 http://dx.doi.org/10.14500/aro.11051

Employing Neural Style Transfer for Generating Deep Dream Images

Lafta R. Al-Khazraji[1,2], Ayad R. Abbas[1], and Abeer S. Jamil[3]

[1]Department of Computer Science, University of Technology-Iraq, Baghdad, Iraq

[2]General Directorate of Education of Salahuddin Governorate, Iraq

[3]Department of Computer Technology Engineering, Al-Mansour University College, Baghdad, Iraq

***Abstract*—In recent years, deep dream and neural style transfer emerged as hot topics in deep learning. Hence, mixing those two techniques support the art and enhance the images that simulate hallucinations among psychiatric patients and drug addicts. In this study, our model combines deep dream and neural style transfer (NST) to produce a new image that combines the two technologies. VGG-19 and Inception v3 pre-trained networks are used for NST and deep dream, respectively. Gram matrix is a vital process for style transfer. The loss is minimized in style transfer while maximized in a deep dream using gradient descent for the first case and gradient ascent for the second. We found that different image produces different loss values depending on the degree of clarity of that images. Distorted images have higher loss values in NST and lower loss values with deep dreams. The opposite happened for the clear images that did not contain mixed lines, circles, colors, or other shapes.**

***Index Terms*—Deep dream, Gradient ascent, Gram matrix, Neural style transfer.**

I. Introduction

Deep learning (DL) has the ability to achieve complicated cognitive tasks exceeding the performance of humans. Because the performance of DL algorithms, such as deep CNN, gives great results compared to other machine learning machine learning (Abedi, et al., 2020; Alzubaidi, et al., 2021).

The deep dream had been developed as the newest DL technology that was produced by Google, which is a technique that works to improve the visual attributes of images (Mordvintsev, et al., 2015).

The deep dream is created through the repeating feeds of the image to the CNN model, where first, the low-level features (i.e., edges and lines) are detected by the first layer. After that, high-level features such as faces and trees appeared. In the end, all those features are collected for configuring combined effects such as trees or the whole structure (Khan, et al., 2020).

Neural style transfer (NST) is a technology that works under the umbrella of deep learning and is considered one of the attractive deep learning applications (Singh, et al., 2021).

In NST, the input consists of two images; a content image and a style image, whereas there is only one output image that combines the contents of the content images with the style of the style image (Li, 2018).

II. Motivations Behind the Study

Many people have a great passion for drawing, but unfortunately, they lack the skill to do it, and this is what is known today as NST. In recent times, deep dream emerged as a hot topic and are used in many important fields as a simulation tool; simulating hallucinations among psychiatric patients and drug addicts have become a very atheistic necessity to enable doctors and those responsible for psychiatric clinics to imagine a vision approaching the reality that these patients see.

Many cases combine hallucinations with seeing artistic images in a certain style, and therefore, this study came to combine the use of these two techniques.

III. Related Works

This section presents some of the most closely related works to NST and deep dream, starting with NST.

A. NST

Chen, et al. (2020) proposed a lightweight network to enable video style transfer by a knowledge distillation model. Two teacher networks were used; the former was used during the inference by taking the optical ow, whereas the latter did not. The output variation between those two

ARO-The Scientific Journal of Koya University
Vol. X, No. 2 (2022), Article ID: ARO.11051. 8 pages
DOI: 10.14500/aro.11051
Received 22 August 2022; Accepted: 17 November 2022
Regular research paper: Published: 01 December 2022
Corresponding author's e-mail: cs.20.15@grad.uotechnology.edu.iq

networks indicates that optical ow had improved, which is then embraced to distill the target student network. They employed the low degree distillation loss to constrain the student network output, which is stylized videos to simulate the low degree of the input video. They measured their work quantitatively and qualitatively; in the former, they compared their method with style Candy from other databases by calculating the temporal error and employing it for calculating the temporal consistency. They got the minimum error compared to the other methods. In the qualitative analysis, also, their method got higher temporal consistency.

Kotovenko, et al. (2019) presented a model based on a block that transforms the content; it works as a dedicated portion of the network to change in a content- and style-specific way. They focused on learning how it is necessary to transform the details of the content image by utilizing objects from the same category in both content and style images. In general, they improved that their model has the ability to stylize the content details from a single complex object class. Their model was evaluated by persons who had no art information and art experts, and both of them voted that the current model is better than and preferred compared to other previous studies. In addition, the objective and subjective evaluations indicated that their model is better than the listed methods in their related works section in terms of the quality of stylization.

Choi (2022) presented a model that used second-order statistics of the encoded features to construct an optimal arbitrary image style transfer technique. Their study had two contributions; in the former, they proposed a new technique for correlation-aware loss and feature alignment. This regular merging of loss and feature alignment techniques robustly match the statistics of the second order of the content features and the target style features, thus, increasing the decoder network style capacity. Whereas in the latter, they proposed a new component-wise style controlling technique. Their method could generate diverse styles depending on singular or multistyle images by utilizing style-specific components from second-order feature statistics. They proved that their model accomplishes improvements in the decoder network style capacity and style diversity without losing the capability to process in real time on GPU devices (under 200 ms). The pre-trained VGG16 network was used for the encoder, whereas for the decoder, trainable VGG16 was used. MS-COCO dataset was used as a content image dataset, whereas a small dataset that consisted of 22 images represents style images. Finally, they relied on increasing the training data to measure the losses of the networks.

B. Deep Dream

Yin, et al. (2020) presented a model called DeepInversion. Their model consisted of two sections; the first is teacher, whereas the second is called student. The teacher started with random noise, where the training dataset did not use any extra information. The DeepInversion model is built based on deep dream by improving the quality of images of the deep dream by adding a new feature to the image regularization. ImageNet and CIFAR-10 datasets had used to train the model.

Kiran (2021) presented a deep dream algorithm. The training is started as soon as the image is entered into the model. With this algorithm, the input image is modified by firing particular neurons. Thus, the layers had the ability to choose particular neurons to fire. This process is continued until all the required features by the input image become in the targeted layer. Hence, the more feeding images, the more able to extract more features. ReseNet, CNN, and ANN pre-trained model had used to extract the image features. At each layer, they executed gradient ascent to maximize the loss function.

El-Rahiem, et al. (2022) presented a multi-biometric cancellable scheme (MBCS) using deep dream. The purpose was to create secure and effective fingerprint cancellable patterns depending on the veins of the finger. The Inception v3 model had used to accomplish this study. They maximized the loss function using gradient ascent, as large the number of iterations as the loss maximized.

IV. BACKGROUND

This section browses the most important concepts relative to our study. As mentioned above, the model is divided into two parts; style transfer and deep dream. Some concepts related to style transfer, whereas others are used with the deep dream.

A. NST General Concepts

The concepts related to NST are VGG-19, Gram matrix, and gradient descent.

VGG-19 network

VGG-19 network is used as a pre-trained model, consisting of 19 layers to perform the convolution process; it has 16 convolutional layers and 5 max pooling layers. Each convolutional layer is followed by a rectified linear unit (ReLU) which works as an activation function. These layers (convolutional and ReLU) are fully connected with the max pooling layer. Unlike traditional convolution, the multiple layers of convolutional and non-linear activation functions enable to extraction of additional image features, whereas max-pooling layers are used to downsampling and select the most valued features. Finally, the fully connected layer is the last in this network. It takes the results of the previous layers, makes flattens, and delivers its output to the activation function (often SoftMax function). Fig. 1 shows the structure of the VGG-19 model (Rashid, et al., 2020; Sudha and Ganeshbabu, 2021; Xiao, et al., 2020).

Gram matrix

First, the features of content and style images have been extracted using the convolutional neural network (CNN). To make the style features that have been extracted more useful, one pre-processing additional step is required. Therefore, the Gram matrix is an important step to make the extracted style features more effective. All the extracted features by the CNN still retain the information related to the image's content, such as object structure and positioning. The gram matrix is then applied to those extracted features to remove

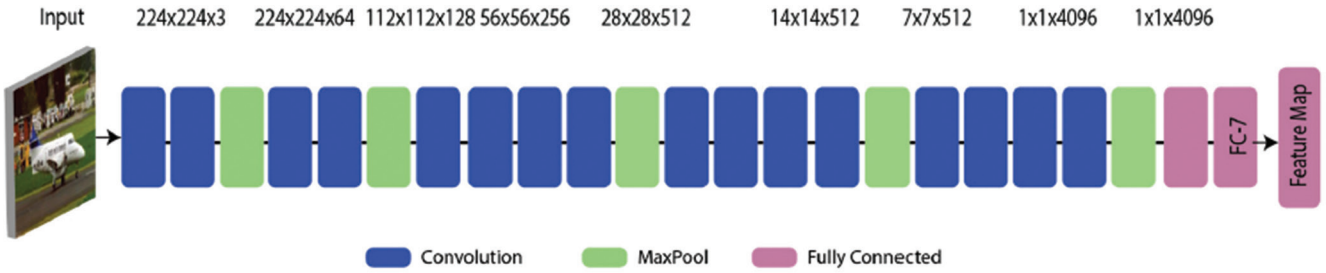

Fig. 1. VGG-19 Architecture.

the content-related information without affecting the style information. Hence, the Gram matrix assists for creating information on the texture related to the data when applied to the extracted features from CNN (Tuyen, et al., 2021).

3. Gradient decent

Gradient descent is an optimization mechanism that aims to minimize the loss function by computing the gradients required to update the parameters of the network. For deep learning, the most familiar algorithm to update the parameters that are used with the gradient is backpropagation, so, they work together as an effective learning algorithm for adjusting the errors by backward propagation through the network layers from the last one going back to the first layer (Wani, et al., 2020).

B. Deep Dream General Concepts

The concepts that relate to deep dream are Inception v3 and gradient ascent.

Inception v3

The Inception model works under the architecture of deep CNN, which aims to decrease the influence of computational effectiveness and low parameters in cases of application. The Inception-v3 model adopts a convolutional filter of various sizes; thus, different areas of receptive fields have been enabled (Cao, et al., 2021). Table I summarizes the Inception-v3 network architecture, where each module takes the output size of the previous one as its input (Cao, et al., 2021; Szegedy, et al., 2016).

Gradient ascent

The process of creating Deep Dream is based on using gradient ascent, which is used at each step of this process no matter the size of the image (it is used from the smallest to the largest image). The goal of gradient ascent is maximizing the loss function (El-Rahiem, et al., 2022).

V. Methodology

The model consists of two main techniques that work together to produce the desired system. These techniques are NST and deep dream. Hence, the model works as illustrated in the following steps:

1. Input the main image (content image) to the model.
2. Apply the NST model as shown below:
   - Choose one image among the style images to extract the desired style.
   - Extract content and style from the content and style image, respectively.
   - Compute the loss for each one of the images.
   - Compute the total loss.
   - Output the resulting image.
3. Input the resulting image from the previous model to the deep dream model, then apply the following steps:
   a. Use the Inception v3 pre-trained model as a deep CNN model to extract the image features (The extraction of the image features starts from low-level features and ends with high-level features).
   b. Compute the error loss.
   c. Compute the gradient using gradient ascent.
   d. Repeat until the required steps.
   e. Output the resulting image.
4. Present the final image.

Table I
Network Structure of the Inception-v3 Model

Type	Patch size/Stride	Input size
Conv	3×3/2	299×299×3
Conv	3×3/1	149×149×32
Conv	3×3/1	147×147×32
Pool	3×3/2	147×147×64
Conv	3×3/1	7373×73×64
Conv	3×3/2	71×71×80
Conv	3×3/1	35×35×192
3×Inception	-------	35×35×288
3×Inception	-------	17×17×768
2×Inception	-------	8×8 × 1280
Pool	8×8	8×8 × 2048
Linear	logits	1×1 × 2048
SoftMax	classifier	1×1 × 1000

In the below two subsections, we illustrate each one of the model components.

A. NST

First, the images must be resized to fit the input of VGG-19, which is 224 × 224. The NST aims to split the content and style of the image, and then, the content of one image is merged with the style of the other image creating a new image. The objects are similar to the content image, whereas colors and structures are similar to style images. Fig. 2 shows our NST proposed system.

VGG-19 pre-trained model is used to extract high- and low-level features from content and style images,

respectively, where it is trained on the ImageNet dataset that contains millions of images.

Gram matrix has been applied to retain the important features of content images like objects and remove the others with keeping the style of the style image. Equation 1 defines the Gram matrix (Tuyen, et al., 2021).

$$Gram = V^T.V \quad (1)$$

Where, V is an arbitrary vector and multiplied by its transpose V^T.

For more explanation, when $V = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, then $V^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$

The stochastic gradient descent (SGD) is used to compute the gradient descent, in which we compute the gradient for one sample at a time then the values of the parameters are updated according to it. Equation 2 shows how SGD computes the parameters (Wani, et al., 2020).

$$w = w - \mu.\nabla E(w; x(i); y(i)) \quad (2)$$

The $\nabla E(w; x(i); y(i))$ represents a gradient of the error loss, whereas the $\{x(i); y(i)\}$ is the training sample. μ is the learning rate.

After that, we compute the loss for both content and style images. Gatys, et al., 2015, proposed a way to compute the loss by computing the square of error loss between two features in the content image calculated from Equation 3.

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2}\sum_{ij}(F_{ij}^l - P_{ij}^l)^2 \quad (3)$$

Here, $\vec{p}, \vec{x}$ are the original image, F_{ij}^l is the activation of the i^{th} mask at the position j in the layer l. Whereas F^l, P^l are the representative feature of both $\vec{x}$ *and* $\vec{p}$ respectively.

The correlation of features is computed by the Gram matrix G^l as in Equation 4.

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l \quad (4)$$

Here, G_{ij}^l represents the inner product between vectorized feature map i and j in the layer l.

The texture of the style image that matches a given image style is generated using gradient descent to find another image matching the representation of the original image. To do this, it must reduce the difference between entries of the Gram matrix of the original and the generated image Gram matrix must be minimized by minimizing the mean-squared distance between them. The style loss calculated from Equation 5.

$$\mathcal{L}_{style}(\vec{\alpha}, \vec{x}) = \sum_{l=0}^{L} w_l E_l \quad (5)$$

Here, w_l represent the weighting factors that each layer contributes to the total loss and $\vec{\alpha}, \vec{x}$ are the original and generated image respectively.

Finally, we can get the output image by mixing the content from the content image and the style from the style image by minimizing the loss of content representation of one layer of the content image and the style representation of several layers of the style image. Equation 6 represents the total loss of the resulting image.

$$\mathcal{L}_{total}(\vec{p}, \vec{\alpha}, \vec{x}) = \alpha\,\mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta\,\mathcal{L}_{style}(\vec{\alpha}, \vec{x}) \quad (6)$$

α and β represent the weighting factors for the content and style representation, respectively.

B. Deep Dream

The deep dream model was built using deep CNN, where the Inception v3 pre-trained model had used to build this model. Fig. 3 shows the steps of the deep dream proposed system.

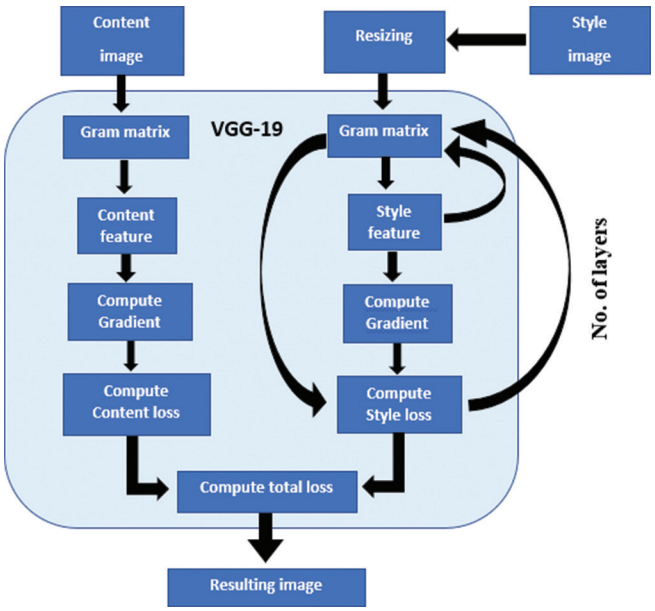

Fig. 2. Neural style transfer proposed system.

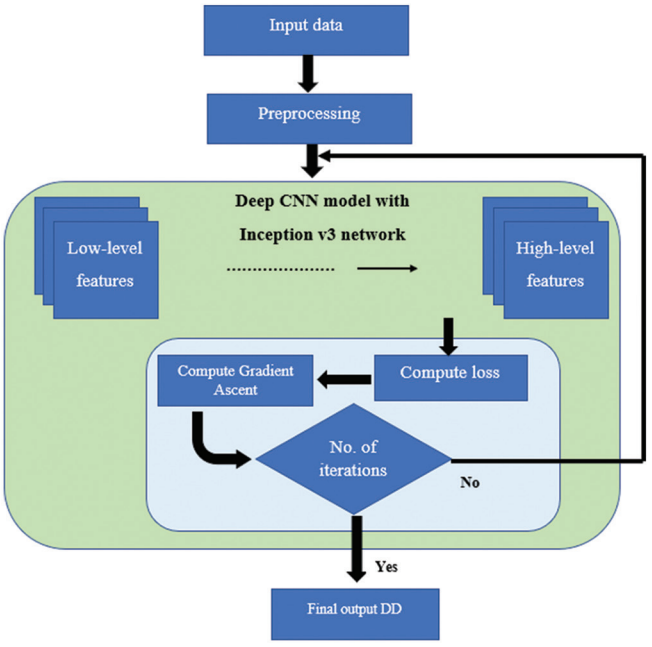

Fig. 3. Main steps of deep dream proposed system.

http://dx.doi.org/10.14500/aro.11051

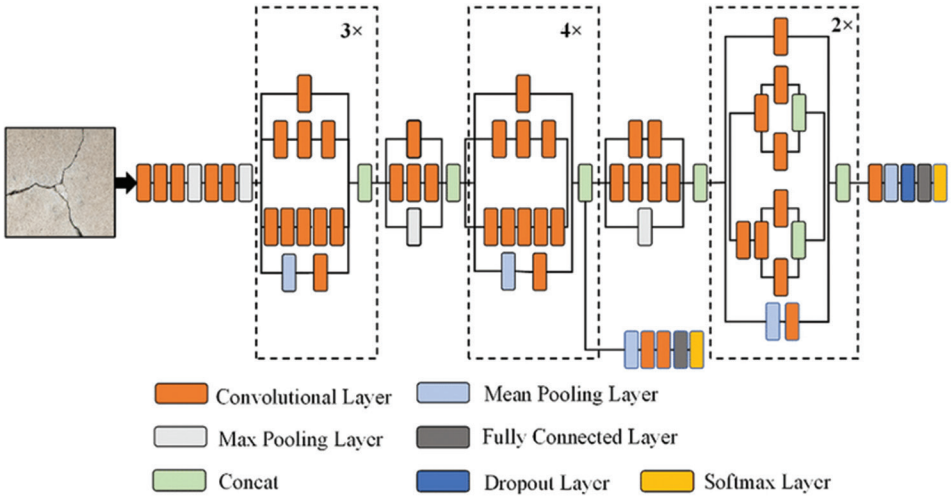

Fig. 4. Inception V3 layer architecture.

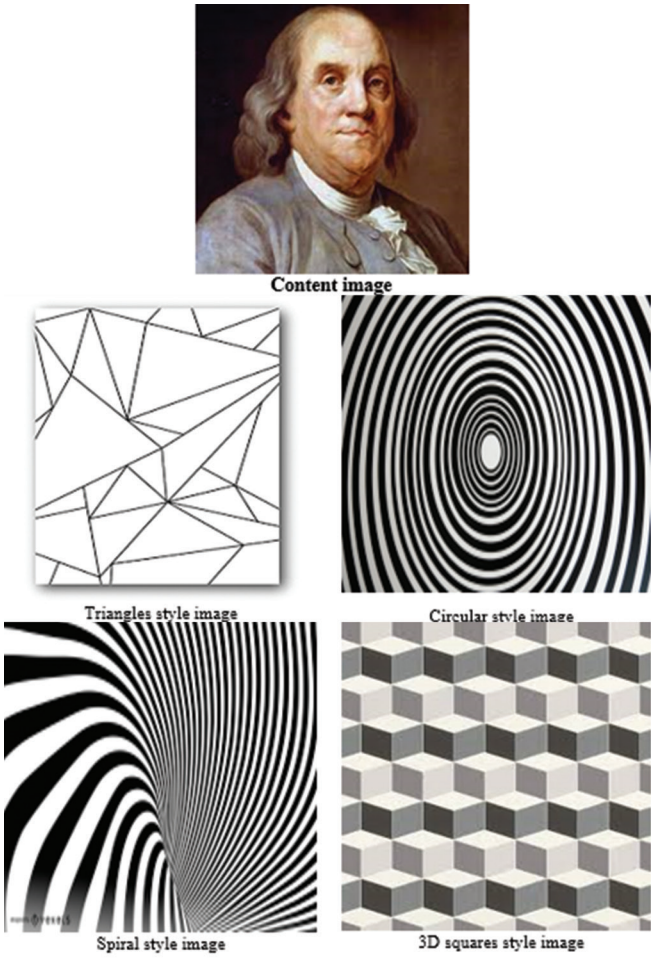

Fig. 5. Content and style images.

The input of the deep dream system is the image that resulted from the NST system. This image must be pre-processed by resizing it to be the size of that appropriate to the Inception v3 model. The size of the image was edited to 299 × 299. Then, the resized image had entered into the pre-trained CNN model, which is Inception v3.

The Inception v3 network consists of 42 layers of depth. It is a pre-trained network with an ImageNet dataset and works to reduce the dimensions of the network by reducing the number of parameters in each layer (Szegedy, et al., 2016).

Using Inception v3, we extract the low-level features at first; then, high-level features are extracted. The Inception v3 network is implemented as shown in Fig. 4 (Ali, et al., 2021), where the first block has been repeated 3 times, the second repeated 4 times, and the third block repeated 2 times.

Then, the loss is computed by adjusting the weights. To do that, at the chosen layers, we summed the activations. All layers have the same contribution, no matter whether it is large or small layers, because, at each layer, we normalized the loss. Our goal is to maximize the value of the loss; this had done using gradient ascent.

After calculating the loss for all layers that had been chosen, we should calculate the gradient w.r.t. the image, the result added to the original image. Each time we add the gradient to the original image, we contribute to creating an enhanced image by increasingly activating particular layers in the network. Adjusting weight to maximize the loss is the core process in a deep dream. The equation of computing the gradient ascent is the same equation of the gradient descent with only one difference represented by flipping the sign of the equation of the gradient descent. Hence, we adjusted the new weights based on the old weights and the gradient of the error function. The overall process was repeated many times; we got different results according to the number of repeated iterations as described in the next section.

VI. Results and Discussion

Our model used pre-trained networks; VGG-19 for NST and Inception v3 for a deep dream, both of them are trained on the ImageNet dataset.

We start with NST implementation. Here, there are two types of images, which are content images and style images, as shown in Fig. 5.

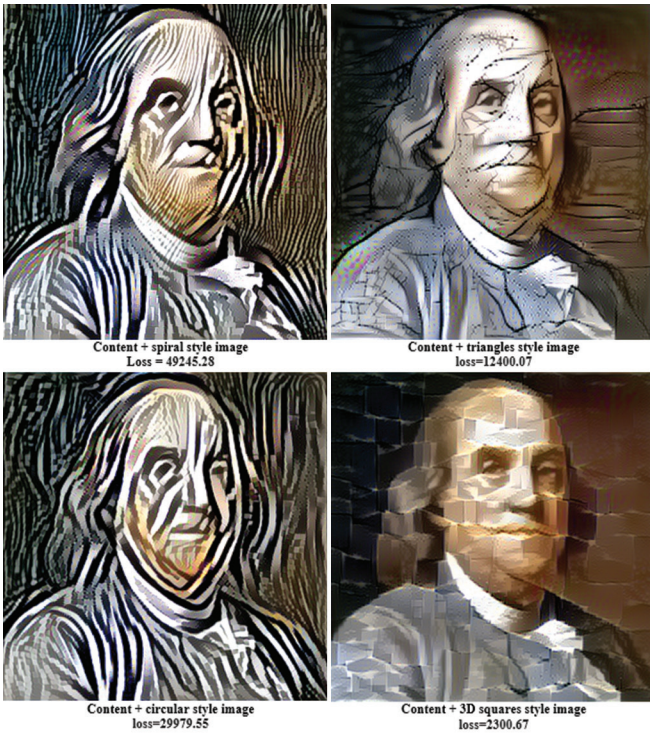

Fig. 6. The resulting images and their losses after 1000 iterations.

The resulting images after 1000 iterations with the loss of every image are shown in Fig. 6.

The loss is decreased as we increase the number of iterations because of using gradient ascent, which minimizes the loss function, thus, enhancing the accuracy of the resulting images. Furthermore, as going deeper through increasing the iterations, it is noticeable that the style of the style originated more and more in the content image, and the content of the content image still preserves the high-level features whereas some of the low-level features such as edges and lines are lost. Fig. 7 shows the resulting images after 10,000 iterations and the loss of every image.

All these output images had entered into the deep dream model after resizing it to 299×299 pixels to make the dreamed image.

In the deep dream algorithm, we used a 0.005 learning rate with 1000 steps to get the final results as shown in Fig. 8.

We measure the quality of the deep dream based on the loss value. The loss value must be balanced; a very high value of the loss may make the dreamed image unrecognizable, whereas a very low loss value does not achieve the purpose of the deep dream. The loss value of the NST works in opposite to the deep dream one, where we work to minimize the loss value. The value of the loss after the 10,000 iterations was 10,431.64 for the triangle style image, 23,225.48 for the circular style image, 40,709.7 for the spiral style image, and finally, 1788.9 for the 3D squares style image. It is clear that the lowest loss value, the better the image quality in NST.

Figs. 9 and 10 represent the loss values of NST and deep dream, respectively.

The deep dream algorithm maximizes the value of loss by increasing the number of steps. The loss value after 900

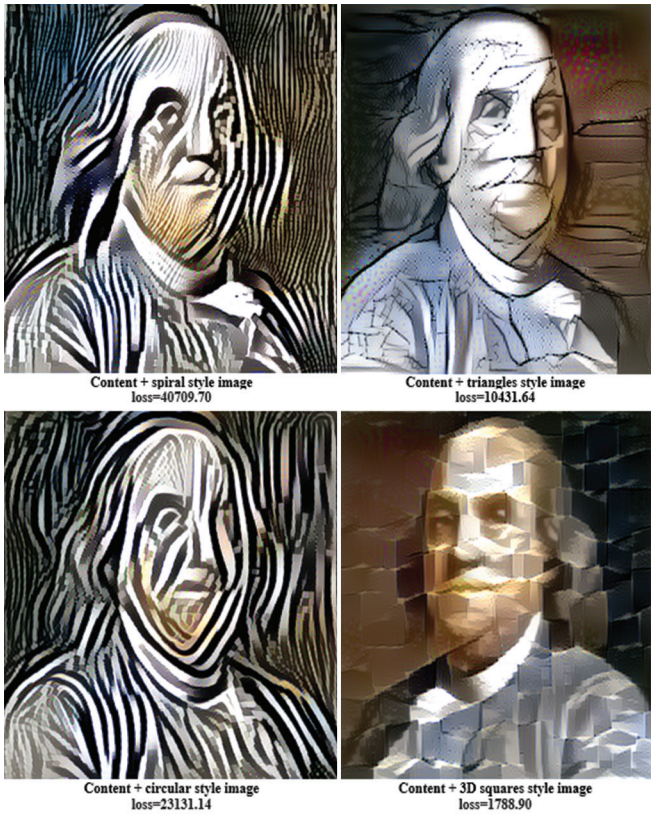

Fig. 7. The resulting images and their losses after 10,000 iterations.

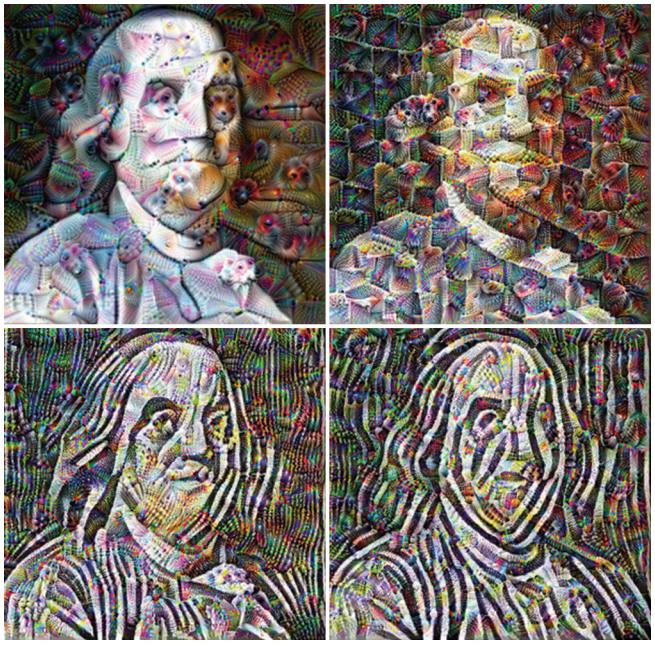

Fig. 8. Stylised images after applying the deep dream algorithm.

steps for the images that are stylized under the NST and then processed in the deep dream algorithm are 3.816556454 for the stylized triangle image, 3.1598 for the 3D squares stylized image, 3.0736 for the circular stylized image, and finally, 3.0087 for the stylized spiral image. The loss in Fig. 10 is for images that passed in the NST model and then entered into the deep dream model.

http://dx.doi.org/10.14500/aro.11051

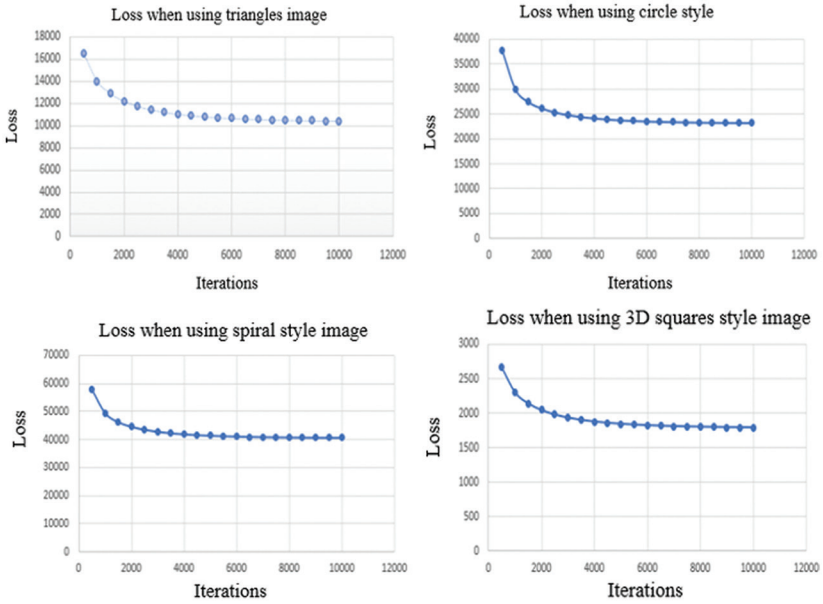

Fig. 9. The loss of the images in the style transfer model.

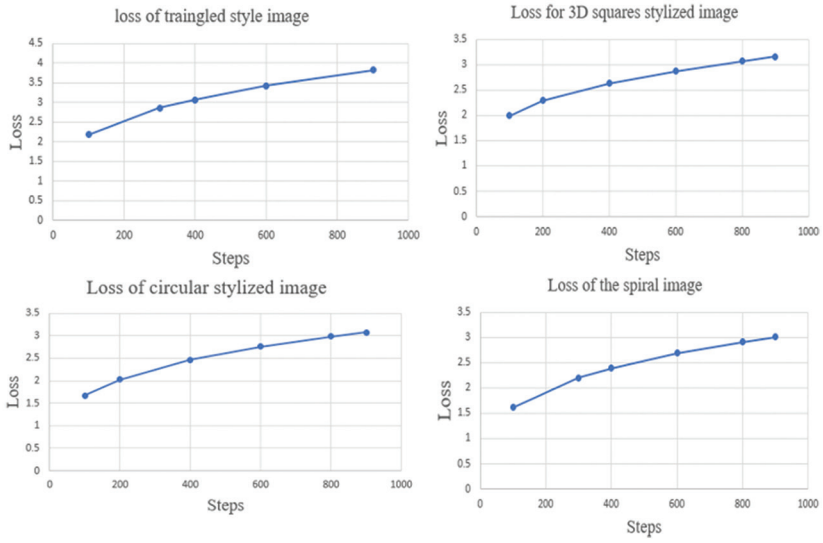

Fig. 10. The loss of the stylized images in the deep dream model.

Fig. 11. Applying deep dream on the original image.

Fig. 11 shows the output of the original image that passed in the deep dream model, with a loss value of 3.6324.

VII. Conclusion

Deep dream and NST represent the most modern deep learning techniques in recent years. This study presents a deep dream model that takes the output of NST images as input and then applies the deep dream algorithm. VGG-19 pre-trained network is used as a deep CNN network to implement the NST based on the Gram matrix that represents the core of this model and the gradient descent, which minimizes the loss function when the image is cleared and increases with the distorted images.

The results of the NST are the input to the deep dream; the loss varies among the different images in the deep dream.

The Inception v3 network is a pre-trained model used to build the deep dream algorithm. The loss here is maximized based on gradient ascent. The loss of the distorted images (i.e., circular and spiral stylized images) is less than the clear images (i.e., 3D squares and triangles images).

References

Abedi, W.M., Nadher, I., Sadiq, A.T. and Al., E., 2020. Modified deep learning method for body postures recognition. *International Journal of Advanced Science and Technology*, 29, pp.3830-3841.

Ali, L., Alnajjar, F., Jassmi, H.A, Gochoo, M., Khan, W. and Serhani, M.A., 2021. Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures. *Sensors,* 21, p.1688.

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L., 2021. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8, pp.1-74.

Cao, J., Yan, M., Jia, Y., Tian, X. and Zhang, Z., 2021. Application of a modified Inception-v3 model in the dynasty-based classification of ancient murals. *EURASIP Journal on Advances in Signal Processing*, 2021, p.49.

Chen, X., Zhang, Y., Wang, Y., Shu, H., Xu, C. and Xu, C., 2020. Optical flow distillation: Towards efficient and stable video style transfer. In: *Lecture Notes in Computer Science (LNCS)*. Springer Science, Germany.

Choi, H.C., 2022. Toward exploiting second-order feature statistics for arbitrary image style transfer. *Sensors(Basel)*, 2022, p.2611.

El-Rahiem, B.A., Amin, M., Sedik, A., Samie, F.E. and Iliyasu, A.M., 2022. An efficient multi-biometric cancellable biometric scheme based on deep fusion and deep dream. *Journal of Ambient Intelligence and Humanized Computing*, 13, pp.2177-2189.

Gatys, L.A., Ecker, A.S. and Bethge, M., 2015. A Neural Algorithm of Artistic Style. arXiv Prepr. arXiv1508.06576.

Khan, A., Sohail, A., Zahoora, U. and Qureshi, A.S., 2020. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53, pp.5455-5516.

Kiran, T.T., 2021. Deep inceptionism learning performance analysis using TensorFlow with GPU-deep dream algorithm. *Journal of Emerging Technologies and Innovative Research*, 8, pp.322-328.

Kotovenko, D., Sanakoyeu, A., Ma, P., Lang, S. and Ommer, B., 2019. A content transformation block for image style transfer. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, United States, pp. 10024-10033.

Li, H., 2018. *A Literature Review of Neural Style Transfer*. Princeton University Technical Report, Princeton NJ, p.085442019.

Mordvintsev, A., Olah, C. and Tyka, M., 2015. Inceptionism: Going Deeper into Neural Networks. Available from: https://googleresearch.blogspot.co.uk/2015/06/inceptionism-going-deeper-into-neural.html [Last accessed on 2022 Aug 03].

Rashid, M., Khan, M.A., Alhaisoni, M., Wang, S.H., Naqvi, S.R., Rehman, A. and Saba, T., 2020. A sustainable deep learning framework for object recognition using multi-layers deep features fusion and selection. *Sustain*, 12, p.1-21.

Singh, A., Jaiswal, V., Joshi, G., Sanjeeve, A., Gite, S. and Kotecha, K., 2021. Neural style transfer: A critical review. *IEEE Access*, 9, pp.131583-131613.

Sudha, V. and Ganeshbabu, T.R., 2021. A convolutional neural network classifier VGG-19 architecture for lesion detection and grading in diabetic retinopathy based on deep learning. *Computers, Materials and Continua*, 66, pp.827-842.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, United States, pp.2818-2826.

Tuyen, N.Q., Nguyen, S.T., Choi, T.J. and Dinh, V.Q., 2021. Deep correlation multimodal neural style transfer. *IEEE Access*, 9, p.141329-141338.

Wani, M.A., Bhat, F.A., Afzal, S. and Khan, A.I., 2020. Advances in Deep Learning. Springer Nature, Singapore.

Xiao, J., Wang, J., Cao, S. and Li, B., 2020. Application of a novel and improved VGG-19 network in the detection of workers wearing masks. *Journal of Physics: Conference Series*, 1518, 012041.

Yin, H., Molchanov, P., Alvarez, J.M., Li, Z., Mallya, A., Hoiem, D., Jha, N.K. and Kautz, J., 2020. Dreaming to distill: Data-free knowledge transfer via deepinversion. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR*). IEEE, United States, pp.8712-8721.