Occlusion Robust Face Recognition Based on Mask Learning With Pairwise Differential Siamese Network

Lingxue Song¹², Dihong Gong¹, Zhifeng Li*¹, Changsong Liu² and Wei Liu¹
¹Tencent AI Lab
²Tsinghua University
songlx15@mails.tsinghua.edu.cn gongdihong@gmail.com michaelzfli@tencent.com
lcs@tsinghua.edu.cn wl2223@columbia.edu

Abstract

Deep Convolutional Neural Networks (CNNs) have been pushing the frontier of face recognition over past years. However, existing general CNN face models generalize poorly for occlusions on variable facial areas. Inspired by the fact that the human visual system explicitly ignores the occlusion and only focuses on the non-occluded facial areas, we propose a mask learning strategy to find and discard corrupted feature elements from recognition. A mask dictionary is firstly established by exploiting the differences between the top conv features of occluded and occlusion-free face pairs using innovatively designed pairwise differential siamese network (PDSN). Each item of this dictionary captures the correspondence between occluded facial areas and corrupted feature elements, which is named Feature Discarding Mask (FDM). When dealing with a face image with random partial occlusions, we generate its FDM by combining relevant dictionary items and then multiply it with the original features to eliminate those corrupted feature elements from recognition. Comprehensive experiments on both synthesized and realistic occluded face datasets show that the proposed algorithm significantly outperforms the state-of-the-art systems.

1. Introduction

Deep Convolutional Neural Networks (CNNs) have recently made a remarkable improvement in unconstrained face recognition problem. Researchers are racing in ways to boost the performance using advanced network architectures [25, 3, 27, 6, 32] or designing new loss functions to facilitate discriminative feature learning [24, 33, 13, 31, 2, 42, 30]. Some of them even surpass human recognition ability on certain benchmark database [7].

Despite the huge success of deep learning models under general face recognition scenario, the deep features still show imperfect invariance to uncontrollable variations like pose, facial expression, illumination, and occlusion. Among all these factors, occlusion has been considered a highly challenging one. In real-life images or videos, facial occlusions can often be observed, e.g., facial accessories including sunglasses, scarves, and masks or other random objects like books and cups. As indicated in [17], without specifically trained with a large number of occluded face images, deep CNN-based models indeed cannot function well because of the larger intra-class variation and higher inter-class similarity that caused by occlusions.

One possible approach to improve the performance of CNN models under partial occlusions is to train the network with occluded faces. Daniel et al. [28] proposed to augment training data with synthetic occluded faces in a strategic manner and observed improved performance. However, it does not solve the problem intrinsically because it only ensures the features are more locally and equally extracted, as analyzed in [21]. The inconsistency between features of two faces with different occlusion situations still exists. For example, features of an occlusion-free face bear much more information in eyes area than that of a face wearing a pair of sunglasses unless the network is trained not to utilize the eyes area at all, which is unreasonable.

Inspired by the fact that the human visual system pays attention to the non-occluded facial areas for recognition (and ignores the occluded areas), we propose to discard feature elements that have been corrupted by occlusions. A core question would be: given a face image with random partial occlusions, how to locate those corrupted feature elements? It is not a big deal for traditional low-level features like LBP, HOG or SIFT because there is a clear correspondence between image pixels and final feature elements, but what about the deep CNN features? Therefore, the core of this work is to find corrupted feature elements under random partial occlusion and eliminate the response of these
Figure 1. An overview of the proposed framework. Based on a trunk CNN model trained for face recognition, we propose the pairwise differential siamese network (PDSN) structure to learn the correspondence between occluded facial blocks and corrupted feature elements. Then a mask dictionary is established accordingly, which is used to composite the feature discarding mask (FDM) for a test face with random partial occlusions. Finally, we multiply the FDM with original face features to eliminate corrupted feature elements from recognition.

elements from the recognition process. It is worth stated that the facial occlusion detection problem is not the focus of this paper, thus we directly adopt a similar way as [23] to detect the occlusion location in image space.

To learn the correspondence between occluded facial regions and corrupted feature elements, we develop a novel pairwise differential siamese network (PDSN) structure with a mask generator module that takes pairwise images (one is a clean face and the other is an occluded one of the same identity) as input. The differential signal between the conv features of clean and corresponding occluded faces is fed into the mask generator module. It acts as a role of attention mechanism which encourages the model to focus on those feature elements that have deviated from its true values owing to partial occlusions. Moreover, we propose to learn the mask generator by minimizing a combination of two losses: a pairwise contrastive loss that penalizes the large differences between the masked conv features of clean and occluded faces, and a classification loss which ensures those feature elements that harm the recognition are masked out. With these two losses, our mask generator will identify those feature elements that are harmful for the recognition as well as far from its genuine values as corrupted ones.

In this dictionary is a binary mask, named Feature Discarding Mask (FDM), which indicates the feature elements that should be set to zero when one facial block is occluded. In the testing phase, the FDM of a face with random partial occlusions is derived by element-wise logical “AND” of relevant dictionary items, which is then multiplied with the original face feature to discard those corrupted feature elements from the recognition. Figure 1 gives an overview of the proposed framework.

The main contributions of this paper are two-fold: (1) we propose a novel PDSN framework to explicitly find correspondence between occluded facial blocks and corrupted feature elements for deep CNN models, which is innovative and inspiring; (2) based on the PDSN, we develop a face recognition system that is robust for occlusions. Our system demonstrates superior performance on face datasets with both realistic and synthesized occlusions and generalizes very well on general face recognition tasks.

2. Prior Work

Partial occlusion is one of the major challenges in face recognition that has received much attention in the era of hand-crafted features. Before the emergence of deep CNNs, face recognition under partial occlusions has been typically handled using two types of algorithms, namely, (i) methods that extract local face descriptors only from the non-occluded facial areas or (ii) methods that recover clean faces from the occluded ones. The first type usually explicitly
the middle conv layer is not discriminative enough and the CNN models which is expected to assign lower weights to propose to add a MaskNet branch to the middle layer of features are more equally and locally extracted. W an discriminative features from are covered. In this way, fea-

that only specific regions where a CNN extracts the most set scenario. Daniel recognition on the recovered face images. But there is no to restore occluded facial areas in the wild and carried out features extracted from 25 face patches further improves the ro-

ruption in face verification task. Combining DeepID2+ fea-

tured showing certain degree of robustness to image cor-

ruption in face verification task. Combining DeepID2+ fea-

tures extracted from 25 face patches further improves the ro-

bustness. Cheng et al. [43] present an LSTM-Autoencoder to restore occluded facial areas in the wild and carried out recognition on the recovered face images. But there is no guarantee the recovered part indeed matches the identity of the individuals to be recognized especially under the open-

set scenario. Daniel et al. [28] tackle the occlusion problem by augmenting training data with synthetic occluded faces that only specific regions where a CNN extracts the most discriminative features from are covered. In this way, features are more equally and locally extracted. Wan et al. [29] propose to add a MaskNet branch to the middle layer of CNN models which is expected to assign lower weights to hidden units activated by the occluded facial areas. But the middle conv layer is not discriminative enough and the MaskNet branch lacks additional supervision information to ensure the functionality.

In a word, the discriminative ability of traditional low-

level feature-based methods is limited, and the existing few deep learning-based methods lack the awareness of how partial occlusions truly affect the CNN models. The in-

consistency between features of two faces with different occlusion situations has not been carefully considered yet. The proposed method complements the missing piece of the puzzle and is able to explicitly locate corrupted feature elements for trained CNN models and discard them from the recognition, to ensure a fair comparison. Therefore, our approach is an intrinsic way with good generalization ability compared to the aforementioned studies.

3. Proposed Approach

The overall pipeline of the proposed approach is shown in Figure 1, which decomposes the problem of face recognition under random partial occlusions into three stages. Stage I: Learning mask generators using the proposed pairwise differential siamese network (PDSN) to capture the correspondence between occluded facial blocks and corrupted feature elements. Stage II: Establishing a mask dictionary from the learned mask generators. Stage III: In the testing phase, combining the feature discarding mask (FDM) of random partial occlusions from this dictionary, which is then multiplied with the original feature to eliminate the effect of partial occlusions from recognition.

3.1. Stage I: Learning Mask Generators

3.1.1 Problem Analysis

Face images fed into the CNN model are mostly well-

aligned by detected facial keypoints, we divide the aligned face into non-overlapping \( N \times N \) blocks, denoted as \( \{b_j\}_{j=1}^{N^2} \), and aim to learn a mask generator for every \( b_j \) to find the corrupted feature elements when this block is oc-

cluded. In our implementation, we set \( N = 5 \) according to the input image size so that the facial components like eyes, nose tip and mouth are appropriately associated with a block. The face (a) in Figure 1 gives the division example.

Then we define our core problem in Stage I as: given the feature of a face image with block \( b_j \) occluded, denoted as \( f(x_j) \), how to learn a mask generator \( \mathcal{M}_\theta \) whose output is multiplied with the \( f(x_j) \) to mask out those corrupted elements. Let the purified feature be denoted as \( \hat{f}(x_j) = \mathcal{M}_\theta(\cdot)f(x_j) \). There are two choices to be decided before running into the learning process:

The choice of \( f \). For the CNN-based face recognition model, the face feature usually refers to the output of the fi-

nal fully-connected (fc) layer before the classification layer. However, every neuron in the fc layer integrates information from all the output elements of the previous layer, so the oc-

...
in 8 channels of the top conv feature maps under three types of occlusions. Obviously, the feature values are altered in a different way for different channels, elements of some channels change very little while elements of some channels change drastically in the same spatial locations. This is interesting because in view of the receptive field, the same spatial location of different conv channels gather information from the same region of the input image, but they actually react quite differently to occlusions. Therefore, we believe the output dimension of \( \mathcal{M}_\theta \) should be the same as the top conv feature maps, which is \( C \times W \times H \).

3.1.2 Pairwise Differential Siamese Network

Given the analysis in Sec. 3.1.1, we propose the pairwise differential siamese network (PDSN) structure to learn the relations between occluded facial blocks and corrupted feature elements. As illustrated in Figure 4, it consists of a trunk CNN and a mask generator branch, forming a siamese architecture. The trunk CNN is responsible for extracting base face representation, which is shared by the clean and occluded face pairs and could be any CNN architecture. The core mask generator module \( \mathcal{M}_\theta \) in our PDSN is expected to output a mask whose element is a real value in \([0, 1]\) and is multiplied with the input contaminated feature to diminish its corrupted elements: \( f(x_j^i) = \mathcal{M}_\theta(\cdot) f(x_j^i) \), where \( f(\cdot) \) is top conv feature and \( x_j^i \) denotes occluded face image of the \( i^{th} \) pair. The two faces inside an input pair belong to the same identity \( y^j \) and the only difference is that one of them has partial occlusion on the facial block \( b_j \). The key requirement for learning the mask generator is that the remaining part of the feature \( f(x_j^i) \) after masking should be as similar to its corresponding clean feature \( f(x^i) \) as possible while guarantees a successful recognition.

To this end, we propose to learn \( \mathcal{M}_\theta \) by minimizing a combination of two losses:

\[
L_\theta = \sum_i \ell_{cls}(\theta; \tilde{f}(x_j^i), y^i) + \lambda \ell_{diff}(\theta; \tilde{f}(x_j^i), \tilde{f}(x^i))
\]

(2)

The first part of the cost, \( \ell_{cls} \), is accounting for evaluating the importance of each feature element for recognition, and the second part, \( \ell_{diff} \), assesses how far the feature of an occluded face is away from its true value. We will expand this formulation in the following part.

The classification loss \( \ell_{cls} \). To find the corrupted feature elements, an intuitive idea is that, these feature elements contribute little to identifying the input face and may instead cause higher classification loss. Therefore the most straightforward supervision signal is the identity information, that is, the occluded face should be correctly classified by the classifier of the trunk CNN after masking, which gives us the first loss item (softmax loss for example):

\[
\ell_{cls}(\theta; \tilde{f}(x_j^i), y^i) = -log(p_{y^i}(F(\tilde{f}(x_j^i))))
\]

(3)
The $\bar{f}(x^i_j)$ is the top conv feature of an occluded face after masking. $F$ is the fc layer(s) of the trunk CNN model next to the top conv layer, and it could also be the average pooling layer in models like [13].

The differential signal and pairwise loss $\ell_{diff}$. The results shown in Figure 3 inspired us that the differential signal between the top conv activation values of an occluded face and its corresponding clean one could be a good indicator of which feature elements are potentially corrupted ones. To put it another way, the differential input signal acts as a role of attention mechanism which encourages the mask generator to focus on those feature elements that have deviated from its true values owing to partial occlusion. Therefore, we feed our mask generator module with the absolute difference between features of an occlusion-free face and its occluded counterpart.

To further make use of the supervision information of what this subject’s occlusion-free feature looks like, we propose a pairwise contrastive loss that minimizes per-element differences between the masked features of the occluded and occlusion-free faces:

$$\ell_{diff}(\theta; \bar{f}(x^i_j), \bar{f}(x^i)) = \|\mathcal{M}_\theta(\cdot)f(x^i) - \mathcal{M}_\theta(\cdot)f(x^i_j))\|_1$$  \hspace{1cm} (4)

where $\mathcal{M}_\theta(\cdot) = \mathcal{M}_\theta(|f(x^i_j) - f(x^i))|$, and $\|\cdot\|_1$ is the L1 norm. Obviously, this contrastive loss will punish those feature elements of the occluded face which are largely different from its occlusion-free one. Together with the classification loss, our mask generator will identify those feature elements that are harmful for the recognition as well as far from its genuine values as corrupted ones.

Thus, the overall object function in Eq. (2) used in our implementation is:

$$L_0 = -\sum_{i} \log(p_y(x^i_j, F(\mathcal{M}_\theta(\cdot)f(x^i)))) + \lambda \|\mathcal{M}_\theta(\cdot)f(x^i) - \mathcal{M}_\theta(\cdot)f(x^i_j))\|_1$$  \hspace{1cm} (5)

The $\lambda$ is set to 10 to make the different components of the object function have the same scale in our experiments.

We implement $\mathcal{M}_\theta$ as a module with several conv blocks and learn the different $\theta$ for occlusions on different facial blocks. The different $\theta$ is accounting for the distinct property of different facial components. For example, the eyes bear much more significance than the cheek area, therefore the input distribution of the mask generator varies accordingly. When learning mask generator $j$, in addition to the faces that only the target block $b_j$ is occluded, we augment samples with other blocks also occluded, which are the 4-neighbors of the target block $b_j$, to capture the dependency of adjacent blocks, as shown in Figure 4.

3.2. Stage II: Establishing the Mask Dictionary

In the testing phase, we don’t have the paired images of a probe face and its occlusion location is random. Therefore, the trained PDSNs cannot be directly used to output the feature discarding mask(FDM) of a probe face. In Stage II, we would like to extract a fixed mask from every trained mask generator $\mathcal{M}_\theta$ and build a dictionary accordingly.

$$M_j[k] = \begin{cases} 0 & \text{if } \tilde{m}_j[k] \in \{\tilde{m}_j[1], \ldots, \tilde{m}_j[r \times K]\}, \\ 1 & \text{else.} \end{cases}$$ \hspace{1cm} (6)

For a mask generator $\mathcal{M}_{\theta_j}$, we first feed the trained network with large amount of face pairs, one of which is occluded on the $j^{th}$ facial block and obtain the output masks of this generator, forming a large set of $m_j^1, m_j^2, \ldots, m_j^P$, where $P$ (about 200k in our experiment) is the number of the face pairs. After Min-Max normalizing each $m_j^i$, we calculate the element-wise mean value of these $m_j^i$s and get a mean mask $\tilde{m}_j$. It’s possible to directly use this $\tilde{m}_j$ as the FDM when the $j^{th}$ block is occluded (referred to as soft weight schema). But this will reserve feature elements with very low mask values, which is inappropriate since the facial components inside this block have been totally lost. Therefore we believe setting those feature elements to zero to completely remove the noise is critical. We’ll validate this in Sec. 4.2. The binarized FDM $M_j^b \in \mathbb{R}^{C \times W \times H}$ for this mask generator is derived by setting the feature loca-
Occlusion Detection.

Figure 6 shows some detection results. Finally, our occlusion detection model works pretty well with sufficient face images to provide a good initialization.

Figure 5. Examples of the feature discarding masks of two occlusion types combined from our mask dictionary.

Figure 6. Occlusion detection results of our FCN-8s segmentation network on the occluded Facesrub and AR test images.

Network Structure. We employ the refined ResNet50 model proposed in recently published ArcFace [2] as our trunk CNN model. The mask generator is simply implemented as a CONV-PReLU-BN structure with a sigmoid function to map the output into \([0, 1]\).

Training. The training procedure includes three stages. Stage 1: Train the trunk CNN on the CASIA-WebFace [40] dataset with the large margin cosine loss [31]. Stage 2: Fixing the model parameters of the trunk CNN, and train the mask generator modules with specifically designed face pairs as shown in Figure 4. We discovered that occlusions on the peripheral blocks of the faces rarely affect the recognition accuracy (less than 0.1% drop), therefore we narrow down the number of needed mask generators from 25 to 9, which correspond to the central \(3 \times 3\) blocks that cover the main facial components. Stage 3: After establishing our mask dictionary, we generate face samples with various random partial occlusions and calculate their corresponding FDMs using this dictionary. Then finetune the trunk CNN using these (face, mask) pairs with a small learning rate. This stage is designed for relieving the inconsistency between the real-value mask output by the mask generator and the final binarized version, so a few epochs are enough.

Testing. In the testing stage, the similarity score is computed by the cosine distance of the fc features of two faces. The nearest neighbor classifier and thresholding are used for face identification and verification respectively. Considering the fact that, when recognizing an occluded face, we have lost the information from the occluded part of the face. Therefore it is necessary to also exclude this portion from the other faces comparing with it, to ensure that the similarity scores are computed based on equivalent information.

Baseline Models. Two baseline models are considered. The first one is the state-of-the-art face recognition model trained on CASIA-WebFace dataset. We will refer to it as Trunk CNN. The second one has the same configuration with the first one but finetuned with synthetic occluded CASIA-WebFace dataset (average occluder area is 25% of the face images), which will be referred to as Baseline.

4.2. Ablation Study

The Effect of \(\tau\). We conduct exploratory experiment to investigate the effect of \(\tau\) used in binarization. By varying \(\tau\) from 0 to 0.45, we evaluate our method on the AR dataset.

4.3. Stage III: Occlusion Robust Recognition

With this mask dictionary, the FDM of a face with arbitrary partial occlusions could be derived by combining relevant dictionary items. By relevant we mean that if the occlusion area in a probe face has at least 0.5 IoU with a predefined facial block from the dictionary, we count this block as an occluded one for this face. For example, for the face (a) wearing sunglasses in Figure 1, its occlusion region covers block \(\{b_{12}\}_{12}^{14}\), therefore its FDM is calculated by \(M = M_{12} \land M_{13} \land M_{14}\), where \(\land\) denotes the element-wise logical “AND” and the result \(M\) is still a binary mask. Figure 5 shows 8 channels of the FDM composited from the dictionary for sunglasses and scarf occlusions respectively.

4. Experiments

4.1. Implementation Details

Preprocessing. The standard MTCNN [41] is used to detect 5 face landmarks for all the images. After performing similarity transformation accordingly, we obtain the aligned face images and resize them to be \(112 \times 96\) pixels.

Occlusion Detection. We train an FCN-8s [14] segmentation network to detect the occlusion location. The training data includes the synthetic occluded CASIA-WebFace dataset and images of 26 subjects (outside the test subjects) from the AR dataset. The vgg16 backbone is firstly trained with sufficient face images to provide a good initialization. Finally, our occlusion detection model works pretty well with a mean IU of 98.51 on our synthetic occluded Facesrub dataset [19]. Figure 6 shows some detection results.
The probe set contains faces with sunglasses and scarf occlusions, and the gallery set contains 1 clean face for every subject. The rank-1 identification accuracy is given in Table 1. As $\tau$ being increased, the accuracy first rises up and then moves down as $\tau$ approaching 0.45. The best accuracy is achieved at $\tau = 0.25$ and the performance is not highly sensitive to this threshold.

**Mask Type.** To further explore the importance of binarization, we performed additional experiments with results shown in Table 2. First, by comparing the “Binary” and “Soft weight”, we see that “Soft weight” noticeably decreases performance. We speculate that it’s due to the excessive participation of features with very low mask values. Then we performed another experiment “Soft+Binary” to remove those features with mask values lower than the threshold (setting these mask values to 0) while keeping mask values above threshold unchanged (rather than setting them as 1). This version achieved comparable performance to the Binary version. Obviously, the importance of binarization is to completely eliminate the noise by setting a feature element with a very low mask value to zero. At the same time, the binary mask is highly efficient in terms of both calculation and storage.

**The Differential Supervision.** To investigate the importance of the differential input and pairwise loss. We set $\lambda$ in the loss function in Eq. (5) to zero, and learn mask generators only from occluded face features. The mask dictionary is established with the same data and threshold $\tau$. The performance comparison is shown in Table 3. The model trained with the pairwise supervision consistently outperforms the model that only trained with classification loss. In Figure 7 we visualize several channels of the mean masks of the left eye and nose blocks respectively under these two conditions. With our full PDSN, the mask elements with much lower weights (highlighted part in Figure 7) could reflect the occlusion location in image space to some extent, which is reasonable since the top conv layer still preserves spatial information. While the mean masks generated by classification loss only are in chaos. As discussed above, the differential input and contrastive loss help the model concentrate on the feature elements that have been altered a lot by partial occlusions, while the classification loss alone is likely to also diminish feature elements that are affected by some other factors unrelated to occlusion.

### 4.3. Performance on LFW Benchmark

LFW [7] is a standard face verification benchmark dataset under unconstrained conditions. We evaluate our models strictly following the standard protocol of unrestricted with labeled outside data and report the mean accuracy on the 6,000 testing image pairs.

As shown in Table 4, the baseline model actually decreases the accuracy of the original trunk CNN by 0.52% when it is trained to gain more robustness to partial occlusions because most of the face images in the LFW dataset are not occluded. This phenomenon is consistent with [21], where performance encountered a drop when they tested a model that functions well for occluded objects on non-occluded objects. While our method can keep the performance of the trunk CNN since our design principle is just discarding those corrupted feature elements from comparison under partial occlusion condition, instead of forcing the trunk CNN to specifically accustom to partial occlusions.

### 4.4. Performance on MegaFace Challenge1

MegaFace Challenge [8] is a testing benchmark to evaluate the performance of face recognition algorithms at the million scale distractors. It contains a gallery set with more than 1 million face images. And the probe set consists of two datasets: Facescrub [19] and FGNet. In this study, we use the Facescrub dataset as our probe set. The training set is viewed as small if it is less than 0.5M. We evaluate the

<table>
<thead>
<tr>
<th>Mask type</th>
<th>Binary</th>
<th>Soft weight</th>
<th>Soft+Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunglasses</td>
<td>98.19</td>
<td>96.67</td>
<td>98.19</td>
</tr>
<tr>
<td>Scarf</td>
<td>98.33</td>
<td>97.22</td>
<td>99.03</td>
</tr>
</tbody>
</table>

Table 2. The rank-1 identification accuracy (%) in AR dataset Protocol 2. The results in the Protocol 1 have a similar conclusion.

<table>
<thead>
<tr>
<th>Differential</th>
<th>AR sunglass</th>
<th>AR scarf</th>
<th>MF1 occ</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>95.97</td>
<td>97.92</td>
<td>54.80</td>
</tr>
<tr>
<td>Yes</td>
<td>98.19</td>
<td>98.33</td>
<td>56.34</td>
</tr>
</tbody>
</table>

Table 3. Rank-1 identification accuracy(%) of our method with and without differential supervision information. “MF1occ” refers to the occluded Facescrub probe set we synthesized.

---

**Figure 7. Illustration of the mean masks learned by our full PDSN and only by classification loss. $\bar{m}_i$ corresponds to occlusion on the left eye block and $\bar{m}_j$ corresponds to occlusion on the nose block.**
Table 4. Face verification(%) on the LFW benchmark. “#Models” is the number of models used in the method for evaluation.

Table 5. Face identification accuracy(%) on MegaFace Challenge 1. “MF1occ” refers to the occluded Facescrub probe set.

Table 6. Rank-1 face identification accuracy(%) on the AR dataset with natural occlusions.

5. Conclusions

In this paper, we propose an occlusion robust face recognition method based on the pairwise differential siamese network (PDSN) that explicitly builds correspondence between occluded facial blocks and corrupted feature elements. Competitive results on synthesized and realistic occluded face datasets demonstrate the superiority of the proposed method, especially the great generalization ability on general face recognition tasks.
References


