Pattern Recognition Letters 192 (2025) 136–143

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Extended excitation backprop with gradient weighting: A general visualization solution for understanding heterogeneous face recognition

Yan Wang [a,1], Sivapriyaa Kannappan [a,1], Fangliang Bai [a,1], Stuart Gibson [b,a,*], Christopher Solomon [a]

[a] VisionMetric Ltd., UK, Unit 24-25, University Rd, Canterbury, CT2 7FG, United Kingdom
[b] University of Kent, Giles Ln, Canterbury, CT2 7NZ, United Kingdom

ARTICLE INFO

Editor: Ajay Kumar

Keywords:
Visualization
Evaluation
Metric learning
Embedding networks
Heterogeneous face recognition

ABSTRACT

Visualization methods have been used to reveal areas of images which influence the decision making of machine learning models, thereby helping to understand and diagnose the learned models and suggest ways to improve their performance. This concept is termed Explainable Artificial Intelligence. In this work, we focus on visualization methods for metric-learning based neural networks. We propose a gradient-weighted extended Excitation Back-Propagation (gweEBP) method that integrates the gradient information during its backpropagation for the accurate investigation of embedding networks. We perform an extensive evaluation of our gweEBP, and seven other visualization methods, on two neural networks, trained for heterogeneous face recognition. The evaluation is performed over two publicly available cross-modality datasets using two evaluation methods termed the "hiding game" and the "inpainting game". Our experiments showed that the proposed method outperforms the competing methods in both games in most cases. Additionally, our comprehensive study also provides a benchmark for comparing visualization techniques, which may help other researchers develop new techniques and perform comparative studies on them.

1. Introduction

Recently, deep neural networks (DNN) have produced significant performance improvements in the cross-modality domain [1–4]. Since they are usually regarded as black-boxes when deployed, it is difficult to understand how and why the networks make certain decisions, giving rise to questions such as: Can we trust the model? Does it focus on the foreground features or simply use irrelevant background information for matching decisions? Visualization techniques can help to identify significant areas of the input image which contribute to the decision. In various literature, they are also called attribution methods, a branch of Explainable AI.

Visualization methods in deep learning were originally proposed for visualizing DNNs for classification of which the output is the predicted scores as per classes. They usually produce a saliency map that highlights the attention of the neural network to local regions of the input image that are important for predicting a certain class label. There are in general two types of visualization methods in deep learning: white-box methods and black-box (model-agnostic) ones. The white-box methods have access to the parameters of and feature maps from the internal layers of the network, whilst the black-box methods, without knowledge about the network's internal structure and parameters, can only access the output of the network.

White-box methods involve back-propagation of the output of deep classifiers or its derivatives with respect to feature maps in a top-down fashion through layers [5–7]. Bach et al. [5] proposed pixel-wise decomposition of the prediction of classifiers into relevance scores and propagate them through layers in order to visualize contributions of image pixels to the prediction. Class Attention Mapping (CAM) based methods [8–11] rely on gradient back-propagation of the classification score w.r.t. the feature map from the final convolution layer of the classifier. RISE [12] is a black-box method which randomly masks the input image and generates corresponding classification scores for the class of interest. A saliency map is obtained by summing these masks weighted by the scores. In [13], Concept Relevance Propagation shows not only where the salient pixels are but what concepts they are related to.

The above methods cannot be directly applied to metric-learning based networks trained for face verification, as these networks learn to

* Corresponding author at: University of Kent, Giles Ln, Canterbury, CT2 7NZ, United Kingdom.
E-mail addresses: yan.wang@visionmetric.com (Y. Wang), siva.kannappan@visionmetric.com (S. Kannappan), fangliang.bai@visionmetric.com (F. Bai), s.j.gibson@kent.ac.uk (S. Gibson), chris.solomon@visionmetric.com (C. Solomon).
[1] The first three authors have equal contributions.

https://doi.org/10.1016/j.patrec.2025.03.032
Received 20 February 2023; Received in revised form 19 February 2025; Accepted 22 March 2025
Available online 1 April 2025

map the input data into a latent space and output an embedding vector instead of per-class scores. The similarity between two samples can thus be measured by their distance in the latent space. We call such networks "embedding networks". Later works for explaining the decisions of embedding networks [14–16] make use of triplets, which consists of a probe, a mate (a ground-truth match to the probe) and a non-mate image (a mismatch to the probe). Given a triplet, a visualization technique highlights the regions of the probe with high excitation, indicating that features extracted from the regions are more similar to the mate than the non-mate. This explains why the embedding network favours the mate as a match over the non-mate in terms of similarities. Zhang et al. [17] proposed a back-propagation based approach called Excitation Back-Propagation (EBP) and its variant cEBP, which incorporates a probabilistic Winner-Take-All (WTA) process. Later Castanon et al. [18] proposed tcEBP to highlight regions that are both significant and unique to a particular subject compared with other identities. Williford et al. [19] proposed two methods Subtree EBP and DISE. They also defined an evaluation protocol called the "inpainting game" for comparing EBP based methods. Zhong et al. [20] investigated features computed by neurons using deep visualization techniques by exploring characters of features like diversity, invariance and discrimination and they found that high-level features resemble complex face attributes. In [14], the Grad-CAM method is adapted for embedding networks by replacing the classification score with the triplet loss and uses the gradients of the loss to calculate weights of feature maps to explain the network's decisions. Stylianou et al. [15] adopted pairwise similarities between feature maps from the last convolutional layer to understand the similarity metric learnt by embedding networks. Based on their work, Zhu et al. [16] produced the point-specific activation map for fine-grained visual explanation of metric learning. More recently, Knoche et al. [21] introduced confidence scores and X-Maps for explainable face recognition systems where the input facial image pair is systematically occluded and a similarity map is derived from cosine similarity between the embedding vectors of the occluded pair. Lu et al. [22] adapted RISE for explanation of face verification where similarity maps are derived from Pearson correlation between cosine similarity scores and random masks. However, these black-box methods are computationally costly, because a large amount of masks are required for computing similarity maps. A summary of the existing visualization approaches for embedding networks is provided in Table 1 in the supplementary material.

Although there has been extensive research on explainable machine learning, evaluation of visualization methods remains challenging [23]. In [14,15,17] the evaluation of the proposed visualization methods is subjective, which is based on their capability of object localization. However, localization is just an alternative to human explanation and may not precisely portray the model's decision. Zhang et al. [17] proposed a pointing game evaluation where a hit is counted when the maximum attention lies inside the human-annotated bounding box of the object in the image, whilst others [14,24] used the mean fraction of the attention inside the bounding box or segmentation mask as the evaluation metric.

In contrast, Petsiuk et al. [25] proposed two objective evaluation metrics which are not only free from human bias but also save time and resources: deletion and insertion metrics. The intuition behind the "deletion metric" is that removing significant pixels highlighted by the correctly visualized network attention should decrease the probability of the predicted class with a lower Area under the ROC Curve (AUC). On the other hand, inserting significant pixels should increase the probability resulting in higher AUC. Alternatively, Zhu et al. [16] used cosine similarity and showed empirically that it works well for visual explanation.

Our interest in this study is to adapt and evaluate visualization techniques for embedding networks for cross-modality face matching. Here we use the specific example of matching facial composites to real photographs. A facial composite is a pictorial likeness to a suspect's face based on an eyewitness' description [26,27]. Since human faces bear similar structures and all face images in our study have been aligned, the location based evaluation is not relevant to our study. Hence, we intend to evaluate the visualization techniques over two performance metrics: the "hiding game" [18] and the "inpainting game" [19]. The main contributions of this study can be summarized as:

1. Proposing gradient-weighted extended EBP (gweEBP) for embedding networks which takes into account negative neuron responses and integrates gradient information.
2. Evaluating visualization methods via two evaluation metrics for VGG16 [28] and LCNN9 [29] networks trained for cross-modality face recognition.
3. The observation that the inpainting game is a more reliable metric than the hiding game.

The rest of this paper is organized as follows: Section 2 details our proposed gradient-weighted extended EBP based method for embedding networks; Section 3 describes the performance metrics we use in this study; Section 4 presents intensive evaluations of several visualization techniques using two publicly available datasets and finally we draw conclusions and indicate future work in Section 5.

2. Gradient-weighted extended EBP

In this section, we firstly review the original EBP algorithm and then detail our method which is motivated by the incapability of the original EBP to handle negative activation properly in the neural network and the importance of gradient information for highlighting salient features contributing to the network's decisions.

2.1. Preliminaries

EBP relies on a probabilistic Winner-Take-All process to highlight the response of neurons at each CNN layer which contribute most to the network's prediction. It back-propagates marginal winning probabilities (MWP) of neurons through all layers in a top-down fashion based on which saliency maps are generated. EBP works with two assumptions [17]: (1) The response of activated neurons must be non-negative, and (2) The response of activated neurons is positively correlated to the prediction. Let L_{n-1} and L_n be the top and bottom layers respectively. Given a neuron a_i at the top layer and its probability $P(a_i)$, our aim is to calculate $P(a_j)$ where a_j is a child neuron of a_i at the bottom layer. The weight of their connection is denoted as w_{ji}. For a_i and a_j being positive, the excitation is passed down through excitatory connections via the conditional winning probability as

$$P(a_j|a_i) = \mathbf{1}_{w_{ji}\geq 0} Z_i \hat{a}_j w_{ji} \tag{1}$$

$Z_i = 1/\sum_{j:w_{ji}\geq 0} \hat{a}_j w_{ji}$ is the normalization factor so that $\sum_{a_j \in C_i} P(a_j|a_i) = 1$ where $\hat{a}_j$ is the response of a_j and C_i is the child neuron set of a_i. $\mathbf{1}_*$ is the indicator function. Considering the parent neuron set $\mathcal{P}_j$ of a_j, we obtain the MWP of a_j as

$$P(a_j) = \sum_{a_i \in \mathcal{P}_j} P(a_j|a_i) P(a_i) \tag{2}$$

We convert the embedding network to a binary classifier for EBP to work on it. Given a triplet that contains a sketch sample as the probe, a photo of the same identity as the mate, and a photo of a different identity as the non-mate, we add a fully connected (FC) layer fc_2 on top of the last FC layer fc_1 of the network of which the weight matrix is set by stacking the mate and non-mate embedding. Taking the probe as input, fc_2 will output class scores which are indeed the probe-mate and probe-non-mate cosine similarities. The aforementioned assumptions restrict EBP to be only applied to CNNs with non-negative neuron responses. For brevity, we term neurons giving positive/negative responses as pos/neg-neurons. A naive workaround

is to ignore neg-neurons and negative weights, or subtract the lower bound of the activation function [17], but doing so will exclude important negative features. Instead, we extend EBP to consider neg-neurons, so that the EBP assumptions can be ignored.

2.2. Extended EBP

Let Θ^+_{n-1} be the pos-neuron set and Θ^-_{n-1} the neg-neuron set at the top layer. Θ^+_n and Θ^-_n are defined similarly for the bottom layer. If a_i is negative, there are two cases where the connection is excitatory: either a_j is negative and w_{ji} positive or a_j is positive and w_{ji} negative, since in these cases a_j contributes to the negative excitation of a_i. Likewise, if a_i is positive, either a positive a_j and a positive weight or a negative a_j and a negative weight will make the connection excitatory. Therefore, the top-down signal from a_j to a_i is composed of two separate streams: one from pos-neurons Θ^+_{n-1} and the other from neg-neurons Θ^-_{n-1}, leading to two components of MWP for a_j as

$$\begin{aligned} P^+(a_j) &= \sum_{a_i \in \mathcal{P}_j \cap \Theta^+_{n-1}} P(a_j|a_i)P(a_i) \\ &= \mathbf{1}_{\hat{a}_j \geq 0} \sum_{a_i \in \mathcal{P}_j \cap \Theta^+_{n-1}} Z^+_i w^+_{ji} \hat{a}_j P(a_i) + \\ &\quad \mathbf{1}_{\hat{a}_j < 0} \sum_{a_i \in \mathcal{P}_j \cap \Theta^+_{n-1}} Z^+_i w^-_{ji} \hat{a}_j P(a_i) \end{aligned} \tag{3}$$

$$\begin{aligned} P^-(a_j) &= \sum_{a_i \in \mathcal{P}_j \cap \Theta^-_{n-1}} P(a_j|a_i)P(a_i) \\ &= \mathbf{1}_{\hat{a}_j \geq 0} \sum_{a_i \in \mathcal{P}_j \cap \Theta^-_{n-1}} Z^-_i w^-_{ji} \hat{a}_j P(a_i) + \\ &\quad \mathbf{1}_{\hat{a}_j < 0} \sum_{a_i \in \mathcal{P}_j \cap \Theta^-_{n-1}} Z^-_i w^+_{ji} \hat{a}_j P(a_i) \end{aligned} \tag{4}$$

where $Z^+_i = 1/(\sum_{a_j \in C_i \cap \Theta^+_n} w^+_{ji}\hat{a}_j + \sum_{a_j \in C_i \cap \Theta^-_n} w^-_{ji}\hat{a}_j)$, $w^+_{ji} = \max\{0, w_{ji}\}$, $w^-_{ji} = \min\{0, w_{ji}\}$ and $Z^-_i = 1/(\sum_{a_j \in C_i \cap \Theta^+_n} w^-_{ji}a_j + \sum_{a_j \in C_i \cap \Theta^-_n} w^+_{ji}a_j)$. The sum of Eq. (3) and (4) gives the final MWP $P(a_j)$.

2.3. Gradient weighting

We introduce another essential improvement which integrates gradients of triplet gains into the back-propagation of the MWP. We firstly define the triplet gain as:

$$G_{tri}(a, p, n) := max(0, \|a - n\| - \|a - p\|) \tag{5}$$

where a, p and n represent the embedding of the anchor (probe), mate and non-mate image respectively. We then compute its gradient w.r.t. neuron responses for every layer. The neuron responses are weighted using their corresponding gradients before computing MWP as follows:

$$\tilde{a}_j = \mathbf{1}_{\hat{a}_j \geq 0}\hat{a}_j \left[\frac{\partial G_{tri}}{\partial a_j}\right]^+ + \mathbf{1}_{\hat{a}_j < 0}\hat{a}_j \left[-\frac{\partial G_{tri}}{\partial a_j}\right]^+ \tag{6}$$

$\frac{\partial G_{tri}}{\partial a_j}$ is the triplet gain gradient and $[*]^+ = \max\{0, *\}$. Finally, we replace the neuron $\hat{a}_j$ with the gradient weighted neuron $\tilde{a}_j$ in Eq. (3) and (4) to compute MWP. The motivation behind Eq. (6) is to promote neurons giving a large magnitude of responses as well as incurring a large magnitude of gradients. Furthermore, Eq. (6) respects the fact that negative neurons contribute to the triplet gain if their gradients are negative. Accordingly, their responses should be scaled by the magnitude of their gradients.

Gradient weighting offers several benefits. It complements the EBP method by boosting the significance of neg/pos-neurons which have negative/positive gradients of a large magnitude and thus contribute to the triplet gain. Conversely, those neurons of which the gradient has a different sign from their responses are suppressed in the computation of MWP, which may remove false salience in the saliency map. Moreover, the gradients contain contrastive information regarding both the mate and non-mate, so that the method possesses similar strength as contrastive EBP. Moreover, the EBP method combines gradient information at various depths of the neural networks, thus reinforcing the salience of image regions that consistently receive strong gradients at various levels. We revise Eq. (3) and (4) similar to [17] and show that our method can be implemented as matrix operations which can take advantage of parallel computing. The matrix formulation can be found in the supplementary material. We summarize the main steps of gweEBP for a CNN layer as Alg. 1 there.

3. Evaluation metrics

3.1. Hiding game

The hiding game is conducted on a set of test triplets for which all the probes are correctly matched to their respective mate. We gradually mask out the image content of the probe with some predefined values, starting from the least to the most significant pixels according to the saliency map obtained from the evaluated visualization method. We then observe the true positive matching rate (TPR) throughout this hiding procedure. It is expected that the TPR should remain high until all the truly significant pixels are hidden. There are various choices for the values to fill in the masked regions of the probe, which can be either zeros, ones, random values or the mean of the image [18]. However, none of these meet our requirements, because they will not necessarily make the probe look more similar to the non-mate. Even after all pixels have been hidden, the TPR may not drop to zero. A more sensible choice is to fill the masked region with the value of corresponding pixels from the non-mate. If the region is important for matching the probe to its mate, doing so will effectively push the probe towards its non-mate in the latent space and eventually force it being classified as non-mate after hiding a sufficient amount of pixels. The earlier the modified probe is matched to the non-mate, the less accurate the visualization method should be.

3.2. Inpainting game

The inpainting game is designed to evaluate visualization techniques quantitatively by synthesizing a non-mate doppelganger from the mate image via inpainting where a specific face part is changed resulting in an identity different from the probe according to the network's prediction. Since the inpainted face part of the non-mate image is the only cause for the identity change, it can be used as the ground truth saliency map for evaluation.

Similarly, a new probe image is also synthesized from the original probe by inpainting the same face part so that its identity matches the inpainted non-mate. Hence, a quadruplet of images is formed containing: the original probe (sketch) image (OP), the original mate (photo) image (OM), the inpainted probe (sketch) image (IP) and the inpainted non-mate (photo) image (IN). They must fulfil the following criteria: (1) The identity of the OP should be similar to that of the OM, subject to a face verification threshold τ. (2) The identity of the IP should be similar to that of the IN according to τ.

Given a set of quadruplets, the saliency map of each OP produced by a visualization technique, and a face verification threshold τ, we run the inpainting game as follows: For each quadruplet, we (1) convert the saliency map into a saliency mask according to a saliency threshold; (2) mask the pixels of the OP using the saliency mask and replace them with that of the IP, giving a blended probe; (3) verify if the blended probe is matched to the IN by the network according to τ. We then calculate the non-mate classification successful rate (SR) as the proportion of blended probes matched to their respective IN in all quadruplets. We identify the masked pixels not belonging to the inpainted facial region as false positives (FPs) and calculate the false alarm rate (FAR) as the proportion of FPs in the total amount of pixels outside the inpainted region. Finally, we repeat this procedure for a range of saliency thresholds to obtain a ROC-like performance indicator by plotting SR against FAR. A good visualization method should have high SR at low FAR.

4. Experimental results

We compare our method gweEBP with Grad-CAM, EBP, cEBP, tcEBP, Pairwise Similarity Map (PairwiseSIM) [15], XFace [21] and CorrRISE [22] using the two aforementioned metrics. Grad-CAM is a popular visualization technique for analysing embedding networks by back-propogating gradients of class scores or losses. Similar to gweEBP, it requires access to the internal neuron responses (i.e., feature maps) and gradients. PairwiseSIM is readily applicable to embedding networks which requires access to feature maps from the last convolutional layer (usually after pooling). XFace and CorrRISE are the most recent visualization methods for explaining face verification results. To obtain general results, we performed deep learning tasks on two publicly available cross-modality face datasets namely the University of Malta Software Generated Face-Sketch (UoM-SGFS) dataset [30] and the Chinese University of Hong Kong (CUHK) face sketch dataset [31]. For each dataset, we fine-tune two pretrained CNNs: VGG16 and LCNN9. In VGG16, the ReLU activation function is used, prohibiting negative responses from all convolutional layers, whilst LCNN9 allows negative responses. Thus, VGG16 satisfies the EBP assumptions whilst LCNN9 does not.

4.1. Datasets

4.1.1. UoM-SGFS dataset

The UoM-SGFS database contains 1200 software generated face sketches of 600 subjects selected from the Color FERET database [32]. There are two sets: Set A contains sketches created using the EFIT-V software [26] and Set B an improved version of Set A with more realistic skin effects. For experiments, we use Set A containing 598 identities after face alignment via the DLIB face aligner [33].

4.1.2. CUHK dataset

The CUHK sketch dataset contains hand-drawn face sketches and photos of 606 identities. For each identity, there is one sketch drawn by an artist based on a photo featuring a frontal face with a neutral expression under normal lighting conditions. For evaluation, we use image data of 188 identities from the CUHK dataset.

4.2. Training and implementation details

4.2.1. Embedding networks

For VGG16, we removed its original FC layers and add a new one producing embeddings of 1024 dimensions. For LCNN9, we retain its first FC layer to output embeddings of 256 dimensions. For both datasets, we used 75% for training (20% of which for validation) and the remaining 25% for testing. Both networks were trained using triplet losses upon triplets obtained from mini-batches of 32 samples (16 identities, one sketch and one photo per identity). For testing, we constructed galleries containing 1871 and 2053 subjects for UoM-SGFS and CUHK respectively. We achieved 50.43% and 98.90% rank-1 accuracies on UoM-SGFS and CUHK respectively with VGG16 and 57.26% and 98.35% on UoM-SGFS and CUHK respectively with LCNN9.

4.2.2. Grad-CAM

Our implementation is based on [14] which originally propagates the triplet loss gradient [34] back to the last convolutional layer of the network. A saliency map is then generated based on a weighted sum of the feature map channels. However, using the gradient of the triplet loss is problematic. In Grad-CAM, only positive gradients are concerned. Accordingly, the grad-weights should correlate positively to an objective function such as the class-score [8], which renders the triplet loss inadequate. Besides, a well-trained embedding network often gives zero triplet losses resulting in zero gradients and empty saliency maps. Therefore, we propose to use the gradients of triplet gains (Eq. (5)) to compute grad-weights. Similar to [8], we kept the top 50 largest grad-weights and multiply them with their corresponding channels of the feature maps from the last convolutional layer. The saliency map has a size of 16×16 pixels for LCNN9 and 14×14 pixels for VGG16.

4.2.3. Pairwise similarity map

The similarity maps were computed based on the feature map from the last convolutional layer which has the same size as the saliency maps computed using Grad-CAM.

4.2.4. EBP based methods

Our implementation of all the EBP based methods is based on [19]. We generated saliency maps based on feature maps extracted at deep layers of the networks. This is because visualization of image features at semantic level makes more sense for attribution purposes than lower-level features and the image features extracted at deep layers tend to be semantic. However, back-propagating MWP to shallow layers may improve the precision of localizing contributing image features. Therefore, we decided to back-propagate the MWP up to an intermediate layer of the embedding network (i.e., the sixth/eleventh convolutional layer of LCNN9/VGG16), where the extracted features are still semantic and the computed saliency map has the same size as those produced using Grad-CAM and PairwiseSIM (i.e., 16×16 pixels for LCNN9 and 14×14 pixels for VGG16). For tcEBP, the truncation rate is set to 20%.

4.2.5. XFace and CorrRISE

We adopt the official implementation of XFace to compute saliency maps. To use it in contrastive settings, we subtract the saliency map of the probe compared with the non-mate from that of the probe compared with the mate. The resultant contrastive saliency map highlights the region of the probe image due to which the embedding network predicts the probe's identity as the mate's rather than the non-mate's. For CorrRISE, we implemented it based on the official implementation of RISE and generated contrastive saliency maps similarly as XFace.

4.3. Hiding game evaluation

4.3.1. UoM-SGFS

Our UoM-SGFS test data for the hiding game contains 59 and 67 Set-A sketches for VGG16 and LCNN9 respectively, satisfying the condition that their rank-1 match from the gallery is their corresponding ground truth photo. We use their rank-1 match as mates and the rank-2 match from the gallery as non-mates. Without hiding any pixels, we have 100% TPR initially. Fig. 1 shows the hiding game results of eight visualization techniques for both VGG16 and LCNN9. For both networks, the top-3 best methods are gweEBP, xFace and CorrRISE, whilst gweEBP performs slightly better than the other two for VGG16 and xFace and CorrRISE are better than gweEBP for LCNN9. A visual comparison of the saliency maps from the competing techniques is presented in the supplementary material. We will discuss about the limitation of hiding games in Section 4.6.

4.3.2. CUHK

We prepared CUHK test data containing 28 triplets for both VGG16 and LCNN9. The hiding game results from all the compared visualization techniques for both VGG16 & LCNN9 are shown in Fig. 2. We also include the saliency maps for a few triplets in the supplementary material. The gweEBP method is superior over the others (Fig. 2(a)). The second best method for VGG16 and LCNN9 is CorrRISE and EBP, respectively. (Fig. 2(b)).

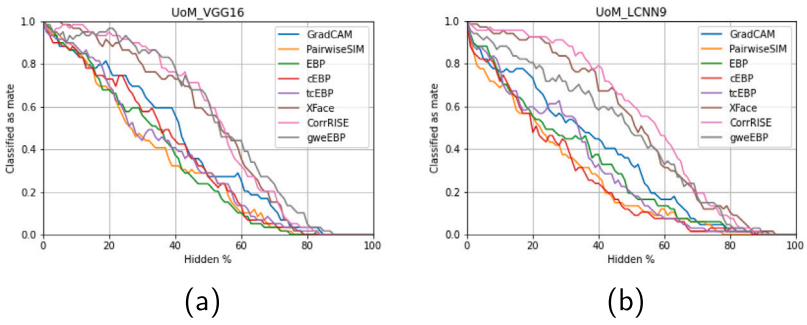

Fig. 1. Hiding game results on UoM-SGFS. (a) VGG16. (b) LCNN9.

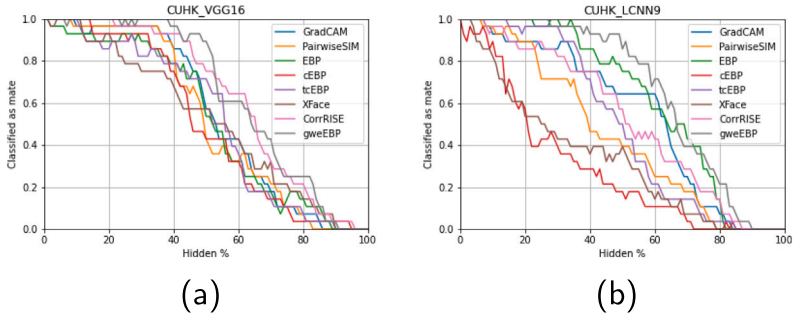

Fig. 2. Hiding game results on CUHK. (a) VGG16. (b) LCNN9.

4.4. Inpainting game evaluation

4.4.1. UoM-SGFS

We generated the inpainted UoM-SGFS dataset for the inpainting game with sketch and photo pairs of 598 identities after face alignment [33]. We defined six facial regions for inpainting: hair, eyebrows, eyes, nose, mouth, and face skin. A pre-trained CelebAMask-HQ [35] face parser[2] was deployed to identify these facial regions of the image samples which were then saved as facial part masks. Following the same way of preparing inpainted data as [19], we used the pluralistic inpainting network [36] to synthesize various facial parts of both sketches and photos, which produced six quadruplets (OP, OM, IP, IN for six facial regions) per identity. Examples are provided in Fig. 1 in the supplementary material. Qualified quadruplets were then selected using the following criteria based on the L2 distance between their embeddings:

$$\|OP - OM\| \leq \tau \text{ and } \|OP - IN\| > \tau \tag{7}$$

$$\|IP - IN\| \leq \tau \text{ and } \|IP - OM\| > \tau \tag{8}$$

where τ is the face verification threshold. It is set to $0.14@FPR = 0.039$ and $0.245@FPR = 0.0465$ according to the ROC curve of VGG16 and LCNN9 respectively. These requirements ensure that the synthesized face parts by the inpainting network have changed the identity of the original sketch and photo pair, whilst the identity of the inpainted sketch still matches that of its paired inpainted photo. Consequently, there are 308 qualified quadruplets for VGG16 and 273 quadruplets for LCNN9 respectively. We present the ROC curves from the inpainting game in Fig. 3 showing that the gweEBP method outperforms the other methods by a big margin regardless of networks and datasets.

[2] https://github.com/zllrunning/face-parsing.PyTorch

4.4.2. CUHK

Similar to UoM-SGFS, there are six regions accounting for various face parts subject to inpainting. However, unlike UoM-SGFS, we face two difficulties in creating the inpainted images: (1) As the hand-drawn sketches are only line drawings and lack rich texture, the face segmentation network does not work properly on them; (2) The face inpainting network does not work properly on them for the same reason. To circumvent these problems, we trained a CycleGAN model [37] on the CUHK dataset and used it to convert all sketches to photo-realistic face images. We then parsed the synthesized mages to obtain masks of face parts for each sketch. To inpaint a sketch, we firstly inpainted various face parts of its mate photo and converted the inpainted photos into fake sketches using CycleGAN. We then replaced various face parts of the sketch with their counterparts cut from the fake sketches, resulting in a set of natural-looking inpainted sketches. We show some examples synthesized using CycleGAN in Fig. 2 in the supplementary material. For VGG16, we generated 58 quadruplets that satisfy Eq. (7) and (8), where τ is set to $0.06@FPR = 0$ estimated from the Receiver Operating Characteristic (ROC) curve of VGG16. For LCNN9, 67 qualified quadruplets were generated. τ is set to $0.645@FPR = 0.0434$ estimated from the ROC of LCNN9. As shown in Fig. 4, gweEBP performs the best on both the networks. In the supplementary material, we include visualization of saliency maps obtained from the compared methods on a set of typical test examples from both datasets for the respective networks in Figure 7-10. We observed that the gweEBP is often more accurate than the other competitors regardless of datasets and networks.

4.5. Ablation study

We show the capability of the extended EBP methods (without gradient-weighting) for capturing important negative responses by comparing them with original EBP methods via inpainting game evaluation. We implemented the extended versions of EBP, cEBP and tcEBP denoted as eEBP, ecEBP and etcEBP, respectively. In order for original EBP methods to work on LCNN9, we simply ignore all negative responses when back-propagating MWP. As shown in Fig. 5 and

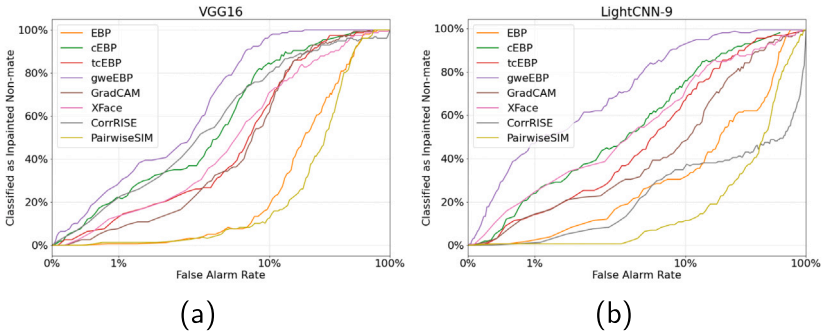

Fig. 3. Inpainting game evaluation on UoM-SGFS: ROC curves. (a) VGG16. (b) LCNN9.

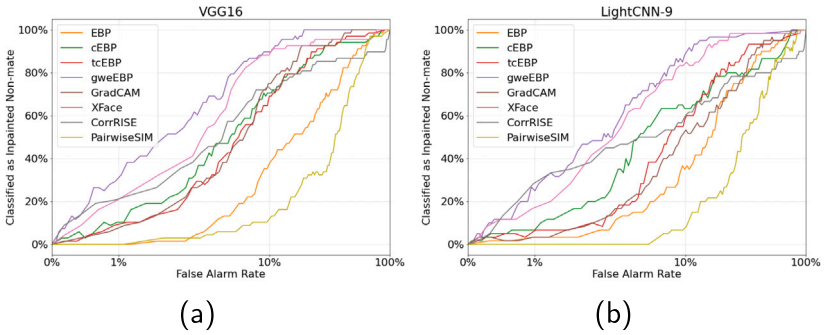

Fig. 4. Inpainting game evaluation on CUHK: ROC curves. (a) VGG16. (b) LCNN9.

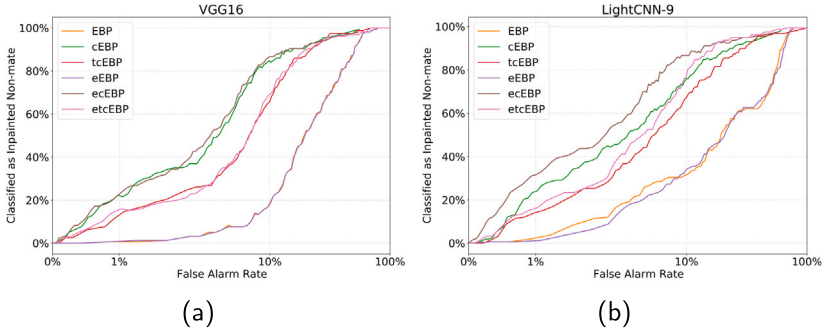

Fig. 5. Comparing the extended EBP and original methods using the inpainting game and UoM-SGFS: ROC curves. (a) VGG16. (b) LCNN9.

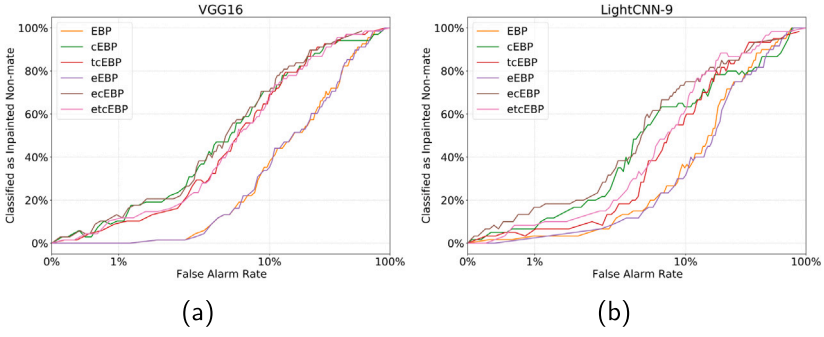

Fig. 6. Comparing the extended EBP and original methods using the inpainting game and CUHK: ROC curves. (a) VGG16. (b) LCNN9.

Fig. 6, the ecEBP and etcEBP outperform their respective counterparts cEBP and tcEBP for LCNN9 on both datasets. However, no noticeable improvements are observed for VGG16. This is because the ReLU activations have suppressed negative responses throughout VGG16.

In contrast, the Max-Feature-Map (MFM) activation in LCNN9 allows negative responses to propagate through all layers. Hence the contribution of negative responses to the classification scores is significant. Therefore, the extended EBP methods can identify significant negative features contributing to the network's decision.

4.6. Discussions

As revealed in both games, gweEBP is the best performing method. It integrates gradients at multiple layers, a major factor we believe for it to surpass Grad-CAM. Moreover, Grad-CAM performs global average pooling of the gradients across the dimensions of the feature maps to compute grad-weights, therefore losing the strength for differentiating features with respect to their individual contributions to the network's decision. EBP and PairwiseSIM perform poorly in general, since they do not take into account the dissimilarities between the probe and the non-mate.

The hiding game is easier to implement than the inpainting game, since the former does not require preparation of image quadruplets using inpainting techniques. However, it does not provide verifiable evaluation due to lack of ground truth saliency maps. For instance, according to the hiding game, EBP outperforms Grad-CAM and cEBP as shown in Fig. 2(b). As EBP is not contrastive, its estimated saliency invariably lies in the whole face region of the probe images (see Figure 8 in the supplementary material). Consequently, we found no clues from the saliency maps produced by EBP for explaining why LCNN9 favours the mate over the non-mate. Hence, a method highly ranked by the hiding game may not necessarily be a good one. Moreover, filling the hidden pixels of the probe (a sketch) with pixel values from the non-mate (a photo) leads to images of mixed modality falling outside the distribution of eligible inputs for the HFR models, which could introduce unexpected disturbance to the decision of the models. In contrast, the inpainting game evaluates attribution methods based on ground truth saliency maps. A top-ranked method is truly the best one, since it produces the most consistent saliency map as the ground truth. In addition, the blended probe still conforms to the distribution of eligible inputs (since both the IP and OP are sketches) for the HFR models. Therefore, we regard the inpainting game as a better evaluation protocol than the hiding game in the context of HFR.

5. Conclusions and future work

In this paper, we proposed a gradient weighted EBP-based method for accurately visualizing embedding networks and extended its applicability to CNNs with negative responses. We compared it with other techniques via two evaluation methods: the hiding game and the inpainting game. Our evaluation is based on two networks VGG16 and LCNN9 trained on heterogeneous face recognition and two popular cross-modality datasets UoM-SGFS and CUHK. Results show that our method in general performs the best regardless of the networks and datasets. In the future, we will explore the possibility of applying our method to other network architectures including recurrent or self-attention mechanisms.

CRediT authorship contribution statement

Yan Wang: Writing – original draft, Software, Methodology, Data curation. **Sivapriyaa Kannappan:** Writing – review & editing, Writing – original draft, Validation, Software, Data curation, Conceptualization. **Fangliang Bai:** Writing – review & editing, Writing – original draft, Validation, Software, Investigation, Conceptualization. **Stuart Gibson:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Christopher Solomon:** Writing – review & editing, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patrec.2025.03.032.

Data availability

Data will be made available on request.

References

[1] S. Saxena, J. Verbeek, Heterogeneous face recognition with CNNs, in: European Conference on Computer Vision, Springer, 2016, pp. 483–491.

[2] C. Galea, R.A. Farrugia, Matching software-generated sketches to face photographs with a very deep CNN, morphed faces, and transfer learning, IEEE Trans. Inf. Forensics Secur. 13 (6) (2017) 1421–1431.

[3] C. Peng, N. Wang, J. Li, X. Gao, DLFace: Deep local descriptor for cross-modality face recognition, Pattern Recognit. 90 (2019) 161–171.

[4] S. Yu, H. Han, S. Shan, A. Dantcheva, X. Chen, Improving face sketch recognition via adversarial sketch-photo transformation, in: 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition, IEEE, 2019, pp. 1–8.

[5] S. Lapuschkin, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLoS ONE 10 (2015) e0130140, http://dx.doi.org/10.1371/journal.pone.0130140.

[6] O. Tursun, S. Denman, S. Sridharan, C. Fookes, SESS: Saliency enhancing with scaling and sliding, 2022, arXiv:2207.01769.

[7] S. Gur, A. Ali, L. Wolf, Visualization of supervised and self-supervised neural networks via attribution guided factorization, 2020, URL: http://arxiv.org/abs/2012.02166.

[8] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[9] J.R. Lee, S. Kim, I. Park, T. Eo, D. Hwang, Relevance-CAM: Your model already knows where to look, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 14939–14948, http://dx.doi.org/10.1109/CVPR46437.2021.01470.

[10] H. Li, Z. Li, R. Ma, T. Wu, FD-CAM: Improving faithfulness and discriminability of visual explanation for CNNs, in: 2022 26th International Conference on Pattern Recognition, ICPR, IEEE Computer Society, Los Alamitos, CA, USA, 2022, pp. 1300–1306, http://dx.doi.org/10.1109/ICPR56361.2022.9956466.

[11] Y. Ha, C.-H. Youn, Collection-CAM: A faster region-based saliency method using collection-wise mask over pyramidal features, IEEE Access 10 (2022) 112776–112788, http://dx.doi.org/10.1109/ACCESS.2022.3215534.

[12] V. Petsiuk, A. Das, K. Saenko, RISE: Randomized input sampling for explanation of black-box models, 2018, arXiv:1806.07421.

[13] R. Achtibat, M. Dreyer, I. Eisenbraun, S. Bosse, T. Wiegand, W. Samek, S. Lapuschkin, From attribution maps to human-understandable explanations through concept relevance propagation, Nat. Mach. Intell. 5 (9) (2023) 1006–1019, http://dx.doi.org/10.1038/s42256-023-00711-8.

[14] L. Chen, J. Chen, H. Hajimirsadeghi, G. Mori, Adapting grad-CAM for embedding networks, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 2794–2803.

[15] A. Stylianou, R. Souvenir, R. Pless, Visualizing deep similarity networks, in: 2019 IEEE Winter Conference on Applications of Computer Vision, WACV, IEEE, 2019, pp. 2029–2037.

[16] S. Zhu, T. Yang, C. Chen, Visual explanation for deep metric learning, 2019, arXiv preprint arXiv:1909.12977.

[17] J. Zhang, S.A. Bargal, Z. Lin, J. Brandt, X. Shen, S. Sclaroff, Top-down neural attention by excitation backprop, Int. J. Comput. Vis. 126 (10) (2018) 1084–1102.

[18] G. Castanon, J. Byrne, Visualizing and quantifying discriminative features for face recognition, in: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2018, IEEE, 2018, pp. 16–23.

[19] J.R. Williford, B.B. May, J. Byrne, Explainable face recognition, in: European Conference on Computer Vision, Springer, 2020, pp. 248–263.

[20] Y. Zhong, W. Deng, Exploring features and attributes in deep face recognition using visualization techniques, in: 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2019, IEEE, 2019, pp. 1–8.

[21] M. Knoche, T. Teepe, S. Hörmann, G. Rigoll, Explainable model-agnostic similarity and confidence in face verification, in: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, WACVW, 2023, pp. 1–8, http://dx.doi.org/10.1109/WACVW58289.2023.00078.

[22] Y. Lu, Z. Xu, T. Ebrahimi, Towards visual saliency explanations of face verification, in: 2024 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV, IEEE, 2024, pp. 4714–4723, http://dx.doi.org/10.1109/WACV57701.2024.00466.

[23] F. Poursabzi-Sangdeh, D.G. Goldstein, J.M. Hofman, J.W. Wortman Vaughan, H. Wallach, Manipulating and measuring model interpretability, in: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–52.

[24] R.R. Selvaraju, P. Chattopadhyay, M. Elhoseiny, T. Sharma, D. Batra, D. Parikh, S. Lee, Choose your neuron: Incorporating domain knowledge through neuron-importance, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 526–541.

[25] V. Petsiuk, A. Das, K. Saenko, Rise: Randomized input sampling for explanation of black-box models, 2018, arXiv preprint arXiv:1806.07421.

[26] C.J. Solomon, S.J. Gibson, J.J. Mist, Interactive evolutionary generation of facial composites for locating suspects in criminal investigations, Appl. Soft Comput. 13 (7) (2013) 3298–3306.

[27] S.J. Gibson, C.J. Solomon, A. Pallares-Bejarano, Synthesis of photographic quality facial composites using evolutionary algorithms, in: Proceedings of the British Machine Vision Conference 2003, 2003, pp. 221–230.

[28] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: British Machine Vision Association, 2015.

[29] X. Wu, R. He, Z. Sun, T. Tan, A light cnn for deep face representation with noisy labels, IEEE Trans. Inf. Forensics Secur. 13 (11) (2018) 2884–2896.

[30] C. Galea, R.A. Farrugia, A large-scale software-generated face composite sketch database, in: 2016 International Conference of the Biometrics Special Interest Group, BIOSIG, IEEE, 2016, pp. 1–5.

[31] X. Wang, X. Tang, Face photo-sketch synthesis and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (11) (2008) 1955–1967.

[32] P.J. Phillips, H. Wechsler, J. Huang, P.J. Rauss, The FERET database and evaluation procedure for face-recognition algorithms, Image Vis. Comput. 16 (5) (1998) 295–306.

[33] D.E. King, Dlib-ml: A machine learning toolkit, J. Mach. Learn. Res. 10 (2009) 1755–1758.

[34] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.

[35] C.-H. Lee, Z. Liu, L. Wu, P. Luo, Maskgan: Towards diverse and interactive facial image manipulation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5549–5558.

[36] C. Zheng, T.-J. Cham, J. Cai, Pluralistic image completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1438–1447.

[37] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017, pp. 2242–2251.