International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I10P202

Volume 71 Issue 10, 11-19, October 2023
© 2023 Seventh Sense Research Group®

Original Article

An Efficient Automatic Image-to-Image Transformation using a Deep Learning Model

Anupama K Ingale[1], A. Anny Leema[2]

[1]*School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, TamilNadu, India.*
[2]*School of Computer science and Engineering, Vellore Institute of Technology, Vellore, TamilNadu, India.*

[2]*Corresponding Author : annyleema.a@vit.ac.in*

Received: 17 October 2022 Revised: 10 February 2023 Accepted: 12 September 2023 Published: 03 October 2023

Abstract *- Image-to-image (i2i) transformation is a class of vision and illustration issues where the objective is to map between an original image and a resultant image utilizing a training set of aligned image pairs. This technique is mainly used for movie post-production, computational photography, face recognition, etc; therefore, deep learning model-based image-to-image translation is proposed in this model. The proposed model initially generates the target image's blended shape expression and then combines the input and blended shape image to produce a new expression image. The model is trained based on the attention mask loss. In this paper, the deep learning model is designed based on the Convolution Neural Network (CNN) and Improved Penguin Optimization (IPO) algorithm called Optimal Convolution Neural Network (OCNN). For experimental analysis, different sets of images are analyzed, and performance is compared with different methods.*

Keywords *- Image to image, Transformation, Optimal Convolution Neural Network, Penguin optimization and Blend shape expression.*

1. Introduction

Suppose you take a selfie and need to find it with additional imagination as an illustration through an illustrator; how might you naturally accomplish that through a PC? This sort of examination effort can be comprehensively considered the picture-to-picture interpretation (I2I) ([1], [2]) issue. As an overall rule, the objective of I2I is to change over an info picture XA through a source space A to an objective space B through the inherent cause satisfied saved and the outward objective elegance moved.

Given the input source image, we must train a mapping that creates an image that cannot be distinguished from the target domain image. In the mathematical model, this conversion procedure can be represented as

$$x_{AB} \in B: x_{AB} = G_{A\leftrightarrow B}(x_A) \qquad (1)$$

From the above essential meaning of I2I, we see that changing a picture starting with one source area over completely and then onto the next target space can cover numerous issues in picture handling, PC illustrations, PC vision, etc. In particular, I2I has been extensively applied in semantic picture combination [1-3], picture division [4-5], style move [6-7], picture in-painting [8-9], 3D posture assessment [10], [11], picture/video colorization [12], picture super-resolution[13], space transformation [14-15], animation age [16] and picture enlistment [17], emotion analysis[18].

With rapid progress in deep learning computations, efforts to break down and understand advanced images for some PC vision applications certainly stand out enough to be noticed in the new year because of the unique execution of such computations and access to a wealth of information. Such computations directly process raw information with removal to essential aimed at spatial hand-crafted variables [1-3]. The strong capability of profound elements to discover how to use multifaceted with indisputable equal component demonstrations naturally has completely progressed the presentation of sophisticated techniques across PC applications, for example, object recognition and face recognition. Both basic constructs and fuzzy variables are detected by deep learning-depend on techniques that can be further categorized into discriminative component knowledge computations with multiplicative component learning computations. Discriminatory models focus on the experience of character development by learning a controllable probability p (x|y) to map an input x to a class name y. One of the most popular techniques used for image processing is using Convolutional Brain Networks (CNN) for variable identification and image organization. Models such as LeanNet [19], AlexNet [20], VGGNet [21], and ResNet [22], all of which run learning computations. Again, building models centered on information exchange to discover fundamental elements from multiple pieces of information in a single system. Such models can generate new examples by predicting the joint probability distribution p (x,y) and y [23] in settings, for example, picture super-goal [24], text-to-picture age [25], and image.

2. Literature Survey

There are several approaches to image analysis and expression transfer. Among them, some works are analyzed here; Pumarola et al. [24] analyzed automatically aware facial animation using animation. It can create a discrete number, still up in the air by the substance of the dataset. Here, they presented a Generative Adversarial Networks (GAN) modeling plan given Action Units (AU) comments, which depicts in a constant complex the physical facial developments characterizing a human expression. Their methodology permits controlling the extent of enactment of every AU and consolidates a few of them. The broad assessment shows that their methodology goes past contending restrictive generators in the capacity to blend a much more extensive scope of expressions controlled by physically practical muscle developments, as in the limit of natural picture manipulation. Isola et al. [28] explained a conditional adversarial network. It is not just gaining the planning through input picture to yield a picture, but also becoming familiar with a misfortune capability to prepare this planning.

This makes it conceivable to apply a very convenient way to deal with issues that customarily would require altogether different misfortune details. Their exhibit is compelling at orchestrating photographs among different assignments. Ververas and Zafeiriou [29] had developed image-to-image translation. Specifically, they presented the SliderGAN, which changes an info face picture into another one as indicated by the nonstop upsides of a measurable blend shape model of facial movement. They demonstrate it is feasible to alter a facial picture as per appearance and discourse blend shapes, utilizing sliders that control the consistent upsides of the blend shape model. Pranavi et al. [30] developed semantic image-to-image translation using a machine learning algorithm. Here, the Cycle GAN algorithm is introduced to translate an image to generate an image. This method applies to both linked and unlinked images. In this paper, they used rotational stability loss. Wiles et al. [31] developed an NN model to control the pose and expression.

This method is mainly used for image editing and lightweight and sophisticated video. Here, they debuted X2Face, which can be used to produce a framing identity, framing identities of the source frame while the form and emotion of the face in the driving frames using a source face in the driving frame. Second, they developed a technique that allows the network to fully self-monitor utilizing a sizable quantity of video data gathering. Third, we demonstrate that alternative techniques like audio or posture codes can drive the generating mechanism without further system training. Zhu et al. [32] had presented unpaired image-to-image translation. To accomplish this idea, a cycle-consistent adversarial network was proposed. Subjective outcomes are introduced on a few undertakings where matched preparing information does not exist, including assortment style move, object change, season move, photograph improvement, etc. Quantitative examinations against a few earlier strategies exhibit the prevalence of our methodology. Booth et al. [33] developed a 3D reconstruction of "in-the-wild" faces in images and videos. In this study, we combine statistical modeling of face detection and identification and expression characteristics using "in-the-wild" texture modeling to create an initial "in-the-wild" 3DMM. They demonstrate how, since no lighting variable tuning is necessary, a fitting method for photos and videos may be created using this technique. They have created three brand-new databases, the first of their type, that pair "in-the-wild" pictures, videos, and 3D facial shapes.

In this paper, we propose an image generation model using CNN and optimize the performance using improved penguin optimization to transfer and create new facial expressions.

3. Expression Blend Shape Model Creation

Although they provide a useful quantitative method for describing facial movement, mixed contour structures are frequently employed in computer vision tasks. Transparent 3D face models that have been individually put together form the basis of the design. In greater detail, each emotive mesh was subtracted with the neutrality mesh of each associated sequence to create a matrix that includes m difference vectors. $d_i \in R^{3n}$. This process is known as sparse Principal Component Analysis (PCA). Therefore, in the following minimization problem, the sparse blend shape components $H \in R^{h \times 1}$ are recovered.

$$argmin\|P - DH\|_F^2 + \Omega(H)S.t\upsilon(B) \qquad (2)$$

Where the constraint υ can either be $max(|D_k|) = 1$, $\forall k$or$max(|D_k|) = 1, \quad D \geq 1, \quad \forall k,$ with$D_k \in R^{3n \times 1}$denoting the k^{th} component of the sparse weight matrix $D = [D_1, \dots, D_h]$. A graphical representation of the five blend-shape models S_{exp} is given in Figure 1. The 3D faces of this figure are created by adding the multiplied elements to the mean face.

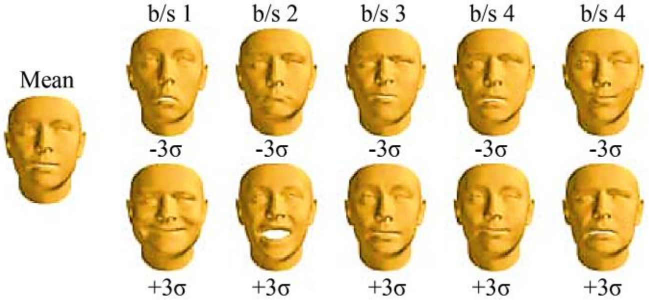


Fig. 1 Blend shape model

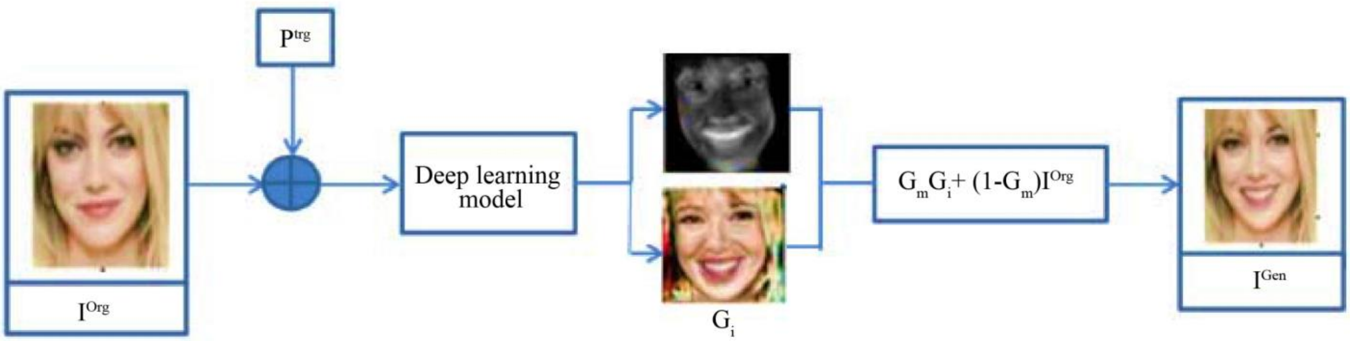


Fig. 2 Overall structure of the proposed methodology

4. Proposed Methodology

The primary objective of the proposed methodology is to effectively transmit the image into another image using a blend shape model. To achieve this concept, a convolution neural network is developed. CNN is effectively transmitting the image. The overall structure of the proposed methodology is given in Figure 2.

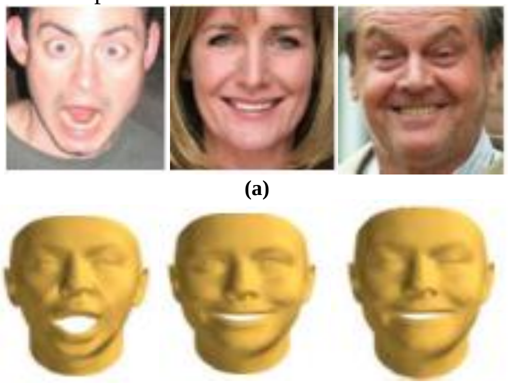
4.1. Problem Definition

Here, we first look at the problem under analysis. Consider the input image. $I^{org} \in R^{H \times W \times 3}$ it represents the face of spontaneous appearance. Also, we accept the slight facial distortion in the image.I^{org}, can be determined by a variable vector$P^{org} = [P^{org1}, P^{org2}, \dots, P^{orgN}]^T$, of N continuous scalar values$P^{org,i}$, normalized in the range$[-1,1]$.

Moreover, the same vector P^{org}establishes the variable of a linear 3D blend shape classic S^{exp} is given in Figure 3 and calculated using Equation (3).

$$S^{exp(P^{\bar{o}rg})^{exp^{org}}} \quad (3)$$

Where, $\bar{S}$ represent the mean 3D face constituent, the appearance Eigen basis of the 3D blend shape classic expression is represented U^{exp}.



(a)

(b)

Fig. 3 3D representation of different input expressions (a) Input image and (b) blend shape model

Our aim is to create a generation classic that, assuming an input image I^{org} and a target exposure variable vectorP^{trg}, can produce a new variety of the input image I^{gen} through the replicated exposure assumed by the 3D exposure event $S^{exp(P^{trg})}$.

4.2. Convolution Network-Based Generator

The convolution network is used to generate a new generator image I_{gen}, trained on a vector of 3D blend shape parameters P^{trg}. The service contains two parallel output layers to reserve the content as a shade of the unique image. The first layer produces a smooth deformation mask. $G_m \in R^{H \times W}$ and an additional distorted image $G_i \in R^{H \times Wx3}$. By requiring a sigmoid activation, the parameters are constrained to the range [0, 1]. The target expressive images are then created by combining the source image.

$$I_{Gen} = G_m G_i + (1 - G_m) I_{org} \quad (4)$$

To enhance the performance of CNN, the weight values are optimally selected using an adaptive penguin optimization algorithm. The working principle of CNN is explained below;

4.2.1. Convolution Neural Network

The convolution layer is a multi-layer perceptron used for many applications, namely, object detection, image processing, and compression. The network mainly consists of three layers: the convolution layer, the pooling layer, and the fully connected layer. Each layer has a different performance. In this paper, we use two output layers with different loss functions.

Convolution Layer

The initial layer of CNN is the convolution layer, which consists of kernels. In this, input data are convoluted with the kernel to generate the output feature maps. The size of the kernels is initialized at the beginning itself. However, the kernel is much smaller than the input image. The mathematical function of the transformation process is given in Equation (5).

$$Y_i^l = B_j^l \sum_{i=1}^{in} K_{ji}^l * x_i^{l-1} \quad (5)$$

Where, $*$ represent the Convolution operator, the index of the output channel is represented as i, the index of the input channel is represented as j, B represents the Bias, k represents the Kernel weight, and the previous layer feature map is represented as x_i^{l-1}.

In the above Equation (5), the IPO algorithm optimally selects the weight values. The first convolution layer in CNN captures features such as color, gradient orientation, and margin. The convolution effect lessens the dimension of the info signal, which is observed by imposing extra bits around the input image.

Activation Function: The output of the first convolution layer output is given to the activation function. This middle function converts the output of the first convolution layer to the next. Two types of activation functions are utilized in this paper, namely, hyperbolic tangent (Tanh) and Rectified Linear Unit (ReLU). This function eliminates negative values from the input and changes them to zero. The function of ReLU is max (0, x), which makes it highly computable. Tanh is a non-linear function executed in the range (-1, + 1), which is calculated mathematically using Equation (6).

$$F(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \tag{6}$$

Pooling Layer

This layer extracts features such as positional, translational, and rotational invariants. Two polling operations are available, namely, average pooling and max pooling. With the former, the elements of a fixed part are replaced with the maximum element of the part, and with the latter, the elements are replaced with the average or arithmetic sum of the elements of the part. The experimental used maximum pooling layer kernel size = (2, 2).

Fully Connected Layer

It is used for the classification process. This layer is also called the output layer. In this paper, two output layers with different loss functions are used. The first layer output gives the smooth deformation mask, and the other a deformation image. To enhance the performance of the CNN classifier, the weight values are optimally selected utilizing an adaptive sunflower optimization algorithm. The basic structure of CNN is given in Figure 4.

4.2.2. Weight Optimization using IPO algorithm

The IPO algorithm is utilized to regulate the weight standards of the CNN. The Penguin Optimization Algorithm is a new transformation mechanism inspired by the eating behavior of penguins. The purpose of this method is to find optimal solutions. The penguins of a group dive simultaneously, work as a group, and collectively feed the fish, the power satisfied of which resembles the merits of solutions. Fish stocks focus more on the best part of the arrangement and then begin the pursuit. The step in weight optimization is explained below;

Step 1: Solution Initialization

For every optimization problem, Solution initialization is essential. The solution consists of weight values. In the proposed penguin optimization, the populations are split into numerous groups. Each group consists of a number of penguins that change their position based on food availability. Here, the solutions are called penguins. The solution format is given in Equation (7).

$$S_j = \{P_j^1, P_j^2, \dots P_j^n\} \tag{7}$$

Where, P_j^1 represent the penguins or solutions. The solution consists of a different set of weight values.

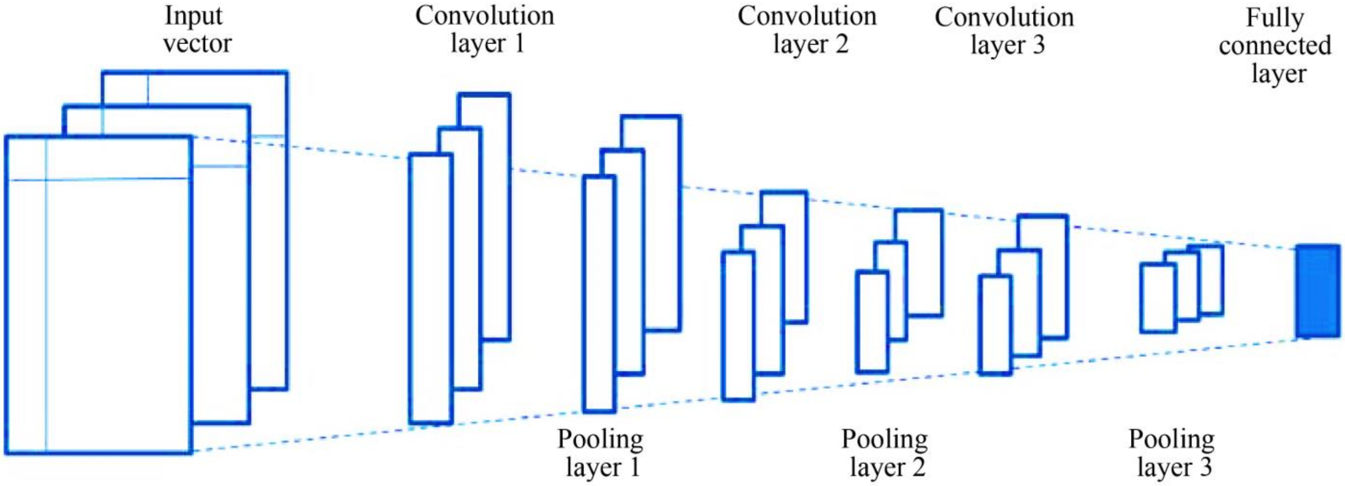


Fig. 4 Structure of convolution neural network

Step 2: Opposite Solution Generation

Once we initialize the solution, the opposite solutions are created. This step is used to increase the searching ability of the algorithm. Where, $X \in [a, b]$ be a real number, the opposite solution $\overline{S_j}$ can be calculated as follows;

$$\overline{S_j} = a + b - S \tag{8}$$

Step 3: Fitness Calculation

Applying fitness value, the solution's goodness is evaluated. We assess the efficiency of each strategy after initiation. The fitness function is thought to be the greatest resemblance. The fitness function is given in Equation (9).

$$Fitness = Max(Similarity) \tag{9}$$

$$Similarity = T\,arg\,e\,t - output \tag{10}$$

Step 4: Swimming Course Update

After fitness calculation, the penguin's positions are updated. The penguin swims to a fresh location at time $t + 1$in the whole solution space is expressed in the following Equation (11);

$$P_j^i(t+1) = P_j^i(t) + X_j^i(t) \times Rand() \times \left(P_{localbest}^i - P_j^i(t)\right) \tag{11}$$

Where, $P_j^i(t+1)$ is the new location of the penguin, $P_j^i(t)$ is the old location of the penguin, $P_{localbest}^i$is the location of the local best, and $Rand()$is represented as a random number [0, 1].

Step 5: Oxygen Reserve Update

After each diving, the penguin's oxygen reserve is efficient. It is expressed in the following Equation (12);

$$O_j^i(t+1) = O_j^i(t) + \left(F\left(P_j^i(t+1)\right) - F\left(P_j^i(t)\right)\right) \times \left|P_j^i(t+1) + P_j^i(t)\right| \tag{12}$$

Where, $O_j^i(t+1)$ is the new oxygen reserve and$F$is represented as the objective function. If the updated solution is better than the old one, the oxygen reserve increases; if the new solution is worse, the oxygen reserve decreases.

Step 6: Food abundance update: Penguin's updated function is given in Equation (13);

$$QEF^i(t+1) = QEF^i(t) + \sum_{j=1}^{d_i}\left(O_j^i(t+1) - O_j^i(t)\right) \tag{13}$$

Where, $QEF^i(t)$ it represents the quantity of eaten fish of the ith group at the tth instance.

A large QEF value means that the area provides adequate food for the entire group and demands migratory penguins from other groups.

Step 7: Group Membership Update

The membership of the penguin is updated according to the degree of food abundance in various groups. The membership function is calculated using Equation (14).

$$R_i(t+1) = \frac{QEF^i(t)}{\sum_{j=1}^{k} QEF^j(t)} \tag{14}$$

Step 8: Termination Criteria

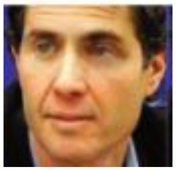
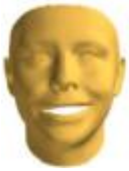
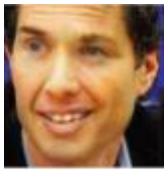
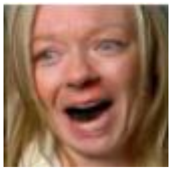
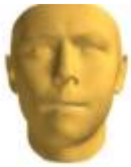
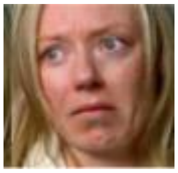
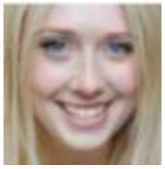
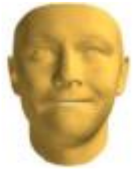
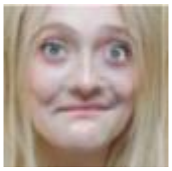
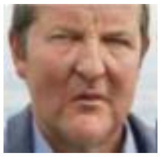
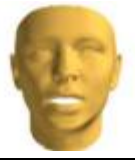
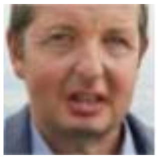
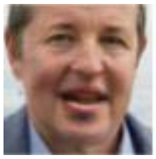
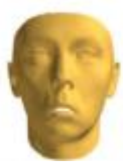
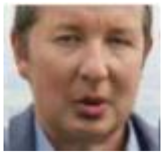
The procedure is carried on till the best sunflower or weight value is selected. The algorithm will be terminated once the best flower has been found. The CNN is provided with this improved weight value.

4.2.3. Attention Mask Loss Calculation

We use a sparsity loss L^{att} to induce the convolution network to create a spatial attention mask G_m that focuses on the distortion regions and does not saturate to 1. In other words, for the created and recreated images, we estimate and minimize the L1-norm of the constructed masks, defining the loss as:

$$L^{att} = \frac{1}{W \times H}\left(\|G_m(I^{org}, P^{trg})\|_1 + \|G_m(I^{gen}, P^{org})\|_1\right) \tag{15}$$

Table 1. Experimental results of proposed image-to-image translation

Input image	Blendshape	output
		
		
		
		
		

5. Experimental Results

The performance of the recommended approach is discussed in this section. The recommended work is implemented using Python. We use an Intel i5/core i7 processor, 4GB RAM, and 3.20 GHz CPU speed for implementation. The performance of the recommended approach compared with different methods. The input image and corresponding blend shape model are given in Table 1.

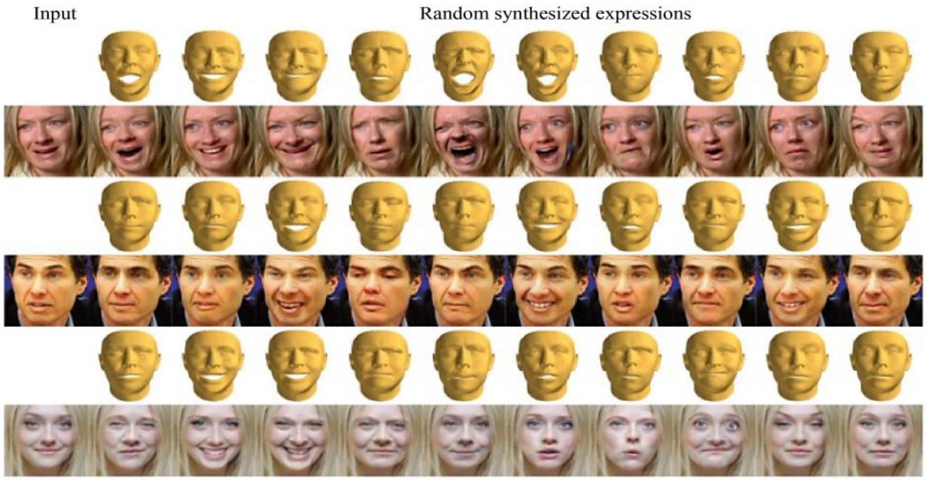


Fig. 5 Expressive faces generated by proposed deep learning model (b/s) parameters in the range [−1, 1]

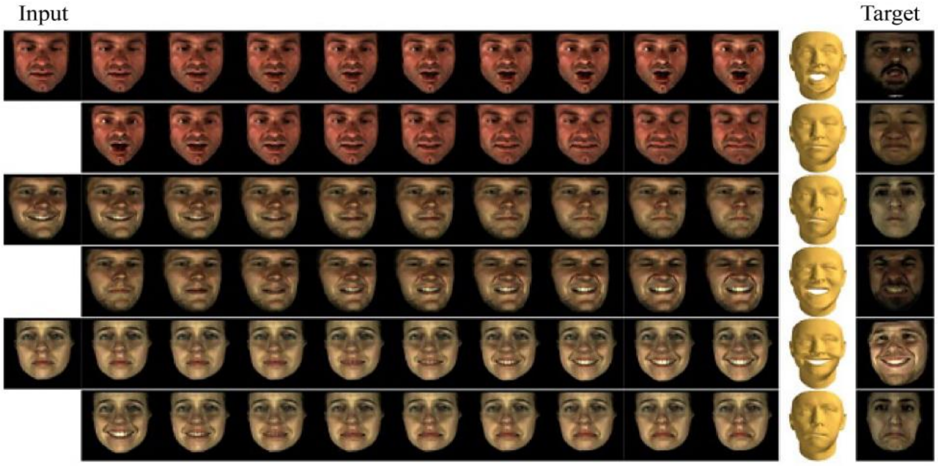


Fig. 6 Expressive faces generated by proposed deep learning model (b/s) parameters in the range [−1, 1]

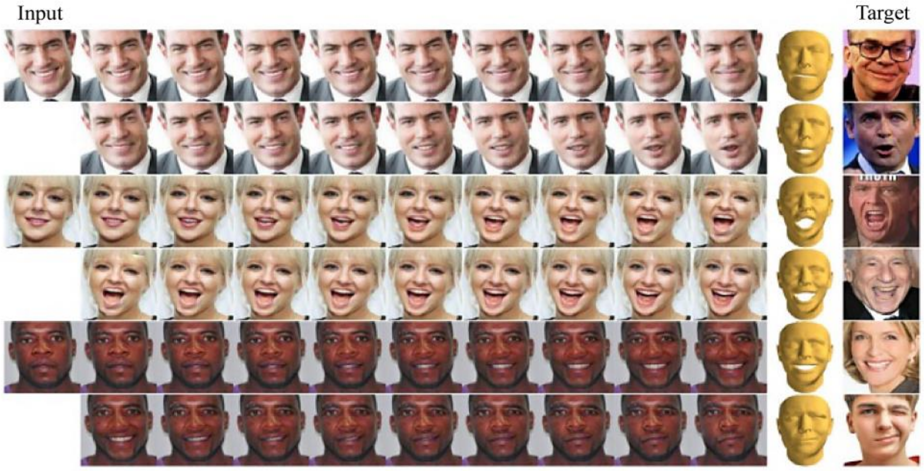


Fig. 7 Expressive faces generated by proposed deep learning model (b/s) parameters in the range [−1, 1]

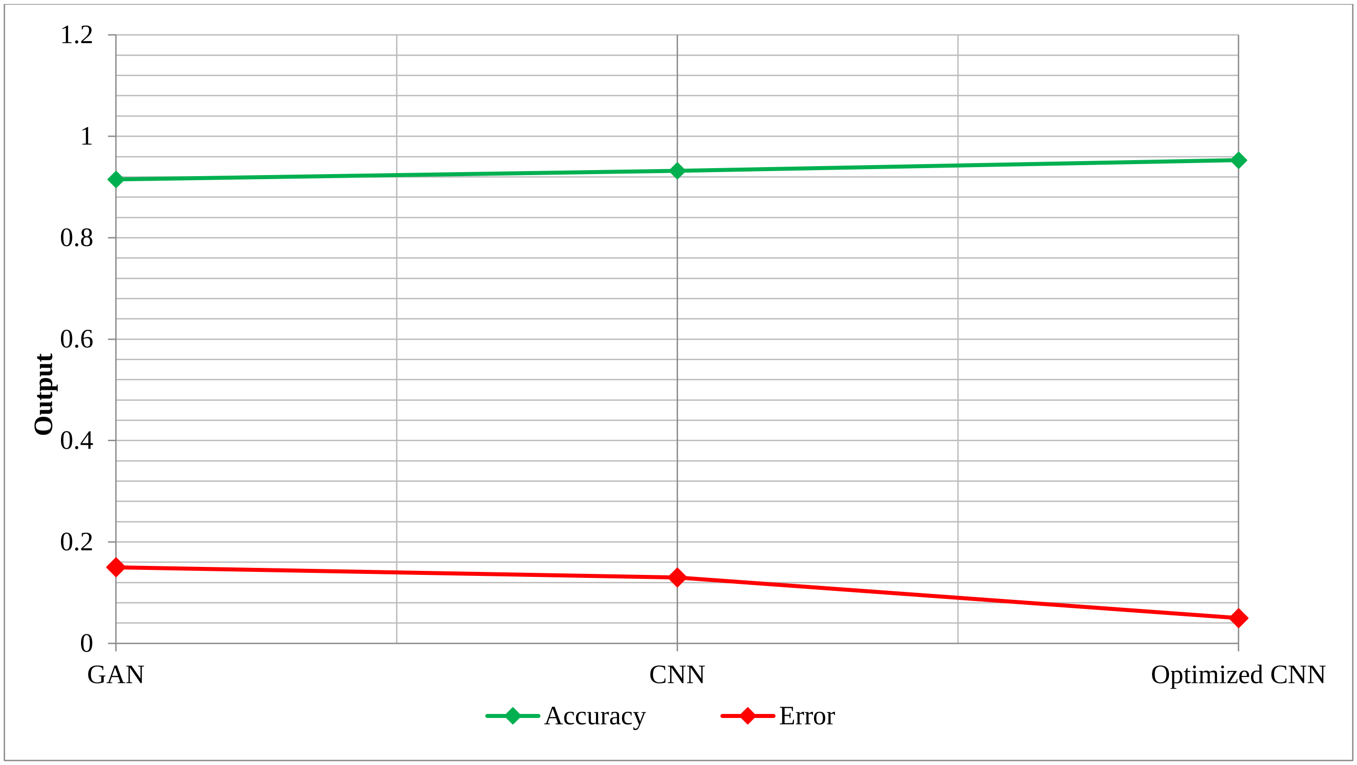


Fig. 8 Comparative analysis output

Table 2 discusses the effectiveness of the recommended method based on accuracy and error. As illustrated in Table 2, our recommended optimized CNN technique achieved a higher accuracy of 95.3%, which is 91.2% for CNN-based image-to-image translation and 92.5% for GAN-based image-to-image translation. According to the results, it is clear that the recommended approach reached a superior output compared to another method. Moreover, our recommended approach attained a minimum error rate of 0.05%, which is low compared to other methods. The error value is calculated based on the target and obtained image. The visual plot of Table 2 is given in Figure 8.

To evaluate the processing time of the proposed and other methods, we run this experiment with an input image resolution of 1280x720. Table 3 shows that our method shows a significant advantage in processing speed. It is the IPO algorithm that brings this advantage. The table corresponding visual representation is given in Table 3.

Table 4 discusses the effectiveness of the recommended technique based on PSNR. The quality of the output image is measured using PSNR. The high PSNR value should be a good system. When analyzing Table 4, our recommended approach reached a higher PSNR of 23.71 dB, which is 19.5 dB for CNN-based image translation and 20.2db for GAN-based image translation. The visual representation of Table 4 is given in Figure 10.

Table 2. Performance analysis based on accuracy and error

Methods	Accuracy	Error
GAN	92.5	0.15
CNN	91.2	0.13
Optimized CNN	95.3	0.05

Table 3. Comparative analysis based on the processing time

Methods	Processing time (ms)
GAN	954
CNN	1256
Optimized CNN	506

Table 4. Comparative analysis based on PSNR

Methods	PSNR
GAN	20.2
CNN	19.5
Optimized CNN	23.71

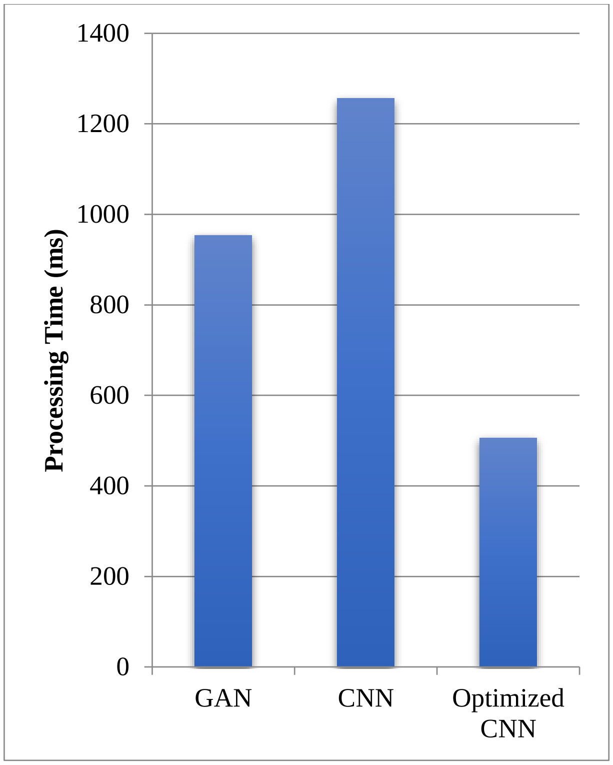


Fig. 9 Processing time

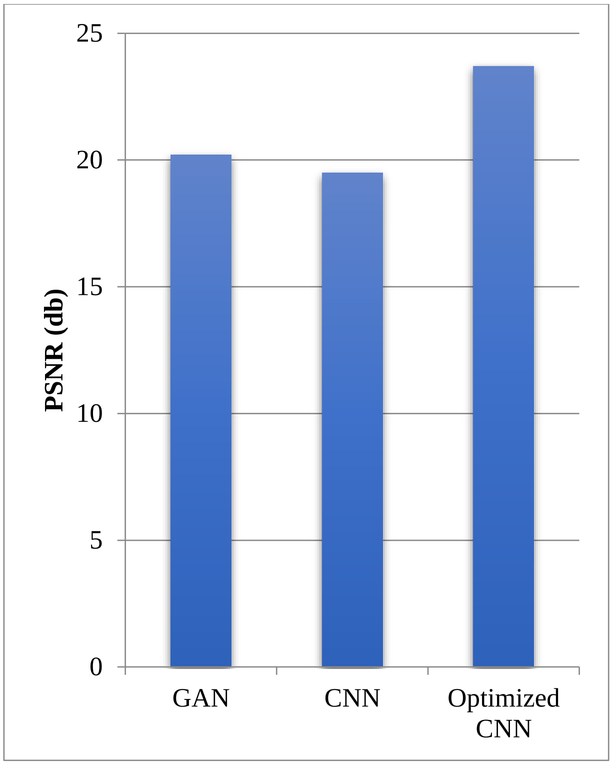


Fig. 10 PSNR measurement

From the results section, we clearly understand that we attained a higher output compared to the other method.

6. Conclusion

An efficient image-to-image translation using a deep learning model has been presented. Here, we generate the blend shape model for different input expressions. A by-product of the learning process is an exceptionally strong regression network that maps the image into various blend shape boundaries, which can then be utilized for conditioning the inputs of the OCNN classifier, which generates the new set of images. The proposed model has been designed based on the input image, blend shape expression, and target image. To train the optimized CNN, neural network attention loss is utilized. The effectiveness of the recommended approach is analyzed based on PSNR, Error, and processing time. In the future, we will develop real-time implementation of image-to-image translation.

References

[1] Phillip Isola et al., "Image-to-Image Translation with Conditional Adversarial Networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125-1134, 2017. [Google Scholar] [Publisher Link]

[2] Jun-Yan Zhu et al., "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223-2232, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[3] Krishna Regmi, and Ali Borji, "Cross-View Image Synthesis using Conditional Gans," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3501-3510, 2018. [Google Scholar] [Publisher Link]

[4] Qianye Yang et al., "MRI Cross-Modality Neuroimage-to-Neuroimage Translation," *arXiv preprint*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] Xi Guo et al., "Gan-based Virtual-to-Real Image Translation for Urban Scene Semantic Segmentation," *Neurocomputing*, vol. 394, pp. 127-135, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] Taeksoo Kim et al., "Learning to Discover Cross-Domain Relations with Generative Adversarial Networks," *International Conference on Machine Learning*, vol. 70, pp. 1857-1865, 2017. [Google Scholar] [Publisher Link]

[7] Zili Yi et al., "DualGAN: Unsupervised Dual Learning for Image-to-Image Translation," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2849-2857, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[8] Deepak Pathak et al., "Context Encoders: Feature Learning by Inpainting," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[9] Jun-Yan Zhu et al., “Toward Multimodal Image-to-Image Translation,” *Advances in Neural Information Processing Systems*, pp. 465-476, 2017. [Google Scholar] [Publisher Link]

[10] Hsiao-Yu fish Tung et al., “Adversarial Inverse Graphics Networks: Learning 2d-to-3d Lifting and Imageto-Image Translation from Unpaired Supervision,” *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. [Google Scholar] [Publisher Link]

[11] Bowen Li et al., “Manigan: Text-Guided Image Manipulation,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. [Google Scholar] [Publisher Link]

[12] Patricia L. Suarez, Angel D. Sappa, and Boris X. Vintimilla, “Infrared Image Colorization based on a Triplet Dcgan Architecture,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 18-23, 2017. [Google Scholar] [Publisher Link]

[13] Yuan Yuan et al., “Unsupervised Image Super-Resolution using Cycle-in-Cycle Generative Adversarial Networks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 701-710, 2018. [Google Scholar] [Publisher Link]

[14] Jinming Cao et al., “Dida: Disentangled Synthesis for Domain Adaptation,” *arXiv preprint*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] Alexander H.Liu et al., “A Unified Feature Disentangler for Multi-Domain Image Translation and Manipulation,” *Advances in Neural Information Processing Systems*, pp. 2590-2599, 2018. [Google Scholar] [Publisher Link]

[16] Yichun Shi, Debayan Deb, and Anil K.Jain, “Warpgan: Automatic Caricature Generation,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10762-10771, 2019. [Google Scholar] [Publisher Link]

[17] Moab Arar et al., “Unsupervised Multi-Modal Image Registration via Geometry Preserving Image-to-Image Translation,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [Google Scholar] [Publisher Link]

[18] K.V Sriram, and R.H. Havaldar, “Convolutional Neural Network Based Data Security in Image Steganography,” *SSRG International Journal of Electrical and Electronics Engineering*, vol. 10, no. 7, pp. 102-109, 2023. [CrossRef] [Publisher Link]

[19] Y.LeCun et al., “Gradient-Based Learning Applied to Document Recognition,” *Proceedings of the IEEE*, vol. 86, pp. 2278-2324, 1998. [CrossRef] [Google Scholar] [Publisher Link]

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E.Hinton, “Imagenet Classification with Deep Convolutional Neural Networks,” *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1097-1105, 2012. [Google Scholar] [Publisher Link]

[21] Karen Simonyan, and Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv preprint*, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[22] Kaiming He et al., “Deep Residual Learning for Image Recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016. [Google Scholar] [Publisher Link]

[23] Andrew Ng, and Michael Jordan “On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes,” *Proceedings of the Advances in Neural Information Processing Systems*, pp. 841-848, 2001. [Google Scholar] [Publisher Link]

[24] Xintao Wang et al., “Esrgan: Enhanced Super-Resolution Generative Adversarial Networks,” *Proceedings of the European Conference on Computer Vision*, pp. 63-79, 2018. [Google Scholar] [Publisher Link]

[25] Scott Reed et al., “Generative Adversarial Text to Image Synthesis,” *arXiv preprint*, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[26] Deblina Bhattacharjee et al., “DUNIT: Detection-Based Unsupervised Image-to-Image Translation,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4787-4796, 2020. [Google Scholar] [Publisher Link]

[27] Deblina Bhattacharjee et al., “Ganimation: Anatomically-Aware Facial Animation from a Single Image,” *Proceedings of the European Conference on Computer Vision*, pp. 818-833, 2018. [Google Scholar] [Publisher Link]

[28] Phillip Isola et al., “Image-to-Image Translation with Conditional Adversarial Networks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125-1134, 2017. [Google Scholar] [Publisher Link]

[29] Evangelos Ververas, and Stefanos Zafeiriou, “Slidergan: Synthesizing Expressive Face Images by Sliding 3d Blendshape Parameters,” *International Journal of Computer Vision*, vol. 128, no. 10, pp. 2629-2650, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[30] G.Pranavi et al., “Semantic Image to Image Translation using Machine Learning Algorithms,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 9, no. 6, pp. 1973-1976, 2020. [CrossRef] [Publisher Link]

[31] Olivia Wiles, A.Sophia Koepke, and Andrew Zisserman, “X2face: A Network for Controlling Face Generation by using Images, Audio, and Pose Codes,” *European Conference on Computer Vision*, 2018. [Google Scholar] [Publisher Link]

[32] Jun-Yan Zhu et al., “Unpaired Image-Toimage Translation using Cycle-Consistent Adversarial Networks,” *IEEE International Conference on Computer Vision*, 2017. [Google Scholar] [Publisher Link]

[33] James Booth et al., “3d Reconstruction of “In-the-Wild” Faces in Images and Videos,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 11, pp. 2638-2652, 2018. [CrossRef] [Google Scholar] [Publisher Link]