Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Masked face recognition with convolutional visual self-attention network

Yiming Ge a, Hui Liu a,⇑, Junzhao Du a, Zehua Li a, Yuheng Wei a

a School of Computer Science and Technology, Xidian University, Xi'an 710071, China

A R T I C L E   I N F O

Article history:
Received 11 November 2021
Revised 18 June 2022
Accepted 9 October 2022
Available online 30 October 2022
Communicated by Zidong Wang

Keywords:
COVID-19
Masked face recognition
Convolutional
Self-attention

Abstract

With the global outbreak of COVID-19, wearing face masks has been actively introduced as an effective public measure to reduce the risk of virus infection. This measure leads to the failure of face recognition in many cases. Therefore, it is very necessary to improve the recognition performance of masked face recognition (MFR). Inspired by the successful application of self-attention in computer vision, we propose a Convolutional Visual Self-Attention Network (CVSAN), which uses self-attention to augment the convolution operator. Specifically, this is achieved by connecting a convolutional feature map, which enforces local features, to a self-attention feature map that is capable of modeling long-range dependencies. Since there is currently no publicly available large-scale masked face data, we generate a Masked VGGFace2 dataset based on the face detection algorithm to train the CVSAN model. Experiments show that the CVSAN algorithm significantly improves the performance of MFR compared to other algorithms.

© 2022 Published by Elsevier B.V.

1. Introduction

In recent years, with the rapid development of deep learning, face recognition models based on deep learning have been proved to have excellent accuracy [1–5] and widely used in the industrial fields [6], such as security, surveillance and mobile applications. The great success in face recognition can be attributed to three aspects: 1) From the first widely used training set CASIA Webface [7] to MS-Celeb-1M [8] and VGGFace2 [9], a large number of face datasets are proposed. These datasets are conducive to training deep models. 2) Classical network structures, such as AlexNet [10], VGGNet [11], GoogleNet [12], ResNet [13], are widely used as the backbone network of the face recognition model. These network structures shape the most advanced deep face recognition. 3) Various loss functions, such as loss based on euclidean distance [4] or angular/cosine margin [1–3], softmax loss and its variations [5], are used as monitoring signals during model training. These loss functions can guide the network to learn more distinguishable features with inter-class discrepancy and intra-class compactness.

Although deep face recognition algorithms have made great progress, low-quality face images [14] are still a challenging problem. These low-quality face images are usually caused by different face angles, lighting conditions, partial occlusion, low resolution, and noise. Researchers can eliminate most of these negative factors by preprocessing the face images and choosing the appropriate placement angle for the hardware [15]. However, partial occlusion caused by clothes and accessories is unavoidable [16]. With the global outbreak COVID-19, this has become a major challenge in the field of face recognition.

As a public measure to avoid being infected and reduce the spread of diseases, wearing masks reduces the recognition accuracy of face recognition systems deployed in public places. The reason for the decline of model performance is that previous face recognition models are designed to focus on overall facial information without considering the loss of face information in the case of occlusion. In fact, the National Institute of Standards and Technology (NIST) recently presents a study [17] where 89 major commercial facial recognition algorithms are examined. The results show that when matching masked and unmasked face photos, the error rates of these algorithms are between 5% and 50%. To make the face recognition method more universal, the task of this paper is to make the masked face recognition (MFR) model effectively distinguish different identities with masked or unmasked faces.

In face recognition and other recognition tasks, a major factor that affects model performance is training data. Deep face recognition models are usually trained on large-scale training datasets, such as CASIA Webface [7], MS-Celeb-1M [8] and VGGFace2 [9]. These datasets are usually constructed by collecting images from the Internet in a semi-automatic way. However, it is extremely difficult to collect large-scale masked face images with identity information. Moreover, researchers need to carefully select the...
masked faces for each target identity, a time-consuming process. In this work, we use the tools proposed in [18] to generate the masked version of the face recognition data to train the model. Specifically, we generate the Masked VGGFace2 dataset on the basis of the original VGGFace2 dataset. Compared with a model trained with only full face data, a model trained with masked face data achieves better MFR performance. But the performance still has room for further improvement.

Recently, a number of methods [19–21] have been proposed to solve the problem of MFR. These methods only use pure convolution operations. The local nature of the convolution kernel prevents it from capturing the global information in the masked face image, which is usually necessary for better face recognition performance [22]. On the other hand, self-attention [23] has emerged as the latest development in capturing long range interactions and has been applied to computer vision tasks, such as image classification [24], object detection [25–27] and semantic segmentation [28]. The key idea of visual self-attention is to perform a weighted average of the output value of the previous layer. This allows information at different locations in the entire image to interact with each other, that is, long range interactions can be captured. However, self-attention has never been applied to MFR. In this paper, we propose a convolutional visual self-attention network (CVSAN) by using a self-attention [29] structure to augment the convolutional operators. Specifically, we fuse the convolution feature map to the self-attention feature map. The advantages are that it cannot only capture the long range interactive information in the masked face images through the self-attention structure, but also learn the local features through the convolution operations.

The main contributions of our work are summarized as follows:

1. The Transformer structure is introduced for the first time into the masked face recognition. 
2. In the proposed model, the convolutional feature map with access to local feature information is fused to the self-attention feature map capable of modeling long-range dependencies. 
3. Experimental results show that the proposed model offers satisfactory MFR performance and thus demonstrate the feasibility of the Transformer structure in MFR.

The rest of this paper is organized as follows. Section 2 reviews related works. Section 3 details the proposed convolutional visual self-attention network. The experimental results are reported and analyzed in Section 4. Finally, conclusions are given in Section 5.

2. Related work

In this section, we first survey some deep face recognition algorithms based on deep learning. Then, we review the applications of the Transformer in computer vision. Finally, the recent advances of masked face datasets and algorithms are presented.

2.1. Deep face recognition

Deep face recognition algorithms based on convolutional neural networks (CNNs) [30] have achieved satisfactory performance. They can learn discriminative features from face images. These features are embedded in \(n\)-dimensional vector space with small intra-class distances and large inter-class distances.

Deep face recognition models contain two key elements in the training process: 1) Network Architecture. Researchers use classical network structures or new network structures to design face recognition models [31]. These classical network structures, such as AlexNet [10], VGGNet [11], GoogleNet [12], ResNet [13] and SENet [32], are widely used as the backbone networks of the face recognition models. Such as Taigman et al. [5] are the first to use a nine-layer CNN with several locally connected layers for face recognition. In addition to the classical network architectures, there are some novel architectures designed for face recognition to improve the performance of the model [31,33,34]. 2) Loss Function. Researchers propose new loss functions or modify the existing ones to force convolutional networks to learn separable features to distinguish different people. Early deep face recognition algorithms [35,5] use cross-entropy loss for feature learning. Schroff et al. [4] use triplet loss to directly optimize the embedded features, which further improves the performance of the model. Most of the recent researches [2,3,1] are based on the loss of angle/cosine margin, in which a penalty term of angular margin is introduced to enhance the discriminative ability of features. These loss functions are usually used as the supervision signals and encourage the separability of features.

Although deep face recognition has achieved encouraging results, these methods are often used to learn discriminative features from full face images. When there are masked faces in the data, the results are not satisfactory. Our method focuses on the task of MFR.

2.2. Vision transformer

The Transformer [23] is a widely used architecture in Natural Language Processing (NLP), which is based on the self-attention architecture. Self-attention uses content-based addressing mechanism to implement pairwise entity interaction, thereby learning a rich hierarchy of associated features across long sequences [36]. Due to the great success of the Transformer in NLP, researchers begin to try to solve computer vision learning tasks relying on the Transformer model [24–28].

The Visual Transformer (ViT) [29] is an early application of the Transformer in computer vision tasks and achieves almost the same performance as CNNs in terms of image classification accuracy. In ViT, each input image is segmented into a series of patches, and each patch is converted into a sequence of tokens. These tokens are processed by the Transformer block. Recently, a series of Vision Transformers have been proposed. Compared with the ViT model, which uses patch-level feature, The Transformer in Transformer (TNT) [37] projects pixel-level features into patch embedding space through linear transformation layers and then adds this pixel-level features to patches. The Data-efficient image Transformer (DeiT) [38] relies on a distillation token to ensure that the student can learn from the teacher through attention. Compared with the ViT, it can get competitive results only by training on ImageNet. The Swin Transformer [39] introduces non-overlapping window partitions and limits self-attention within each local window, while allowing cross-window connections. This hierarchical structure provides the flexibility to model at different scales and has linear computational complexity with respect to the size of the image. The Bottleneck Transformer (BoT) [36] regards the ResNet bottleneck block with self-attention as a Transformer block to process computer vision tasks. It significantly improves the performance of instance segmentation and object detection, as well as reducing the parameters of the model.

For the self-attention structure to be position aware, position coding needs to be added to the input image features. In the original Transformer [23], sine and cosine functions of different frequencies are used for position encoding. However, this position coding is not helpful for image classification and object detection [22] because it does not satisfy the translation equivariant, which is a required property when processing images. Therefore, in this work, we use two-dimensional relative position coding [22], which is an improvement of the relative positional coding [40] that can achieve the translation equivariant while preventing the permutation equivariant.
2.3. Masked face datasets and recognition

Since the global health crisis originated with COVID-19, partial occlusion caused by masking has become a major challenge in the field of face recognition. Recently, some related work on MFR has emerged.

**Masked Face Datasets.** The lack of large-scale masked face datasets is one of the reasons for the poor performance of MFR. Fortunately, some researchers have tried to solve this problem. Geng et al. [41] collect a masked face segmentation and recognition (MFSR) dataset from the Internet and the real world. Each identity in this dataset includes masked faces and full faces. However, this dataset contains only 11,615 images and 1,004 identities, not enough to train complex networks. Other researchers directly utilize the existing face dataset to synthesize masked face images using face key point detection techniques. Montero et al. [21] use a MS1MV2 dataset to generate masked face images. In addition, for the performance evaluation of masked faces, Deng et al. [42] organize the Masked Face Recognition challenge, and the authors collect a test set of masked faces for evaluating deep face recognition methods. Anwar and Raychowdhury [18] also provide a public real mask face dataset MFR2 for model evaluation.

**Masked Face Recognition.** To improve the performance of MFR, some MFR algorithms have been proposed. Haririi [19] directly extracts features from the upper part of the face for training and evaluation. By using only the upper part of the face, the influence of the mask on face recognition can be ignored. However, when dealing with unmasked face images, the available information on the lower half of the faces will be lost. Therefore, this method is not suitable for face recognition scenarios where there are both masked and unmasked faces. Li et al. [20] train a Generative Adversarial Network (GAN) that can restore masked faces to full faces. Then, these recovered face images are identified by using a face recognition model. However, this model requires real-world face data for training and increases the time consumption in the prediction phase.

In this work, we propose a convolutional visual self-attention algorithm by fusing convolution and self-attention operators. Besides, we use the face key point detection technology to generate a masked version of face recognition datasets for training and evaluation. In the prediction stage, our model can improve the face recognition performance in scenarios with both masked and unmasked faces, without the need of extra prediction time.

3. Masked face recognition with convolutional visual self-attention network

In processing masked face images, convolution is used to extract local features, leading to an inability to learn global information. Self-attention recently showed the ability to capture long-range interactive information in images. Therefore, we use self-attention as a supplement to convolution in visual tasks. We first describe the details of self-attention in CVSAN, and then introduce the implementation of our proposed CVSAN structure, including the overall design, pure convolution stage, Conv-MHSA stage and loss function.

3.1. Multi-head self-attention over image

Given an input feature map with shape of \((W, H, C)\), where \(H, W\) and \(C\) represent the height, width and number of channels of a feature map, respectively. We first use pointwise convolution to process the input to get the queries \(Q \in \mathbb{R}^{WH \times d_q}\), keys \(K \in \mathbb{R}^{WH \times d_k}\) and values \(V \in \mathbb{R}^{WH \times d_v}\) required by the self-attention mechanism. \(d_q, d_k, \) and \(d_v\) respectively refer to the dimensions of queries, keys and values in self-attention and \(d_v = d_k\). Then, \(QK^T\) calculates the attention scores \(S \in \mathbb{R}^{WH \times WH}\) between vectors at different positions in the feature maps. These scores determine the degree of attention we give other vectors when encoding the vector at the current position. The scores \(S\) need to be normalized to enhance gradient stability for improved training. Then, we translate the scores into probabilities with the softmax function. The specific formula is:

\[
S = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)
\]

where the softmax function is applied over each row of the input matrix. Finally, the output of attention layer is the weighted sum of a set of value vectors (packed into \(V \in \mathbb{R}^{WH \times d_v}\)). The output matrix can be calculated by

\[
A(Q, K, V) = SV.
\]

Since there is no clear information on positions, the self-attention is permutation equivariant. Therefore, it is necessary to use position coding to clarify the position information. We use the standard learnable two-dimensional relative position coding, because recent studies have shown that relative distance sensing position coding [40] is more suitable for visual tasks [22, 43, 44]. Suppose that \(i = (i_x, i_y)\) and \(j = (j_x, j_y)\) are the positions of two points in the feature map considered. The relative position information of \(i\) and \(j\) is computed by

\[
p_{ij} = r^W_{i_x - j_x} + r^H_{i_y - j_y}
\]

where \(r^W_{i_x - j_x}\) and \(r^H_{i_y - j_y}\) are learned embeddings for relative width \(j_x - i_x\) and relative height \(j_y - i_y\), respectively. We can rewrite the attention logit as:

\[
t_{ij} = \frac{q_i}{\sqrt{d_k}} (k_j^T + p_{ij}^T),
\]

where \(q_i\) is the query vector for \(i\) and \(k_j\) is the key vector for \(j\). Now, the output of head \(h\) becomes:

\[
A_h(X) = \text{softmax}\left(\frac{QK^T + L}{\sqrt{d_k}}\right)V,
\]

where \(L \in \mathbb{R}^{WH \times WH}\) is the matrices of two-dimensional relative position logits, which satisfies \(L_{ij} = q_i p_{ij}^T\).

The output results of multiple heads are spliced to form the overall output of the attention layer. The specific formula is:

\[
\text{MHSA}(X) = \text{Concat}(A_1, \ldots, A_h)W^\theta,
\]

where \(W^\theta \in \mathbb{R}^{d_h \times c}\) is a learned linear transformation. Finally, \(\text{MHSA}(X)\) is reshaped into a tensor of shape \((W, H, C)\) to match the original spatial size and number of channels.

3.2. Overall design

Many improved convolutional architectures [32, 45, 46] proposed earlier show that convolution operators are limited by their local nature and inability to capture global context in images. However, the global context is usually necessary for better recognition of objects in images. For masked face images, a large number of local facial features are lost due to partial occlusion of the face. Therefore, it is more necessary to supplement face information by learning global context information. To better learn local and global information, we propose a masked face recognition (MFR) algorithm, namely convolutional visual self-attention network (CVSAN). The overall framework of the 2-stage CVSAN is shown in Fig. 2(a).
The first stage, which includes pure convolution operations, is called the Conv Stage. Due to the high computational complexity and time consumption of the self-attention mechanism, we use pure convolution operations to extract facial features in the early stages of the model. A $3 \times 3$ convolution of stride 1 is used to improve the spatial connection of the feature map ($W \times H \times 3$). The remaining part of the convolution stage is divided into three modules, and the operation of each module will cause the feature map to change. The output of the pure convolution stage is a feature map of shape ($W/8 \times H/8 \times 256$).

The second stage is to use the self-attention to augment convolution operation, which is called Conv-MHSA Stage. The Conv-MHSA stage includes the multiple residual convolutional layers and the multi-head self-attention layers. They capture the local feature information of the input feature and the remote interaction information, respectively. The residual convolutional layer is similar to that in the pure convolution stage, which includes the residual connections and layer normalization. The multi-head self-attention layer integrates the self-attention structure into the residual structure. It uses $1 \times 1$ convolution to process the feature map and input the features into the multi-head self-attention. Finally, we can obtain more useful features which include local and global information by feature fusion.

### 3.3. Convolutional stage

In order to augment the spatial connection, we first apply a $3 \times 3$ convolution of stride 1 operation on the input face images. The output features are processed by 3 successively connected modules (each contains multiple residual blocks). Each module reduces the width $W$ and height $H$ of the image feature by half and doubles the number of channels. As shown in Fig. 2(a), each module of the pure convolution stage includes two versions of the residual structure. Fig. 2(b) is the specific implementation process of these two versions. When the input and output are of different dimensions, the residual block is defined as:

$$y = f(x, \{W_i\}) + W.x.$$  

Here $x$ and $y$ are the input and output of the layers considered. The function $f(x, \{W_i\})$ represents the residual mapping to be learned, $W.$ is used to match dimensions (done by $1 \times 1$ convolutions). When the input and output are of the same dimension, the residual block is defined as:

$$y = f(x, \{W_i\}) + x.$$  

Compared with Eq. 7, there is no convolution operation for matching dimensions. It is easy for the deep residual net to obtain accuracy gains from the greatly increased depth while well preventing overfitting of the model.

In the pure convolution stage, the local nature of the convolution kernel prevents it from capturing the global context in the image. Inspired by the application of self-attention mechanism in computational vision, in the next stage, we use the self-attention to augment convolution operation to further process the image features.

### 3.4. Conv-MHSA stage

In order to improve the ability of learning global information from the input, we employ the self-attention layer, which can capture remote interactions and make the overall model focus on global information. Self-attention is introduced in the Conv-MHSA stage as a supplement to convolution. Module 4 of Fig. 2(a) summarizes our proposed Conv-MHSA stage, which can capture long-range dependencies while focusing on local details.

**MHSA in Convolution Block.** As shown in Fig. 2(c), the multi-head self-attention is incorporated into the residual structure. Considering that self-attention performed globally across entities requires $O(n^2d)$ memory and computation, we only use self-attention in the Conv-MHSA stage of the proposed model. Moreover, the feature map in the final stage has higher semantic information, which is beneficial to self-attention. There are three MHSA blocks in the MHSA stage.

In the MHB1_out block, suppose that $M_1 \in R^{W \times H \times C}$ is the input tensor of the layer considered. We first use $1 \times 1$ convolution with stride of 1to produce feature map $M_2 \in R^{W \times H \times C}$. Next, $M_2$ is input into the MHSA shown in Fig. 1. We use pointwise convolution to process $M_2$ to get the queries $Q$, keys $K$ and values $V$ required by the self-attention. Specifically, $Q, K$ and $V$ are calculated as follows:

$$Q = \text{Conv}(M_2, W^Q),$$
$$K = \text{Conv}(M_2, W^K),$$
$$V = \text{Conv}(M_2, W^V),$$

where $W^Q \in R^{1 \times 1 \times d_q}, W^K \in R^{1 \times 1 \times d_k}$ and $W^V \in R^{1 \times 1 \times d_v}$ represent $1 \times 1$ convolution kernels, and Conv means the convolution operation. Since these three convolution kernel matrices are initialized randomly, $M_2$ can be projected into different representation subspaces after training. The $Q, K$ and $V$ are flattened into two-dimensional sequences of shape $(WH, C)$ and then we get feature map $M_3 \in R^{WH \times C}$ according to Eq. 5 and Eq. 6. Finally, we apply an average pooling layer followed by $1 \times 1$ convolution with stride of 2 to produce the feature map $M_4 \in R^{WH/2 \times C}$. In the MHB1_out block, since the size of the feature maps of $M_1$ and $M_4$ are different, a $1 \times 1$ convolution with stride of 2 is needed for the input and output feature maps to have equal size and number of channels when doing the shortcut.

Next, the feature output from MHB1_out are sequentially input into two MHB2_out blocks. The specific process of MHB2_out is shown in the second picture in Fig. 2(c). Its implementation process is similar to that of MHB1_out. The difference is that this part no longer uses the pooling layer, so the feature map does not changed in size.

**Feature Fusion.** In the Conv-MHSA stage, in addition to 3 MHSA blocks, 3 residual blocks are also used to extract local features. The implementation process of each block is introduced in Section 3.3. We connect the convolution and self-attention feature maps to yield the final output feature maps, which can be written as:

**Fig. 1.** Multi-Head Self-Attention (MHSA) structure. We show a head of the MHSA in the figure. $W^Q$, $W^K$ and $W^V$ represent pointwise convolution, and the output features are flattened into two-dimensional sequences, $\oplus$ and $\odot$ represent elementwise sum and matrix multiplication respectively. The attention logits are $QK^T + I$ where $Q, K, I$ represent queries, keys and position logits respectively.
\[
\frac{1}{2} \left( \text{BN} \left( \text{Conv} \left( \text{Fin} \right) \right) + \text{MHSA} \left( \text{Fin} \right) \right)
\]

where \( \text{Fin} \) is the input feature of the Conv-MHSA stage. \( \text{f Conv} \) and \( \text{f MHSA} \) represent multiple residual blocks and MHSA blocks, respectively. \( \text{BN} \) is the batch normalization operation on the tensor after feature fusion. The CVSAN model after feature fusion can capture long range interactive information while focusing on local details.

3.5. Loss

The additional angular margin loss, being effective, easy and efficient [1], is used to optimize the entire model. It is an engaging loss that directly optimizes the geodesic distance margin, which has the exact correspondence to the angular margin, and results a better geometric attribute than the previous margin penalty. Specifically, it utilizes the arc–cosine function to calculate the angle \( \theta_i \) between the current feature and the target weight. After that, an additional angle margin \( m \) is added to the calculated target angle \( \theta_i \) and the target logit is obtained by the cosine function. Then the target logit is re-scaled by a fixed feature norm \( s \). The logits then go through the softmax function and contribute to the cross entropy loss:

\[
\text{Loss} = \frac{1}{N} \sum_{i=1}^{N} \log \left( \frac{e^{\cos(\theta_i + m)}}{\sum_{j=1}^{N} e^{\cos \theta_j}} \right),
\]

where \( N \) denotes the number of samples.

4. Experiments

In this section, the proposed CVSAN is evaluated on masked face datasets. First, we introduce the data, which consist of one training dataset and six test datasets. Then the evaluation metrics and experimental settings are introduced. Finally, we evaluate the effectiveness of CVSAN by a series of experiments.

4.1. Dataset

4.1.1. Training data

We select the large-scale face dataset VGGFace2 [9], which contains about 3 million images of 9,131 identities. There are approximately 362 images for each identity, which are different in posture, age, race, and light. There are 8,631 identities in the training part and 500 identities in the verification part. This dataset only contains the unmasked images for each identity. We generate masked face data based on the masked face generation tool [18].

It utilizes an existing face detection algorithm to recognize face tilt and identify six key features of the face. Then, the corresponding mask template is selected based on the inclination of the face and transformed according to the six key features to make it completely consistent with the face. The masked face data and the original data are mixed together to form the Masked VGGFace2 dataset. The statistics of Masked VGGFace2 are shown in Table 1. More specifically, we apply a randomly selected mask (surgical, N95, cloth) to each image. The three types of masks we choose are typi-

1. https://github.com/aqeelanwar/MaskTheFace.
cally used in daily life. Some sample images of training data are shown in Fig. 3(a). Fig. 4(a) shows the distribution of unmasked and different types of masked images in Masked VGGFace2.

4.1.2. Test Data

To evaluate the performance of different algorithms, we generate five masked versions (SM-LFW, Masked LFW, Masked AgeDB, Masked CFP-FF, and Masked CFP-FP) of the commonly used face recognition datasets. In addition, LFW without masked face data is also used to evaluate different algorithms.

Labelled Faces in the Wild (LFW) [47] is the most commonly used benchmark dataset to evaluate the performance of face recognition algorithms. It contains 5,749 identities and 13,233 images. By running the masked face generation program, the original LFW face data is synthesized into a masked face data, named SM-LFW. SM-LFW includes 12,769 face images, each with a mask. To simulate the randomness of whether people wear masks in the real environment, we mix the original face data and the masked face data to form a new dataset called Masked LFW. The total number of Masked LFW images is the sum of images of LFW and SM-LFW. We use the standard protocol mentioned in [47] to evaluate 6,000 image pairs.

AgeDB [48] is the first manually collected in-the-wild age dataset. It includes 568 famous characters (e.g., writers, actors, and scientists) and 16,488 images. Each image has accurate identity, age, and gender attributes. The youngest and oldest are 1 and 101 years old, respectively. The Masked AgeDB is generated in the same way as the Masked LFW. It contains 568 identities and a total of 24,521 images. Generally, the original AgeDB contains four verification schemes, where the compared faces have an age difference of 5, 10, 20 and 30 years, respectively. In our experiments, we select the most challenging scheme (Masked AgeDB30) for evaluation. It contains 6,000 comparisons.

Celebrities in Frontal-Profile in the Wild (CFP) [49] is also a commonly used dataset to evaluate the performance of face recognition algorithms. It contains the face images of 500 celebrities in frontal and profile views. We generate the Masked CFP dataset using the original CFP. It contains 500 identities and a total of 12,557 images. The dataset contains two verification protocols: one comparing only frontal faces (CFP-FF), the other comparing frontal and profile faces (CFP-FP). Each protocol consists of 7000 comparisons. In our experiments, both verification protocols (Masked CFP-FF and Masked CFP-FP) are considered.

In addition, we also consider a real masked face dataset MFR2 [18], which contains 53 identities with a total of 269 images. The

| # Identities | # Images | # Avg(images/identity) | # Test pairs |
|--------------|----------|------------------------|--------------|
| VGGFace2     | 8,631    | 1,016,744              | 350          |
| Masked VGGFace2 | 8,631 | 4,747,011              | 550          |
| LFW          | 5,749    | 13,233                 | 2.3          |
| Masked LFW   | 5,749    | 12,769                 | 2.2          |
| Masked AgeDB30 | 568  | 24,521                 | 43.17        |
| Masked CFP-FF | 500   | 12,557                 | 2.511        |
| Masked CFP-FP | 500   | 12,557                 | 2.511        |
| MFR2         | 53       | 269                    | 5            |

Fig. 3. Some examples in the training set and test sets. (a) Masked VGGFace2 dataset, which contains the original faces and the generated masked faces, (b) Test datasets, from top to bottom are Masked LFW, Masked AgeDB30, Masked CFP, Masked CFP-FP and MFR2.

Fig. 4. The proportions of different types of face images in the five datasets.
protocol used in [18] contains 848 comparisons. The number of comparisons used in our experiments is five times larger than that in [18]. Specifically, 2120 positive pairs and 2120 negative pairs are randomly selected in our experiments to better match the real-world scenarios.

Some sample images of test data are shown in Fig. 3(b). Important statistics of these datasets are summarized in Table 1. In addition, we show the proportions of unmasked and different types of masked images in test datasets in Fig. 4(b–e).

4.2. Performance metrics and experimental settings

4.2.1. Performance metrics

To analyze the performance of the trained model more accurately, Max Accuracy, Accuracy @ FAR = 0.1% and TPR @ FAR = 0.1% are employed as the evaluation metrics. Next, we describe the calculation process of them in detail. TP, TN, FP and FN are true positive, true negative, false positive, and false negative, respectively.

**Max Accuracy (%)**: The maximum accuracy in identifying image pairs as the same identity or different identities.

\[ \text{Max Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \]  

**Accuracy @ FAR = 0.1% (%)**: The accuracy in identifying image pairs as the same identity or different identities at the selected threshold for which the false acceptance rate (FAR) is 0.1%.

\[ \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad \text{s.t.} \quad \frac{FP}{TP+FP} = 0.1%. \]  

**TPR @ FAR = 0.1% (%)**: The ratio of the true positive samples detected by the network to the input image pairs to all the true positive samples at the selected threshold for which the false acceptance rate (FAR) is 0.1%.

\[ \text{TPR} = \frac{TP}{TP+FN}, \quad \text{s.t.} \quad \frac{FP}{TP+FP} = 0.1%. \]  

4.2.2. Experimental settings

For preprocessing, the input face images are cropped and adjusted to 160 × 160. Each pixel ([0, 255]) in the RGB images is normalized by subtracting 127.5 and divided by 128. We use the SGD optimizer with a momentum of 0.9 and weight decay of 5e-4. The learning rate starts from 0.01 and is divided by 10 at 132 k, 264 k steps. We set the training batch size to 36, and complete the training process after 600 K iterations. The dimensional-ity of the masked face features of the models is set to 512. We set the angular margin to 0.5. The CVSAN is implemented using the Pytorch framework and trained on one GPU of NVIDIA GeForce GTX2080.

We compare our method with some baseline face recognition methods. **FaceNet** [4] learns the maps from the face images to the compact euclidean space directly, where the distance directly corresponds to the face similarity measure. **ArcFace** [1] proposes an additive angular margin loss to obtain high discriminative features for face recognition. **SEResNet50** embeds SE [50] blocks in the ResNet50 structure to enhance the interdependencies between channels. **BoTNet** [36] replaces spatial convolution with global self-attention in the bottleneck blocks of ResNet50. **MTArcFace** [16] combines ArcFace loss with mask-usage classification loss. **CropFace** [51] is a masked face recognition method which is a combination of the cropped face recognition method and the convolution block attention module.

4.3. Comparisons to the state-of-the-art

**Result on LFW and SM-LFW**: The experimental results of the proposed CVSAN and baselines on LFW and SM-LFW are shown in Table 2. All models are trained with the Masked VGGFace2 dataset, which includes both unmasked and masked face images. These baselines can be divided into three categories: traditional face recognition algorithms (FaceNet and ArcFace), attention-based algorithms (SEResNet50 and BoTNet) and masked face recognition algorithms proposed recently (MTArcFace and CropFace).

On LFW, our model outperforms ArcFace by 0.8% (Max Accuracy), 3.53% (Accuracy@FAR = 0.1%) and 7.07% (TPR@FAR = 0.1%). Compared with FaceNet, CVSAN can also achieve satisfactory results. This shows that the traditional models suffer performance degradation when trained on masked face data and tested on LFW.

On SM-LFW, CVSAN improves the TPR@FAR = 0.1% from 89.90% to 95.16% compared to the benchmark model ArcFace. The performance improvements clearly show that the proposed algorithm can effectively bridge the gap between training data and test data on the masked face images, which effectively improves the performance of MFR.

For the comparison with two attention-based models, CVSAN significantly improves the performance. In LFW, CVSAN has improved by 0.52% (Max Accuracy), 1.87% (Accuracy@FAR = 0.1%) and 3.67% (TPR@FAR = 0.1%) compared with the best-performing BoTNet. In SM-LFW, the performance of CVSAN has also been improved to varying extents. This result shows that CVSAN can learn more discriminative representations and is robust to face recognition with or without a mask.

In addition, two recently proposed MFR methods are also used as comparison algorithms. Because CorpFace crops the lower half of the face to eliminate the influence of mask occlusion, its performance on the full face data is very poor. On the SM-LFW data, compared with the performance of MTArcFace, the proposed method increases the three performance metrics by 1.02%, 3.33% and 6.59%, respectively.

**Result on several different masked datasets**: Further, in order to be more in line with the random scenario of whether people wear masks in reality, we evaluate our model on four masked datasets including Masked LFW, Masked AgeDB30, Masked CFP-FF, Masked CFP-FP and MFR2 [18]. The CVSAN is also compared with

### Table 2

| Metrics       | Max Accuracy (%) | Accuracy@FAR = 0.1% (%) | TPR@FAR = 0.1% (%) |
|---------------|------------------|-------------------------|--------------------|
| Datasets      | LFW              | SM-LFW                  | LFW                | SM-LFW            | LFW                | SM-LFW            |
| FaceNet       | 98.97            | 98.55                   | 96.93              | 86.33             | 93.97              | 72.77             |
| ArcFace       | 98.55            | 98.00                   | 95.27              | 94.90             | 90.63              | 89.90             |
| SEResNet50    | 98.82            | 97.63                   | 96.80              | 91.95             | 93.70              | 84.03             |
| BoTNet        | 98.83            | 98.18                   | 96.93              | 95.52             | 94.03              | 91.13             |
| MTArcFace     | 99.10            | 98.08                   | 97.53              | 94.22             | 95.17              | 88.57             |
| CropFace      | 97.60            | 97.53                   | 89.53              | 90.92             | 79.20              | 81.93             |
| CVSAN         | 99.35            | 99.10                   | 98.80              | 97.55             | 97.70              | 95.16             |
other baselines and the specific experimental results are shown in Table 3. The overall experimental results show that the proposed method outperforms other baselines in masked face recognition tasks. In Masked LFV, the performance of CVSAN is about 0.4% higher than that of SEResNet50. This shows that introducing self-attention to augment convolutional structure is more conducive to extract discriminative masked face features. In Masked AgeDB30, the available information of face is reduced due to age differences. In this case, the proposed model significantly improves the performance of MFR compared to other baselines. For example, our model has an accuracy improvement of 3% compared to the best baseline (MTArcFace), while the number of parameters is only increased by 2.1 M. When the Masked CFP-FF is used to evaluate the performance of the models, the available facial information is reduced due to the inclusion of profile faces. Here, the performance of CVSAN is nearly 8% higher than that of FaceNet. In addition, compared with the MFR algorithms proposed recently, the proposed model also has significant performance improvements. As shown in Table 3, compared with the MTArcFace, although the performance on Masked LFV is almost the same, our model has significant improvements on the other test datasets. Regarding the performances of different models on real mask data, we use MFR2 [18] for validation. Our model improves performance by 1.57% compared to the best baseline model MTArcFace. This shows that our model handles real-world masked faces well by fusing convolutional features with self-attention features. We also compare the performance of the CVSAN and baselines on five test datasets under the performance metric of Max Accuracy. As shown in Fig. 5, we draw the ROC curves of different models on these five datasets.

4.4. Ablation studies

Effect of mask occlusion on face recognition. Two models (FaceNet and ArcFace) that achieve state-of-the-art performance on common face recognition are evaluated on the original LFV and SM-LFW. They use loss based on Euclidean distance and loss based on angular/cosine margin respectively. The results are shown in Table 4. Compared with LFV, the performance of FaceNet on SM-LFW has decreased, and Max Accuracy, Accuracy @ FAR = 0.1% and TPR @ FAR = 0.1% have decreased by 0.60%, 11.95% and 24.12% respectively. The performance of ArcFace is similar to FaceNet. These results show that when the test data contains masked faces, the classic face recognition model cannot achieve satisfactory results. Therefore, the model designed for common face recognition is not suitable for MFR tasks. This shows that mask occlusion will cause loss of local feature information, and the traditional model that only relies on the convolution operation lacks the ability to capture remote interaction information of the face. This information can usually be used as a supplementary feature to distinguish different face images to improve recognition performance.

Effects of different branches on the Conv-MHSA stage. We investigate the effectiveness if only one of the two paths in the Conv-MHSA stage is used. In Fig. 2, the Conv-MHSA stage can be divided into two paths: 1) Conv-C using only convolution blocks and 2) Conv-M using MHSA blocks that add self-attention to the convolution. In Table 5, the experimental results show that the performance of the proposed model decreases by about 1%–2% when one of the two paths is removed from the Conv-MHSA stage.

Table 3
Under Accuracy @ FAR = 0.1%, the performances (%) of the proposed method and the baselines on the masked face datasets.

| Dataset         | FaceNet | ArcFace | SEResNet50 | BoTNet | MTArcFace | CropFace | CVSAN |
|-----------------|---------|---------|------------|--------|-----------|----------|-------|
| Masked LFV      | 98.60   | 98.47   | 98.40      | 98.84  | 99.23     | 98.18    | 99.22 |
| Masked AgeDB30  | 61.65   | 69.83   | 66.88      | 71.62  | 74.12     | 63.85    | 77.63 |
| Masked CFP-FF   | 95.31   | 92.81   | 91.37      | 90.51  | 95.07     | 85.00    | 97.29 |
| Masked CFP-FP   | 78.13   | 73.61   | 68.56      | 67.17  | 72.59     | 64.56    | 86.27 |
| MFR2            | 91.24   | 90.75   | 91.44      | 94.72  | 94.79     | 92.95    | 96.36 |
| # params.       | 22.8 M  | 25.6 M  | 57.2 M     | 45.5 M | 56.8 M    | 44.3 M   | 58.9 M |

Table 4
Results of FaceNet and ArcFace on the original LFV and SM-LFW.

| Metrics      | LFW | SM-LFW |
|--------------|-----|--------|
| Methods      | Max Accuracy (%) | Accuracy@FAR = 0.1% (%) | TPR@FAR = 0.1% (%) |
| FaceNet      | 99.15 | 61.65 | 95.31 | 78.13 | 91.24 | 22.8 M |
| ArcFace      | 99.48 | 73.61 | 91.44 | 73.61 | 90.75 | 25.6 M |
| SEResNet50   | 98.40 | 68.56 | 91.44 | 68.56 | 90.75 | 57.2 M |
| BoTNet       | 98.84 | 67.17 | 94.72 | 67.17 | 91.44 | 45.5 M |
| MTArcFace    | 99.23 | 72.59 | 94.79 | 72.59 | 91.44 | 56.8 M |
| CropFace     | 98.18 | 64.56 | 92.95 | 64.56 | 90.75 | 44.3 M |
| CVSAN        | 99.22 | 86.27 | 96.36 | 86.27 | 91.24 | 58.9 M |

Fig. 5. The ROC curves of all models on the five test datasets under the performance metric of Max Accuracy.
We also find that the performance of the model using only the convolution blocks is higher than that of using only the MHSA blocks in Conv-MHSA stage. This suggests that the Conv-MHSA stage, which incorporates convolutional feature maps (local features) and self-attentive feature maps (global features), is a possible method to improve the model performance.

Effect of training data augmentation. In the experiment, we use the masked face generation tool to generate a masked version of Masked VGGFace2 from the original VGGFace2 to augment the training data. An example of the mask image we generated is shown in Fig. 3(a). We compare the CVSAN model trained using Masked VGGFace2 with the original VGGFace2. Fig. 6 shows the experimental results. Normally, it is unfair to compare the performance of the model trained on datasets of different sizes, because the model trained on larger datasets can generally enjoy more information and tends to make better decisions. In our case, the additional images in Masked VGGFace2 are generated from the original images and they do not add any additional information for training models, so the comparison is fair enough.

In Fig. 6, the blue bar represents the performance of the models trained on VGGFace2, and the yellow bar represents the performance of the models trained on Masked VGGFace2. We use four masked datasets to evaluate the performance of the model. When the test data contains both unmasked face and masked face data, the performance of the models trained on Masked VGGFace2 is better than that of the models trained on VGGFace2. These experimental results show that the network trained with masked face data is robust and effectively compensates for the performance degradation of the model under partial occlusion.

Attention Area Visualization. To better understand how MHSA focuses on face images in the CVSAN model, we visualize the attention map $S$ (Eq. 1) of the last MHB2_out layer in Fig. 7. We select four key points of the face in the image for visualization. The white and green points are the areas near the left eye and the right eye in the face images, respectively. From the position of the visualized image, it is found that MHSA seems to focus on the features of the area around the two eyes and the upper part of the face. The orange-red and red points are located near the nose and mouth, respectively. Although they are covered by masks, MHSA still recognizes effective facial features by learning long-range features (near the two eyes and the upper part of the face). In addition, we also show the attention maps of the four key points of the face image occluded by both mask and sunglasses in the fourth row of Fig. 7. When the eyes are occluded by sunglasses and the mouth is occluded by mask, the attention maps of these three points seems to pay more attention to the contour information of the face. Furthermore, we also notice that the attention map in the nose area pays attention to vague facial contour information, which still concerns the global information. The visualization of the attention area in Fig. 7 further shows that our model takes advantage the ability of self-attention to learn long range interactive information to more accurately locate the forehead-eye area, and capture the global information of the image.

4.5. Results on MFR challenge protocol

To further validate the model performance under real masked faces, we test our model and competitive baseline models on the InsightFace track of the Masked Face Recognition (MFR) challenge [42]. For the InsightFace track, researchers manually collected a large-scale test set of masked faces with 7 K identities. We follow the relevant regulations of the competition and employ MaskTheFace [18] to generate masked face images. Moreover, we modify the input size of our model and baselines from 160 × 160 to 112 × 112 to satisfy the requirement of competition. Based on the performance of the experiments in Section 4.3, we select three baselines (i.e., SEResNet50, BoTNet and MTArcFace) for comparisons. As shown in Fig. 8, we provide the final evaluation results on the InsightFace track. We can see that the proposed method has an accuracy improvement of 1.86% compared to the best baseline MTArcFace. This shows that our model is robust compared to baselines on the large-scale real-mask face test dataset.
5. Conclusions

In this work, we combine the visual Transformer and convolution to design a masked face recognition model: the Convolutional Visual Self-Attention Network (CVSAN). In our model, the convolutional feature map with local information is fused to the self-attention feature map capable of modeling long-range dependencies. In view of the lack of large-scale masked face datasets for training, we use the face key point technology to generate a dataset containing masked faces from the original VGGFace2. Experimental results on multiple test datasets show that the proposed model achieves satisfactory performance in masked face recognition. Our future work will focus on extending the applicability of this method to a wider range of mask types and considering more occlusion situations.

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.

Acknowledgement

This work is partially supported by a grant from the National Science Foundation of China (No. 62032017, No.62272368), the Innovation Capability Support Program of Shaanxi, the Key Research and Development Program of Shaanxi (No. 2021ZDLG03-09, No. 2021ZDLG07-02, No. 2021ZDLG07-03), and The Youth Innovation Team of Shaanxi Universities.

References

[1] J. Deng, J. Guo, N. Xue, S. Zafeiriou, Arfacce: Additive angular margin loss for deep face recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019, Computer Vision Foundation/IEEE, 2019, pp. 4690–4699.

[2] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, Squeeze-and-excitation networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, 40 (2018) 922–937.

[3] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, W. Liu, Cosface: Large margin cosine loss for deep face recognition, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018, IEEE Computer Society, 2018, pp. 5265–5274.

[4] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7–12, 2015, IEEE Computer Society, 2015, pp. 815–823.

[5] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23–28, 2014, IEEE Computer Society, 2014, pp. 1701–1708.

[6] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39 (2017) 1137–1149.

[7] D. Yi, Z. Lei, S. Liao, S.Z. Li, Learning face representation from scratch, CoRR abs/1411.7923 (2014).

[8] Y. Guo, L. Zhang, Y. Hu, X. He, J. Gao, Ms-celeb-1m: A dataset and benchmark for large-scale face recognition, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision - ECCV 2016 –14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part III, volume 9907 of Lecture Notes in Computer Science, Springer, 2016, pp. 87–102.

[9] Q. Cao, L. Shen, W. Xie, O.M. Parkhi, A. Zisserman, Vggface2: A dataset for recognising faces across pose and age, in: 13th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2018, Xi’an, China, May 15–19, 2018, IEEE Computer Society, 2018, pp. 67–74.

[10] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (2017) 84–90.

[11] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, 2015.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7–12, 2015, IEEE Computer Society, 2015, pp. 1–9.

[13] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, IEEE Computer Society, 2016, pp. 770–778.

[14] X. Tan, B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, IEEE Trans. Image. Process. 19 (2010) 1635–1650.

[15] S.T. Chaudhari, A. Kale, Face normalization: Enhancing face recognition, in: 3rd International Conference on Emerging Trends in Engineering and Technology, ICTET 2010, Goa, India, November 19–21, 2010, IEEE Computer Society, 2010, pp. 520–525.

[16] D. Montero, M. Nieto, P. Leskovsky, N. Aginako, Boosting masked face recognition with multi-task arface, CoRR abs/2104.09874 (2021).

[17] Nutt finds flaws in facial checks on people with mask bioms, Biometric Technology Today 2020 (2020) 2.

[18] A. Anwar, A. Raychowdhury, Masked face recognition for secure authentication, CoRR abs/2008.11104 (2020).

[19] W. Harir, Efficient masked face recognition method during the COVID-19 pandemic, CoRR abs/2105.03026 (2021).

[20] C. Li, S. Ge, D. Zhang, J. Li, Look through masks: Towards masked face recognition with de-occlusion distillation, in: C.W. Chen, R. Cucchiara, X. Hou, G. Qi, E. Ricci, Z. Zhang, R. Zimmermann (Eds.), MM ‘20: The 28th ACM International Conference on Multimedia, Virtual Event/ Seattle, WA, USA, October 12–16, 2020, ACM, 2020, pp. 3016–3024.

[21] D. Montero, M. Nieto, P. Leskovsky, N. Aginako, Boosting masked face recognition with multi-task arface, CoRR abs/2104.09874 (2021).

[22] I. Bello, B. Zoph, Q. Le, A. Vaswani, J. Shin, Attention augmented convolutional networks, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 – November 2, 2019, IEEE, 2019, pp. 3285–3294.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA, 2017, pp. 5998–6008.

[24] C. Rysa, K. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, Deformable DETR: deformable transformers for end-to-end object detection, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021, OpenReview.net, 2021.

[25] M. Zheng, P. Cao, X. Wang, H. Li, H. Dong, End-to-end object detection with adaptive clustering transformer, CoRR abs/2011.09315 (2020).

[26] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, H. Xia, End-to-end video instance segmentation with transformers, CoRR abs/2011.14503 (2020).

[27] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021, OpenReview.net, 2021.

[28] Y. LeCun, Y. Bengio, Convolutional networks for images, speech, and time series, 1998.

[29] M. Wang, W. Deng, Deep face recognition: A survey, Neurocomputing 429 (2021) 215–244.

[30] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2020) 2011–2023.

[31] C. Xiong, X. Zhao, D. Tang, J. Karlekar, S. Yan, T. Kim, Conditional convolutional neural network for modality-aware face recognition, in: 2015 IEEE
International Conference on Computer Vision, ICCV 2015, IEEE Computer Society, pp. 3667–3675.

[3] J. Sun, X. Wang, X. Tang, Sparsefyng neural network connections for face recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, IEEE Computer Society, 2016, pp. 4856–4864.

[4] J. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8–13 2014, Montreal, Quebec, Canada, 2014, pp. 1988–1996.

[5] A. Srivinas, T. Lin, N. Parmar, J. Shi, P. Abbeel, A. Vaswani, Bottleneck transformers for visual recognition, CoRR abs/2101.11605 (2021).

[6] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, Y. Wang, Transformer in transformer, CoRR abs/2103.00112 (2021).

[7] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention, in: M. Meila, T. Zhang (Eds.), Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event, volume 139 of Proceedings of Machine Learning Research, PMLR, 2021, pp. 10347–10357.

[8] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, CoRR abs/2103.14030 (2021).

[9] P. Shaw, J. Uszkoreit, A. Vaswani, Self-attention with relative position representations, in: M.A. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1–6, 2018, Volume 2 (Short Papers), Association for Computational Linguistics, 2018, pp. 464–468.

[10] M. Gong, P. Peng, Y. Huang, Y. Tian, Masked face recognition with generative data augmentation and domain constrained ranking, in: C.W. Chen, R. Cucchiara, X. Hua, G. Qi, E. Ricci, Z. Zhang, R. Zimmermann (Eds.), The 28th ACM International Conference on Multimedia, Virtual Event/ Seattle, WA, USA, October 12–16, 2020, ACM, 2020, pp. 2246–2254.

[11] J. Deng, J. Guo, X. An, Z. Zhu, S. Zafeiriou, Masked face recognition challenge: The insightface track report, in: ICCCW, IEEE, 2021, pp. 1437–1444.

[12] N. Parmar, P. Ramachandran, A. Vaswani, I. Bello, A. Levskaya, J. Shlens, Stand-alone self-attention in vision models, in: H.M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E.B. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada, 2019, pp. 68–80.

[13] H. Zhao, J. Jia, V. Kolm, Exploring self-attention for image recognition, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020, IEEE, 2020, pp. 10073–10082.

[14] J. Hu, L. Shen, S. Albanie, G. Sun, A. Vedaldi, Gather-excite: Exploiting feature context in convolutional neural networks, in: S. Bengio, H.M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3–8, 2018, Montréal, Canada, 2018, pp. 9423–9433.

[15] S. Woo, J. Park, J. Lee, I.S. Kweon, CBAM: convolutional block attention module, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VI, Volume 11211 of Lecture Notes in Computer Science, Springer, 2018, pp. 3–19.

[16] G. Huang, M. Mattar, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database forstudying face recognition in unconstrained environments, Tech. rep. (2008).

[17] S. Moschoglou, A. Papaioannou, C. Sagounas, J. Deng, I. Kotsia, S. Zafeiriou, AgeDB: The first manually collected, in-the-wild age database, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21–26, 2017, IEEE Computer Society, 2017, pp. 1997–2005.

[18] S. Sengupta, J. Chen, C.D. Castillo, V.M. Patel, R. Chellappa, D.W. Jacobs, Fronto ngal profile face verification in the wild, in: 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2016, Lake Placid, NY, USA, March 7–10, 2016, IEEE Computer Society, 2016, pp. 1–9.

[19] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018, Computer Vision Foundation/ IEEE Computer Society, 2018, pp. 7132–7141. URL: http://openaccess. thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR2018.html. doi: 10.1109/CVPR.2018.00745.

[20] Y. Li, K. Guo, Y. Lu, L. Liu, Cropping and attention based approach for masked face recognition, Appl. Intell. 51 (2021) 3012–3025.