MobileFaceSwap: A Lightweight Framework for Video Face Swapping

Zhiliang Xu[1], Zhibin Hong[1]*, Changxing Ding[2], Zhen Zhu[3], Junyu Han[1], Jingtuo Liu[1], Errui Ding[1]

[1] Baidu Inc.
[2] South China University of Technology
[3] University of Illinois at Urbana-Champaign
{xuzhiliang, hongzhibin, hanjunyu, liujintuo, dingerrui}@baidu.com,
chxding@scut.edu.cn, zhenzhu4@illinois.edu

Abstract

Advanced face swapping methods have achieved appealing results. However, most of these methods have many parameters and computations, which makes it challenging to apply them in real-time applications or deploy them on edge devices like mobile phones. In this work, we propose a lightweight Identity-aware Dynamic Network (IDN) for subject-agnostic face swapping by dynamically adjusting the model parameters according to the identity information. In particular, we design an efficient Identity Injection Module (IIM) by introducing two dynamic neural network techniques, including the weights prediction and weights modulation. Once the IDN is updated, it can be applied to swap faces given any target image or video. The presented IDN contains only 0.50M parameters and needs 0.33G FLOPs per frame, making it capable for real-time video face swapping on mobile phones. In addition, we introduce a knowledge distillation-based method for stable training, and a loss reweighting module is employed to obtain better synthesized results. Finally, our method achieves comparable results with the teacher models and other state-of-the-art methods.

Introduction

Recently, face swapping has drawn much attention from the research community, and it has many applications in visual effects. Face swapping means transferring the identity information of the source image to the target image while keeping the other attributes like the expression and background of the target image unchanged. Face swapping has achieved rapid progress with deep learning. However, most of these methods require many parameters and involve high computation costs. For example, FaceShifter (Li et al. 2020a) contains two stages for face swapping. The first stage alone has nearly 421M parameters and 97.4G FLOPs. FSGAN (Nirkin, Keller, and Hassner 2019) proposes a more complicated algorithm, which has over 226M parameters and 2440G FLOPs in total. Despite SimSwap (Chen et al. 2020) claims that it presents an efficient network, it still has 107M parameters and 55.7G FLOPs. It is not only challenging to deploy these methods on edge devices, but these methods also require plenty of time for video face swapping, which means we cannot deploy them in real-time applications.

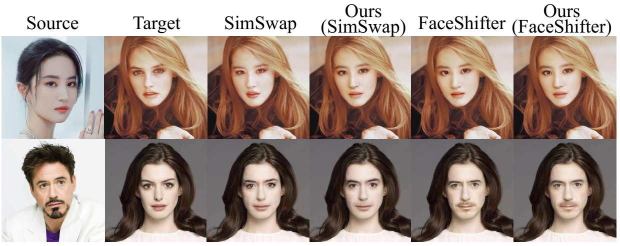

Figure 1: Face swapping results of our and the teacher models by swapping the face of the target image with the source image. Our method decreases the computations of SimSwap and FaceShifter by 146 and 207 times while preserving the visual fidelity.

A natural idea to address the challenges above is using model compression techniques to produce a lightweight network for face swapping. However, when we apply the channel pruning technique and shrink the network to significantly compress the SimSwap or FaceShifter, we observe that it is troublesome to obtain pleasant face swapping results. In particular, some of the generated images have apparent artifacts, or the identities of the generated images may not be similar to the source images as the compressed model is insufficient to inject identity information with the limited network capacity. Inspired by the subject-aware face swapping technique (Perov et al. 2020), we also find that the better swapped images can be obtained if we fix the source image and train a lightweight model for a specific identity. Therefore, to achieve subject-agnostic and real-time face swapping, an intuitive idea is to adjust the parameters of a neural network according to the identity information.

Inspired by the dynamic neural network techniques, we propose a lightweight Identity-aware Dynamic Network (IDN) for real time face swapping. To efficiently inject identity information, we also design an Identity Injection Module (IIM) using weights prediction (De Brabandere et al. 2016) and weights modulation (Karras et al. 2020) to adjust the parameters of IDN. In this way, the IDN can be updated given the needed identity information, and then we can achieve fast face swapping for any target image or video. Our method can significantly reduce the parameters and computations using these designs and achieve compa-

*Corresponding author.

rable results with other state-of-the-art methods. The proposed IDN has only 0.50M parameters and 0.33G FLOPs per frame for video face swapping when the input size is 224×224. This means that we reduce the parameters and computations of the recent face swapping methods by more than 100 times. Without further optimization such as quantization, our model can achieve real-time face swapping on the mobile phone with MediaTek Dimensity 1100 chip, arriving at 26 FPS.

Generally, training a neural network for face swapping is unstable, and the generated images may have obviously artifacts. We notice that if we transfer the face swapping to a paired training task using knowledge distillation (Hinton, Vinyals, and Dean 2015), we can achieve a more stable training process and get better results. Therefore, we employ a well-trained network as the teacher and train our lightweight network as the student. However, the teacher model may also produce some failure cases, such as the generated image having a low identity similarity with the source image. The student model can be misled by these failure cases and produce suboptimal results. In this paper, we propose a loss reweighting module to alleviate this problem. In particular, we evaluate the quality of the teacher outputs and adaptively adjust the weighting of distillation loss simultaneously. In this way, the student network can learn from better teacher outputs and therefore obtain better generated results with fewer artifacts and higher identity similarity.

The contributions of our paper are listed as follows:

1. We propose a real-time framework for video face swapping. It contains only 0.50M parameters and 0.33G FLOPs, and arrives at 26 FPS on the mobile phone.

2. We present an Identity Injection Module (IIM), which utilizes the weights prediction and weights modulation for more efficient identity information injection to build an Identity-aware Dynamic Network (IDN).

3. To stabilize the learning process, we train the proposed network using a knowledge distillation framework and propose a loss reweighting module to improve the generated results qualitatively and quantitatively.

Related Work

Face swapping. Deep learning based face swapping has achieved significant improvement. The popular DeepFaceLab (Perov et al. 2020) trains an Encoder-Decoder for subject-aware face swapping and has achieved appealing results. However, this method only supports face swapping for two specific identities. Recently, additional subject-agnostic methods have been proposed that are more convenient to be employed. These methods can be roughly divided into two categories: source-oriented and target-oriented methods.

The source-oriented methods first transfer the pose and expression of the source image to the target image, and then they apply a blending method to obtain the swapped face image. For example, (Nirkin et al. 2018) employ 3DMM (Blanz and Vetter 1999) to align the source image with the target image. FSGAN (Nirkin, Keller, and Hassner 2019) trains two respective models to implement the face reenactment and blending process. However, these methods are unstable and prone to generating artifacts, such as unnatural color. Target-oriented methods blend the features of the source image and target image to obtain the swapped face. FaceShifter (Li et al. 2020a) leverages a two-stage framework. The first stage is for face swapping, and the second stage is for occlusion processing. SimSwap (Chen et al. 2020) utilizes AdaIN (Huang and Belongie 2017) to inject the identity information into the target feature map. FaceController (Xu et al. 2021) proposes a face representation based on 3DMM coefficients, as well as style and identity embedding, which can achieve more attribute editing than face swapping. HifiFace (Wang et al. 2021) utilizes 3DMM coefficients to preserve the face shape of the source image for face swapping. However, the above methods have many parameters and computations that restrict the usage of these methods for face swapping.

Dynamic neural networks. A dynamic neural network refers to one that adapts its structure or parameters to the input during inference, which can result in greater computational efficiency. This was first proposed by (De Brabandere et al. 2016). This technique has been applied to other applications such as style transfer (Shen, Yan, and Zeng 2018), super-resolution (Jo et al. 2018; Hu et al. 2019), and image-to-image translation (Liu et al. 2019; Park et al. 2019), etc.

Knowledge distillation. Knowledge distillation (Hinton, Vinyals, and Dean 2015) was proposed for transferring the knowledge in a larger teacher network to a smaller student network, which is widely used for model compression (Chen et al. 2017; Wang et al. 2019). Recently, (Yim et al. 2017; Li et al. 2020b; Jin et al. 2021) use knowledge distillation to compress the generative adversarial network (Goodfellow et al. 2014) for image-to-image translation. These methods have also confirmed that utilizing knowledge distillation can improve the training stability for an unpaired training task by transferring it to a paired learning. Therefore, we introduce knowledge distillation into face swapping.

Method

In this section, we will first describe the network architecture of our MobileFaceSwap, including the details of the Identity Injection Module (IIM), Identity-aware Dynamic Network (IDN), and the weakly semantic fusion module. Then, we introduce the knowledge distillation and our loss reweighting module to address the training stability problem and synthesis better swapped results. The overall framework is illustrated in Fig. 2.

Network Architecture

The main architecture of our method contains two neural networks. One is an Identity Injection Network (IIN) that has several Identity Injection Modules (IIM) to obtain the parameters required, while the other uses these parameters to construct a lightweight network named Identity-aware Dynamic Network (IDN) for the inference process. Given the identity representation of the source image by ArcFace (Deng et al. 2019), the IIM contains two dynamic neural network techniques to inject identity information to the IDN.

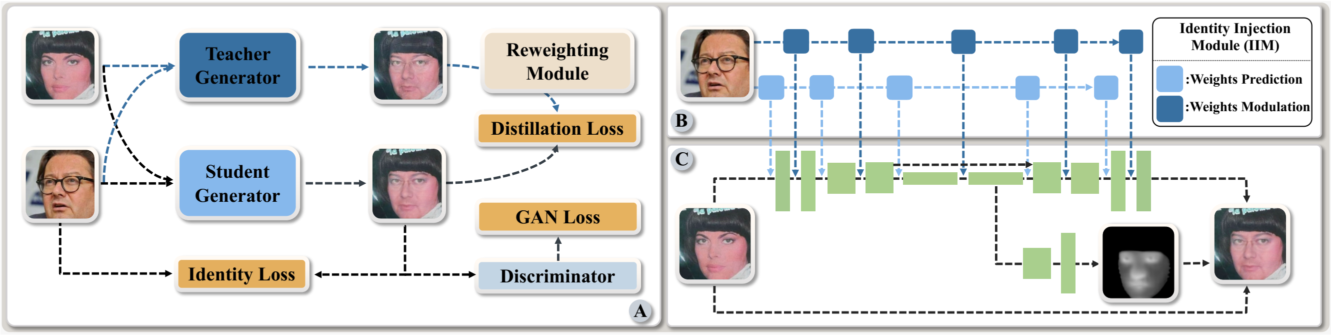

Figure 2: MobileFaceSwap framework: (a) The overall training process. (b) The Identity Injection Network (IIN) of MobileFaceSwap, which contains several Identity Injection Modules (IIM) and utilizes the identity information to predict or modulate the weights of the IDN. (c) The architecture of the Identity-aware Dynamic Network (IDN) and a weakly semantic fusion module for face swapping, that contain only 0.50M parameters and 0.33G FLOPs in total.

Once we finish the injection process, we can swap faces with any target image or video using this lightweight IDN.

The IDN is simplified from U-Net (Ronneberger, Fischer, and Brox 2015) by replacing the standard convolution with the depthwise, and pointwise convolution (Chollet 2017). To modify the parameters of IDN according to a given source image, we introduce weights prediction (De Brabandere et al. 2016), and weights modulation (Karras et al. 2020) for depthwise and pointwise convolutions, respectively. Let C_{in}, C_{out}, and K denote the input channels, output channels, and kernel size of a convolution layer, respectively. We utilize the identity embedding z_{id} as the input to predict the weights of a depthwise convolution layer as follows:

$$\mathcal{W}_d = \mathcal{F}_p(z_{id}), \tag{1}$$

where $\mathcal{W}_d \in \mathbb{R}^{C_{in} \times 1 \times K \times K}$ represents the weights of a depthwise convolution layer in IDN, and $\mathcal{F}_p$ represents the prediction module that contains several convolution layers of the IIM. Predicting the weights $\mathcal{W}_p \in \mathbb{R}^{C_{in} \times C_{out} \times 1 \times 1}$ for pointwise convolution needs more parameters since C_{out} is generally far outweigh than $K \times K$. Also, we observe that it can obtain better results by employing the weights modulation to inject the identity information into the weights of a pointwise convolution layer. Let $\mathcal{W}_p$, $\hat{\mathcal{W}}_p$, and $\tilde{\mathcal{W}}_p$ represent the origin, modulation, and demodulation weights of a pointwise convolution respectively, and i, j, and k enumerate the input/output feature maps and spatial footprint of the convolution, respectively. Then, the weights modulation technique is formulated as follows:

$$\begin{aligned} \hat{\mathcal{W}}_p^{(i,j,k)} &= \mathcal{F}_m(z_{id})^{(i)} \cdot \mathcal{W}_p^{(i,j,k)}, \\ \tilde{\mathcal{W}}_p^{(i,j,k)} &= \frac{\hat{\mathcal{W}}_p^{(i,j,k)}}{\sqrt{\sum_{i,k} (\hat{\mathcal{W}}_p^{(i,j,k)})^2 + \epsilon}}, \end{aligned} \tag{2}$$

where $\mathcal{F}_m$ is the modulation module of the IIM, which includes several fully-connected layers, and ϵ is used for number stability. By applying IIM, the IDN can achieve subject-agnostic face swapping, which contains 0.413M parameters and 0.328G FLOPs only.

In practice, it is hard to keep the background and hair of the generated images being the same as the target images. This problem also causes jitter for video face swapping. Generally, it can be solved by adding computationally intensive post-processing such as face segmentation (Yu et al. 2018). In this paper, we propose a weakly semantic fusion module to merge the background of the target image. We predict a weakly fusion mask by reusing the feature maps of the IDN. Our semantic fusion module contains only 0.083M parameters and 0.005G FLOPs. Even though our semantic fusion module is exceptionally lightweight, the predicted fusion mask is pretty well as demonstrate in Fig. 2. Finally, we have built the entire network for video face swapping, which has 0.495M parameters and needs 0.333G FLOPs in total when the input size is 224×224.

Training Objectives

Generally, training a face swapping network requires many loss functions to guarantee that the generated result meets the definition of face swapping. The competition of these different losses makes the training process unstable and easier to generate artifacts as there is no paired ground truth for the constraint. In this paper, we transfer face swapping to a paired training by introducing a knowledge distillation (Hinton, Vinyals, and Dean 2015) framework. Given a well-trained face swapping network as the teacher and the proposed network as the student, we utilize L1 loss and perceptual loss (Johnson, Alahi, and Fei-Fei 2016) between the student output I_g and the teacher output I'_g as follows:

$$\begin{aligned} \mathcal{L}_{rec} &= ||I'_g - I_g||, \\ \mathcal{L}_{per} &= \sum_{i=1}^{L} \frac{1}{N_i} ||\mathcal{F}_{VGG}^{(i)}(I'_g) - \mathcal{F}_{VGG}^{(i)}(I_g)||^2, \end{aligned} \tag{3}$$

where $\mathcal{F}_{VGG}^{(i)}$ denotes the i-th layer with N_i elements of the VGG network (Simonyan and Zisserman 2014). Our method can achieve more stable training process and better synthesized results by knowledge distillation.

However, the teacher is not perfect and some bad cases can be found in the teacher outputs. Specifically, we roughly divide these failure cases into two categories of face swapping. First, some teacher outputs cannot keep the identity

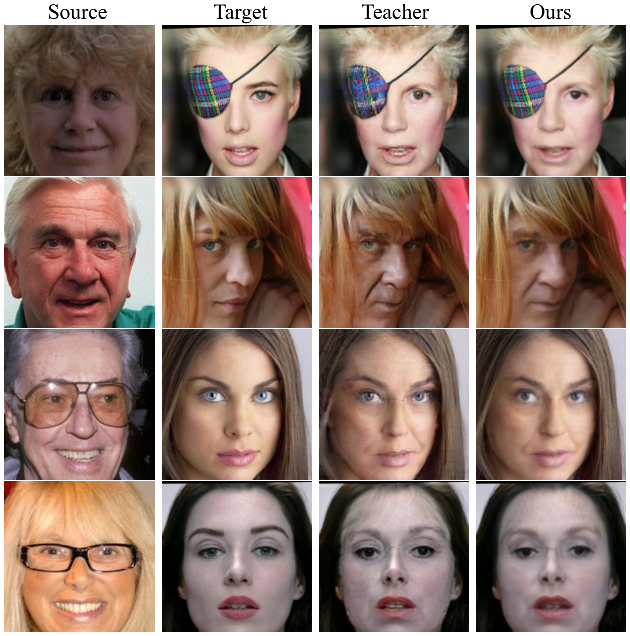

Figure 3: Comparison with the teacher (SimSwap) results.

well of the source images. Second, some teacher outputs may have unnatural results or artifacts, such as noise and a dirty forehead. If we assign an equal weight for each teacher output in Equ. 3, the student network will also learn to generate these bad cases. In this paper, we propose a loss reweighting module to alleviate this problem. We consider the identity similarity and image quality for each teacher output. Specifically, we use the square of the cosine distance between the identity representations of the teacher output and source image to measure the identity similarity. However, evaluating the quality of the image is non-trivial. Since we observe that the swapped results gradually become better during the training process, to assess the image quality of the teacher outputs properly, we assign each teacher output with a score between 0 to 1, according to the percentage of the completed iterations in the teacher training process. Subsequently, we use a ResNet-18 (He et al. 2016) to regress these scores by supervision with L2 Loss. Finally, we employ this model Q to evaluate the image quality of the teacher outputs. The final distillation loss reweighting coefficient α is calculated as follows:

$$\alpha = Cos(z_{id}, z'_{id})^2 \times Q(I'_g), \tag{4}$$

where z'_{id} denotes the identity representation of the teacher output. By introducing the loss reweighting module, not only can we improve the identity similarity between the generated image and the source image, but we can also improve the quality of swapped results.

Although we transfer the teacher knowledge to the student model using distillation loss, the supervision on identity is insufficient. The quantitative results about identity similarity drop significantly compared with the teacher. Therefore, we add a supplementary identity loss for better identity supervision following FaceShifter (Li et al. 2020a). We employ ArcFace (Deng et al. 2019) as the extractor $\mathcal{F}_{id}$ and calculate the cosine similarity between the identity representations of the source image I_s and generated image I_g.

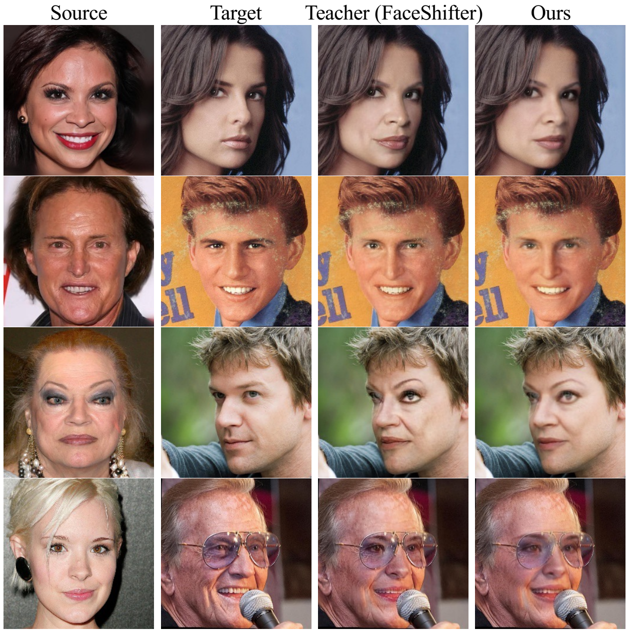

Figure 4: Comparison with the teacher (FaceShifter) results.

$$\mathcal{L}_{id} = 1 - Cos(\mathcal{F}_{id}(I_s), \mathcal{F}_{id}(I_g)), \tag{5}$$

For the fusion mask prediction, we employ weak supervision. Specifically, we only constrain the background of the generated image to keep it being the same as the target image. Compared with full supervision, our weak mask loss can better retain the attributes of the target image, such as local textures. The mask loss is defined as follows:

$$\mathcal{L}_{mask} = ||M_t[bg] - M_g[bg]||, \tag{6}$$

where M_t and M_g represent the mask of the target and generated image, respectively, $M[bg]$ denotes the background elements of the mask. We use (Yu et al. 2018) to obtain M_t by combing the semantic labels corresponding to the background.

The total loss is defined as a sum of the above losses.

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{adv} &+ \alpha(\lambda_{rec}\mathcal{L}_{rec} + \lambda_{per}\mathcal{L}_{per}) \\ &+ \lambda_{id}\mathcal{L}_{id} + \lambda_{mask}\mathcal{L}_{mask},\end{aligned} \tag{7}$$

where $\mathcal{L}_{adv}$ denotes the GAN loss, and we set $\lambda_{id} = 3$, $\lambda_{rec} = 30$, $\lambda_{per} = 5$, and $\lambda_{mask} = 10$, respectively.

Experiments

Implementation details. The training images are collected from VGGFace2 (Cao et al. 2018). We select the landmarks between the two eyes larger than 70 pixels and get 550K images. We conduct experiments on two famous face swapping algorithms as the teacher models, including SimSwap (Chen et al. 2020) and FaceShifter (Li et al. 2020a). The student model uses the same face alignment algorithm as the teacher model. The image sizes are 224×224 and 256×256 for SimSwap and FaceShifter, respectively.

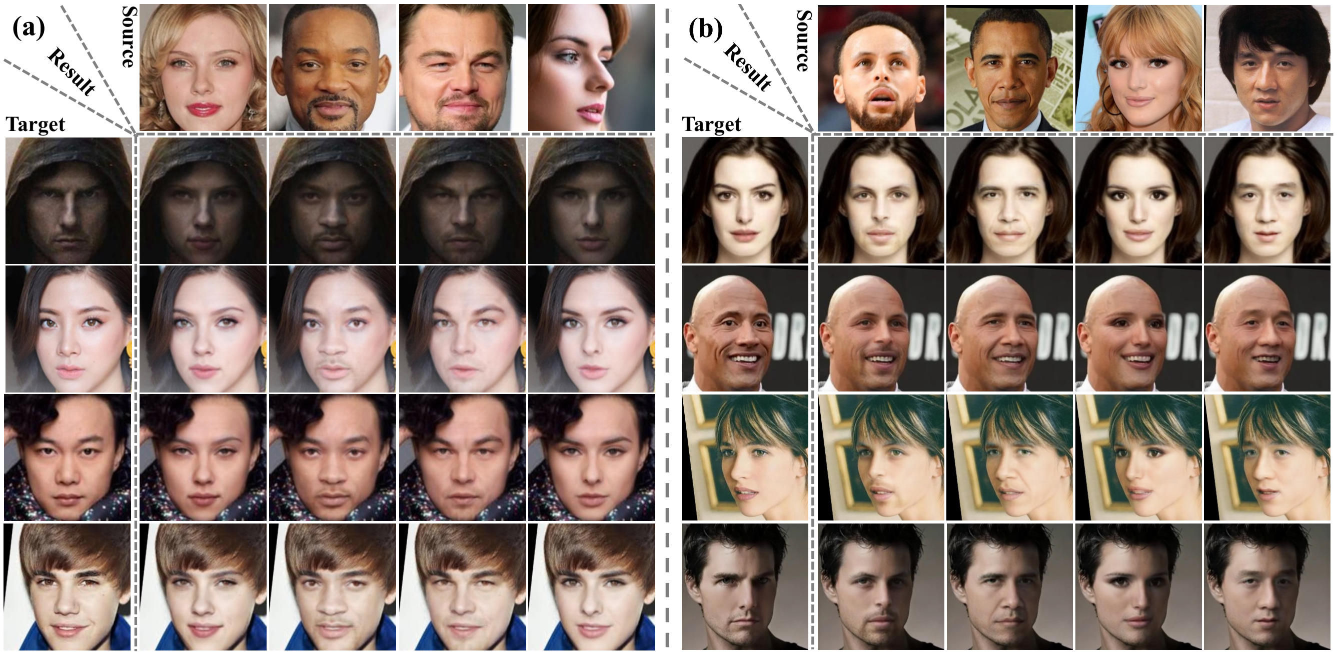

Figure 5: Further face swapping results of our models that are distilled from SimSwap (a) and FaceShifter (b) respectively.

Qualitative Results

Comparison with teachers. First, we compare our method with the teacher methods. The model of SimSwap is derived from the official repository, while the FaceShifter has been implemented by ourselves since the source codes are not released. The test images are collected from CelebA-HQ (Karras et al. 2018). As shown in Fig. 3, our results have fewer artifacts than those of SimSwap. Specifically, employing our semantic fusion module, our method can keep the hair and background more stable than SimSwap, as shown in the first and second rows of Fig. 3. In addition, The result of SimSwap has artifacts around the eyes when the subject of the source image wears glasses, while our method can obtain clearer results. We demonstrate the comparison with FaceShifter in Fig. 4. As shown in the first row, our method can keep the expression of the target image better than FaceShifter. At the same time, our method can also generate impressive results under uncommon conditions like occlusions or large poses.

Demonstration. We demonstrate further results of our models in Fig. 5. These superstar images are collected from the internet. As we can see, our models generate appealing results for different types of source and target images.

Comparison with state-of-the-art methods. We also compare our method with other face swapping methods, including DeepFakes[1], FSGAN (Nirkin, Keller, and Hassner 2019), FaceShifter (Li et al. 2020a), SimSwap (Chen et al. 2020), and recently proposed FaceController (Xu et al. 2021) on the FaceForensics++ (Rossler et al. 2019) dataset, which is a widely used dataset for deepfakes creation and detection. For a fair comparison, we use the official results that are derived from the FaceForensics++ dataset for DeepFakes and FaceShifter. While for FSGAN and SimSwap, these results are generated by the official codes and models. The results of FaceController are cropped from its paper. The comparison results are shown in Fig. 6. As we can see, the results of DeepFakes and FSGAN have noticeable artifacts like unnatural color. Compared with FaceController, our method can keep the expression of the target image better. For SimSwap and FaceShifter, our method achieves comparable results at these images. We also compare our method with SimSwap and FaceShifter using the images from FaceShifter for more qualitative results as shown in Fig. 7. Our method can retain the attributes of the target image better than FaceShifter, and generate fewer artifacts than SimSwap.

Quantitative Results

We compare our method with other methods about parameters and computations in more detail as shown in Table 1. We report the parameters and computations at image-based and video-based conditions, respectively. As we can see, our method has significant advantages in relation to the parameters and computations. Specifically, for video face swapping, our method contains only 0.50M parameters and 0.44G FLOPs when the input size is 256×256, which are fewer than 100 times as compared to other face swapping methods like SimSwap and FaceShifter. For image-based face swapping, most of the parameters and computations of our method are consumed by the identity network (Deng et al. 2019). DeepFakes also has advantages with computations. However, it requires training a different model when given a new identity, and the size of the input is 64×64. Never-

[1] https://github.com/deepfakes/faceswap

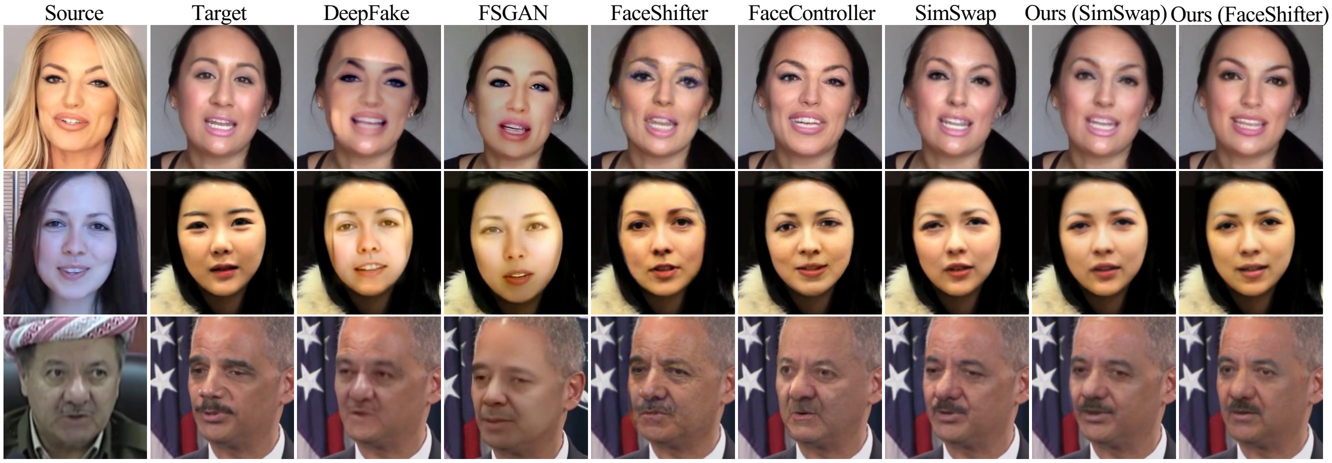

Figure 6: Comparison with DeepFakes, FSGAN, FaceShifter, FaceController, and SimSwap on the FaceForensics++ dataset.

Method	Size	(I) Params (M)	(I) FLOPs (G)	(V) Params (M)	(V) FLOPs (G)	(V) FPS	Id↑	Pose↓
DeepFakes	64	82.1	**1.90**	82.1	1.90	9.5	81.96	4.14
FSGAN	256	226	2440	226	2240	-	57.34	3.81
SimSwap	224	107	55.7	45.6	48.2	0.64	92.83	1.53
FaceShifter	256	421	97.4	350	91.1	-	97.38	2.96
FaceController	224	306	192	236	177	-	**98.27**	2.65
Teacher (SimSwap)	224	107	55.7	45.6	48.2	0.64	95.94	1.39
Teacher (FaceShifter)	256	421	97.4	350	91.1	-	97.15	1.76
Ours (SimSwap)	224	**72.8**	8.07	**0.50**	**0.33**	**25.6**	95.98	**1.32**
Ours (FaceShifter)	256	**72.8**	8.18	**0.50**	0.44	19.7	96.10	1.70
Id Network	112	52.2	7.52	-	-	-	-	-

Table 1: The comparison of different face swapping methods, where I and V represent the image and video scenes, respectively. Note that we count the parameters and computations of the Id Network for image face swapping if these algorithms need it, but not for video scenes. The Size means the size of the input image. FPS is tested under the mobile phone with MediaTek Dimensity 1100 chip. In the last two columns, we report the accuracies of identity retrieval and pose errors on the FaceForensics++ dataset.

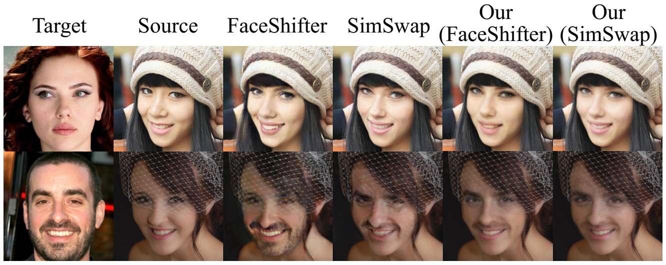

Figure 7: More comparison with SimSwap and FaceShifter (Zoom view better).

theless, our method still has fewer computations than DeepFakes for video face swapping. Then, we test our approach and SimSwap at the mobile phone with the MediaTek Dimensity 1100 chip to demonstrate our superiority in speed. Without further optimizations, our method arrives at 25.6 FPS, 40 times faster than SimSwap. We did not test others methods as these models are too heavy for mobile phones.

The quantitative comparisons with respect to the quality of face swapping are shown in the last two columns of Table 1. We follow the evaluation metrics used in FaceShifter and SimSwap, including identity retrieval accuracy and posture similarity on the FaceForensics++ dataset. For identity retrieval, we use the CosFace (Wang et al. 2018) to extract the identity embedding and retrieve the closest face using cosine similarity. For pose evaluation, we employ a pose estimator (Ruiz, Chong, and Rehg 2018) to estimate head pose and then report the L2 distance of pose vectors. Note that the results of Teacher (SimSwap) are different from that reported in the original SimSwap paper since the authors released a different model in their official repository. As we can see, the obtained models using our method can achieve comparable results with their teachers. The identity retrieval accuracy and pose similarity of our results are better than the Teacher (SimSwap). Compared with other methods, our method can better balance identity retrieval accuracy and pose similarity.

Ablation Study

Effectiveness of the network. We conduct all ablation studies using SimSwap as the teacher model to verify the efficiency and necessity of our network architecture designs. The qualitative and quantitative results are shown in Fig. 8 and Table 2, respectively. First, suppose we drop out the weights prediction or replace the weights modulation with

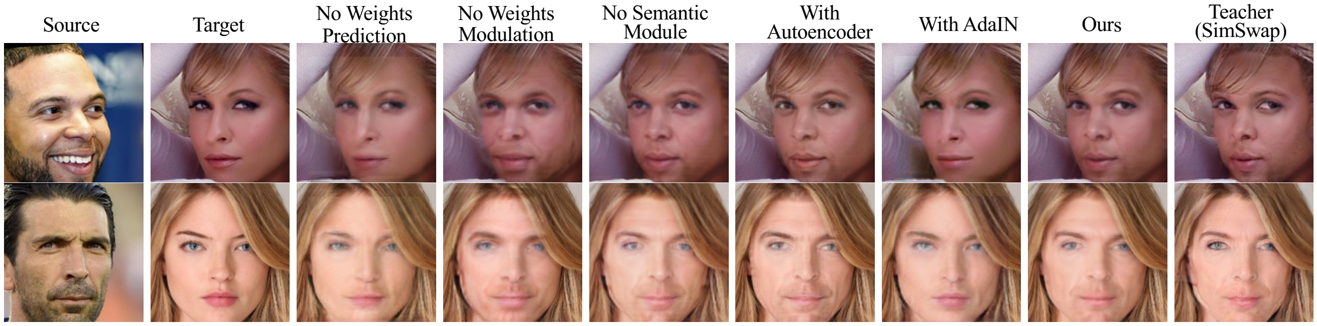

Figure 8: Ablation study about network architecture.

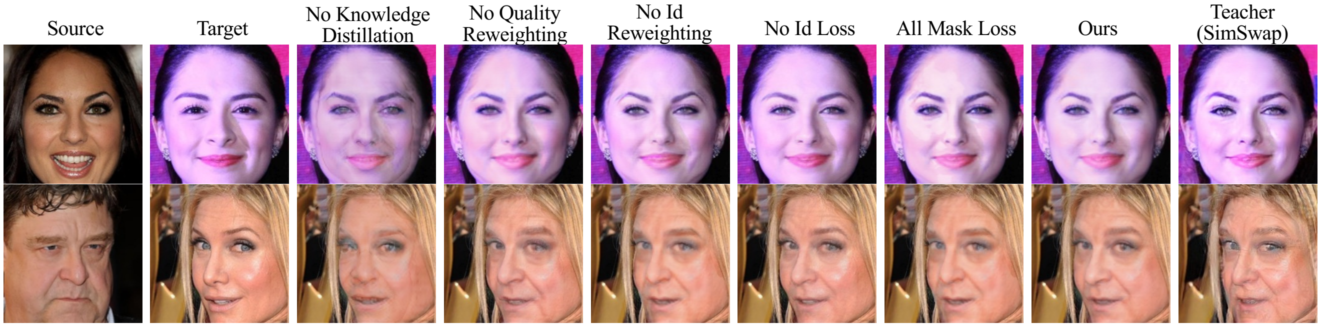

Figure 9: Ablation study about training objectives.

Method	Id↑	Pose↓	FID↓
No Weights Prediction	16.29	1.40	16.75
No Weights Modulation	86.20	2.12	9.92
No Semantic Module	**96.10**	1.69	12.13
With AutoEncoder	92.78	1.43	7.77
With AdaIN	18.34	1.30	12.04
No Knowledge Distillation	88.48	3.21	17.37
No Quality Reweight	95.28	1.50	8.12
No Id Reweight	93.71	1.52	7.38
No Id Loss	28.13	**0.98**	**4.76**
All Mask Loss	95.46	1.42	7.44
Teacher (SimSwap)	95.94	1.39	10.26
Ours	95.98	1.32	6.79

Table 2: Ablation study results of different network architectures and training objectives on the FaceForensics++ dataset.

AdaIN (Huang and Belongie 2017), the revised method can be regarded as a compression version of SimSwap. In these cases, the training processes are unstable and cannot generate acceptable swapping results. The identity retrieval accuracy significantly drops as shown in Table 2. If we drop out of the semantic module, there is a clear difference in the backgrounds between the generated and target image, which is similar to SimSwap. Then, the qualitative and quantitative results drop slightly by removing the weighting modulation or replacing the U-Net-based with AutoEncoder-based network, which results in more artifacts than ours.

Effectiveness of training objectives. To verify the advantages of the training objectives, we have conducted many ablation studies on loss functions, and the results have been shown in Fig. 9 and Table 2. First, if we drop out the knowledge distillation, the generated results have noticeable artifacts and worse quantitative results. Then, we evaluate our loss reweighting module, including the identity reweighting and image quality reweighting. The identity retrieval accuracy increases by 2.27% when adding the identity reweighting module. When adding image quality reweighting, our model can generate slightly better results. Then, if we drop out of the identity loss, the generated results have fewer artifacts than those of the proposed method. However, the identity retrieval accuracy is also pretty low. Last, as demonstrated in Fig. 9, we can see that using a weakly semantic module can clearly decrease the color difference between the generated image and the target image.

Conclusion

In this work, we propose MobileFaceSwap for real-time subject-agnostic face swapping. We design an efficient Identity Injection Module (IIM) to adjust the parameters of the Identity-aware Dynamic Network (IDN) adaptively. Then, we use the knowledge distillation and design a loss reweighting module to obtain better swapped results. Our method can be deployed on mobile phones, perform real-time face swapping. Besides, we can generate some forgery samples by MobileFaceSwap and hope these will have a little impact on forgery detection, as some forgery techniques are likely to be abused for malicious purposes.

Acknowledgement

This work is supported by CCF-Baidu Open Fund.

References

Blanz, V.; and Vetter, T. 1999. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 187–194.

Cao, Q.; Shen, L.; Xie, W.; Parkhi, O. M.; and Zisserman, A. 2018. Vggface2: A dataset for recognising faces across pose and age. In *13th IEEE international conference on automatic face & gesture recognition*, 67–74. IEEE.

Chen, G.; Choi, W.; Yu, X.; Han, T.; and Chandraker, M. 2017. Learning efficient object detection models with knowledge distillation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 742–751.

Chen, R.; Chen, X.; Ni, B.; and Ge, Y. 2020. SimSwap: An Efficient Framework For High Fidelity Face Swapping. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2003–2011.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1251–1258.

De Brabandere, B.; Jia, X.; Tuytelaars, T.; and Van Gool, L. 2016. Dynamic filter networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 667–675.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4690–4699.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing*, 2672–2680.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hu, X.; Mu, H.; Zhang, X.; Wang, Z.; Tan, T.; and Sun, J. 2019. Meta-SR: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1575–1584.

Huang, X.; and Belongie, S. J. 2017. Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, 1510–1519.

Jin, Q.; Ren, J.; Woodford, O. J.; Wang, J.; Yuan, G.; Wang, Y.; and Tulyakov, S. 2021. Teachers Do More Than Teach: Compressing Image-to-Image Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 13600–13611.

Jo, Y.; Oh, S. W.; Kang, J.; and Kim, S. J. 2018. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3224–3232.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, 694–711. Springer.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations*.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8110–8119.

Li, L.; Bao, J.; Yang, H.; Chen, D.; and Wen, F. 2020a. Advancing high fidelity identity swapping for forgery detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5074–5083.

Li, M.; Lin, J.; Ding, Y.; Liu, Z.; Zhu, J.-Y.; and Han, S. 2020b. Gan compression: Efficient architectures for interactive conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5284–5294.

Liu, X.; Yin, G.; Shao, J.; Wang, X.; and Li, H. 2019. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 570–580.

Nirkin, Y.; Keller, Y.; and Hassner, T. 2019. FSGAN: Subject Agnostic Face Swapping and Reenactment. In *Proceedings of the IEEE International Conference on Computer Vision*, 7184–7193.

Nirkin, Y.; Masi, I.; Tuan, A. T.; Hassner, T.; and Medioni, G. 2018. On face segmentation, face swapping, and face perception. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, 98–105. IEEE.

Park, T.; Liu, M.; Wang, T.; and Zhu, J. 2019. Semantic Image Synthesis With Spatially-Adaptive Normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2337–2346.

Perov, I.; Gao, D.; Chervoniy, N.; Liu, K.; Marangonda, S.; Umé, C.; Dpfks, M.; Facenheim, C. S.; RP, L.; Jiang, J.; et al. 2020. Deepfacelab: A simple, flexible and extensible face swapping framework. *arXiv preprint arXiv:2005.05535*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Rossler, A.; Cozzolino, D.; Verdoliva, L.; Riess, C.; Thies, J.; and Nießner, M. 2019. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE International Conference on Computer Vision*, 1–11.

Ruiz, N.; Chong, E.; and Rehg, J. M. 2018. Fine-grained head pose estimation without keypoints. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2074–2083.

Shen, F.; Yan, S.; and Zeng, G. 2018. Neural style transfer via meta networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8061–8069.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; and Liu, W. 2018. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5265–5274.

Wang, J.; Bao, W.; Sun, L.; Zhu, X.; Cao, B.; and Philip, S. Y. 2019. Private model compression via knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 1190–1197.

Wang, Y.; Chen, X.; Zhu, J.; Chu, W.; Tai, Y.; Wang, C.; Li, J.; Wu, Y.; Huang, F.; and Ji, R. 2021. HifiFace: 3D Shape and Semantic Prior Guided High Fidelity Face Swapping. *arXiv preprint arXiv:2106.09965*.

Xu, Z.; Yu, X.; Hong, Z.; Zhu, Z.; Han, J.; Liu, J.; Ding, E.; and Bai, X. 2021. FaceController: Controllable Attribute Editing for Face in the Wild. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 3083–3091.

Yim, J.; Joo, D.; Bae, J.; and Kim, J. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4133–4141.

Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; and Sang, N. 2018. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European conference on computer vision*, 325–341.