

Proposal of image generation model using cGANs for sketching faces

Phat Nguyen Huu ^{a,1}, Nguyet Giap Thi ^{a,2}

^a School of Electronics and Telecommunications, Hanoi University of Science and Technology, Vietnam

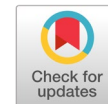
¹ phat.nguyenhuu@hust.edu.vn; ² nguyet.GT202445M@sis.hust.edu.vn

* Corresponding Author

Received 12 April 2021; accepted 7 May 2021; published 30 May 2021

ABSTRACT

The transition from sketches to realistic images of human faces has an important application in criminal investigation science to find criminals as depicted by witnesses. However, the difference between the sketch image and the real face image is image detail and color. Therefore, it is challenging and takes time to transform from hand-drawn sketches to actual faces. To solve this problem, we propose an image generation model using the conditional generative adversarial network with autoencoder (cGANs-AE) model to generate synthetic samples for variable length and multi-feature sequence datasets. The goal of the model is to learn how to encode a dataset that reduces its vector size. By reducing the dimension, the autoencoder will have to recreate the image similar to the original image. The purpose of the autoencoder is to produce output as input and focus only on the essential features. Raw sketches over the cGANs create realistic images that make the sketch images to raw images quickly and easily. The results show that the model achieves high accuracy of up to 75%, and PSNR is 25.5 dB, potentially applicable for practice with only 606 face images. The performance of our proposed architecture is compared with other solutions, and the results show that our proposal obtains competitive performance in terms of output quality (25.5 dB) and efficiency (above 75%).



KEYWORDS

GANs
cGANs
CNN
Sketching faces
Image processing



This is an open-access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license

1. Introduction

In criminal investigation science, automatic retrieval of suspect photos from the database can enable quick authorities to narrow potential suspects. However, in practice, the suspect's photos are often difficult to obtain. It is known that commercial software or experienced artists are looking to create sketches of suspects based on witnesses' descriptions. In addition to applications in security, facial sketch photosynthesis also has many applications in digital entertainment. Therefore, synthesizing and comparing sketch images are important and practical issues.

In this paper, we solve generating realistic face images from corresponding sketch images using cGAN. The recently proposed for translate human face produces impressive results. Face image synthesis methods based on convolution neural networks (CNN) [1] minimize the Euclidean distance between predicted and ground truth pixels. However, the generated results are blurred. Several works have successfully used Generative adversarial network GANs [2]–[5], allowing for automatic image creation from sketches to various subjects, including human faces. GAN is an unsupervised learning algorithm that belongs to DL. GANs are expected to create highly accurate systems. They consist of two networks, namely Generator and Discriminator. The generator tries to produce the most realistic data with noise input. Otherwise, the discriminator tries to distinguish what is the data generated by the generator and proves the fake images to improve them. The process is repeated until the generator produces the perfect data. However, we will not have control over category images generated in the dataset while training GAN. Therefore, a conditional generative adversarial network (cGAN) is proposed to control the image generating by the

generator under a specific category. The paper aims to build an algorithm that uses cGANs to create a real face applied in criminal investigation science when the input image is raw without a full face.

The rest of the paper includes five parts and is organized as follows. Section 2 presents several related works. Section 3 presents the proposed algorithm. Section 4 will evaluate the proposed model and analyze the results. In the final section, we give conclusions and future research directions.

2. Method

2.1. Related Work

Many studies are being developed based on converting from image to image using variations of GANs [2], [6]–[10]. In [2], the authors proposed GANs for image processing. Isola et al. [6] propose using pix2pix for solving the image-to-image conversion problem. The results show that represent images are improved their color sketches. Wang et al. [7] introduce of pix2pixHD technique. It is an improved version of pix2pix that synthesizes images from sketches and produces images with higher resolution. However, these papers all often assume good conditions for the training process. Therefore, sketches achieve good results when edge maps are input.

Jun-Yan Zhu et al. [8] proposed an algorithm using CycleGAN to reduce heterogeneity between input and output images. However, there is no guarantee for correspondence between inputs and outputs. To examine the known structure of human faces, researchers perform with a method based on facial components. Wu and Dai [9] first retrieve the most suitable face composition from the database of face images with input sketch. They then combine those elements and eventually transform the image to create sketches of similarity. Because of the synthesis and transformation strategy, their solution required an input sketch to be drawn professionally.

There are several topics on creating images of human faces in crime science [10]. In [10], the authors build a database based on Vietnamese human identity characteristics. The authors have collected portrait images from the national identity. They then filter about 14 thousand images and extract about 98 thousand facial details such as ears, nose, eyes, and eyebrows. As a result, they create 1000 unique masks and accessories like birthmarks, scars, moles, and earrings. As a result, the accuracy of paper gains 90% that can apply for real applications.

2.2. Overview of GAN

GANs are proposed for image generation models. They produce a characteristic vector z using either a normal or uniform distribution. We will use the generator (G) to generate the same output image with a feature vector z [11]. G network will work to generate fake data. The goal is to generate the most realistic data where the input is a random vector of fixed length and produces a pseudo-entity in the data domain. The vector is randomly distributed from the Gaussian distribution and is used to initiate the generation process. Discriminator (D) network is responsible for distinguishing between real and fake data. It performs the binary classification problem to determine whether these examples are real or fake.

The real and fake data are taken from the training set and output of the G network, respectively. Otherwise, cGANs maps from input (x) and noise vector (z) to produce the output image as shown in Fig. 1 [11]–[14]. In Fig. 1, there are several parameters and variables: D (Discriminator), G (Generator), θ_d (Parameter of Discriminator), θ_g (Parameter of Generator), $P_z(z)$ (Input noise distribution), $P_{data}(x)$ (Original data distribution), $P_g(x)$ (Generator distribution).

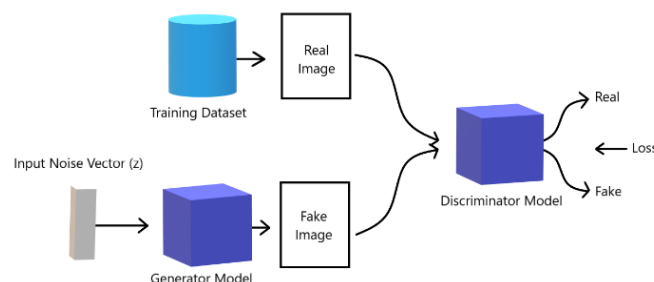


Fig. 1. Basic model structure of GANs [11]

2.3. Lost Function

GANs can have two loss functions: one for the generator and one for the discriminator. Generator and Discriminator loss functions are different, although they are created from the same formula. The loss function can be derived from the binary cross-entropy calculating as (1).

$$L(\hat{y}, y) = [y \cdot \log \hat{y} + (1 - y) \cdot \log(1 - \hat{y})], \quad (1)$$

where y and \hat{y} are real and predictive values, respectively.

Let us define several notations that we used throughout the paper: X is real data, Z is a latent vector. $G(z)$ is fake data, $D(x)$ is a discriminator of evaluation of real data, $D(G(z))$ is a discriminator of evaluation of fake data. $L(a, b)$ is loss between a and b , $E_{x \sim p_{data}(x)}$ is the expected value over all real data instance, $E_{z \sim p_z(z)}$ is the expected value over all random inputs to the generator.

Discriminator loss:

When the discriminator is trained, it will classify both real and fake data generating by the generator. Using data labels $P_{data}(x)=1$ (real data) and $\hat{y} = D(x)$ for (1), we have (2).

$$L(D(x), 1) = \log(D(x)). \quad (2)$$

If data is created from Generator $y = 0$ (fake data) and $\hat{y} = D(G(z))$, we will have (3).

$$L(D(G(z)), 0) = \log(1 - D(G(z))). \quad (3)$$

The object of the discriminator is to classify the fake and real datasets accurately. Since (2) and (3) must be maximized, and the final loss function for the discriminator is defined as (4).

$$L^D = \max[\log(D(x)) + \log(1 - D(G(z)))]. \quad (4)$$

Generator loss:

Otherwise, the generator is competing against the discriminator. Therefore, it tries to minimize (4), and the loss function is (5).

$$L^G = \min[\log(D(x)) + \log(1 - D(G(z)))]. \quad (5)$$

GANs loss:

To combine the (4) and (5), we have (6)

$$L = \min_G \max_D [\log(D(x)) + \log(1 - D(G(z)))]. \quad (6)$$

However, the loss function is only suitable for single-point data. Therefore, to consider all data, we need to use the (7).

$$\min_G \max_D L_{GAN}(G, D) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))]. \quad (7)$$

Regularization L1

The regression model uses the L1 calling least absolute shrinkage and selection operation (LASSO). It is a technique to prevent over-matching in neural network and to improve the accuracy of training by (8).

$$L_{L1}(G) = E_{x,y,z} [\|y - G(x,z)\|]. \quad (8)$$

2.4. Conditional Generative Adversarial Network (cGANs)

CGANs are used to learn a multimodal model. It creates descriptive labels. It is the attributes associating with the specific image that is not part of original training data. The loss function of cGANs is similar to that of GANs [15]–[18]. The only difference is that the conditional probability is used for both Generator and Discriminator as (9).

$$\min_G \max_D L_{cGAN}(G, D) = E_{x \sim p_{data}(x)} [\log D(x/y)] + E_{z \sim p_z(z)} [\log (1 - D(G(z/y)))]. \quad (9)$$

The loss of cGANs and L1 is (10).

$$L = \min_G \max_D L_{cGAN}(G, D) + \lambda L_{L1}(G), \quad (10)$$

where λ is called the regularization rate.

2.5. Autoencoder

Autoencoder is a neural network used to learn to encrypt data in an unsupervised learning method efficiently. In the paper, we use cGAN with autoencoder (cGANs-AE) model to generate synthetic samples for variable length and multi-feature sequence datasets based on [17]–[20]. The goal of the model is to learn how to encode a dataset that reduces its vector size. By a vector to reduce the dimension, the autoencoder will have to recreate the image similar to the original image. The purpose of the autoencoder is to produce output as input and focus only on the essential features.

An autoencoder has two subnets:

- Encoder: Taking the data and compressing it into latent space. If we represent input x and encoder is E , and output of latent-space is s , we will have $s = E(x)$.
- Decoder: Decoder is the task of taking latent spaces and then recreating the input data. If we represent the decoder as D and the output is o , we will have $o = D(s)$.

Fig. 2 is shown an example of an autoencoder. The detailed operation can be seen in [17].

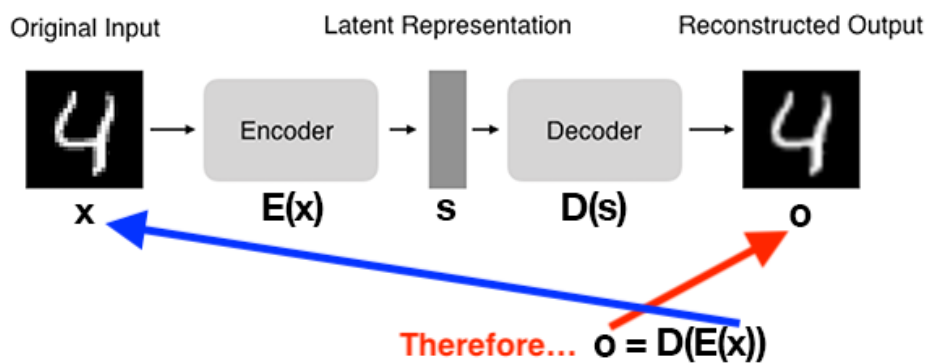


Fig. 2. An example of autoencoder for cGANs [17]

2.6. cGANs model

An extension of GANs is used to generate conditional output, as shown in Fig. 3. At this point, both Generator and Discriminator have added additional information. The information can be any supplemental information such as labels or data from another method.

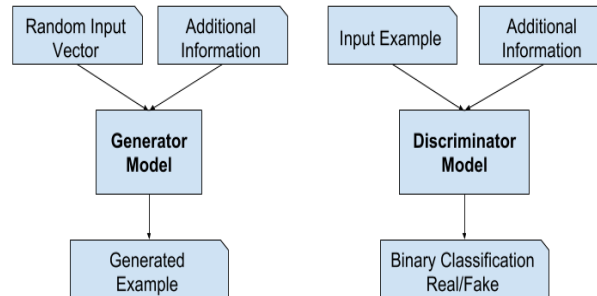


Fig. 3. Structure proposal using cGANs network model.

With the Generator and Discriminator, we train with real and fake images generating by the generator. The discriminator is responsible for determining that images are real or fake images from the generator. The generator is trained to trick the discriminator into distinguishing between its image and real image.

The flowchart of cGANs is shown in Fig. 4. Details of the cGANs algorithm are as follows:

Step 1: Identifying the problem

Step 2: Determining the operating model of cGANs

Step 3: Training Discriminator on real data with n Epochs

Step 4: Creating fake images for Generator and train Discriminator on fake datasets

Step 5: Training Generator with Discriminator output

Step 6: Repeating steps 3 and 5

Step 7: Checking if the fake image is valid. If it is a fake image, it will stop the training and go back to step 3.

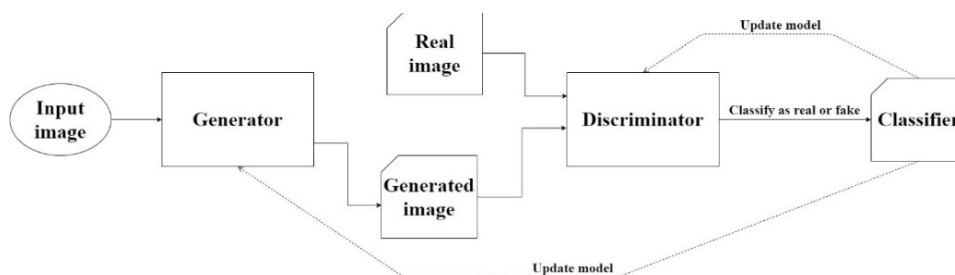


Fig. 4. Operational proposal model using cGANs.

3. Results and Discussion

3.1. Setup–Simulation

Our paper uses cGAN to learn the mapping from input to output images, as shown in Fig. 5.

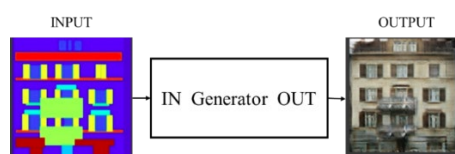


Fig. 5. Generator operation.

An example of a dataset is architectural labels. In this case, the generator will try to learn how to convert input into real images. A discriminator looks at converting to the real Generator image and trying to figure out the difference between the generating image from the generator provided from the dataset. Operation of Discriminator is shown in Fig. 6.

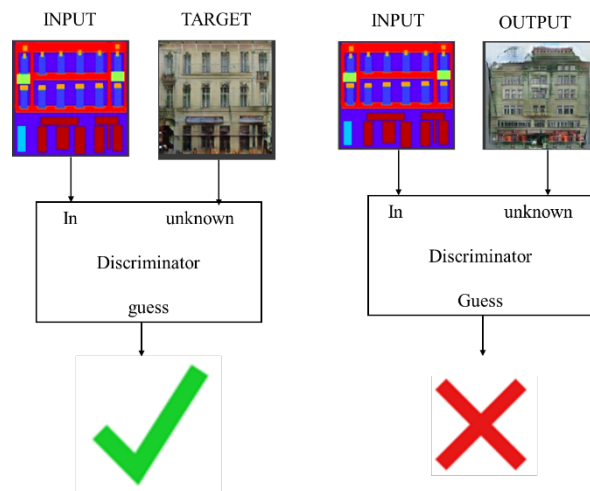


Fig. 6. Discriminator operation.

The generator is responsible for taking the input image and performing the transformation to create the target. For example, an architectural facade image of the house is created from the color shape, and the purpose is converted into a real house facade. The structure of the generator is called Encoder-Decoder (Autoencoder), as shown in Fig. 7.

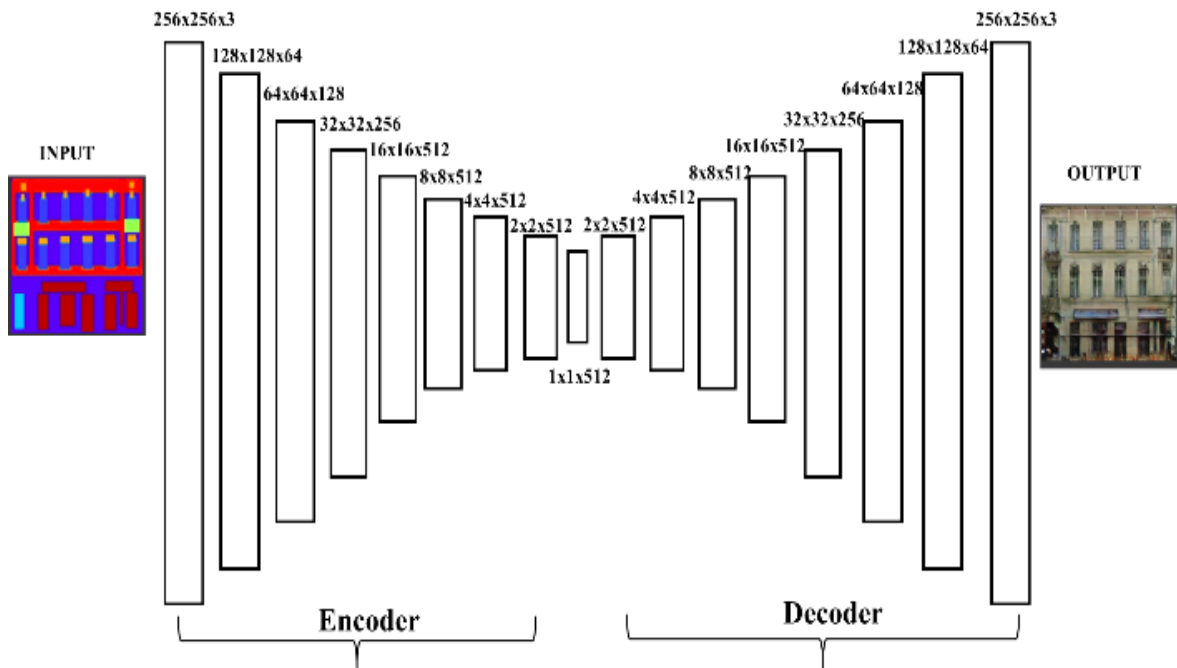


Fig. 7. Generator structure.

Each block is (Conv → Batchnorm → Leaky ReLU) in the encoder. Likewise, each block is (Transposed ConV → Batchnorm → Dropout (applying the first three blocks) → ReLU) in the decoder.

The discriminator task takes two images (an input and unknown images). First, it is an image of disability from the dataset that is generated by the generator. It then determines if the second image is generated from the generator, as shown in Fig. 8.

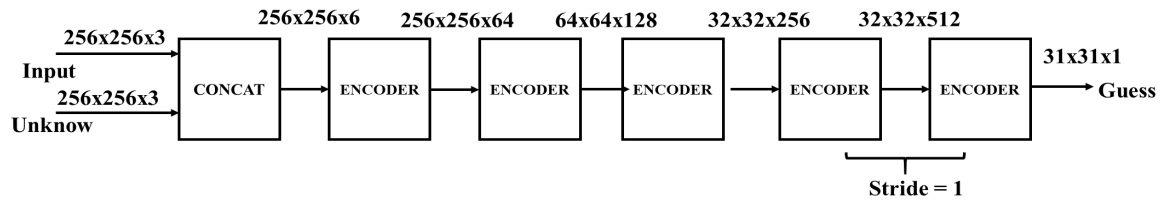


Fig. 8. Structure of Discriminator.

3.2. Result of this Research

To train the cGANs network, we perform the approach with multiple datasets such as CMP Facades [21]. The dataset includes 600 facades images editing from different sources. The images have various architectural styles, including molds, roofs, pillars, windows, doors, sills, curtains, balconies. According to the house model and their entry condition, the input images are color labels that are the actual architecture of the house. The generator will learn the input features and create a facade image similar to their color label.

From the data we have collected as listed above. First, we divide the dataset into three parts: the training set, the validation set, and the test set. Next, we make the model using the Pytorch framework, train the model on a Tesla T4 GPU with 15109MiB memory. Finally, we select the learning rate as 0.0002 for 150 epochs and Adam optimization.

To improve over-matching and rapid convergence, we add the loss parameter λ as shown in (10). The parameter is used to evaluate the complexity of the model. The more layer the parameter is, the more complex model is. In our model, we choose $\lambda = 150$. We found that the generator only produced noise in the first Epochs. Results are shown in Fig. 9.

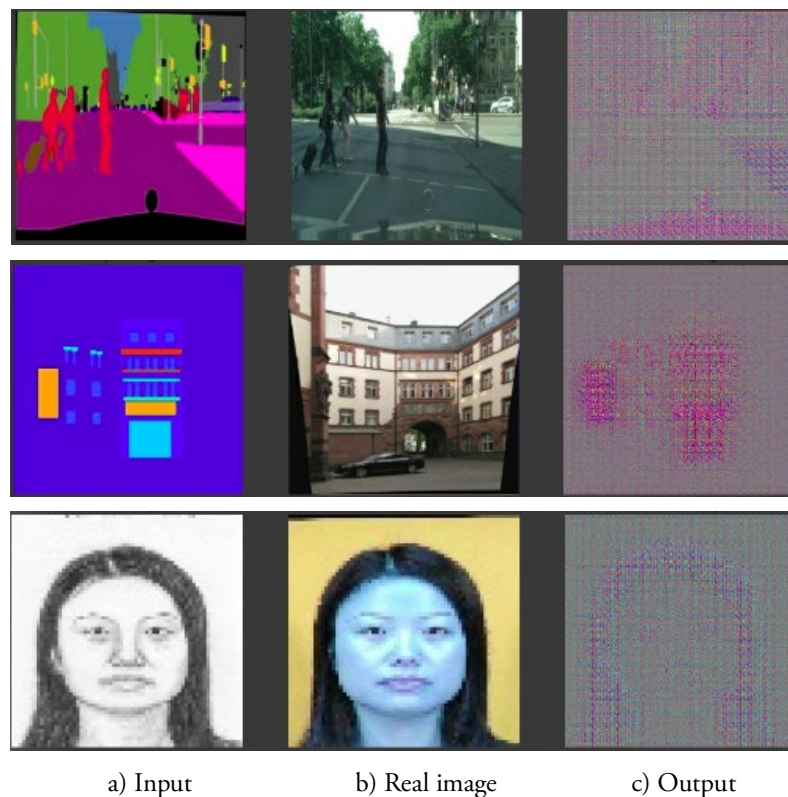


Fig. 9. Results in the first epochs.

After many iterations and classifying by Discriminator, Generator is learned and created output image similar to the input. Results are shown in Fig. 10.

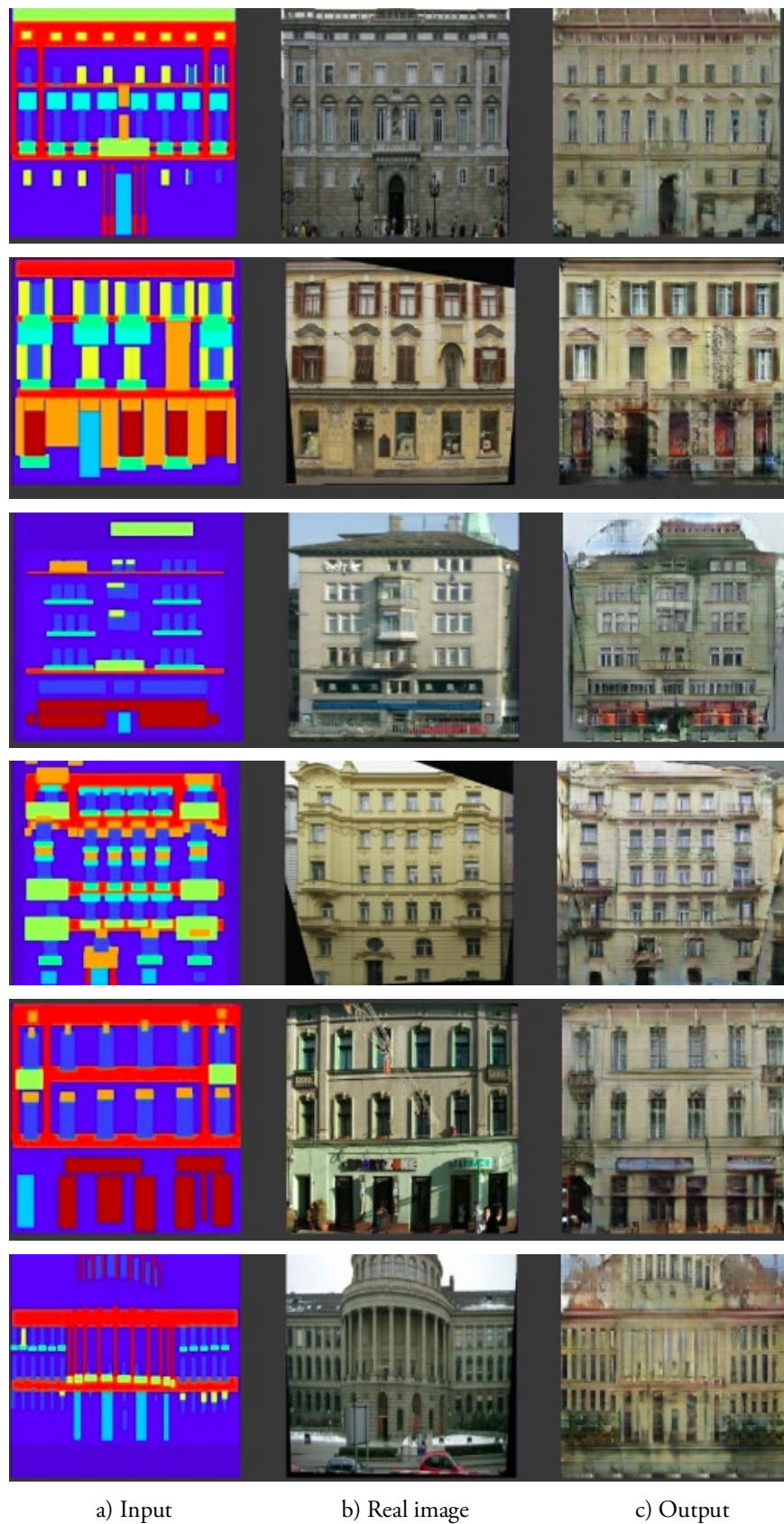


Fig. 10. The result is generated with the CMP Facades dataset.

We aim to create an image of the human face from sketches. We use CUHK Face Sketch Database (CUFS) datasets, as shown in Fig. 11. This dataset is for face sketch synthesis and recognition. It consists

of 188 student faces (CUHK), 130 faces from AR data [22], and 295 faces from the XM2VT database [23]. There are a total of 606 faces. Each face has a sketch sketched by the artist based on portrait shots under normal lighting conditions. The data set is described in Table 1.

Table 1. Description of the input data set

Dataset	Description	Viewing	Number of images	Source
AR Face Database	70 male faces and 60 female faces (26 images/person) in terms of the smile, wearing sunglasses, wearing a scar, etc.	Face to face	3380	Many countries
CUHK Face Sketch Database (CUFS)	188 faces of Hong Kong University students with age from 18 - 24 years old	Face to face	188	Hongkong
Our dataset	556 faces of the students of Nam Ha High School, Bien Hoa, Dong Nai with age 15-18 years old	Face to face	556	Vietnam

Fig. 11 shows pair of images (portrait and sketch) provided by ARFACE, CUHK, and XM2VTS.

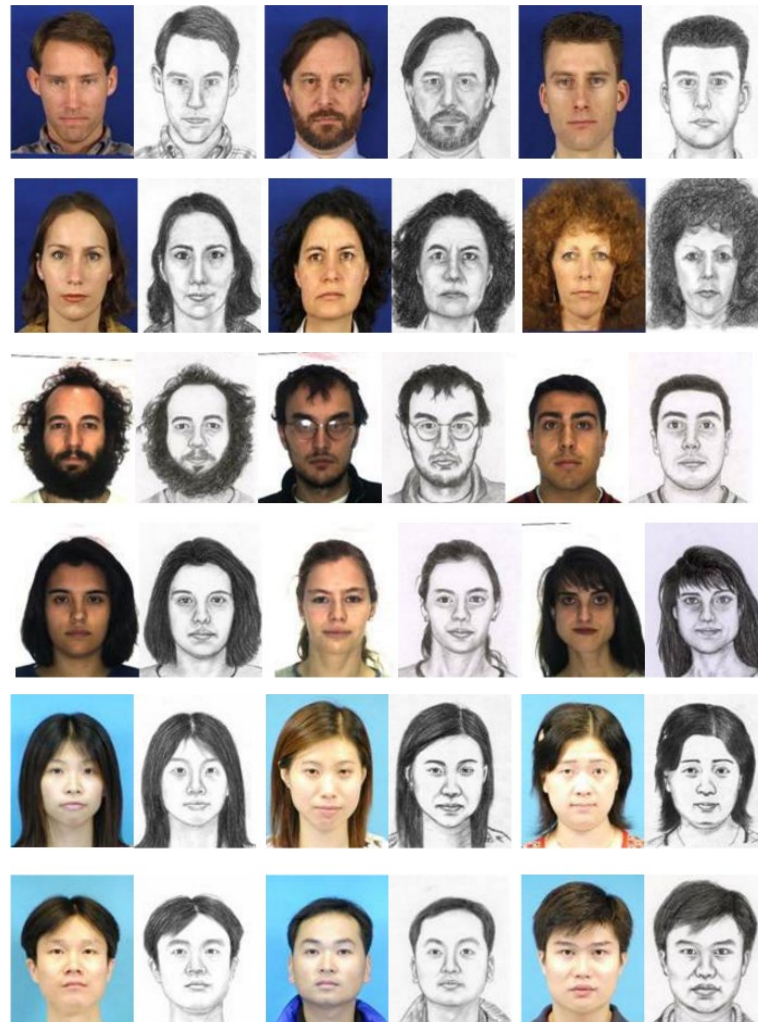


Fig. 11. Sample CUFS face database.

From face datasets and sketches, we use cGANs to train and learn facial features. Based on the sketch and portrait, we will create human faces. Results are shown in Fig. 12.

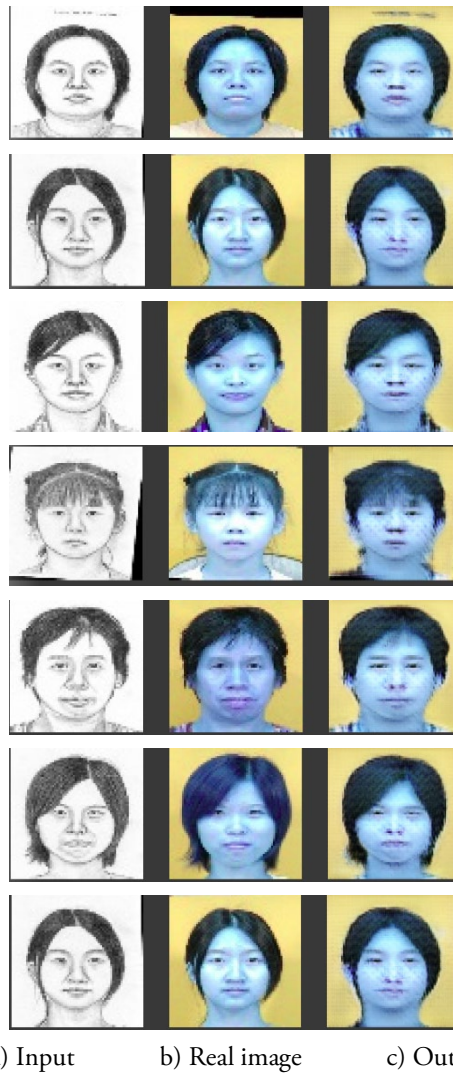


Fig. 12. Results are generated from the CUFS dataset.

Besides, we have also created our dataset, including 495 portraits and incomplete sketches of students, as shown in Fig. 13.

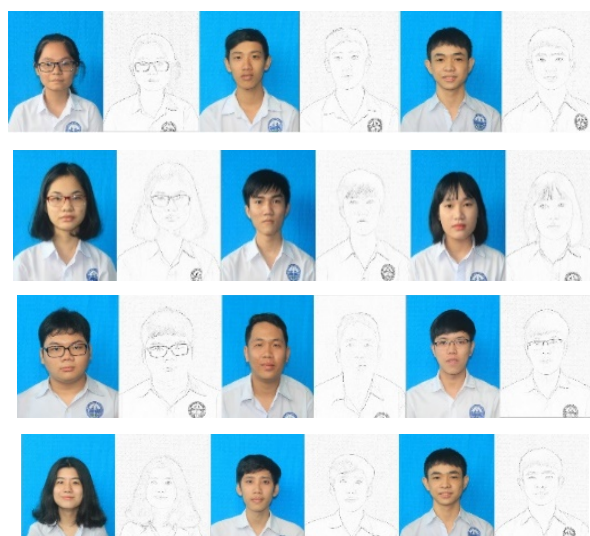


Fig. 13. Images of our dataset.

Our data includes more portraits with accessories such as glasses for network training. Before training, we perform to preprocess image. First, we resize images in the dataset to define size to ensure they are the same aspect ratio. We set the size to 256×256 pixels. The image is then normalized to $[-1, 1]$. Before normalizing the input data, the weights relating to features will be different. The uneven distribution of weights leads the algorithm to oscillate in the optimal region before finding the global minimum position. To avoid time-consuming training, we normalize input features for the same scale and distribution. It helps to improve the accuracy and generalization of the model. The results are shown in Fig. 14.

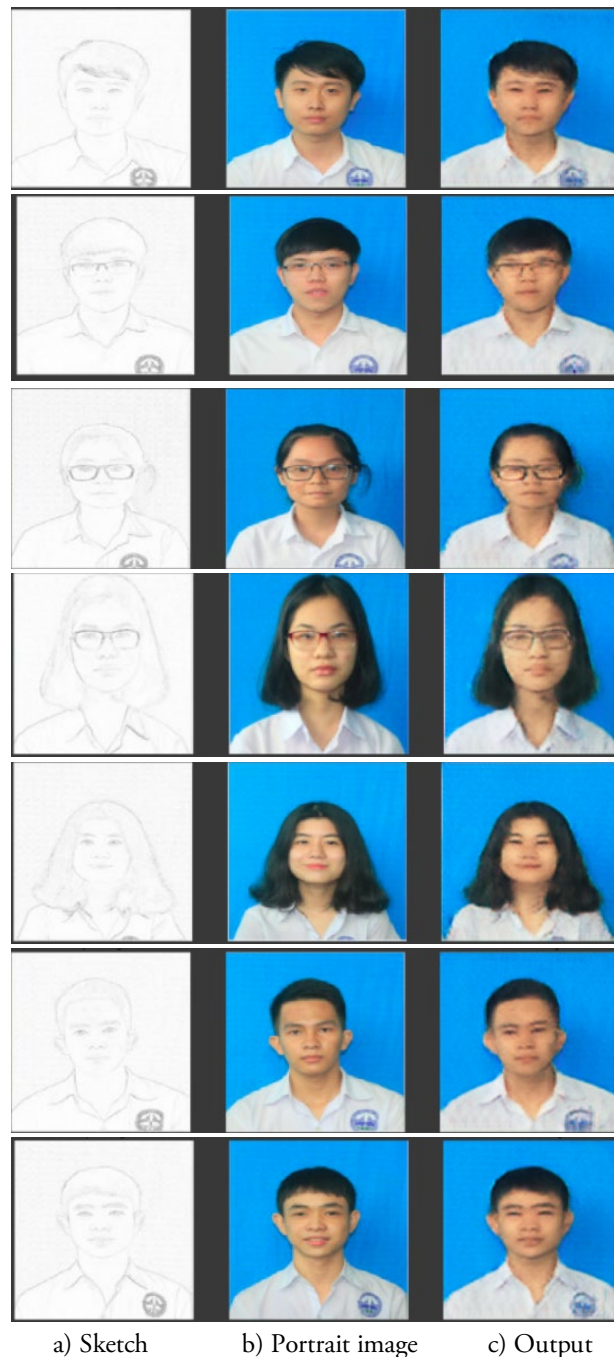


Fig. 14. Results generated from our dataset.

Besides, we compare the results of the proposal method with other methods based on the accuracy of predictions with the input data. The parameters are selected as SSIM (Structural Similarity Index) and PSNR (Peak Signal to Noise Ratio). The SSIM index is calculated on various windows of an image as (11).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (11)$$

where μ_x the average of x , μ_y the average of y , σ_x is the average of x , σ_y is the average of y , c_1 and c_2 are two variables to stabilize the division with a weak denominator.

PSNR is most easily defined via the mean squared error (MSE) between monochrome image I and its noisy approximation K as (12) and (13).

$$PSNR = 20 \log_{10} \left(\frac{2^B - 1}{\sqrt{MSE}} \right), \quad (12)$$

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i, j) - K(i, j)]^2, \quad (13)$$

where, B is selected as 8 bit/sample, M represents the numbers of rows of pixels of the images, i represents the index of that row, and N represents the number of columns of pixels of the image, j representing the column's index. The results are shown in Table 2. The results show that the proposal is the best in terms of accuracy, SSIM, and PSNR.

Table 2. Comparison of accuracy with others.

Method	Accuracy (%)	SSIM	PSNR (db)
Ground Truth	100	1	Infinity
[6]	68.34	0.654	16.588
[24]	73.8	N/A	N/A
[25]	70.55	N/A	N/A
Pix2Pix (DA*) [7]	N/A	0.657	16.617
[26]	67.22	0.695	16.722
[26] (DA*)		0.709	16.843
Proposal	74.65	0.787	25.55

3.3. Discussion

Accuracy: The Discriminator of cGAN is a classification network. So we rely on neural network architecture for image classification for accuracy measure as [27]. We train D with images generated by cGAN and then evaluate its performance on a test set composed of real images. The accuracy of this network is trained on real images and evaluated on the generated images. This measure is similar to precision, with a high value denoting that the generated samples are a realistic approximation of the distribution of real images. When cGAN is not perfect cGAN accuracy will be low.

SSIM and PSNR: we choose two metrics for evaluating model performance, Peak signal-to-noise ratio (PSNR) and Structural index similarity (SSIM), that evaluate the quality of a generated image from sketch images and real images, which aligns with this task. The results are summarized in Table 2. We compare our model with other studies on the CUFS dataset. From Table 2, it can be seen that the method proposed in this paper achieves SSIM is 0.787 and PSNR up to 25.5, which outperforms the listed state-of-the-arts methods [6], [25], [27], [28].

4. Conclusion

In the article, we focus on using the cGANs to create the object from the actual image. We use the human face for training data. The objective of this paper is to apply for criminal investigation science where input data is the sketch. As a result, we create an image of real human face that helps give out the object's identity. The results prove that the algorithm can apply for real applications with up to 75%

accuracy. In the future, we will combine the other methods to improve the accuracy and apply them for real application on the wireless network [26], [28]–[36].

Acknowledgment

This research is carried out in the framework of the project funded by the Ministry of Education and Training (MOET), Vietnam, under grant B2020-BKA-06. The authors would like to thank the MOET for their financial support.

Declarations

Author contribution. All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding statement. None of the authors have received any funding or grants from any institution or funding body for the research.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

- [1] Y. Jo and J. Park, "SC-FEGAN: Face Editing Generative Adversarial Network With User's Sketch and Color," *2019 IEEE/CVF Int. Conf. Comput. Vis.*, pp. 1745–1753, Oct. 2019, doi: [10.1109/ICCV.2019.00183](https://doi.org/10.1109/ICCV.2019.00183).
- [2] I. J. Goodfellow *et al.*, "Generative Adversarial Nets," *Adv. Neural Inf. Process. Syst.*, pp. 2672–2680, Jun. 2014, doi: [10.5555/2969033.2969125](https://doi.org/10.5555/2969033.2969125).
- [3] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "MaskGAN: Towards Diverse and Interactive Facial Image Manipulation," *2020 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 5548–5557, Jun. 2020, doi: [10.1109/CVPR42600.2020.00559](https://doi.org/10.1109/CVPR42600.2020.00559).
- [4] M. Wang *et al.*, "Example-Guided Style-Consistent Image Synthesis From Semantic Labeling," *2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 1495–1504, Jun. 2019, doi: [10.1109/CVPR.2019.00159](https://doi.org/10.1109/CVPR.2019.00159).
- [5] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context Encoders: Feature Learning by Inpainting," *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2536–2544, Jun. 2016, doi: [10.1109/CVPR.2016.278](https://doi.org/10.1109/CVPR.2016.278).
- [6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *2017 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5967–5976, Jul. 2017, doi: [10.1109/CVPR.2017.632](https://doi.org/10.1109/CVPR.2017.632).
- [7] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs," *2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 8798–8807, Jun. 2018, doi: [10.1109/CVPR.2018.00917](https://doi.org/10.1109/CVPR.2018.00917).
- [8] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," *2017 IEEE Int. Conf. Comput. Vis.*, pp. 2242–2251, Oct. 2017, doi: [10.1109/ICCV.2017.244](https://doi.org/10.1109/ICCV.2017.244).
- [9] D. Wu and Q. Dai, "Sketch realizing: lifelike portrait synthesis from sketch," *Proc. 2009 Comput. Graph. Int. Conf.*, pp. 13–20, 2009, doi: [10.1145/1629739.1629741](https://doi.org/10.1145/1629739.1629741).
- [10] H. V. Dinh, "Building database of human to apply to the portrait of the criminal through descriptions of witnesses and victims," *Quang Ninh province police*, 2017. <http://cstc.cand.com.vn>.
- [11] S. A. Israel *et al.*, "Generative Adversarial Networks for Classification," *2017 IEEE Appl. Imag. Pattern Recognit. Work.*, pp. 1–4, Oct. 2017, doi: [10.1109/AIPR.2017.8457952](https://doi.org/10.1109/AIPR.2017.8457952).
- [12] L. Gonog and Y. Zhou, "A Review: Generative Adversarial Networks," *2019 14th IEEE Conf. Ind. Electron. Appl.*, pp. 505–510, Jun. 2019, doi: [10.1109/ICIEA.2019.8833686](https://doi.org/10.1109/ICIEA.2019.8833686).

-
- [13] Y.-J. Cao *et al.*, “Recent Advances of Generative Adversarial Networks in Computer Vision,” *IEEE Access*, vol. 7, pp. 14985–15006, 2019, doi: [10.1109/ACCESS.2018.2886814](https://doi.org/10.1109/ACCESS.2018.2886814).
- [14] M. A. Souibgui and Y. Kessentini, “DE-GAN: A Conditional Generative Adversarial Network for Document Enhancement,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2021, doi: [10.1109/TPAMI.2020.3022406](https://doi.org/10.1109/TPAMI.2020.3022406).
- [15] J. Wang, X. Li, and J. Yang, “Stacked Conditional Generative Adversarial Networks for Jointly Learning Shadow Detection and Shadow Removal,” *2018 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 1788–1797, Jun. 2018, doi: [10.1109/CVPR.2018.00192](https://doi.org/10.1109/CVPR.2018.00192).
- [16] S. Kim and D. Y. Suh, “Recursive Conditional Generative Adversarial Networks for Video Transformation,” *IEEE Access*, vol. 7, pp. 37807–37821, 2019, doi: [10.1109/ACCESS.2019.2906472](https://doi.org/10.1109/ACCESS.2019.2906472).
- [17] N. Hubens, “Deep inside: Autoencoders,” 2018. <https://towardsdatascience.com/deep-inside-autoencoders-7e41f319999f>.
- [18] Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan, “GEE: A Gradient-based Explainable Variational Autoencoder for Network Anomaly Detection,” *2019 IEEE Conf. Commun. Netw. Secur.*, pp. 91–99, Jun. 2019, doi: [10.1109/CNS.2019.8802833](https://doi.org/10.1109/CNS.2019.8802833).
- [19] J. Xue, P. P. K. Chan, and X. Hu, “Experimental study on stacked autoencoder on insufficient training samples,” *2017 Int. Conf. Wavelet Anal. Pattern Recognit.*, pp. 223–229, Jul. 2017, doi: [10.1109/ICWAPR.2017.8076693](https://doi.org/10.1109/ICWAPR.2017.8076693).
- [20] A. Deshpande, J. Lu, M.-C. Yeh, M. J. Chong, and D. Forsyth, “Learning Diverse Image Colorization,” *2017 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2877–2885, Jul. 2017, doi: [10.1109/CVPR.2017.307](https://doi.org/10.1109/CVPR.2017.307).
- [21] R. Tyleček, “The CMP Facade Database,” *Res. Reports C. Czech Tech. Univ. Prague, No. 24, 2012*, pp. 1–8, 2013, [Online]. Available: https://cmp.felk.cvut.cz/~tylecr1/facade/CMP_facade_DB_2013.pdf.
- [22] A. Martinez and R. Benavente, “The AR Face Database: CVC Technical Report, 24,” 1998. Available: [Google Scholar](https://scholar.google.com/citations?user=...).
- [23] K. Messer, J. Matas, J. Kittler, J. Luetttin, and G. Maitre, “XM2VTSDB: The extended M2VTS database,” *Second Int. Conf. audio video-based biometric Pers. authentication*, vol. 964, pp. 965–966, 1999. Available: [Google Scholar](https://scholar.google.com/citations?user=...).
- [24] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. BT - 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, Conference Track Proceedings.” 2016, [Online]. Available: <http://arxiv.org/abs/1511.06434>.
- [25] Q. Chen and V. Koltun, “Photographic Image Synthesis with Cascaded Refinement Networks,” *2017 IEEE Int. Conf. Comput. Vis.*, pp. 1520–1529, Oct. 2017, doi: [10.1109/ICCV.2017.168](https://doi.org/10.1109/ICCV.2017.168).
- [26] K. Shmelkov, C. Schmid, and K. Alahari, “How Good Is My GAN?,” *Ferrari V., Hebert M., Sminchisescu C., Weiss Y. Comput. Vis. – ECCV 2018. ECCV 2018. Lect. Notes Comput. Sci.*, vol. 11206, pp. 218–234, 2018, doi: [10.1007/978-3-030-01216-8_14](https://doi.org/10.1007/978-3-030-01216-8_14).
- [27] S. Gu, J. Bao, H. Yang, D. Chen, F. Wen, and L. Yuan, “Mask-Guided Portrait Editing With Conditional GANs,” *2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 3431–3440, Jun. 2019, doi: [10.1109/CVPR.2019.00355](https://doi.org/10.1109/CVPR.2019.00355).
- [28] V. Carvalho, F. Soares, and R. Vasconcelos, “Artificial intelligence and image processing based techniques: A tool for yarns parameterization and fabrics prediction,” *2009 IEEE Conf. Emerg. Technol. Fact. Autom.*, pp. 1–4, Sep. 2009, doi: [10.1109/ETFA.2009.5347255](https://doi.org/10.1109/ETFA.2009.5347255).
- [29] P. N. Huu, T. Tran Van, and N. G. Thi, “Proposing distortion compensation algorithm for determining distance using two cameras,” *2019 6th NAFOSTED Conf. Inf. Comput. Sci.*, pp. 172–177, Dec. 2019, doi: [10.1109/NICS48868.2019.9023875](https://doi.org/10.1109/NICS48868.2019.9023875).
- [30] P. N. Huu, V. Tran-Quang, and T. Miyoshi, “Energy threshold adaptation algorithms on image compression

- to prolong WSN lifetime,” *2010 7th Int. Symp. Wirel. Commun. Syst.*, pp. 834–838, Sep. 2010, doi: [10.1109/ISWCS.2010.5624318](https://doi.org/10.1109/ISWCS.2010.5624318).
- [31] S. S. Kumar, F. Taheri, and M. R. Islam, “Artificial Intelligence and Image Processing Approaches in Damage Assessment and Material Evaluation,” *Int. Conf. Comput. Intell. Model. Control Autom. Int. Conf. Intell. Agents, Web Technol. Internet Commer.*, vol. 1, pp. 307–313, doi: [10.1109/CIMCA.2005.1631284](https://doi.org/10.1109/CIMCA.2005.1631284).
- [32] S. Shukla, A. Lakhmani, and A. K. Agarwal, “Approaches of artificial intelligence in biomedical image processing: A leading tool between computer vision & biological vision,” *2016 Int. Conf. Adv. Comput. Commun. Autom.*, pp. 1–6, Apr. 2016, doi: [10.1109/ICACCA.2016.7578900](https://doi.org/10.1109/ICACCA.2016.7578900).
- [33] A. K. Rathinam, Y. Lee, D. N. C. Ling, and R. Singh, “A review of image processing leading to artificial intelligence methods to detect instruments in ultrasound guided minimally invasive surgical procedures,” *2017 IEEE Int. Conf. Power, Control. Signals Instrum. Eng.*, pp. 3074–3079, Sep. 2017, doi: [10.1109/ICPCSI.2017.8392290](https://doi.org/10.1109/ICPCSI.2017.8392290).
- [34] X. Jia, “Image recognition method based on deep learning,” *2017 29th Chinese Control Decis. Conf.*, pp. 4730–4735, May 2017, doi: [10.1109/CCDC.2017.7979332](https://doi.org/10.1109/CCDC.2017.7979332).
- [35] J. Ruili, W. Haocong, W. Han, E. O’Connell, and S. McGrath, “Smart Parking System Using Image Processing and Artificial Intelligence,” *2018 12th Int. Conf. Sens. Technol.*, pp. 232–235, Dec. 2018, doi: [10.1109/ICSensT.2018.8603590](https://doi.org/10.1109/ICSensT.2018.8603590).
- [36] W. Chao, L. Chang, X. Wang, J. Cheng, X. Deng, and F. Duan, “High-Fidelity Face Sketch-To-Photo Synthesis Using Generative Adversarial Network,” *2019 IEEE Int. Conf. Image Process.*, pp. 4699–4703, Sep. 2019, doi: [10.1109/ICIP.2019.8803549](https://doi.org/10.1109/ICIP.2019.8803549).