True Black-Box Explanation in Facial Analysis

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

Abstract

When explaining a recognition approach that can be used in facial analysis, e.g. *face verification, face detection, attribute recognition, etc., the task is to answer: how relevant are the parts of a given image to establish the recognition. In many cases, however, the trained models cannot be manipulated and must be treated as "black-boxes". In this paper, we present a saliency map methodology, called MinPlus, that can be used to explain any facial analysis approach with no manipulation inside of the recognition model, because it only needs the input-output function of the black-box 'f_x'. The key idea of the method is based on how the probability of recognition of the given image changes when it is perturbed. Our method removes and aggregates different parts of the image, and measures contributions of these parts individually and in-collaboration as well. We test and compare our method in four different scenarios: face verification (with ArcFace), face expression recognition (with Xception), face detection (with MTCNN) and masked face detection (with YOLOv5s). We conclude that MinPlus achieves saliency maps that are stable and interpretable to humans. In addition, our method shows promising results in comparison with other state-of-the-art methods like AVG, LIME and RISE. This paper presents good insights into any facial analysis approach. It can be used to highlight the most relevant areas that an algorithm takes into account to carry out the recognition process.*

1. Introduction

Explainability can help us, humans, to interpret and visualize the models that are often considered as black-boxes. Usually, these models are very effective, but it is not known what they do internally, sacrificing interpretability for accuracy [15, 22]. The explanation, generally represented through a visualization of a saliency or attention map, can be used to obtain a reliable tool that provides us with information on what a model has learned and what are the possible failures of the model [28]. Thus, the explanation must be interpretable and accurate, in which the limits of the model are known [21].

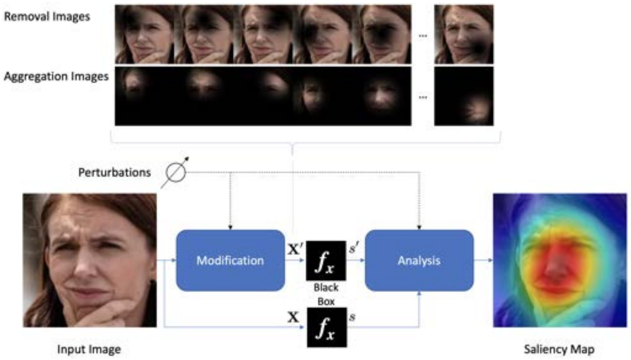

Figure 1. Diagram of our method MinPlus. The key idea is to analyze how the score varies when add or remove a part of the original image. The input-output function of the black-box is 'f_x'. In this example, the saliency map corresponds to the most important parts used by a face verification algorithm.

Explainable facial analysis arises from the need to have an understanding of the facial analysis models. 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 [14], an exhaustive analysis of VGGface [18] to understand the inner work [33], a learned structured face representation that activates relevant face parts based on a Siamese network [30], a model trained on a controlled dataset to understand its behavior [29], 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 [28]. Other approaches that analyze the influence of perturbations in the output, can be used for black-block explanation in face verification [16]. For explanation of face attribute recognition, there are some few works (*e.g.* in expression recognition [12]). All of these methods, with the exception of [16], 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.

More in general, several saliency map approaches have been proposed in the last years as explanations of deep learning networks [3]. We distinguish the following ones: a method that modifies the network with a feedback loop to infer the activation status of hidden layers [5]; methods based on the gradient of the class signal with respect to the input image (Gradient-based attribution and Grad-CAM) [1, 24–26, 34]; trained saliency models [7, 13]; a method that prunes the neural network in order to keep those neurons that contribute to the prediction [11]; methods based on top-down and bottom-up information that estimates the winning probability of each neuron of the model (Excitation Backprop) [6, 31]. 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] 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 [9, 10, 27]. These models have been tested mainly on object detection problems.

In this paper, we present a general approach to explain facial analysis algorithms using a true black-box method without accessing the internal structure of the recognition model. This method can be used in any facial analysis algorithm: face verification, face recognition, face detection, face attribute recognition (*e.g.* expressions, gender, age, etc.) and others like masked face detection.

Our method takes into account the importance of selective face areas (that has been already studied in the past, *e.g.* [4]). In our approach, we only need a score function 'f_x' of the black-box, as shown in Figure 1, that gives the probability that an input image $\mathbf{X}$ is (or contains) *something*. Thus, the method can be applied to any method, not only CNNs.

Our proposed general approach is based on the ideas of the methodology that was originally developed in [16] for face verification only, where six different saliency maps have been proposed for explaining face verification: S_0^-, S_0^+, S_1^-, S_1^+, SEQ (a sequential combination of the first two ones) and AVG (an average of the first four ones). The key idea of the first four saliency maps is to analyze what happens with the face comparison score when removing (methods with minus '-': S_0^- or S_1^-) or aggregating (methods with minus '+': S_0^+ or S_1^+) relevant parts of the image. The index '0' (methods S_0^- or S_0^+) means the operation is performed in a grid manner by evaluating the perturbation when only a single part of the image is removed or aggregated. The index '1' (methods S_1^- or S_1^+) means the operation is performed iteratively by removing or aggregating the most relevant part in each iteration until a maximal number of iteration is achieved or the delta in the comparison score is low enough. The authors in [16] reported that the best performance has been achieved by saliency map AVG, *i.e.* an average of the first four saliency maps S_0^-, S_0^+, S_1^-, S_1^+.

Our proposed method, called MinPlus, is based on the same idea of AVG [16] that takes into account the removal and aggregation of relevant parts of the image to analyze the perturbations, however, the main differences between MinPlus and AVG are: *i)* the way the first four saliency maps are computed, *ii)* the way they are combined, and *iii)* finally, our method MinPlus can be used to explain any facial analysis tasks (on the contrary, AVG was developed for face verification only). Details are given in Section 2.

The main contributions of the paper are the following three:

- A general agnostic methodology that can be used to explain any facial analysis algorithm based on the definition of a function 'f_x' for verification, attribute recognition or detection.
- Explanations of facial analysis algorithms in four scenarios: face verification, face attribute recognition, face detection and masked face detection.
- An objective metric to explain face detection methods that compares the saliency maps with the detected bounding boxes. Thus, subjective evaluation can be avoided.

2. Our Method: MinPlus

Now, we present our general method that can be used to explain any recognition approach. The only function we need from the recognition approach is the black-box function:

$$f_x(\mathbf{X}) = s \tag{1}$$

i.e. a function that computes a score 's' of input image $\mathbf{X}$. Details of how to define this function are given in subsection 2.1. Using the key idea of how $f_x(\mathbf{X})$ changes when replacing $\mathbf{X}$ by $\mathbf{X}'$ (see Figure 1), our general approach analyzes what happens when we remove or aggregate a region of the image $\mathbf{X}$. In following subsections (2.2 and 2.3), we explain, the removal and the aggregation strategies, and how we can combine them (subsection 2.4). We call this method MinPlus for a combination of removing (Minus) and adding (Plus).

2.1. Score function f_x of the Black-Box

The score function can be understood as a probability that $\mathbf{X}$ is (or has) *something*. For instance, 'f_x' can be defined as:

- the probability that the face in image $\mathbf{X}$ is the same that the face in another image, *e.g.* $\mathbf{Y}$,
- the probability that the expression of the face in image $\mathbf{X}$ is happy,
- the probability that a face is detected in image $\mathbf{X}$,
- the probability that the face in image $\mathbf{X}$ wears a mask.

We distinguish three kind of score functions that can be used in facial analysis:

1. Verification: In face verification, the aim is to validate the identification of a probe face image $\mathbf{X}$ against a gallery face image $\mathbf{Y}$. Using an embedding function 'f_e', *i.e.* from ArcFace [8], a unit vector is computed for each image as $\mathbf{x} = f_e(\mathbf{X})$ and $\mathbf{y} = f_e(\mathbf{Y})$ respectively. The score function is computed as the dot product of the embeddings: $\langle \mathbf{x}, \mathbf{y} \rangle$. Thus, the score function, known as comparison score[1], is defined by:

$$f_x(\mathbf{X}) = \langle f_e(\mathbf{X}), f_e(\mathbf{Y}) \rangle. \tag{2}$$

In this case, the larger the score the higher the probability that faces $\mathbf{X}$ and $\mathbf{Y}$ belongs to the same person.

2. Attribute recognition: In attribute recognition, a unit vector $\mathbf{x}$ is extracted from a face image $\mathbf{X}$ using function 'f_a'. Each element of $\mathbf{x} = f_a(\mathbf{X})$ gives the probability that a specific attribute (of a family of attributes) is present. A typical example is the expression recognition, where a 7-element vector is given for the seven universal expressions (angry, disgust, scared, happy, sad, surprised, neutral). Thus, the fourth element, x_4, gives the probability that the face in image $\mathbf{X}$ is "happy". The score function must be computed for a specific attribute k, *e.g.* $k = 4$ for happiness in expression recognition. Another example can be gender recognition. Thus, the score function is computed as:

$$f_x(\mathbf{X}) = f_a(\mathbf{X}, k). \tag{3}$$

In this case, the larger the score the higher the probability that attribute k is present in face $\mathbf{X}$.

3. Detection: For facial analysis, the detection can be used to detect faces in an image, to detect some kind of specific faces (*e.g.* people that are wearing masks), to detect some parts of the face (*e.g.* detection of the nose), etc. In detection, for a given input image $\mathbf{X}$ the output is a list 'L' of detected objects, computed as $L = f_d(\mathbf{X})$. Each detected object is defined by:

- 'loc'– its localization (typically a bounding box)
- 'class' – its category (a number that identifies its class),
- 'sc' – its score that corresponds to the probability that the detected obtect belongs to the detected category.

For example, for the third detected object in the list L of image $\mathbf{X}$, the score of that object is $L(3,\text{'sc'})$. To establish the score function we need *i)* a specific category and *ii)* a given location. That means, we can say that function score 'f_x' corresponds to

$$f_x(\mathbf{X}) = L(\hat{k}, \text{'sc'}) \tag{4}$$

where $L = f_d(\mathbf{X})$, and $\hat{k}$ is the number of the detected object that matches with the given category and the given location. If there is no match, $f_x(\mathbf{X}) = 0$. To establish if the location coincides, we can use the intersection over union criterium.

In this case, the larger the score the higher the probability that image $\mathbf{X}$ contains the defined object in the given location.

2.2. Removal Strategy

In this subsection, we explain the removal strategy that analyzes the relevance of each region of an image by removing it. We define the modified face image as:

$$\mathbf{X}'_{ij} = \mathbf{X} \circ (1 - G(\sigma, i, j)), \tag{5}$$

that is a modified image, called $\mathbf{X}'_{ij}$ of the same size of $\mathbf{X}$, that is computed as a pixel-wise multiplication of image $\mathbf{X}$ 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 $\mathbf{X}$ centered in (i, j). Examples of these images are illustrated in Figure 1 (see 'Removal Images').

In this strategy, the 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:

$$H_0^-(i, j) = f_x(\mathbf{X}) - f_x(\mathbf{X}'_{ij}), \tag{6}$$

that means, the difference between the original score and the modified score. In this saliency map, the larger this difference the more relevant is the removed part. Algorithm 1 shows this approach.

[1] We use the standard vocabulary given by ISO/IEC 2382-37:2017: "comparison score: measurement of similarity between biometric probe and biometric reference". See https://www.iso.org/obp/ui/iso:std:iso-iec:2382:-37:ed-2:v1:en:term:3.5.7

Algorithm 1 :
– Single Removal Saliency Map ($\mathbf{H}_0^-$)

```
Input:
$\mathbf{X}$     : Input image of $N \times M$ pixels
$\sigma$     : Width of Gaussian mask
$d$     : Steps
$f_x$     : Score function
————————————————————————————————
$\mathbf{H}_0^- \leftarrow \text{zeros}(N, M)$        ▷ initialization of saliency map
$s_0 \leftarrow f_x(\mathbf{X})$        ▷ initial score
for $i = 0 : d : N$ do
    for $j = 0 : d : M$ do
        $\mathbf{X}'_{ij} \leftarrow \mathbf{X} \circ (1 - G(\sigma, i, j))$        ▷ removal
        $s' \leftarrow f_x(\mathbf{X}'_{ij})$
        $H_0^-(i, j) \leftarrow s_0 - s'$
————————————————————————————————
Output:
$\mathbf{H}_0^-$        ▷ saliency map
```

Algorithm 2 :
– Accumulated Removal Saliency Map ($\mathbf{H}_1^-$)

```
Input:
$\mathbf{X}$     : Input image of $N \times M$ pixels
$\sigma$     : Width of Gaussian mask
$d$     : Steps
$\theta$     : Minimal incremente allowed
$t_{\max}$  : Maximal number of iterations
$f_x$     : Score function
————————————————————————————————
$\mathbf{H}_1^- \leftarrow \text{zeros}(N, M)$        ▷ initialization of saliency map
$\mathbf{X}_0 \leftarrow \mathbf{X}$        ▷ initial image
$s_0 \leftarrow f_x(\mathbf{X}_0)$        ▷ initial score
$t \leftarrow 0$        ▷ initialization of iteration counter
$\Delta s \leftarrow 1$        ▷ 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
            $\mathbf{X}'_{ij} \leftarrow \mathbf{X}_{t-1} \circ (1 - G(\sigma, i, j))$        ▷ removal
            $s' \leftarrow f_x(\mathbf{X}'_{ij})$
            if $s' < s_t$ then
                $s_t \leftarrow s'$
                $(i^*, j^*) \leftarrow (i, j)$
                $\mathbf{X}_t \leftarrow \mathbf{X}'_{ij}$
    $\Delta s \leftarrow s_{t-1} - s_t$
    $H_1^-(i^*, j^*) \leftarrow \Delta s$
————————————————————————————————
Output:
$\mathbf{H}_1^-$        ▷ saliency map
```

Algorithm 3 :
– Single Aggregation Saliency Map ($\mathbf{H}_0^+$)

```
Input:
$\mathbf{X}$     : Input image of $N \times M$ pixels
$\sigma$     : Width of Gaussian mask
$d$     : Steps
$f_x$     : Score function
————————————————————————————————
$\mathbf{H}_0^+ \leftarrow \text{zeros}(N, M)$        ▷ initialization of saliency map
$\mathbf{X}_0 \leftarrow \text{zeros}(N, M, 3)$        ▷ initial image
$s_0 \leftarrow f_x(\mathbf{X}_0)$        ▷ initial score
for $i = 0 : d : N$ do
    for $j = 0 : d : M$ do
        $\mathbf{X}'_{ij} \leftarrow \mathbf{X}_0 \oplus (\mathbf{X} \circ G(\sigma, i, j))$        ▷ aggregation
        $s' \leftarrow f_x(\mathbf{X}'_{ij})$
        $H_0^+(i, j) \leftarrow s' - s_0$
————————————————————————————————
Output:
$\mathbf{H}_0^+$        ▷ saliency map
```

Algorithm 4 :
– Accumulated Aggregation Saliency Map ($\mathbf{H}_1^+$)

```
Input:
$\mathbf{X}$     : Input image of $N \times M$ pixels
$\sigma$     : Width of Gaussian mask
$d$     : Steps
$\theta$     : Minimal incremente allowed
$t_{\max}$  : Maximal number of iterations
$f_x$     : Score function
————————————————————————————————
$\mathbf{H}_1^+ \leftarrow \text{zeros}(N, M)$        ▷ initialization of saliency map
$\mathbf{X}_0 \leftarrow \text{zeros}(N, M, 3)$        ▷ initial image
$s_0 \leftarrow f_x(\mathbf{X}_0)$        ▷ initial score
$t \leftarrow 0$        ▷ initialization of iteration counter
$\Delta s \leftarrow 1$        ▷ 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
            $\mathbf{X}'_{ij} \leftarrow \mathbf{X}_{t-1} \oplus (\mathbf{X} \circ G(\sigma, i, j))$        ▷ aggregation
            $s' \leftarrow f_x(\mathbf{X}'_{ij})$
            if $s' > s_t$ then
                $s_t \leftarrow s'$
                $(i^*, j^*) \leftarrow (i, j)$
                $\mathbf{X}_t \leftarrow \mathbf{X}'_{ij}$
    $\Delta s \leftarrow s_t - s_{t-1}$
    $H_1^+(i^*, j^*) \leftarrow \Delta s$
————————————————————————————————
Output:
$\mathbf{H}_1^+$        ▷ saliency map
```

Now, we can define a new saliency map by computing (6) iteratively as follows: In each iteration, the most relevant part of image $\mathbf{X}$ is removed. Thus, image $\mathbf{X}$ is replaced by $\mathbf{X}'_{ij}$ where (i,j) is (i^*,j^*), the coordinates that maximize (6). In this approach, we start with $\mathbf{X}_0 = \mathbf{X}$, and after each iteration we obtain image $\mathbf{X}_t$ in which the most relevant part of image $\mathbf{X}_{t-1}$ is removed. 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 comparison score with the previous one.

$$H_1^-(i^*,j^*) = f_x(\mathbf{X}_t) - f_x(\mathbf{X}_{t-1}), \tag{7}$$

The iteration stops when a maximal number of iterations is achieved or the difference of the scores is low enough. The details of this approach are given in Algorithm 2.

2.3. Aggregation Strategy

Instead of removing parts, we can analyze now what happens when we add them. In this strategy, we define the modified face image by considering a circular region of $\mathbf{X}$ in (i,j):

$$\mathbf{X}'_{ij} = \mathbf{X} \circ G(\sigma,i,j), \tag{8}$$

That is, a pixel-wise multiplication of image $\mathbf{X}$ 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 $\mathbf{X}$ to a black image. An example is illustrated in Figure 1 (see 'Aggregation Images'). In this strategy, the 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:

$$H_0^+(i,j) = f_x(\mathbf{X}'_{ij}) \tag{9}$$

That means, the larger this value the more relevant is the aggregated part. The algorithm is shown in Algorithm 3.

Similar to previous section, we repeat this procedure several times to aggregate in each iteration the most relevant part of image $\mathbf{X}$. In this iterative process, we start with $\mathbf{X}_0 = \mathbf{Z}$, a black image, and after each iteration we obtain image $\mathbf{X}_t$ in which the most relevant part of image $\mathbf{X}$ is aggregated to $\mathbf{X}_{t-1}$. 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 comparison score with the previous one:

$$H_1^+(i^*,j^*) = f_x(\mathbf{X}_t) - f_x(\mathbf{X}_{t-1}), \tag{10}$$

The iteration stops when a maximal number of iterations is achieved or the difference of the scores is low enough. The details of this approach are given in Algorithm 4.

2.4. Combining Removal and Aggregation

In Section 2.2, we compute two saliency maps: $\mathbf{H}_0^-$ that gives information about the relevance of removing a single region in the input image, and $\mathbf{H}_1^-$ that gives information about the relevance of removing regions in combinations using an iterative approach. Similarly, In Section 2.3, we compute two saliency maps: $\mathbf{H}_0^+$ that gives information about the relevance of adding a single region in the input image, and $\mathbf{H}_1^+$ that gives information about the relevance of adding regions in combinations using an iterative approach. In our experiments, we observe that all four saliency maps are important. A simple way to consider all of them, giving them equal relevance, is to calculate the average of all of them:

$$\bar{\mathbf{H}} = \frac{\mathbf{H}_0^- + \mathbf{H}_1^- + \mathbf{H}_0^+ + \mathbf{H}_1^+}{4}. \tag{11}$$

The saliency map $\bar{\mathbf{H}}$ has been sparsely computed, that means, $\bar{\mathbf{H}}$ 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 $\bar{\mathbf{H}}$ that were not considered in the grid evaluation. Additionally, we scale the smoothed saliency map between 0 and 1 using the min-max normalization.

Then, the saliency map is computed by *i)* averaging the individual saliency maps using (11), *ii)* smoothing using a Gaussian mask as low pass filtering:

$$\mathbf{D} = \text{conv}(\bar{\mathbf{H}}, G(\sigma)), \tag{12}$$

and *iii)* scaling between zero and one:

$$\text{MinPlus} = \frac{\mathbf{D} - D_{\min}}{D_{\max} - D_{\min}}. \tag{13}$$

This strategy preserves the original relevance of each saliency map. It differs from the original one, called AVG, proposed in [16], where the saliency map was computed (in reverse) by *i)* smoothing and *ii)* scaling each one of the four individual saliency maps and *iii)* averaging the four scaled maps.

The normalization used by MinPlus gives each map the same importance without weighting them, keeping the original values. In the experiments, we will see this advantage when comparing AVG with MinPlus.

3. Experimental Results

In this Section we present the results obtained in four scenarios: face verification, facial expression recognition, face detection and masked face detection. All of them use the strategy based on the definition of the black-box function 'f_x' explained in Section 2.1 for verification, attribute

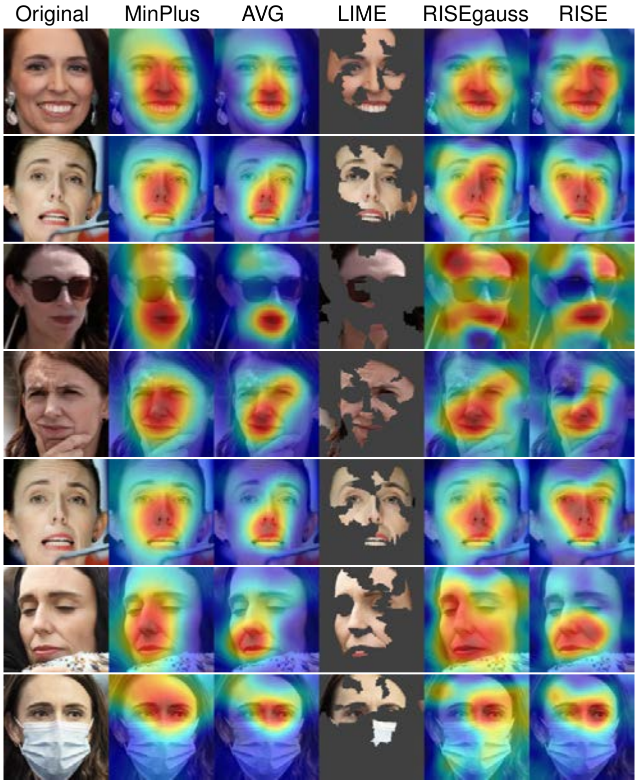

Figure 2. Explainability for face verification using ArcFace. In this case, the verification is performed for each original face image (called $\mathbf{X}$) against the first original face (called $\mathbf{Y}$).

recognition or detection. The experiments have been implemented in Python using Google Colab[2].

As other state-of-the-art methods, we tested AVG for face verification [16]; and 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), RISEgauss (adaption of RISE algorithm with Gaussian masks)[3] in all experiments.

3.1. Face Verification

In these experiments, we show how are the saliency maps in face verification using ArcFace [8] for different kind of faces (with different expressions and occlusions). The results are shown in Figure 2. We can observe that MinPlus achieves stable saliency maps focused on relevant parts of the faces. It is worthwhile to mention that visually, MinPlus shows better results than AVG, because the saliency maps covers the relevant part of the face more homogeneously. In the next experiments, we decided not to include AVG because it was originally developed for face verification, and because MinPlus achieves a better performance.

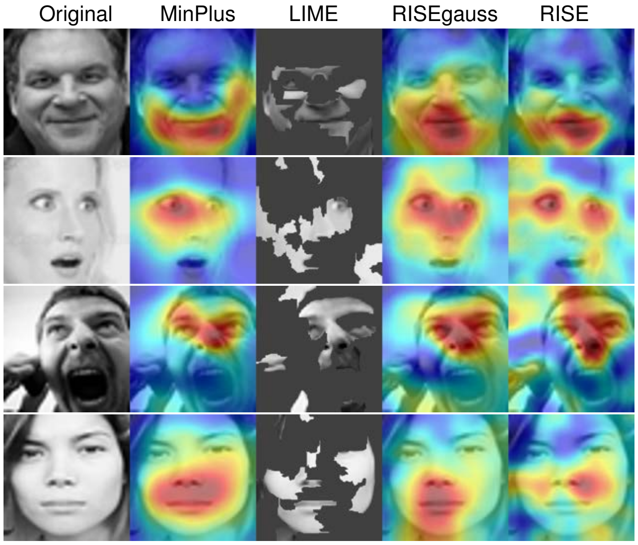

Figure 3. Explainability for recognition of expressions using Xception. The figure shows the performance on four expressions (one per row): happiness, surprise, angry and neutral respectively.

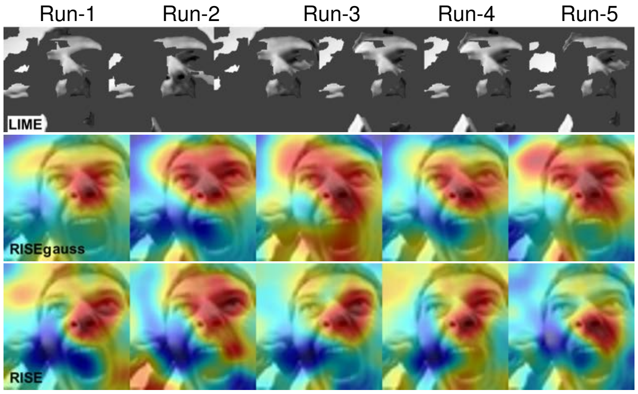

Figure 4. Five different runs of LIME, RISEgauss, RISE for the explanation of the recognition of one expression (see original face in the row that corresponds to 'angry' in Figure 3). Since the random nature of these procedures, the result of each run is very different. In these experiments, each run is the average of 500 random masks.

3.2. Recognition of Facial Expressions

In this experiments, we tested the face expression recognition model based on Xception architecture [2] that is able to recognize the seven universal face expressions: angry, disgust, scared, happy, sad, surprised and neutral. For this

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

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

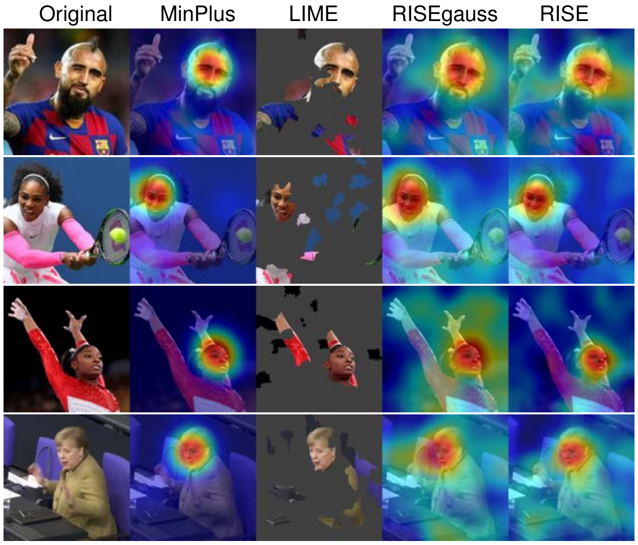

Figure 5. Explainability for face detection using MTCNN.

task we selected some images of FER-2013 dataset[4]. The results are shown in Figure 3. We present the saliency maps for five expressions: happy, surprise, angry and neutral. We observe the most important regions of the face are that highlighted by each method. MinPlus offers stable saliency maps because it is not random in nature like the other ones as shown in Figure 4. We could avoid this random problem by averaging the saliency maps of multiple executions, however, that would substantially increase the execution time of each algorithm. In our experiments, the execution time of each of these methods is approximately 3 minutes. A satisfactory result would bring the run to more than 30 minutes. On the other hand, MinPlus requires 4:30 minutes. In this comparison, the unstable response of LIME and RISE is a disadvantage.

3.3. Face Detection

We evaluate the MTCNN[5] approach for face detection [32]. In the experiments, we tested the performance of the saliency methods in several face images in different scenarios, poses and expressions. The results are shown in Figure 5. We observe how compact is the result of MinPlus, where the saliency map are focused on the faces only.

3.4. Masked Face Detection

With the COVID-19 pandemic, the performance in face detection and face localization has decreased due to the use of face masks [17]. We consider in this Section, how our approach can be used to evaluate a face detection algorithm in faces with and without masks. We trained a YOLOv5s[6] model, to detect three classes of faces: with no-mask, with mask (correctly worn) and , with mask (incorrectly worn)[7]. In this experiment, we evaluated the detection of the second class defining 'f_x' as a detection function (see Section 2.1). Some results are shown in Figure 6.

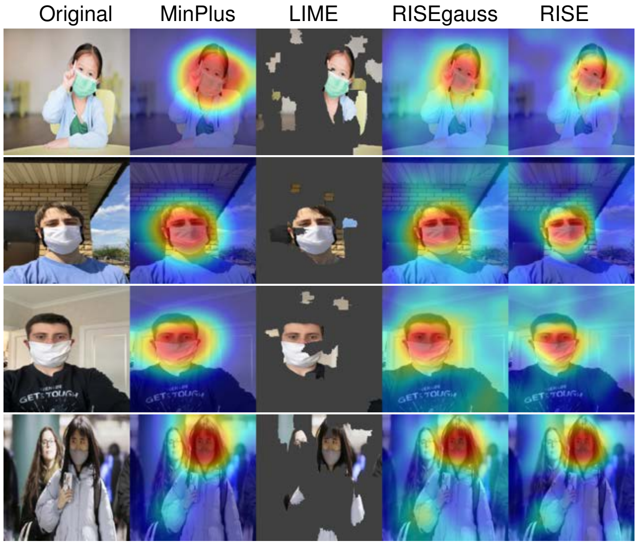

Figure 6. Explainability for detection of faces with masks using YOLOv5s.

In order to evaluate objectively the performance of the saliency maps, we developed a new strategy (as shown in the right figure in Figure 7). We compare the binary image defined by the detected bounding box (blue rectangle) with the binary image defined by thresholding the saliency map (orange shape). Thus, for different thresholds we have different precision-recall values that can be computed from the true positive, false positive and false negative pixels. The area under the precision-recall curve (AP) is presented in Table 1 for the twenty images of this class in the testing subset. The higher the AP value the better the compactness of the saliency map. This is achieved by our method MinPlus.

3.5. Discussion

MinPlus achieves saliency maps that are stable, very focused and interpretable to humans. Our method shows promising results in comparison with other state-of-the-art methods because it is not random in nature like the others. To illustrate the random effect in LIME and RISE, we run the methods five times and show the variation of the results in Figure 4. This phenomenon is not present in MinPlus, because there is no random component. The experiments

[4] See https://www.kaggle.com/msambare/fer2013.

[5] See https://github.com/kpzhang93/MTCNN_face_detection_alignment.

[6] See https://github.com/ultralytics/yolov5.

[7] The dataset was exported via roboflow.ai. It includes 2033 images: 87.5% for training, 8.5% for validation and 4% for testing.

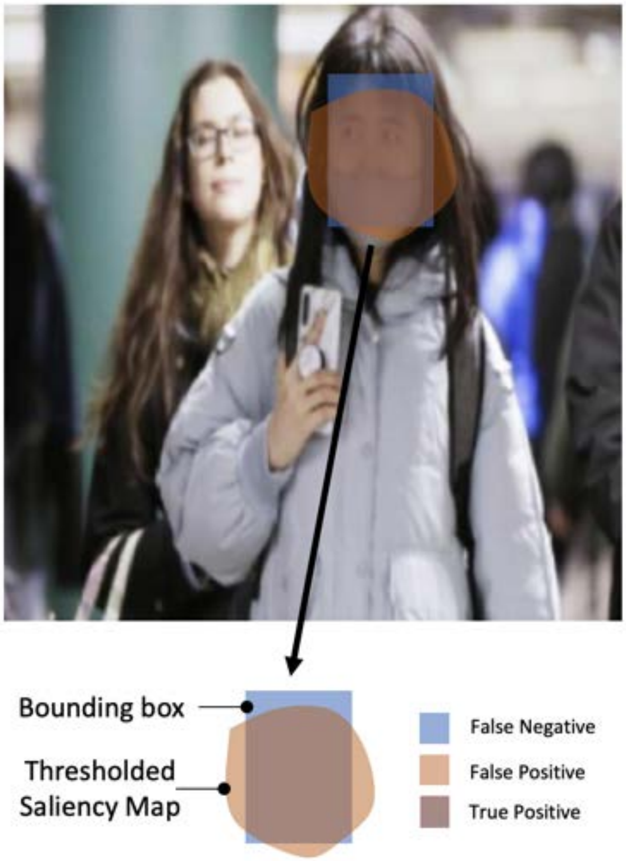

Figure 7. Definition of false negative, false positive and true positive using YOLOv5s.

Table 1. AP in masked face detection using YOLOv5s

Image	MinPlus	LIME-map	RISEgauss	RISE
01	0.93	0.77	0.84	0.92
02	0.94	0.85	0.94	0.92
03	0.93	0.74	0.93	0.93
04	0.92	0.76	0.77	0.52
05	0.80	0.50	0.79	0.74
06	0.90	0.44	0.92	0.83
07	0.86	0.64	0.82	0.88
08	0.95	0.56	0.95	0.82
09	0.91	0.59	0.88	0.83
10	0.78	0.69	0.73	0.83
11	0.91	0.59	0.85	0.84
12	0.97	0.55	0.95	0.72
13	0.90	0.87	0.85	0.85
14	0.84	0.81	0.91	0.94
15	0.97	0.84	0.92	0.93
16	0.88	0.71	0.86	0.82
17	0.90	0.43	0.83	0.88
18	0.91	0.75	0.85	0.85
19	0.85	0.69	0.87	0.85
20	0.83	0.64	0.82	0.77
Mean	0.90	0.67	0.86	0.83

show that MinPlus can be used to evaluate any facial analysis approach as a black-box with a known recognition function ‘f_x’.

4. Conclusions

We presented MinPlus, a saliency map that can be used to explain any facial analysis algorithm. In this method, we only need the input-output function of the black-box ‘f_x’. The key idea is based on how the probability of recognition of a given image changes when it is perturbed. MinPlus removes and aggregates different parts of the image, and measure contributions of these parts individually and in-collaboration as well. We tested and compared our method in four different scenarios: face verification, expression recognition, face detection, and masked face recognition. The results show saliency maps focused on relevant parts. This paper presents a qualitative explanation of any facial analysis approach, in which it can be clearly appreciated which are the most relevant 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 recognition systems as well. We conclude that MinPlus achieves saliency maps that are stable and interpretable to humans. In addition, our method shows promising results in comparison with other state-of-the-art methods like AVG, LIME and RISE. This paper presents good insights into any facial analysis approach. It can be used to highlight the most relevant areas that an algorithm takes into account to carry out the recognition process.

Acknowledgement

This work was supported by Fondecyt Grant No. 1191131 from ANID–Chile and National Center for Artificial Intelligence CENIA FB210017, Basal ANID–Chile.

References

[1] 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.

[2] Octavio Arriaga, Matias Valdenegro-Toro, and Paul Plöger. Real-time convolutional neural networks for emotion and gender classification. *arXiv preprint arXiv:1710.07557*, 2017.

[3] 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.

[4] Manuele Bicego, Enrico Grosso, Andrea Lagorio, Gavin Brelstaff, Linda Brodo, and Massimo Tistarelli. Distinctive-

ness of faces: a computational approach. *ACM Transactions on Applied Perception (TAP)*, 5(2):1–18, 2008.

[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] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] Sunbin Kim and Hyeoncheol Kim. Deep explanation model for facial expression recognition through facial action coding unit. In *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 1–4. IEEE, 2019.

[13] 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.

[14] 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.

[15] 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.

[16] Domingo Mery and Bernardita Morris. On black-box explanation for face verification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3418–3427, 2022.

[17] Mei Ngan, Patrick Grother, and Kayee Hanaoka. Ongoing face recognition vendor test (frvt): Part 6b: Face recognition accuracy with face masks using post-covid-19 algorithms. 2022.

[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] 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.

[25] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

[26] 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.

[27] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1568–1576, 2017.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

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

[33] 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.

[34] 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.