On Black-Box Explanation for Face Verification

Domingo Mery
Department of Computer Science
Pontificia Universidad Católica de Chile
domingo.mery@uc.cl

Bernardita Morris
Department of Computer Science
Pontificia Universidad Católica de Chile
bamorris@uc.cl

Abstract

Given a facial matcher, in explainable face verification, the task is to answer: how relevant are the parts of a probe image to establish the matching with an enrolled image. In many cases, however, the trained models cannot be manipulated and must be treated as "black-boxes". In this paper, we present six different saliency maps that can be used to explain any face verification algorithm with no manipulation inside of the face recognition model. The key idea of the methods is based on how the matching score of the two face images changes when the probe is perturbed. The proposed methods remove and aggregate different parts of the face, and measure contributions of these parts individually and in-collaboration as well. We test and compare our proposed methods in three different scenarios: synthetic images with different qualities and occlusions, real face images with different facial expressions, poses, and occlusions and faces from different demographic groups. In our experiments, five different face verification algorithms are used: ArcFace, Dlib, FaceNet (trained on VGGface2 and CasiaWebFace), and LBP. We conclude that one of the proposed methods achieves saliency maps that are stable and interpretable to humans. In addition, our method, in combination with a new visualization of saliency maps based on contours, shows promising results in comparison with other state-of-the-art art methods. This paper presents good insights into any face verification algorithm, in which it can be clearly appreciated which are the most relevant face areas that an algorithm takes into account to carry out the recognition process.

1. Introduction

Explainable face verification arises from the need to have an understanding of the facial matcher models. This methodology can help us, humans, to interpret and visualize the models that are often considered as black-boxes. These models are very effective, but it is not known what they do internally, sacrificing interpretability for accuracy [22, 17]. From this explanation, usually represented through a visualization of a saliency or attention map, we can obtain a reliable tool that provides us with information on what a model has learned and what are the possible failures of the model [29]. Thus, the explanation must be interpretable and accurate, in which the limits of the model are known [21].

In general, there have been several saliency map approaches proposed in the last years as explanations of deep learning networks [4]. We distinguish the following ones: methods based on the gradient of the class signal with respect to the input image (Gradient-based attribution and Grad-CAM) [26, 27, 28, 35, 3]; a method that modifies the network with a feedback loop to infer the activation status of hidden layers [5]; trained saliency models [8, 15]; methods based on top-down and bottom-up information that estimates the winning probability of each neuron of the model (Excitation Backprop) [33, 7]; a method that prunes the neural network in order to keep those neurons that contribute to the prediction [13]. All of them, however, require the intrinsic model structure to manipulate or observe the outputs of model layers. This is not always available, or often needs specialized knowledge of how the network has been designed. On the other hand, there are general methods that do not require to manipulate the network architecture. In this family of true black-box explanation approaches, we find LIME [23] that uses a random selection of superpixels and a linear decision model; RISE [19] and D-RISE [20]

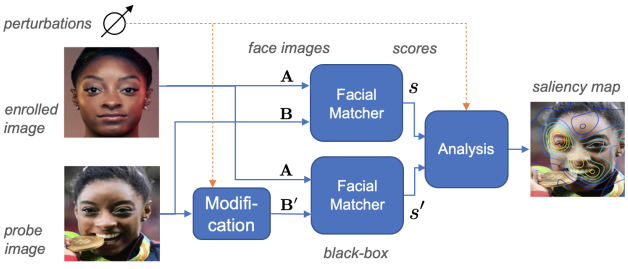

Figure 1. The proposed methods are based on how the matching score between enrolled and probe images changes when the probe is perturbed. The facial matcher is considered as a black-box.

that analyze the response of the model when the input is sampled randomly with square patches; and methods based on perturbed input images that maximally affect the output [11, 10]. However, these models have been tested mainly on object detection rather than face recognition.

For explainable face recognition, the most relevant methods that have been published are: a learnable module, xCos, that can be added into the deep face verification model [16], an exhaustive analysis of VGGface [18] to understand the inner work [34], a learned structured face representation that activates relevant face parts based on a Siamese network [32], a model trained on a controlled dataset to understand its behavior [30], and new evaluation protocols for explainable face recognition based on triplets (probe, mate and nonmate) and white-box saliency methods based on excitation backprop among others [29]. All of them, however, assume that they can access to the layers of the deep learning architecture used by the facial matcher. This is not always possible, especially in commercial software.

In this paper, we present a general approach to explain face verification using true black-box methods. Thus, we are able to analyze any facial matcher as a black-box without accessing the internal structure of the facial matcher model. In our approach, we only need the matching score between two images (or the embedding of the individual images to compute the score) as illustrated in Fig. 1. The contributions of the paper are four-fold:

• A general agnostic methodology that can be used to explain any facial matcher (including algorithms based on handcrafted features or commercial software).

• A new visualization of the saliency maps based on contours as a promising alternative for the explanation.

• Explanations of face verification using five facial matchers, and experiments on synthetic images with different qualities and occlusions, real face images with different facial expressions, poses, and occlusions and faces from different demographic groups.

• Adaption, implementation and comparison of LIME [23] and RISE [19] to explain facial matchers.

2. Proposed Method

In our work, we propose six different methods to explain black-boxes for face verification. The key idea is to evaluate the matching between two face images $\mathbf{A}$ (enrolled image) and $\mathbf{B}$ (probe image) by answering the question: *How relevant are the parts of* $\mathbf{B}$ *to establish the matching with* $\mathbf{A}$*?* To answer this question, our proposed methods analyze how the matching score between face images $\mathbf{A}$ and $\mathbf{B}$ changes when face image $\mathbf{B}$ is replaced by $\mathbf{B}'$, a modification of $\mathbf{B}$. The approaches can analyze any facial matcher as a black-box, *i.e*. with no manipulation inside of the face recognition model, just analyzing the response under certain scenarios as illustrate in Fig. 1.

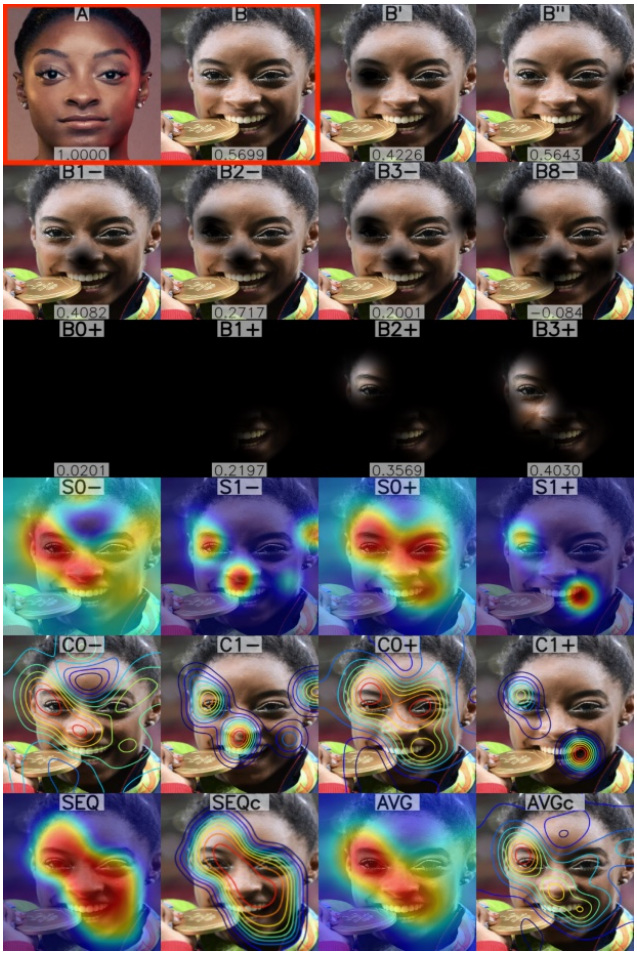

Figure 2. Proposed saliency maps. Row-1: image pair (A,B), and modifications of B removing one eye (B') and one ear (B"). Row-2: outputs of algorithm Greedy Removal after iteration t=1,2,3,8, where the most important t regions are removed. Row-3: outputs of algorithm Greedy Aggregation after iteration t=0,...,3, where the most important t regions are aggregated. The matching score with image A is given in each square. Row-4: saliency maps for algorithms S0−, S1−, S0+, and S1+. Row-5: contour visualization of fourth row. Row-6: saliency maps for algorithms SEQ and AVG, and the corresponding contour visualizations SEQc and AVGc.

.

In face verification, the matching score between face images $\mathbf{A}$ and $\mathbf{B}$ is defined as follows:

$$s = \text{score}(\mathbf{A}, \mathbf{B}). \tag{1}$$

Typically, it is the dot product of vectors $\mathbf{x}_A$ and $\mathbf{x}_B$, the embeddings of images $\mathbf{A}$ and $\mathbf{B}$ respectively, with $\mathbf{x}_A = f(\mathbf{A})$ and $\mathbf{x}_B = f(\mathbf{B})$, and $||\mathbf{x}_A|| = ||\mathbf{x}_B|| = 1$, so the matching score s corresponds to the cosine similarity. The perfect similarity, $s = 1$, occurs when $\mathbf{A} = \mathbf{B}$. In face verification, if $\mathbf{A}$ and $\mathbf{B}$ are face images of the same person, the matching score should be greater than a threshold. An example using ArcFace [9] is shown in Fig. 2, where the

matching score between two face images of Simone Biles ($\mathbf{A}$ and $\mathbf{B}$) is $s = 0.5699$.

We define $\mathbf{B}'$ as a modification of $\mathbf{B}$. For example, $\mathbf{B}'$ can be the image after removing one eye of the face as illustrated in Fig. 2. In this case, the matching score, using (1), decreases to $s' = \text{score}(\mathbf{A}, \mathbf{B}') = 0.4226$. Another example could be $\mathbf{B}''$ after removing one ear of the face as shown in Fig. 2. In this new case, the matching score decreases only slightly to $s'' = \text{score}(\mathbf{A}, \mathbf{B}'') = 0.5643$. Like human visual perception, we can say that face verification is much more difficult when one eye is missing than when one ear is missing [12]. That is what happens in our example: if we remove one eye, the matching score will decrease much more than if we remove one ear.

2.1. Proposed Saliency Maps

Using the key idea of how $\text{score}(\mathbf{A}, \mathbf{B})$ changes when replacing $\mathbf{B}$ by $\mathbf{B}'$, we propose six different approaches to build saliency maps:

• **Single Removal** (S0–): We define the modified face image as:

$$\mathbf{B}'_{ij} = \mathbf{B} \circ (1 - G(\sigma, i, j)), \tag{2}$$

that is, a pixel-wise multiplication of image $\mathbf{B}$ and a mask of the same size with values between 0 and 1, where the elements of the mask corresponds to an inverted Gaussian kernel of width σ centered in (i, j). In this operation, we remove a circular region of $\mathbf{B}$ centered in (i, j). An example is illustrated in Fig. 2 (see B′ and B″). Single Removal is performed for a set of coordinates $\{(i, j)\}$ distributed in a grid manner across the image by steps of d pixels. For each modified image, we define a saliency map value:

$$H_0^-(i, j) = \text{score}(\mathbf{A}, \mathbf{B}) - \text{score}(\mathbf{A}, \mathbf{B}'_{ij}), \tag{3}$$

that means, the different between the original score and the modified score. In this saliency map, the larger this difference the more relevant is the removed part.

The saliency map $\mathbf{H} = \mathbf{H}_0^-$ has been sparsely computed, that means, $\mathbf{H}$ is given only in pixels (i, j) that belong to the grid defined in steps of d pixels. For this reason, we smooth the obtained saliency map using a convolutional Gaussian kernel of width σ. This operation can fill the elements of matrix $\mathbf{H}$ that were not considered in the grid evaluation:

$$\mathbf{D} = \text{conv}(\mathbf{H}, G(\sigma)). \tag{4}$$

Additionally, we scale the smoothed saliency map between 0 and 1 using the min-max normalization:

$$\mathbf{S} = \frac{\mathbf{D} - D_{\min}}{D_{\max} - D_{\min}}. \tag{5}$$

The algorithm is given in further details in Algorithm 1. The output in Fig. 2 is shown as S0–.

• **Greedy Removal** (S1–): We repeat the Single Removal procedure iteratively. In each iteration, the most relevant part of image $\mathbf{B}$ is removed. Thus, image $\mathbf{B}$ is replaced by $\mathbf{B}'_{ij}$ where (i, j) is (i^*, j^*), the coordinates that maximize (3). In this approach, we start with $\mathbf{B}_0 = \mathbf{B}$, and after each iteration we obtain image $\mathbf{B}_t$ in which the most relevant part of image $\mathbf{B}_{t-1}$ is removed. The output of each iteration in our example is show in Fig. 2 as B–(t) for $\mathbf{B}_t$ with $t = 1, 2, 3, 8$ (the reader can observe that B–(0) is shown as B). The saliency map is defined in the pixels (i^*, j^*) where the part is removed, and the saliency map value is the difference of the new matching score with the previous one.

$$H_1^-(i^*, j^*) = \text{score}(\mathbf{A}, \mathbf{B}_t) - \text{score}(\mathbf{A}, \mathbf{B}_{t-1}), \tag{6}$$

The iteration stops when a maximal number of iterations is achieved or the difference of the scores is low enough, that

Algorithm 1 – Removal Saliency Maps ($\mathbf{S}_0^-$ and $\mathbf{S}_1^-$)

Input:
$\mathbf{A}$ – Face image A
$\mathbf{B}$ – Face image B
σ – Width of Gaussian mask
d – Steps
θ – Minimal incremente allowed
$t_{\max}$ – Maximal number of iterations

$N, M \leftarrow \text{size}(\mathbf{A})$ ▷ height and width of face images
$\mathbf{H}_0^- \leftarrow \text{zeros}(N, M)$ ▷ initialization of saliency map
$\mathbf{H}_1^- \leftarrow \text{zeros}(N, M)$ ▷ initialization of saliency map
$\mathbf{B}_0 \leftarrow \mathbf{B}$ ▷ initialization of removing image
$t \leftarrow 0$ ▷ initialization of iteration counter
$s_t \leftarrow \text{score}(\mathbf{A}, \mathbf{B}_t)$ ▷ matching score
$\Delta s \leftarrow 1$ ▷ initialization of difference of scores
while $\Delta s > \theta$ and $t < t_{\max}$ **do**
  $t \leftarrow t + 1$
  $s_t \leftarrow 1$
  for $i = 0 : d : N$ **do**
    for $j = 0 : d : M$ **do**
      $\mathbf{B}'_{ij} \leftarrow \mathbf{B}_{t-1} \circ (1 - G(\sigma, i, j))$
      $s' \leftarrow \text{score}(\mathbf{A}, \mathbf{B}'_{ij})$
      if $t = 0$ **then**
        $H_0^-(i, j) \leftarrow s_0 - s'$
      if $s' < s_t$ **then**
        $s_t \leftarrow s'$
        $(i^*, j^*) \leftarrow (i, j)$
        $\mathbf{B}_t \leftarrow \mathbf{B}'_{ij}$
  $\Delta s \leftarrow s_t - s_{t-1}$
  $H_1^-(i^*, j^*) \leftarrow \Delta s$
$\mathbf{S}_0^- \leftarrow \text{minmax}(\text{conv}(\mathbf{H}_0^-, G(\sigma)))$ ▷ smooth and
$\mathbf{S}_1^- \leftarrow \text{minmax}(\text{conv}(\mathbf{H}_1^-, G(\sigma)))$ ▷ normalization

Output:
$\mathbf{S}_0^-, \mathbf{S}_1^-$ ▷ normalized saliency maps

happens in our example by $t = 8$ where the difference of the scores is 0.0372 only. We use (4) and (5) to smooth and normalize the saliency map. The algorithm is given in further details in Algorithm 1. The output in Fig. 2 is shown as S1−.

• **Single Aggregation** (S0+): We define the modified face image by considering a circular region of $\mathbf{B}$ in (i, j):

$$\mathbf{B}'_{ij} = \mathbf{B} \circ G(\sigma, i, j), \tag{7}$$

That is, a pixel-wise multiplication of image $\mathbf{B}$ and a mask of the same size with values between 0 and 1, where the elements of the mask corresponds to a Gaussian kernel of width σ centered in (i, j). This operation corresponds to aggregate a circular region of $\mathbf{B}$ to a black image. An example is illustrated in Fig. 2 (see in B+(1) a region centered in the right part of the mouth). Single Aggregation is performed for a set of coordinates $\{(i, j)\}$ distributed in a grid manner across the image by steps of d pixels. For each modified image, we define a saliency map value:

$$H_0^+(i, j) = \text{score}(\mathbf{A}, \mathbf{B}'_{ij}) - \text{score}(\mathbf{A}, \mathbf{Z}), \tag{8}$$

where $\mathbf{Z}$ is a black image, *i.e.* an image of zeros of the same size as $\mathbf{B}$. That means, the larger this difference the more relevant is the aggregated part. We use (4) and (5) to smooth and normalize the saliency map. The algorithm is given in further details in Algorithm 2. The output in Fig. 2 is shown as S0+.

• **Greedy Aggregation** (S1+): We repeat the Single Aggregation procedure several times to aggregate in each iteration the most relevant part of image $\mathbf{B}$. In this iterative process, we start with $\mathbf{B}_0 = \mathbf{Z}$, a black image, and after each iteration we obtain image $\mathbf{B}_t$ in which the most relevant part of image $\mathbf{B}$ is aggregated to $\mathbf{B}_{t-1}$. The image of each iteration in our example is show in Fig. 2 as B+(t) for $\mathbf{B}_t$ (the reader can observe that B+(0) is a black image). The saliency map is defined in the pixels (i^*, j^*) where the part is aggregated, and the saliency map value is the difference of the new matching score with the previous one:

$$H_1^+(i^*, j^*) = \text{score}(\mathbf{A}, \mathbf{B}_t) - \text{score}(\mathbf{A}, \mathbf{B}_{t-1}), \tag{9}$$

The iteration stops when a maximal number of iterations is achieved or the difference of the scores is low enough, that happens in our example by $t = 3$ where the difference of the scores is 0.0386 only. We use (4) and (5) to smooth and normalize the saliency map. The algorithm is given in further details in Algorithm 2. The output in Fig. 2 is shown as S1+.

• **Sequential Removal/Aggregation** (SEQ): We start with Greedy Aggregation until the matching score is greater than a threshold, and we switch to Greed Aggregation until the matching score is lower than another threshold. We repeat this sequence several times to find a reduced image $\mathbf{B}_r$ which matching score is very similar to the original one. The reported saliency map is the corresponding saliency map H_0 of the last iteration. We use (4) and (5) to smooth and normalize the saliency map.

• **Average Removal/Aggregation** (AVG): The first four procedures compute different saliency maps that show relevant parts of the face image $\mathbf{B}$ when matching to face image $\mathbf{A}$. The strategies remove and aggregate relevant parts, and measure individual contributions of these parts and in collaboration as well. For these reasons, we think that a average of these four saliency maps can provide a good insight into the significance of each part face image $\mathbf{B}$. The average is computed as $\mathbf{S}_{\text{avg}} = (\mathbf{S}_0^- + \mathbf{S}_1^- + \mathbf{S}_0^+ + \mathbf{S}_1^+)/4$.

Algorithm 2 – Aggregation Saliency Maps ($\mathbf{S}_0^+$ and $\mathbf{S}_1^+$)

Input:
$\mathbf{A}$ – Face image A
$\mathbf{B}$ – Face image B
σ – Width of Gaussian mask
d – Steps
θ – Minimal incremente allowed
$t_{\max}$ – Maximal number of iterations

$N, M \leftarrow \text{size}(\mathbf{A})$ ▷ height and width of face images
$\mathbf{H}_0^+ \leftarrow \text{zeros}(N, M)$ ▷ initialization of saliency map
$\mathbf{H}_1^+ \leftarrow \text{zeros}(N, M)$ ▷ initialization of saliency map
$\mathbf{B}_0 \leftarrow \text{zeros}(N, M, 3)$ ▷ black image
$t \leftarrow 0$ ▷ initialization of iteration counter
$s_t \leftarrow \text{score}(\mathbf{A}, \mathbf{B}_t)$ ▷ matching score
$\Delta s \leftarrow 1$ ▷ initialization of difference of scores
while $\Delta s > \theta$ and $t < t_{\max}$ **do**
  $t \leftarrow t + 1$
  $s_t \leftarrow 0$
  for $i = 0 : d : N$ **do**
    for $j = 0 : d : M$ **do**
      $\mathbf{B}'_{ij} \leftarrow \mathbf{B}_{t-1} \oplus (\mathbf{B} \circ G(\sigma, i, j))$
      $s' \leftarrow \text{score}(\mathbf{A}, \mathbf{B}'_{ij})$
      if $t = 0$ **then**
        $H_0^+(i, j) \leftarrow s' - s_0$
      if $s' > s_t$ **then**
        $s_t \leftarrow s'$
        $(i^*, j^*) \leftarrow (i, j)$
        $\mathbf{B}_t \leftarrow \mathbf{B}'_{ij}$
  $\Delta s \leftarrow s_t - s_{t-1}$
  $H_1^+(i^*, j^*) \leftarrow \Delta s$
$\mathbf{S}_0^+ \leftarrow \text{minmax}(\text{conv}(\mathbf{H}_0^+, G(\sigma)))$ ▷ smooth and
$\mathbf{S}_1^+ \leftarrow \text{minmax}(\text{conv}(\mathbf{H}_1^+, G(\sigma)))$ ▷ normalization

Output:
$\mathbf{S}_0^+, \mathbf{S}_1^+$ ▷ normalized saliency maps

2.2. Proposed Visualization

Typically, the saliency map is visualized using a color heat map (red for high values and blue for low values). The

heat map is superimposed onto the face image, so in the same face image, we can visualize the most relevant parts of the face in red, orange and yellow, and the regions with less significance in green and blue.

Instead of superimposition of heat maps, we propose to superimpose contours because they do not alter significantly the details of the face image. A contour is defined as a continuous line that corresponds to a boundary of the saliency map having the same saliency value. In our visualization, the color of the contours are the same colors of the heat map and we use 8-10 contour lines.

In Fig. 2, the six proposed saliency maps are shown as superimposed heat maps as S0-, S1-, S0+, S1+, SEQ and AVG, and as contours as C0-, C1-, C0+, C1+, SEQc and AVGc respectively.

3. Experimental results

In this section, we show the experimental results that we achieved using the proposed saliency maps. Firstly, as sanity check [1], we computed saliency maps for enrolled faced with random probe images and different facial matchers. The results show random saliency maps, which means that the proposed methods are dependent of the model and data.

3.1. Comparison of Saliency Maps

We compare ten different explanation approaches for face verification in the matching of eleven pairs of the same person using ArcFace [9]. For this person, from MORPH dataset [24], we have eleven face images as shown in the first row of Fig. 3: **A**, **B**, **C**, ... **K**. The first two ones are original face images, the last nine images are synthetic versions of image **B** as follows: images **C**, **D**, **E** are images with different resolutions; images **F**, **G**, **H** with different blurriness; images images **I**, **J**, **K** are images with occlusions (eye patch, eyes with black rectangle, and face-mask respectively). The eleven pairs for the face matching evaluation consist of pairs (**A**,**A**), (**A**,**B**), (**A**,**C**), ... (**A**,**K**), *i.e*. all pairs have image **A** as first image (enrolled image), and each of the eleven images as second image (probe image).

The ten saliency maps to be compared are shown in Fig. 3: LIME [23] (adaption of original algorithm that shows the relevant parts of the face image), LIME-map (saliency map of LIME), RISE [19] (adaption of original algorithm using square masks), RISE-Gauss (RISE algorithm with Gaussian masks)[1] and our six proposed algorithms described in section 2.1: S0–, S1–, S0+, S1+, SEQ and AVG.

For the not-occluded face images (from **A** to **H**) –face images with different degrees of quality–, we distinguish saliency maps based on LIME and RISE approaches do not show always relevance in the periocular region and nose. That is not the case of five of the six proposed methods (excluding SEQ that shows sometimes saliency outside of the face). In particular, AVG is very stable in the center of the face, showing the best results in this experiment.

A similar performance is achieved with occluded faces **I**, **J**, **K**, where the most relevant information of the face is in the not occluded parts of the face. We observe that the saliency maps of the proposed methods focused on the relevant parts of the faces (and not on the occlussion). In these images, again LIME approaches do not offer good results, however, RISE approaches can be used to explain occluded faces. The disadvantage of LIME and RISE methods is the random component in the evaluation, this indicates that the results vary slightly with each run of the algorithm.

In our experiments, the proposed method AVG is very stable and focused on the relevant parts of the face images. In addition, we believe, that the visualization of the saliency map using contours, as can be seen in AVGc in Fig. 3, shows a very promising alternative for the explanation. The contour visualization highlights the most important areas of the face using colored lines, without altering much of the visual information of the face.

For these reasons, in the next experiments, the results of the saliency maps are presented using AVGc, that means, it is the average of the first four proposed saliency maps (S0–, S1–, S0+, S1+) using contour visualization.

3.2. Evaluation of Facial Matchers

In this section, we report the results in five different face verification algorithms achieved by our proposed method AVG using contour visualization. The five algorithms are ArcFace [9], Dlib [14], FaceNet [25] (trained on VGGface2 [6] and Casia-WebFace [31]) and LBP (local binary patterns) [2]. The first four algorithms are methods based on deep learning, and the last one is a method based on handcrafted features developed before the deep learning age.

For the evaluation of the face verification algorithm we use 18 pairs using 18 face images of Angela Merkel, as shown in Fig. 4 in the rows called original: **A**, **B**, **C**, ... **R**. All of them are original face images with different facial expressions, poses, and occlusions (with her hands, face-mask and a glass of wine). The 18 pairs for the face matching evaluation consist of pairs (**A**,**A**), (**A**,**B**), (**A**,**C**), ... (**A**,**R**), *i.e*. all pairs have image **A** as first image (enrolled image), and each of the 18 images as second image (probe image).

From the results shown in Fig. 4, it is clear that the explanation of LBP reveals that this method is making its decisions by paying attention to parts of the face that are obviously not relevant, because many saliency maps are focused on occluded parts and does not consider the periocular area to be of major relevance. On the other hand, the results for the four methods based on deep learning show

[1] We create LIME-map (random selection of superpixels with a saliency map using RISE strategy) and RISE-Gauss (that replaces squares by Gaussian masks) as simple modifications of the original algorithms.

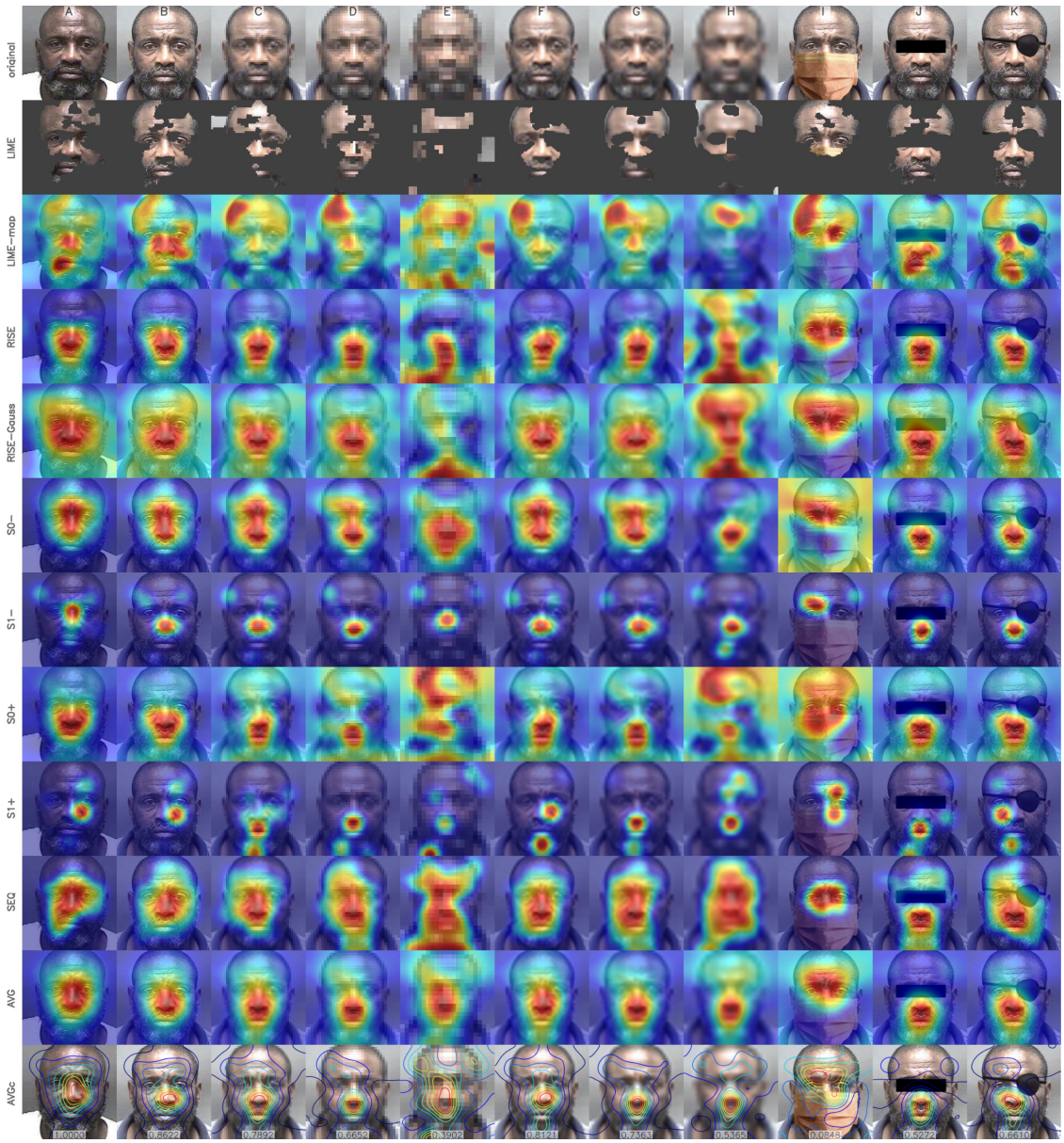

Figure 3. Saliency maps for matching between image A and images A, B, C ... K using ArcFace. Images C, D, ... H are synthetic low quality versions of image B. Images I, J, K are synthetic occluded versions of image B. LIME and RISE are known methods. Proposed methods are S0–, S1–, S0+, S1+, SEQ and AVG. The last row (AVGc) is the contour visualization of AVG.

more interpretable explanations, because the saliency maps are focused on relevant parts. In addition, these methods exclude in the vast majority of cases the occluded zones of the faces. From the four deep learning methods, Dlib is probably the most questioned because the maps are very focused on small regions of the face, downplaying other areas of the face that should contribute to facial recognition. The other three methods (ArcFace, FaceNet trained on VGGface2 and Casia-WebFace) make good use of the areas of the face, highlighting the focus on the periocular area and the nose.

3.3. Explanations on Demographic Groups

In our last experiment, we try to answer the question about what are the most relevant parts of the face when dealing with demographic groups. For this end, we use MORPH data set [24] and select randomly 100 pairs (of 100 different persons) for each of the following demographic groups: FB black female, FW, white female, MB black male, MW, white male, F female, M, male, B black, and W white. For each group, we compute the mean of the 100 saliency maps using our proposed approach AVG. The results for five face

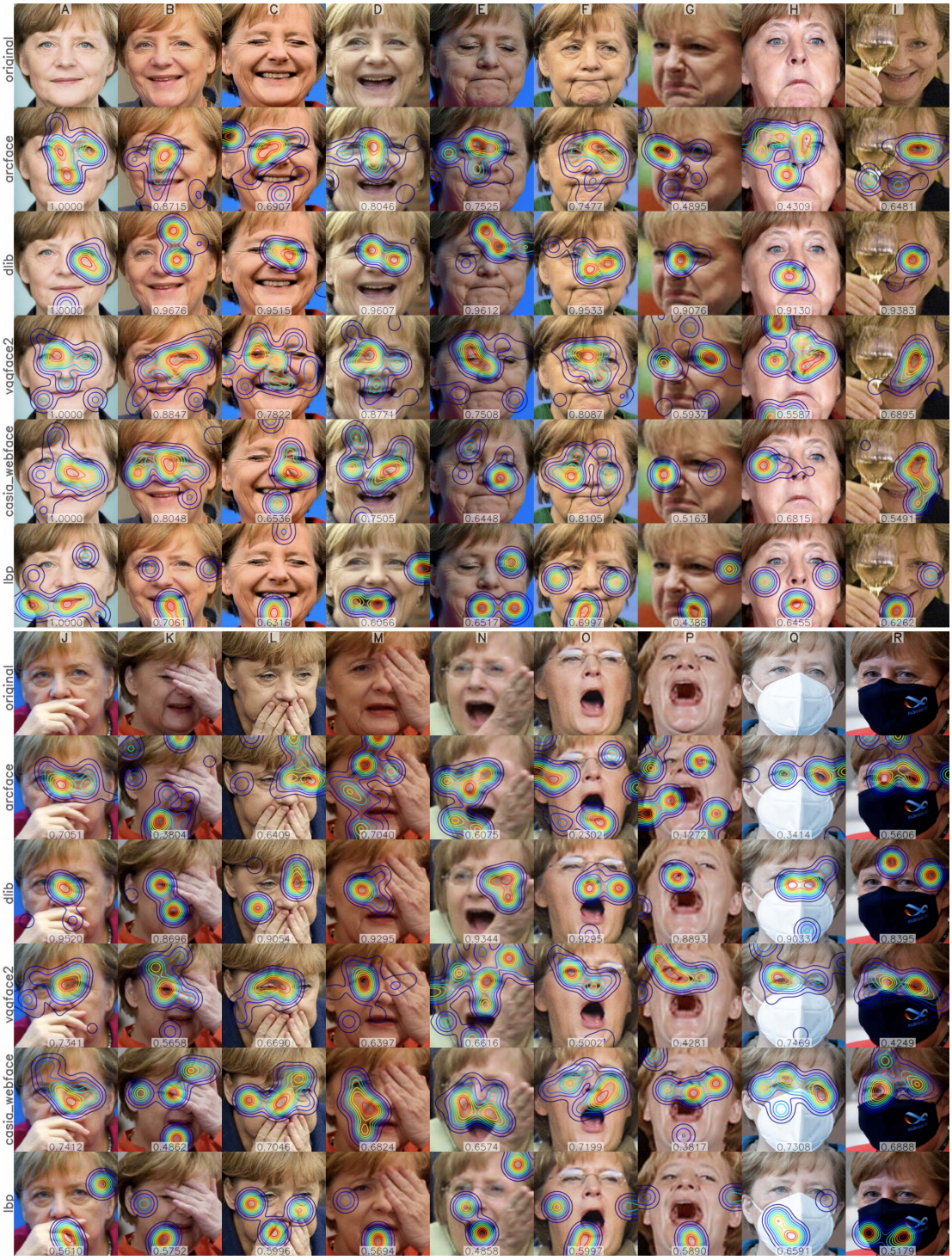

Figure 4. Saliency map using proposed method AVGc for five face verification algorithms between image A and images A, B, ... R.

verification approachs (ArcFace, Dlib, Facenet (trained on VGGface2 and Casia-WebFace) and LBP) are shown in Fig. 5. In this figure, a representative face of the demographic group is presented as background. The most important result in this experiment, is that there is no significative difference across the demographic groups, because the saliency maps (in each face verification method) are very similar to each other. There are differences between the face verification methods, but no major differences between the demographic groups.

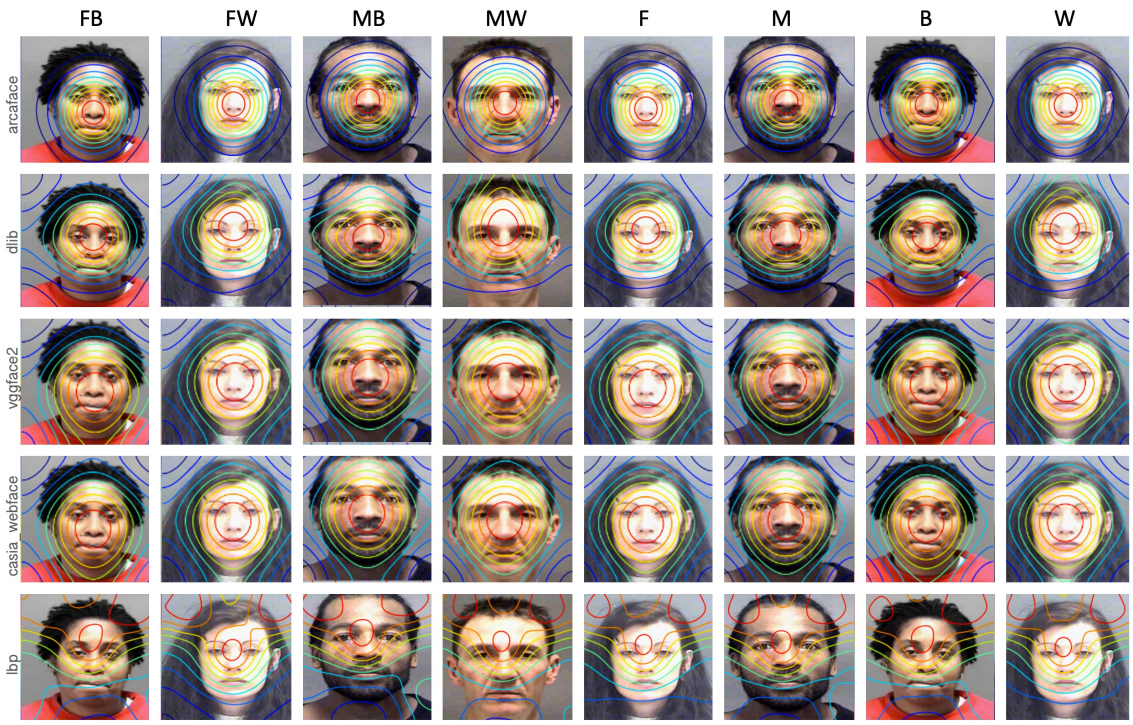

Figure 5. Explanations using AVGc on demographic groups of MORPPH: F/M - Females/Males, B/W - Black/White.

Again, the explanation of LBP reveals that this method is making its decisions by paying attention to parts of the face that are not relevant (for example the gray top corners of the images). Additionally, the lines of saliency maps of LBP are rather horizontal than concentric.

On the other hand, the deep learning methods show ring patterns centered in the center of the face. In ArcFace, the center is clearly in the nose. As in previous experiment, the explanation for the four methods based on deep learning are more interpretable to humans, because the saliency maps are focused on the faces.

3.4. Implementation

The proposed methods have been implemented in Python using Google Colab[2]. Computational times for each pair are approx. 1, 3, 1, 3, 6, 6 minutes for S0–, S1–, S0+, S1+, SEQ and AVG respectively. In our experiments: $N = M = 256$, $d = 8$, $\sigma = 19$, $\theta = 0.05$, $t_{\max} = 20$. The parameters were set manually.

4. Conclusions

In this paper, we present different saliency maps that can be used to explain any face verification algorithm. The key idea of the methods is based on how the matching score of two face images changes when replacing one of the face images with a modification of itself. We propose six different approaches to build saliency maps avoiding any manipulation inside of the face recognition model. Four of these methods remove and aggregate relevant parts of the face, and measure individual contributions of these parts and in collaboration as well. The other two methods use combinations of these four methods, where the most promising result has been achieved by the average of the four mentioned methods, yielding stable saliency maps focused on relevant parts of the faces. We test and compare our proposed methods in three different scenarios: synthetic images with different qualities and occlusions, real face images with different facial expressions, poses, and occlusions and faces from different demographic groups. In our experiments, five different face verification algorithms are used. Moreover, we propose a new visualization of saliency maps based on contours. Instead of superimposition of heat maps, we propose to superimpose contours because they do not alter significantly the details of the face image. The contour visualization shows a very promising alternative for the explanation. This paper presents a qualitative explanation of any face verification algorithm, in which it can be clearly appreciated which are the most relevant face areas that an algorithm takes into account to carry out the recognition. We believe that our approach can be used to evaluate commercial off-the-shelf face recognition systems as well.

Acknowledgments – This work was supported by Fondecyt Grant No. 1191131 from CONICYT, Chile.

[2]Code and images are available in `https://domingomery.ing.puc.cl/material/`.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in Neural Information Processing Systems*, 2018-Decem(NeurIPS):9505–9515, 2018.

[2] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face description with local binary patterns: application to face recognition. *IEEE Trans Pattern Anal Mach Intell*, 28(12):2037–2041, dec 2006.

[3] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Gradient-Based Attribution Methods. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11700 LNCS:169–191, 2019.

[4] Xiao Bai, Xiang Wang, Xianglong Liu, Qiang Liu, Jingkuan Song, Nicu Sebe, and Been Kim. Explainable Deep Learning for Efficient and Robust Pattern Recognition: A Survey of Recent Developments. *Pattern Recognition*, 120:108102, 2021.

[5] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, Deva Ramanan, and Thomas S. Huang. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:2956–2964, 2015.

[6] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[7] Gregory Castañón and Jeffrey Byrne. Visualizing and quantifying discriminative features for face recognition. *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*, pages 16–23, 2018.

[8] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. *Advances in Neural Information Processing Systems*, 2017-Decem:6968–6977, 2017.

[9] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[10] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:2950–2958, 2019.

[11] Ruth C. Fong and Andrea Vedaldi. Interpretable Explanations of Black Boxes by Meaningful Perturbation. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:3449–3457, 2017.

[12] Paweł Karczmarek, Witold Pedrycz, Adam Kiersztyn, and Przemysław Rutka. A study in facial features saliency in face recognition: an analytic hierarchy process approach. *Soft Computing*, 21(24):7503–7517, 2017.

[13] Ashkan Khakzar, Soroosh Baselizadeh, Saurabh Khanduja, Christian Rupprecht, Seong Tae Kim, and Nassir Navab. Improving feature attribution through input-specific network pruning. *arXiv preprint arXiv:1911.11081*, 2019.

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

[15] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9215–9223, 2018.

[16] Yu-Sheng Lin, Zhe-Yu Liu, Yu-An Chen, Yu-Siang Wang, Hsin-Ying Lee, Yi-Rong Chen, Ya-Liang Chang, and Winston H Hsu. xCos: An explainable cosine metric for face verification task. *arXiv preprint arXiv:2003.05383*, 2020.

[17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777, 2017.

[18] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference (BMVC2015)*, 2015.

[19] Vitali Petsiuk, Abir Das, and Kate Saenko. RisE: Randomized input sampling for explanation of black-box models. *British Machine Vision Conference 2018, BMVC 2018*, 1, 2019.

[20] Vitali Petsiuk, Rajiv Jain, Varun Manjunatha, Vlad I Morariu, Ashutosh Mehra, Vicente Ordonez, and Kate Saenko. Black-box explanation of object detectors via saliency maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11443–11452, 2021.

[21] P Jonathon Phillips and Mark Przybocki. Four principles of explainable ai as applied to biometrics and facial forensic algorithms. *arXiv preprint arXiv:2002.01014*, 2020.

[22] Arun Rai. Explainable AI: from black box to glass box. *Journal of the Academy of Marketing Science*, 48(1):137–141, 2020.

[23] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should i trust you?" Explaining the predictions of any classifier. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-Augu:1135–1144, 2016.

[24] Karl Ricanek and Tamirat Tesafaye. Morph: A longitudinal image database of normal adult age-progression. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 341–345. IEEE, 2006.

[25] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823. IEEE, jun 2015.

[26] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, 128(2):336–359, 2020.

[27] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Ba-

tra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

[28] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*, pages 1–8, 2014.

[29] Jonathan R. Williford, Brandon B. May, and Jeffrey Byrne. Explainable Face Recognition. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12356 LNCS:248–263, 2020.

[30] Tian Xu, Jiayu Zhan, Oliver GB Garrod, Philip HS Torr, Song-Chun Zhu, Robin AA Ince, and Philippe G Schyns. Deeper interpretability of deep networks. *arXiv preprint arXiv:1811.07807*, 2018.

[31] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[32] Bangjie Yin, Luan Tran, Haoxiang Li, Xiaohui Shen, and Xiaoming Liu. Towards interpretable face recognition. *Proceedings of the IEEE International Conference on Computer Vision*, 2019-Octob:9347–9356, 2019.

[33] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-Down Neural Attention by Excitation Backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

[34] Yaoyao Zhong and Weihong Deng. Exploring features and attributes in deep face recognition using visualization techniques. *Proceedings - 14th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2019*, 2019.

[35] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:2921–2929, 2016.