Face Image Inpainting with Evolutionary Generators

Chong Han, Junli Wang

***Abstract*—Recently, deep learning has become a mainstream method of image inpainting. It can not only restore the image texture, obtain high-level abstract features of images, but also restore semantic images such as human face images. Among these methods, generative adversarial networks(GANs) using autoencoder as the generator have become the promising model for image inpainting. These models implement the end-to-end image inpainting and also generate visually reasonable and clear image structures and textures. However, GANs often have problems with gradient vanishing and model collapse during training, so we propose a Generative Adversarial Network with Evolutionary Generators (EG-GAN) and apply it in face image inpainting. To stabilize the model training process, EG-GAN trains the generator network by evolution, combines two mutation functions as a training objective to update the parameter of generator networks, and produces offspring generators through crossover, using the matcher assists the discriminator to criticize the generated image. Experiments on various face image datasets such as CelebA-HQ and CelebA show that EG-GAN successfully overcomes the gradient vanishing problem, achieves stable and efficient training, and generates visually reasonable images.**

***Index Terms*—Neuro-evolution, Autoencoder, Generative Adversarial Networks, Face Image Inpainting.**

I. Introduction

IMAGE inpainting is an important task in machine vision, which aims to fill in missing pixels in damaged images. Initially, Pathak et al. proposed an image inpainting model based on an autoencoder: Context Encoder (CE)[1]. Because CE is trained by Euclidean distance, which will inevitably lead to the blurred image[2]. Afterwards,GAN has become one of the most significant research domains because of its various applications in the field of image processing and multi-view works[3][4]. Therefore, researchers began to use GANs. For example, Raymond et al. proposed semantic image inpainting with deep generative models(SIIGAN)[5]. However, this model can't accomplish an end-to-end process, so it takes a lot of time to train. To overcome the above problems, Pathak et al. added an adversarial network into the original CE, which is applied to image inpainting.

Neuro-evolution [6] is a method that uses biological evolution theory or evolutionary computation to generate artificial neural network parameters, structures, and rules. Therefore, deep learning models based on neuro-evolution can be divided into two categories: the first is to optimize the deep learning model by biological evolution theory. For example, Chaoyue Wang et al. proposed EGAN[7]. The second is to use evolutionary computing to optimize deep learning models. Evolutionary computing includes: genetic algorithm (GA), genetic programming (GP), evolutionary strategy (ES), and evolutionary programming (EP). For example, Masanori Suganuma et al. proposed the ES-CAE [8] which applies an evolutionary strategy to autoencoder. Moreover, these experiments prove that deep learning models optimized by neuro-evolution are easier to train and possible to generate some skip connections that are difficult to be designed by a human.

In this letter, EG-GAN is inspired by biological evolution. GAN is trained in the way of neuro-evolution and applied to face image inpainting. EG-GAN mainly consists of three parts: generator, discriminator, and matcher. The generator is regarded as an individual in the population, and the network parameters in the generator are optimized by mutation and crossover. The discriminator is utilized to criticize the quality of the entire generated image. But for image inpainting, the image quality of missing regions is the key to determine the performance of the generator, Therefore, the discriminator has certain limitations. To solve this problem, this letter proposes a matcher to assist the discriminator.The contributions of this letter are summarized as follows:

(1) It is proposed to train the generator by evolution and combine two different mutation functions as the objective function to update the parameters in generators, avoiding the gradient vanishing and model collapse. What's more, a crossover is added to explore the commonality in excellent generator models, producing offspring population, so that elite generators will not be lost in the process of mutation.

(2) A matcher is proposed to assist the discriminator in criticizing the generated image. The matcher can learn to obtain the relevance of each component in the image and focus on criticizing the contextual correlation of the generated regions, avoiding discriminators giving high-value feedback to images with wrong correlation, thereby misleading the generator.

(3) EG-GAN achieves high-quality inpainting results on various face image datasets, including CelebA-HQ, CelebA, PubFig, and non-realistic high-definition face datasets generated by StyleGAN. Meanwhile, the quantitative evaluations on PSNR and SSMI are higher than some traditional and recent image inpainting models.

II. Generative Adversarial Network with Evolutionary Generator

A. Neuro-evolution

We train the entire network by neuro-evolution, regarding the generator as an "individual" in the population, the discriminator and matcher as the "environment" where individuals

Manuscript received July 26, 2020; revised October 2, 2020. This work was supported in part by the National Key R&D Program of China(2017YFA0700602), in part by the National Natural Science Foundation of China(No.61672381); in part by the Fundamental Research Funds for the Central Universities.(Corresponding author: Junli Wang)

C. Han and J. Wang are with the Department of Electronics and Information Engineering, Tongji University, Shanghai 201804, China(e-mail: 496274966@qq.com; e-mail: junliwang@tongji.edu.cn)

can be selected in. The whole process includes four steps: mutation, evaluation, selection, and crossover.

(1)Mutation:The essence of mutation is two different objective functions: heuristic mutation and minimax mutation[7], which aim to update the parameters in generators after each evaluation by discriminator and matcher. It is defined as:

$$M_G^{heuristic} = -\frac{1}{2}E_{G(x)\sim P_g}[log(D(G(X))] \quad (1)$$

$$M_G^{minimax} = \frac{1}{2}E_{G(x)\sim P_g}[log(1 - D(G(X))] \quad (2)$$

where Pg is the data distribution of generated image, G(x) is the generated image, and D(G(x)) is the output of discriminator.

Unlike traditional GAN models use a single objective function, EG-GAN uses the combination of heuristic mutation and minimax mutation as the generator's objective function ,avoiding gradient vanishing and model collapse to some extent. As shown in Fig.1, it can guide the model to Nash balance, stabilizing the training process.

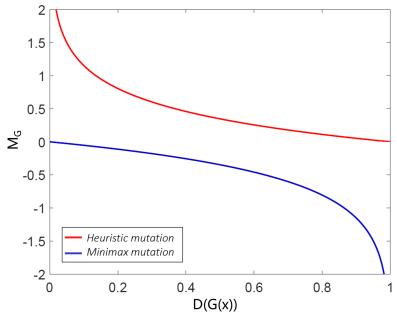

Fig. 1. Mutation function graph: EG-GAN will determine to use minimax mutation or heuristic mutation: when D(G(x)) is bigger than 0.5, choose the minimax mutation; when D(G(x)) is less than or equal to 0.5, choose the heuristic mutation

(2)Evaluation: In the process of training, using least absolute deviations[9](L1loss) as the standard for evaluating the performance of the generators. It reflects the difference between the generated sample and the real sample. It is defined as:

$$L1 = min\sum_{i=1}^{n}\|x^i - G(x^i)\| \quad (3)$$

where x^i is a sample from the real data distribution, $G(x^i)$ is a sample from the generated data distribution, and n is the number of samples.

(3)Selection: According to the value of L1loss, generators will be sorted. Then it will select two generators with the smallest value as a pair of parents and eliminate the rest of individuals. According to the population size, you can keep more than one pair of parents.

(4)Crossover: The generator structure of EG-GAN is encoder-decoder, so the crossover is to produce offspring generators by exchanging the encoder part with each other in a pair of elite parent. In order to avoid the generator's performance fluctuating caused by frequent crossover, the generator executes the crossover only after mutating a certain times(T). The whole process can be seen in the Fig.2

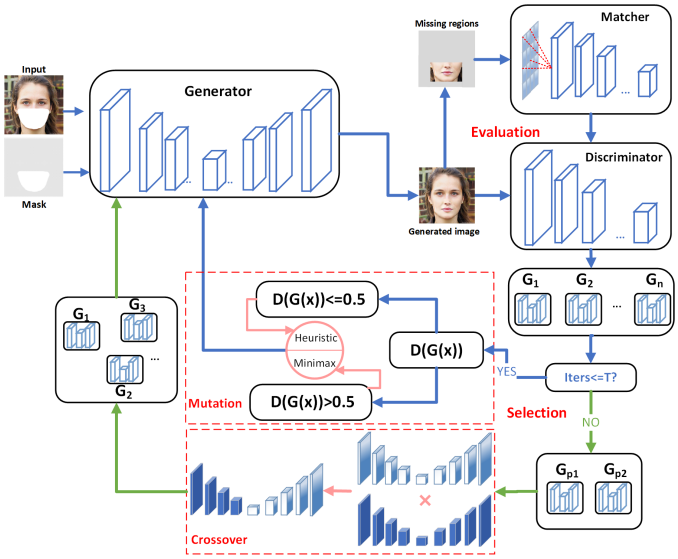

Fig. 2. EG-GAN framework: The blue line represents the process of updating generator parameters by mutation, and the green line represents the process of producing offspring by crossover.

B. EG-GAN

The generator of our model is autoencoder structure which can directly restore the missing regions of images. The input of the generator is a pair of images. It consists of a damaged face image and a mask (filled by white pixels). The size of the two images is 256 * 256, and the function of the mask is to input the position and shape of the missing regions in images. During training, the face image is blocked by random shapes.

In EG-GAN, the generated images are criticized by two adversarial networks-discriminator and matcher. The discriminator mainly criticizes the authenticity of the entire image, and the matcher focuses on criticizing the authenticity and correlation of missing regions in images.

The input of the matcher is the missing regions of images. The structure of matcher is PatchGAN[10] to criticize the correlation of generated content. It calculates the pixel differences between the generated image and the original image by the dot product. The result is the loss of the matcher and is also regarded as the penalty of the discriminator. It is defined as formula 4, where m is the mask image, I is the original image.

$$l_m = E_{G(x)\sim P_g}[m \odot D_m(G(x) - I)] \quad (4)$$

Discriminator criticizes the authenticity of generated images from a global view to ensure the continuity of the missing boundary pixels. The input of the discriminator is the complete face image generated by the generators. The network structure is the discriminator in WGAN-GP[11][12]. In EG-GAN, the gradient penalty is given by the matcher.

After the generated image is criticized by the matcher and discriminator, the two output values are calculated by weighted summation. Because the authenticity of missing regions is more important, it is defined as formula 5, where σ = 0.35 and

γ = 0.65, D_d(G(x)) is the output of discriminator,D_m(G(x)-I) is the output of matcher.

$$D(G(x)) = \sigma D_d(G(x)) + \gamma D_m(G(x) - I) \tag{5}$$

According to the penalty provided by the matcher, the discriminator's loss function is defined as formula 6, where D is a 1-Lipschitz function, which meets the conditions of $\|f(x_1) - f(x_2)\| \leq \|x_1 - x_2\|$, so w is a set of 1-Lipschitz functions, P_{data} is the real data distribution. The detailed algorithm can be seen in Fig.3.

$$\begin{aligned} l_d = \max_{D\in\omega}\{&E_{x\sim P_{data}}[D(x)] - E_{G(x)\sim P_g}[D(x)] - \\ &E_{G(x)\sim P_g}[m \odot D_m(G(x) - I)]\} \end{aligned} \tag{6}$$

Algorithm 1: EG-GAN. Default values: $\sigma = 0.35, \gamma = 0.65, \alpha = 0.0001, \beta_1 = 0.5, \beta_2 = 0.999, P = 60, N = 3, T = 4000$

for generation=1 to P **do**
  if generation = 1 **then** Randomly initializes a population of generators {G^1,G^2,…. G^N};
  for gen =1 to N **do** $G = G^{gen}$
    for iters=1 to T **do**
      Sample a batch of training data {x^1,x^2,x^3 ,…. ,x^n} from the dataset;
      Generate a random mask for x {m^1,m^2,m^3 ,…. ,m^n};
      Get generated images {G(x^1), G(x^2), G(x^3) ,…. , G(x^n)};
      Discriminator and matcher criticize the G(x) and output:
        $D(G(x)) = \sigma D_d(G(x)) + \gamma D_m(G(x) - I)$;
      Update matcher parameters θ_m to minimize:
        $l_m \leftarrow \nabla\{E_{G(x)\sim P_g}[m \odot D_m(G(x) - I)]\}$;
        $\theta_m \leftarrow Adam(l_m, \theta_m, \alpha, \beta_1, \beta_2)$;
      Update discriminator parameters θ_d to maximize:
        $l_d \leftarrow \nabla\left\{E_{x\sim P_{data}}[D(x)] - E_{G(x)\sim P_g}[D(G(x))] - E_{G(x)\sim P_g}[m \odot D_m(G(x) - I)]\right\}$;
        $\theta_d \leftarrow Adam(l_d, \theta_d, \alpha, \beta_1, \beta_2)$;
      if $D(G(x)) > 0.5$ **then**
        $M_G^{minimax} \leftarrow \nabla[\frac{1}{2}E_{G(x)\sim P_g}[\log(1 - D(G(x)))]$; $\theta_g \leftarrow Adam(M_G^{min}, \theta_g, \alpha, \beta_1, \beta_2)$;
      else:
        $M_G^{heuristic} \leftarrow \nabla\{-\frac{1}{2}E_{G(x)\sim P_g}[\log(D(G(x)))]$; $\theta_g \leftarrow Adam(M_G^{heur}, \theta_g, \alpha, \beta_1, \beta_2)$;
    end for
  end for
  Evaluate generators and sort them based on : $L1 \leftarrow \min\sum_{i=1}^{n}\|x^i - G(x^i)\|$;
    $\{L1_{G^a} < L1_{G^b} < L1_{G^c} \ldots\} \leftarrow sort(\{l1_G\})$;
  Crossover to produce offspring:
    $G^{child1} \leftarrow \theta^a_{encoder} \& \theta^b_{decoder} = G^a \times G^b$;
    $G^{child2} \leftarrow \theta^b_{encoder} \& \theta^a_{decoder} = G^a \times G^b$;
    $next\ G^{gen} = \{G^1, \ldots, G^N\} \leftarrow \{G^a, G^{child1}, G^{child2}\}$;
end for

Fig. 3. EG-GAN Algorithm: P is the number of evolutionary generation, N is the number of generators in each generation, and T is the iteration of mutation. When the hardware permits, N can be expanded appropriately. Simultaneously, to keep N the same during the evolution process, the parent individual can be reserved to the next generation.

III. Experiments

In the experiment, we use CelebA-HQ[13] as the training dataset. CelebA-HQ is a high-definition human face dataset generated by Nvidia in 2018, which can increase the pixels of images in CelebA[14] up to 1024*1024. The pixel size of the face image used in our experiment is 256 * 256, the training set contains 24102 images, and the test set contains 2942 images. Two GTX 1080 Ti GPUs run simultaneously, completing the 60 generations evolution. In our experiments, apart from the inpainting results, we also utilize the PSNR and SSMI indexes[15] to estimate our model.

A. Ablation Experiment

In order to prove the effectiveness of introducing evolution and matcher, the face image is restored by four different settings of our model. For M1, we introduce a reconstruction loss l_d to the discriminator and only the heuristic mutation is introduced to the generator. For M2, based on M1, the minimax mutation is added to the generator, thus the generator is optimized by these two mutations. For M3, based on M2, we introduce the matcher and its loss function l_m to the whole network to further optimize the discriminator. For M4, based on M3, we introduce evolutionary optimization to the generator, which is the proposed method EG-GAN. Fig.4 shows the ablation results, and Tab1 shows the PSNR and SSIM results under each setting.

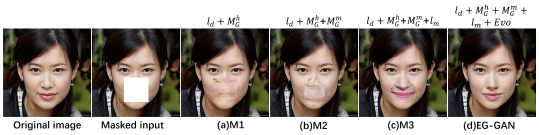

Fig. 4. Ablation results under different settings

TABLE I
Quantitative evaluations in terms of PSNR and SSMI under different settings

M	M1	M2	M3	EG-GAN
SSIM	0.900941	0.913659	0.923995	0.949266
PSNR	25.50357	26.68998	29.37788	34.16005

According to the comparison of the results, M1 and M2 can not generate reasonable and clear results, which may be due to model collapse or gradient vanishing problem, as can be seen in Fig4.(a)(b). When the matcher is added, the model can generate a reasonable result in the missing region, but the generated image is a little unclear, and there is an unnatural color transition at the boundary of the missing region, as shown in Fig4.(c). In Fig4.(d), based on the evolutionary selection, EG-GAN can generate more clear inpainting results and get the highest score on SSIM and PSNR .

B. Evolutionary Process

The image inpainting results are generated by the corresponding generators from the 10th to 60th generation, as shown in Fig5.(a). In addition, the performance of EG-GAN can be reflected by the change of L1loss and the M_G in Fig5.(b). After the 10th, the downward of L1loss tends to be stable. Because the generated image must have a certain error compared with the original image, which indirectly reflects whether the generator is creative or not. Secondly, based on the M_G, we can infer the value of D(G(x)), because the generator chooses minimax mutation or heuristic mutation as the loss function of the generator depending on D(G(x)).As shown in

Fig.1, we can see that when M_G vibrates between 0.5 and -0.25, the corresponding value of D(G(x)) vibrates around 0.5. In Fig5(b), we can see that M_G is basically maintained at 0.5 to -0.25 after the 10th generation, indicating that gradient vanishing can be avoided during training.

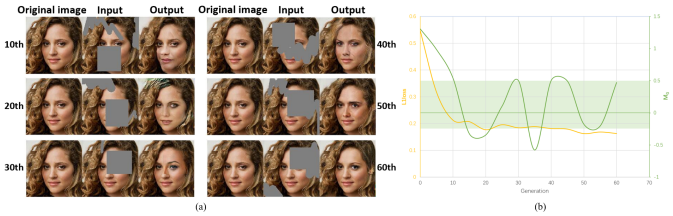

Fig. 5. The image inpainting results from different generations(a); L1loss and M_G change with generations(b)

C. Face Inpainting Results

In order to study the influence of different mask sizes on inpainting results, we input eight different square masks into the EG-GAN, and eight masks are numbered from 1 to 8, in Fig6.(a). To a certain extent, SSIM and PSNR decrease as the size of mask increases, as shown in Fig6.(b). However, the downward of SSIM is not as obvious as PSNR. Thus, even if EG-GAN can not achieve high performance in pixel level for larger mask, it can achieve a higher structural restoration. We also did an experiment using a face with glasses, as shown in Fig7.(a)(b)(c).Meanwhile, to test the generalization ability of EG-GAN, we select some images from other datasets, as shown in Fig7.(d)(e)(f). All of these results show that EG-GAN is creative enough to generate images distinctive from the training set, which is significant to avoid model collapse.

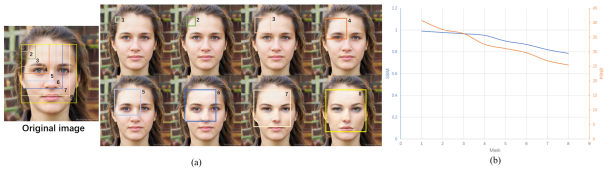

Fig. 6. Inpainting results of different square mask sizes(a): For mask from 1 to 6 which do not cover the eyes or only cover one eye, EG-GAN can restore image by learning the symmetry of the face. Thus, the inpainting results are very similar to the original image. For mask7 and mask8, it shows that EG-GAN has creative imagination. SSIM and PSNR evaluations at different square mask sizes(b)

Finally, we compare the EG-GAN model with several image inpainting models (CE, SIIGAN, E-CAE), as shown in the Fig7.(g)(h). In the Tab2, the PSNR and SSMI of each model tested on half of the missing image.

TABLE II
QUANTITATIVE EVALUATIONS IN TERMS OF PSNR AND SSMI ON DIFFERENT MODELS

Dataset	Models	PSNR	SSMI
CelebA(half)	CE	15.5	0.747
	SIIGAN	13.7	0.582
	ES-GAN	21.1	0.771
	EG-GAN	24.2	0.832

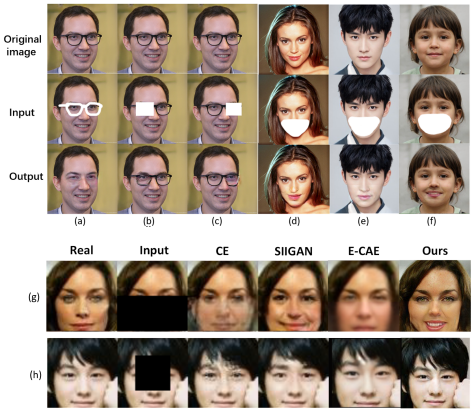

Fig. 7. Inpainting results of a face with glasses(a)(b)(c): (a):EG-GAN can restore an irregular mask well; (b)(c):It can prove the restoration capability of EG-GAN in both left and right sides. Inpainting results on PubFig dataset[16](d). Inpainting results on non-real face dataset generated by StyleGAN[17](e)(f). Image inpainting results of each model(g)(h)

IV. CONCLUSION

This letter proposes a face image inpainting model based on neuro-evolution, which regards the generator as an individual in the population. Generators with excellent performance are selected as the parent individuals, and offspring individuals are generated by crossover. The matcher is designed to assist the discriminator to criticize the image. Both qualitative and quantitative experiments show that EG-GAN can restore various kinds of face images with different masks in sizes, positions, and shapes, achieving reasonable and clear inpainting results.

REFERENCES

[1] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544. I

[2] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon, "Pixel-level domain transfer," in *European Conference on Computer Vision*. Springer, 2016, pp. 517–532. I

[3] P. Hu, D. Peng, Y. Sang, and Y. Xiang, "Multi-view linear discriminant analysis network," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5352–5365, 2019. I

[4] P. Hu, D. Peng, X. Wang, and Y. Xiang, "Multimodal adversarial network for cross-modal retrieval," *Knowledge-Based Systems*, vol. 180, pp. 38–50, 2019. I

[5] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5485–5493. I

[6] D. Floreano, P. Dürr, and C. Mattiussi, "Neuroevolution: from architectures to learning," *Evolutionary intelligence*, vol. 1, no. 1, pp. 47–62, 2008. I

[7] C. Wang, C. Xu, X. Yao, and D. Tao, "Evolutionary generative adversarial networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 921–934, 2019. I, II-A

[8] M. Suganuma, M. Ozay, and T. Okatani, "Exploiting the potential of standard convolutional autoencoders for image restoration by evolutionary search," *arXiv preprint arXiv:1803.00370*, 2018. I

[9] K. De Bot, P. Gommans, and C. Rossing, "L1 loss in an l2 environment: Dutch immigrants in france," *First language attrition*, pp. 87–98, 1991. II-A

[10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134. II-B

[11] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777. II-B
[12] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017. II-B
[13] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017. III
[14] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738. III
[15] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690. III
[16] S. Setty, M. Husain, P. Beham, J. Gudavalli, M. Kandasamy, R. Vaddi, V. Hemadri, J. Karure, R. Raju, B. Rajan *et al.*, "Indian movie face database: a benchmark for face recognition under wide variations," in *2013 fourth national conference on computer vision, pattern recognition, image processing and graphics (NCVPRIPG)*. IEEE, 2013, pp. 1–5. 7
[17] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401–4410. 7