IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 August 2023, accepted 21 August 2023, date of publication 28 August 2023, date of current version 21 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3309419

RESEARCH ARTICLE

A Hybrid Artistic Model Using Deepy-Dream Model and Multiple Convolutional Neural Networks Architectures

LAFTA R. AL-KHAZRAJI[1,2], AYAD R. ABBAS[1], ABEER S. JAMIL[3], AND ABIR JAAFAR HUSSAIN[4,5], (Senior Member, IEEE)

[1]Department of Computer Science, University of Technology, Baghdad 19006, Iraq
[2]General Directorate of Education of Salahuddin Governorate, Salahuddin 34001, Iraq
[3]Department of Computer Technology Engineering, Al-Mansour University College, Baghdad 69005, Iraq
[4]Department of Electrical Engineering, University of Sharjah, Sharjah, United Arab Emirates
[5]School of Computer Science and Mathematics, Liverpool John Moores University, L3 3AF Liverpool, U.K.

Corresponding author: Abir Jaafar Hussain (a.hussain@ljmu.ac.uk; abir.hussain@sharjah.ac.ae)

ABSTRACT The significant increase in drug abuse cases prompts developers to investigate techniques that mimic the hallucinations imagined by addicts and abusers, in addition to the increasing demand for the use of decorative images resulting from the use of computer technologies. This research uses Deep Dream and Neural Style Transfer technologies to solve this problem. Despite the significance researches on Deep Dream technology, there are several limitations in existing studies, including image quality and evaluation metrics. We have successfully addressed these issues by improving image quality and diversifying the types of generated images. This enhancement allows for more effective use of Deep Dream in simulating hallucinated images. Moreover, the high-quality generated images can be saved for dataset enlargement, like the augmentation process. Our proposed deepy-dream model combines features from five convolutional neural network architectures: VGG16, VGG19, Inception v3, Inception-ResNet-v2, and Xception. Additionally, we generate Deep Dream images by implementing each architecture as a separate Deep Dream model. We have employed autoencoder Deep Dream model as another method. To evaluate the performance of our models, we utilize normalized cross-correlation and structural similarity indexes as metrics. The values obtained for those two quality measures for our proposed deepy-dream model are 0.1863 and 0.0856, respectively, indicating effective performance. When considering the content image, the metrics yield values of 0.8119 and 0.3097, respectively. As for the style image, the corresponding quality measure values are 0.0007 and 0.0073, respectively.

INDEX TERMS Deep dream image, neural style transfer, gram matrix.

I. INTRODUCTION

The transformation of artistic images has increased significantly with the development of deep learning techniques. Deep Dream (DD) and Neural Style Transfer (NST) are two of these techniques that have drawn a lot of interest due to their capacities to produce captivating, dream-like images and transfer the artistic style from one image to another [1]. Deep Dream generates new images that mimic the images that schizophrenic patients and drug addicts imagine. These techniques try to reflect hallucinations and delusional perceptions resulting from addiction to narcotics [2], where applying Deep Dream is currently in its early phases, showing how artificial intelligence has the potential to advance the knowledge of complicated mental health issues and enrich the lives of individuals who are affected.

This study provides an investigation of Deep Dream implementation using five different Convolutional Neural Networks (CNNs) architectures; VGG16 [3], Inception v3 [4], VGG19 [5], Inception-ResNet-v2 [6], and Xception [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

In addition, Neural Style Transfer (NST) is applied using VGG16 architecture. In this paper, we proposed a novel hybrid deepy-dream model based on NST and CNN, as NST with Deep Dream provides unique and artistic images by exploiting the hierarchical nature of CNN.

Deep Dream is developed by Google, exploiting the CNN power to visualize the neural network features by maximizing the activation of particular neurons in the intermediate layers [8].

Neural Style Transfer method was developed by Gatys et al. [9] which is an attractive application of deep learning, that has been used in the entertainment industry to generate art like images using CNNs to capture image content while applying the style of another image [4], [5]. In this case, the output of NST is a generated image consisting of a combination of the content and the style of both content and style reference image, respectively [11]. Deep Dream and NST have demonstrated their artistic power while providing valuable insights into how CNNs perform their functions.

Normalization is a generic transformation that ensures the changed data has certain statistical features [12]. Blending images refer to composing images to create visual content. One of the main advantages of image blending is that a blended image depends on both the source and destination [13].

Although Deep Dream is a recent technology, the studies in this field are still limited and need more support. Much of the research needs to manage the quality of the generated images and employ metrics to evaluate the effectiveness of Deep Dream model. In this case, we have improved the quality of the results and maintain the performance by using evaluation metrics.

The motivation behind this work is to propose a Deep Dream model for generating Deep Dream images that mimic the hallucinations imagined by schizophrenia patients and abusers. These generated images look similar to dream-like and/or surreal images.

The proposed deepy-dream model is tested against the blurred images; for both an image with a single face (Franklin's image) and the blended image. The generated images are evaluated using three quality measures including structure similarity index (SSIM) [14], normalized cross-correlation (NCC) [15], and peak signal-to-noise ratio (PSNR) [14].

Our proposed deepy-dream model provides augmentation by enriching the dataset with new dreamed images that have small differences from the original images. In the first stages (where each image has a small variation from the previous one), this assists to make the classification systems that are trained on these datasets more powerful and able to deal with distorted images.

In this paper, we provide the following novelties:

1) Generating Deep Dream images by combining the features of five CNN architectures. As far as the authors are concerned, this is the first time that such a combination is performed.
2) For the first time, a hybrid model of both Deep Dream and Neural Style Transfer and its application for generating images to support schizophrenia patients is implemented.
3) The generation of Deep Dream images using blurred images.

II. RELATED WORK

A. NEURAL STYLE TRANSFER (NST)

Chen, et al. [16] proposed a unique way for increasing the stability and efficiency of video-style transfer networks by employing knowledge distillation and low-rank distillation loss. Accordingly, smaller student network that has an optical flow module for increased stability is trained to replicate the output of a bigger instructor network. The student network is encouraged to learn the residual between output stylized movies generated by the teacher network. A low-rank distillation loss is used to stabilize the student network output by imitating the rank of the input videos. The suggested method is assessed and contrasted with cutting-edge video-style transfer techniques. The authors claimed that their outcomes perform better than the state-of-the-art methods in terms of temporal inaccuracy and running speed.

Choi et al. [17] proposed a model in which they claimed that it created the best arbitrary image style transfer method using second-order statistics of the encoded characteristics; representing correlation among various feature maps to control the style of the resulting image. They made two contributions to their study; the first was the suggestion of a method for correlation-aware loss and feature alignment. The capacity of the decoder network style is increased using a regular blending of loss and feature alignment algorithms, which reliably match the statistics of the second order of the content features and the target style features. They suggested a component-wise style regulating method in the second contribution. Using style-specific components from second-order feature statistics, their method could produce several styles based on single or multi-style images. They claimed that their methodology increases decoder network style diversity and capacity without compromising its ability to process in real-time on GPU devices (under 200 ms). For the encoder, a pre-trained VGG16 network was employed to extract the features of both content and style images, while a trainable VGG16 network was used for the decoder to show the effect of the independent style on the resulting image. A dataset of 22 photos was utilized to represent style images, and the MS-COCO dataset [18] was used for content images. To quantify the losses of the networks, they relied on augmenting the training data.

Al-Khazraji et.al. [19] proposed a model that combines both Deep Dream and NST to improve the art and produce pictures that mirror the hallucinations experienced by psychiatric patients and drug addicts. The pre-trained VGG-19 and Inception v3 networks were used. The Gram matrix had a significant role in the process of transferring styles. Using gradient descent and gradient ascent, the loss was minimized

during style transfer and maximized during a deep dream. The authors indicated that an image lost value was decided by how clear it was. Deep Dream loss values were decreased and neural style transfer loss values were higher for distorted visuals. The opposing patterns appeared in contrast when there were no muddled lines, circles, colors, or other shapes. So, this state of art differs from our current study in that the two works (previous and current) implement Deep Dream and NST but in reverse order, where the previous work implements NST then the stylized images become the input of Deep Dream model. While in the current work, the Deep Dream process is implemented first then the resulting Deep Dream image is the input to the NST model. Also, the current study maintains the quality of generated images since it was implemented by mixing the features of many CNN architectures, while the previous work implement Deep Dream by using Inception v3. Although both the two works used VGG-19 for NST but with different layers. In addition, the current study uses evaluation metrics while the previous work did not.

B. DEEP DREAM

Yin et al. [20] proposed an approach for creating images using deep neural networks. Their approach, known as the DeepInversion, consists of two components: teacher logits and student logits; the first component represents the unnormalized results of a trained neural network for a particular input image, while the second component represents the results of training a different neural network to emulate the teacher. Without using any Deep Dream extra data from the training data set, the teacher depicts the inverted trained network that starts from random noise. They used Deep Dream to construct DeepInversion by enhancing the image quality of Deep Dream and extending the concept of image regularisation to include the concept of feature distribution regularization. Their model was tested using ImageNet and CIFAR-10 datasets [21]. Although their great efforts to build this approach, their study was built only by using two CNN architectures: VGG-16 and ResNet while they were able to implement their work with multiple other variants. This method is time-consuming and very large resources are required. Also, they did not use evaluation metrics to measure the values of similarity or connectivity between the original images and the generated images.

Arthi et al. [22] proposed a biometric authentication system that employs Deep Dream to extract important features. They used gradient ascent to maximize specific layers in the CNN to exclude irrelevant features. The resulting images from the Deep Dream system are matched with the database that contains the biometric images of the authorized person. If matched correctly, the system grants permission, otherwise, no permission will be allowed, and a security control trigger is generated by the system. The authors did not mention the type of architecture they used to build their model, although they used Deep Dream only to extract the features, and building a model that generates dream images was not their purpose. So, they did not concern with the quality of images or anything else, just extracted the features by implementing Deep Dream.

Kiran [23] introduced a new algorithm for computer vision known as Deep Inceptionism learning or the Deep Dream algorithm. As the picture enters the network and causes the neurons to activate, the algorithm training phase starts. The idea behind their suggested technique is to alter the input image to increase the firing of specific neurons (through the boosting or activation of the neurons). With the help of Deep Dream, users can decide which neurons they're willing to fire more noticeably. Such a procedure is repeatedly carried out until an input image has all the attributes needed by a specific layer. As images are fed to the network, recognizing and highlighting unusual elements in the original image are improved. In this case, the first stage of the process is to submit an image to a trained CNN, ANN, ResNet, etc. The next step is to select a layer and describe the activations (or output) that are generated by the layer of interest. The next steps involve calculating the activation gradient for the input image, modifying the image to increase those activations, improving network-detected patterns that produce a trippy hallucinatory visual, and repeating the process across various sizes. Although his good work in implementing Deep Dream algorithm, he did not add or change anything but implemented the original algorithm by using two CNN architectures: Inception v3 and ResNet. So, the quality of the images nor measuring the degree of the quality is his interest.

El-Rahiem et al. [24] used Deep Dream to present a multi-biometric cancellable scheme (MBCS) for the generation of cancellable patterns of the fingerprints that are secure and effective, as well as biometric modalities based on eye iris and finger veins. To create a multi-exposure deep fusion module that produces a fused biometrical template collected as the Deep Dream module final cancel-label template, they harnessed the power of e deep learning models. Inception v3 network that had already been trained is utilized. The loss value is increased during the gradient-ascent phase to visualize the filter. Filter visualisation is used to increase the number of filter activations and maximises the value of a single filter in a certain layer. The gradient ascent method is governed by three primary parameters: gradient step ''S'', maximum loss ''Lmax'', and numerous iterations ''I'', which are used to adjust the loss gradients of the successive layers. Optimization of Deep Dream algorithm performance is based on these factors. The authors introduced a good study, but in the case of Deep Dream, they only used Deep Dream for cancellable patterns of the fingerprints and did not make any comparison of the results with other models. The quality of the resulting images was not their concern.

Al-Khazraji et.al. [25] conducted a study that shows how to use CNN to create images that are representative of deep dreams. The procedure entails looking at the layers in each network block, choosing the necessary layers, and maximizing their features. Entire loss is calculated and the last Deep Dream image is recovered, in which this process is performed

recursively. The authors repeated the procedure twice, once on the low-level layers and then on the high-level layers. High-level layers create profound dream visuals that are more distinct than low-level layers. The loss values of photos from the higher layers range from 20.0704 to 32.1625, and the loss values for images from the lower layers range from 31.1435 to 54.6134. Although the great work introduced by the authors, they did not compare their results with any results of other models. Also, the evaluation was not measured in the case of the quality of images and correlation between pixels of original and generated images.

Sahu et al. [26] proposed a hybrid model that combines the features of both deep learning and machine learning. They implemented the original Deep Dream algorithm using Inception-v3 on the images that have been collected from farms to extract important features of disease lesions existing in the leaves of crops. By maximizing the activations of the selected layers on the Inception v3, the features have become more prominent. Gradient ascent is used to maximize the activations. The authors focused only on extracting the important features but not generating Deep Dream images. Deep Dream was a secondary process employed to achieve another goal.

Table 1 illustrates the limitations of Deep Dream studies. Table 1 shows the limitations of the Deep Dream in literature.

In this study, we implemented Deep Dream in various methods with layers and tested the results in different conditions. The difference from the previous studies are:

- Improving the quality of generated Deep Dream images.
- Comparing the results with the results of the other models.
- Reserving the quality of images by employing evaluation metrics to measure the quality of the results based on similarity and connectivity between the original and generated Deep Dream images which are ignored in the previous studies.
- In addition to the CNN variants, we also implemented the Deep Dream by using the autoencoder Deep Dream model.
- Proposed a hybrid Deep Dream model named proposed deepy-dream model to generate Deep Dream images by mixing the features of all the five CNN architectures that were used in this study.
- Examining the proposed deepy-dream model against blurred and unblurred (pure) images.

TABLE 1. The limitations of the previous deep dream studies.

Study	Model	Limitations
[22]	There is no obvious model	They did not explain how their model is built, on which CNN variant they trained the system, and which layers they maximized. No comparison with any other models, so this is unfair. Also, they did not use any metric to evaluate the resulting Dream image.
[24]	Inception v3	It uses only one workstation for execution, in addition to the small number of biometric modalities employed for evaluation, and the absence of comparison with other cutting-edge cancellable biometric schemes.
[23]	ResNet Inception v3	The work did not add any modification to the original algorithm and it just reimplemented the original DD algorithm.
[20]	VGG ResNet	The produced images may have identical colors and backgrounds due to the usage of a standard Gaussian distribution to initialize every one of the pixels. It took a lot of time and resources to make these images. The style of synthesized images may vary depending on the optimization hyperparameters used. The article just provides a cursory analysis of the proposed method.
[25]	VGG-16	This study only focused on particular layers to maximize their activation functions and ignored the other important parameters like learning rate, and the architecture of the network itself such as depth, width, and layers connectivity.
[26]	Inception v3	This study implements the original Deep Dream without modifications. Also, in the section where Deep Dream is used, there is not any comparison with other models that implement the same task.

III. THEORETICAL BACKGROUND

This section is divided into two parts, in the first part, we will go through various CNN structures utilized in this research. While in the second part, we will move to data augmentation.

Machine learning technology [27] includes deep learning, which is mostly used when dealing with image data [28], [29]. Deep Dream and NST are deep learning techniques, that utilize CNN pre-trained architectures to build various models.

CNNs are deep neural networks created primarily to process image data; they can be benchmarked to feed-forward ANNs with a variety of convolutional and subsampling layers [30], [31], [32], [33], [34]. By using deep CNN, the most representative features are extracted from the images [35], [36]. In a CNN, the layers that extract features get input from neighbouring layers. The features such as ending points, edges, and corners are pooled by the upper layer [37].

A. CNN STRUCTURES USED IN THIS STUDY

1) VGG-16

VGG16 (Visual Geometry Group) is a 16-layer neural network that consists of three fully connected (FC) layers, five blocks of convolutional layers, and one max-pooling layer. In order to guarantee that the final map has the same dimensions as the activation map from the preceding layer, the convolutional layers employ a 3×3 kernel with padding and a stride of 1. The spatial dimensions of the activation map from the preceding layer are cut in half using the max-pooling layer, which has a 2×2 kernel size, no padding,

and a stride of 2 [27]. The spatial dimension is decreased by using a rectified linear unit activation after each convolutional and max-pooling block. For the final classification, three FC layers are utilized.

2) INCEPTION V3

Inception-V3 is a well-known deep neural network with 11 Inception modules, each of which has an activation layer, a convolutional layer, a batch normalization layer, and a pooling layer. In order to maximize feature extraction, these modules, which have been expertly built, are concatenated in the Inception-v3 model. Each module takes advantage of the multi-scale notion and has several branches with various kernel sizes, including 1×1, 3×3, 5×5, and 7×7. In each module, a 1×1 convolution is employed before the computationally demanding 3×3 and 5×5 convolutions to reduce dimensionality and improve performance. Factorization is used by the network's sub-networks (Mixed0 to Mixed10) to lower the total number of deep neural network (DNN) parameters [38].

3) VGG-19

The VGG-19 neural network is a sophisticated system that has 19 connection layers, 16 convolution layers, 1 ReLU layer after each of these layers, and 3 completely connected layers. In the case of a classification process, the convolution layers' primary goal is to extract the features from the input images. These features are then used by the fully connected layers to classify the images. Five max-pooling layers are used to reduce dimensionality and avoid overfitting [39].

4) INCEPTION-RESNET-V2

Inception and ResNet, the two most popular deep convolutional neural networks (CNNs), are combined to generate Inception-ResNet-V2. Batch normalization, however, does not apply to summations; it is only applied to the conventional layers to make the training process faster. The number of Inception blocks and the network's overall depth is increased by using the residual modules [40].

5) XCEPTION

Xception is a modified version of the Inception architecture created by Google researchers [ref.]. For classifying images in a dataset, Xception combines pointwise and depthwise convolution layers, which have been shown to increase accuracy. The feature extraction network of Xception is composed of 36 convolution layers divided into 14 modules. All of these modules, except the first and last, contain linear residual connections. Because Xception is made to be easily adaptable, users can modify its parameters [7].

B. DATA AUGMENTATION

Data augmentation is the process of expanding the quantity of data by employing various tactics (such as scaling, flipping, zooming, rotating, translation, noise injection, etc.). As a result, the size of the training data will be increased as necessary. Data Augmentation plays a significant role in developing a better Deep Learning model [41], [42].

As the Deep Dream model generates images at each iteration that has a small difference from the previous one, it can be employed as an augmentation to increase the images in the dataset.

The augmentation process with Deep Dream is similar to the noise injection process except that noise injection has some limitations such as high computational cost, time-consuming, and additional memory required [42].

IV. METHODOLOGY

Our proposed system is built based on three techniques; image blending, deep dream, and neural style transfer. Figure 1 shows the diagram of the proposed model.

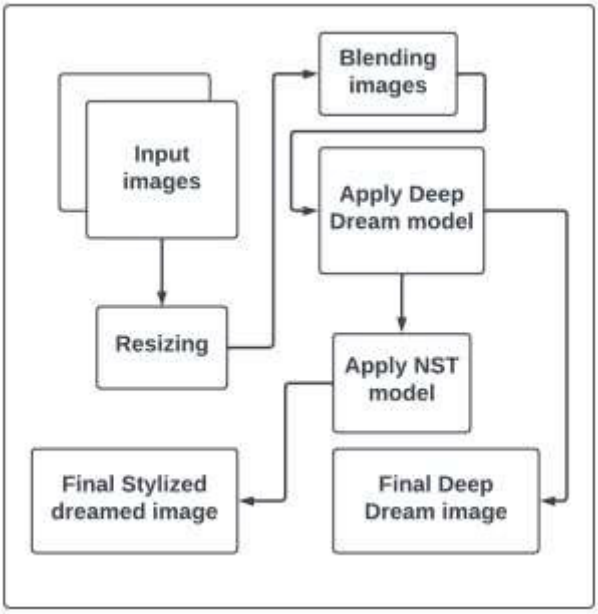

FIGURE 1. Proposed model diagram.

At first, the images are resized to (224×224) or (299×299).

Then, the images are used by the model, and they are blended. Equation 1 shows how the blending operation is performed [43].

$$I_{blended} = I_{source} * M - I_{distination} * (M - 1) \quad (1)$$

where $I_{blended}$ is the output image, I_{source} is the first image, $I_{distination}$ is the second image and M is the binary mask for the objects' location, while $(M-1)$ is the inverse of the mask M.

Deep Dream model is implemented into two separate models, the first is named deepy-dream model which generates dream-like images by combining the extracted features of all five CNN architecture models (VGG-16, VGG-19, Inception v3, Inception-ResNet-v2, and Xception). While the second is to implement each one of those five models individually.

Figure 2 shows our proposed deepy-dream model while Figure 3 illustrates the diagram of every individual component of our model.

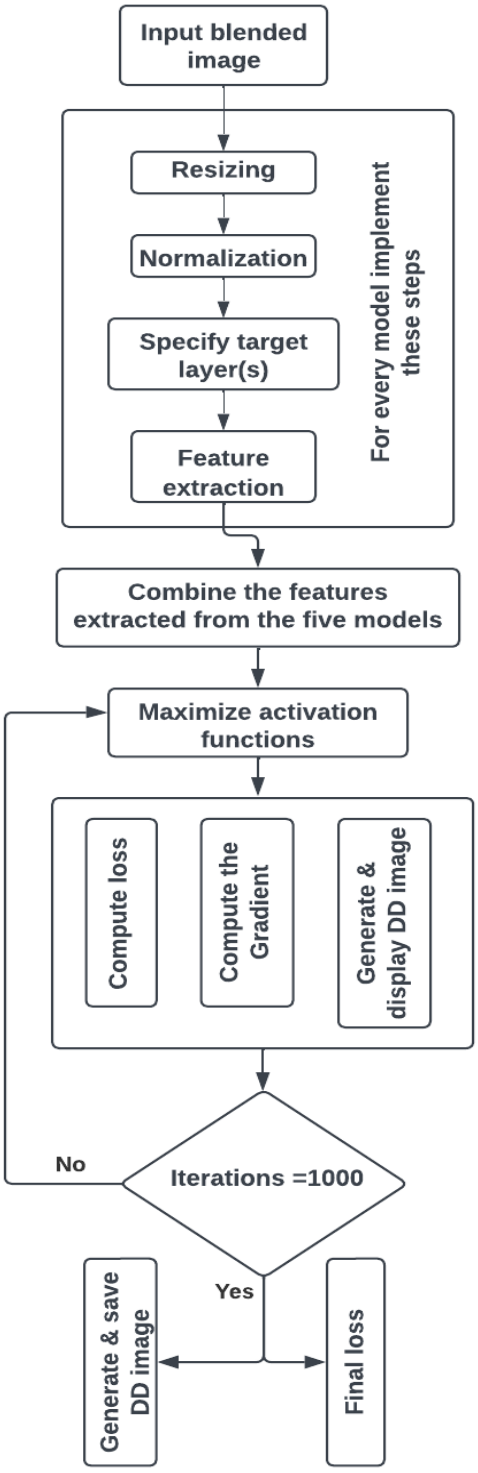

FIGURE 2. Diagram of our proposed deepy-dream model.

The images are resized to minimize their size into (299 × 299) in the case of Xception, Inception v3, Inception-ResNet-v2 models, and (224, 224) when using VGG16 and/or VGG19 models because these models have been built to accept those sizes of images.

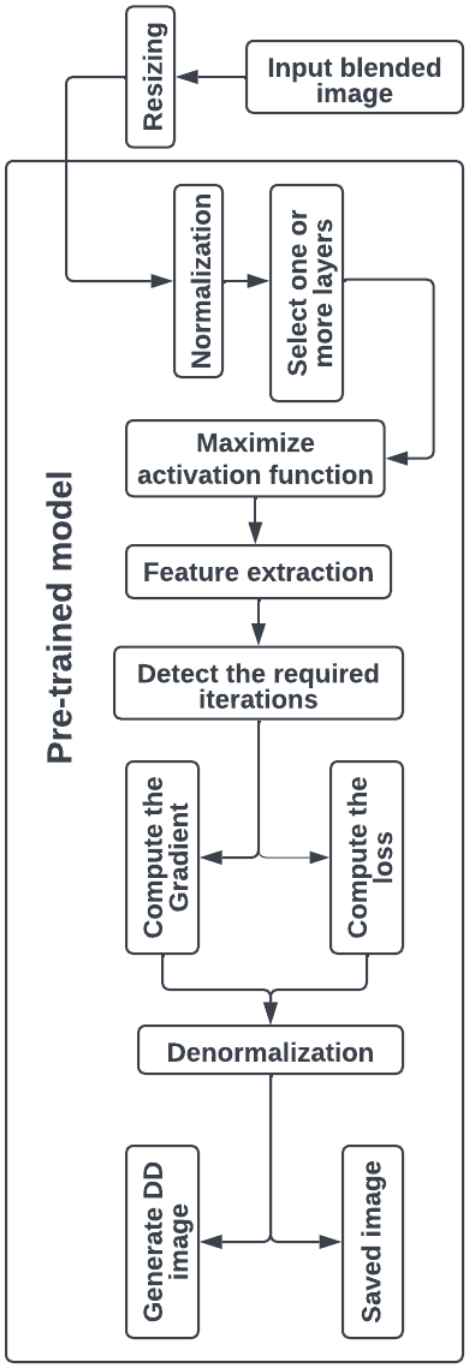

FIGURE 3. Diagram of the individual five deep dream models.

Normalization is performed to the image (0-255) and the three colors are limited in the range between (0-1) [44], [45].

During pre-trained, we monitor lower and upper layers for low-level and high-level features Table 2 illustrates the targeted layers in each one of the CNN architectures.

We extract the most crucial features from the pre-trained models after maximizing the loss of the specified layers.

TABLE 2. The targeted layers in each architecture of the CNN networks.

C NN variant	The activated layers
VGG-16	block5_conv1, block5_conv2, block5_conv3
Inception v3	mixed4, mixed5, mixed6, mixed7, mixed8
VGG-19	(block5_conv1, block5_conv2, block5_conv3, block5_conv4
Inception-ResNet-V2	block35_5_mixed, block35_6_mixed, block35_7_mixed, block35_8_mixed, block35_9_mixed
Xception	block7_sepconv3_bn, block9_sepconv2_act, block8_sepconv3

In every CNN architecture, the features are extracted by combining convolution, activation functions, and pooling procedures. First, the convolution process is implemented. The activation function and max pooling layers extract the network features. Feature extraction starts by applying the convolution process as shown in Equation (2) [46].

$$\boldsymbol{F(i,j) = (A * K)(i,j = \Sigma\Sigma A(i - m, j - n)K(m,n)nm} \tag{2}$$

where A is the input image, K indicates the 2D filter of size $m \times n$. F stands for the 2D feature map resulting from convolving A with K, expressed by $A * K$, *i and j* represent the positions of the row and column in the output feature map F or the input image A.

Rectified Linear Unit (ReLU), which is a hidden layer activation in a deep neural network, is among the most common activation functions. It eliminates the negative values and gives the algorithm non-linearity through computing the activation by thresholding input at zero [47]. In the max pooling operation, only the strongest activations are selected from the pooling region by applying Equation (3) [48].

$$fmax = max\{x_i\}_{i=1}^{N} \tag{3}$$

where x is the input, i represents the pixel's number inside the block of pixels with a range from 1 to N, where N is the number of overall items. F is the output result of the max pooling operation.

In Deep Dream, the features are extracted by running the input image through the CNN up to the desired layer(s), after which the image is optimized to magnify the patterns associated with the features.

A total of 1000 iterations were employed in this study.

Our proposed deepy-dream model is implemented by applying each one of these five Deep Dream models individually, the overall pre-trained models are combined into one hybrid Deep Dream model to generate different Deep Dream images. At each iteration, the loss is computed independently for every model, and these losses are summed and utilized for updating the image. Hence, the features of all these models are combined, and this gives each model a fair contribution to the gradients to be used for updating the image.

Also, we have implemented an autoencoder network to generate Deep Dream images, where the features have been extracted from specific layers of the VGG-19 network, and then built an autoencoder, which is made of encoder layers that perform various convolutional operations to reduce the input image representation. However, decoder layers use transpose convolutions to attempt to recover the original image from the compressed version. The 'same' padding technique helps maintain spatial dimensions, while the 'relu' activation function is employed to introduce non-linearity.

The NST model is used to produce a newly decorated image. The input to this model must include at least two images, one of which must reflect the style image produced by the Deep Dream process and the other must represent the content image. The NST model is shown in Figure 4.

In the pre-trained model, the layers (the same layers that were detected in Table 2) of both content and style images to extract their features by applying the CNN operations including convolution, activation function, and pooling operations are selected.

For both images (content and style), the Gram matrix is computed for each layer to extract the high-level features of these images. The purpose of the Gram matrix is to compute the correlation between features in each layer.

The style information is kept while the content-specific information is removed using the Gram matrix. As a result, the recovered features from the CNN can be used by the Gram matrix to help create information about the texture.

The gram matrix is computed using Equation (4) [9].

$$G_{ij}^{l} = \sum\nolimits_{k} F_{ik}^{l} F_{jk}^{l} \tag{4}$$

where, G_{ij}^{l} is the inner product between vertices i, and j in layer l, and at each layer l, F denotes a collection of features related to the vertices of a graph.

The new image is generated from the content image and the style image. Then the content and style loss are computed to ensure that the new image has the content of the content image and the structure of the style image

The loss of the content image is computed by applying Equation 5.

$$\mathrm{L}_{content}\,(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum\nolimits_{ij} (F_{ij}^{l} - P_{ij}^{l})^2 \tag{5}$$

Here $\mathrm{L}_{content}$ refers to the loss in the content image, $\vec{p}, \vec{x}$ are the original image, F_{ij}^{l} is the activation of the i^{th} mask at. the position j in the layer l. While F^{l}, P^{l} are the representative feature of both $\vec{x}$ *and* $\vec{p}$, respectively. The style loss of the style image is computed by applying Equation 6.

$$\mathrm{L}_{style}\,(\vec{\alpha}, \vec{x}) = \sum\nolimits_{l=0}^{L} w_l E_l \tag{6}$$

where L_{style} is the style loss of the style image, w_l represents the weighting factors that each layer contributes to the total loss, and $\vec{\alpha}, \vec{x}$ are the original and generated image, respectively. While E_l denotes the style loss at every single layer in the network.

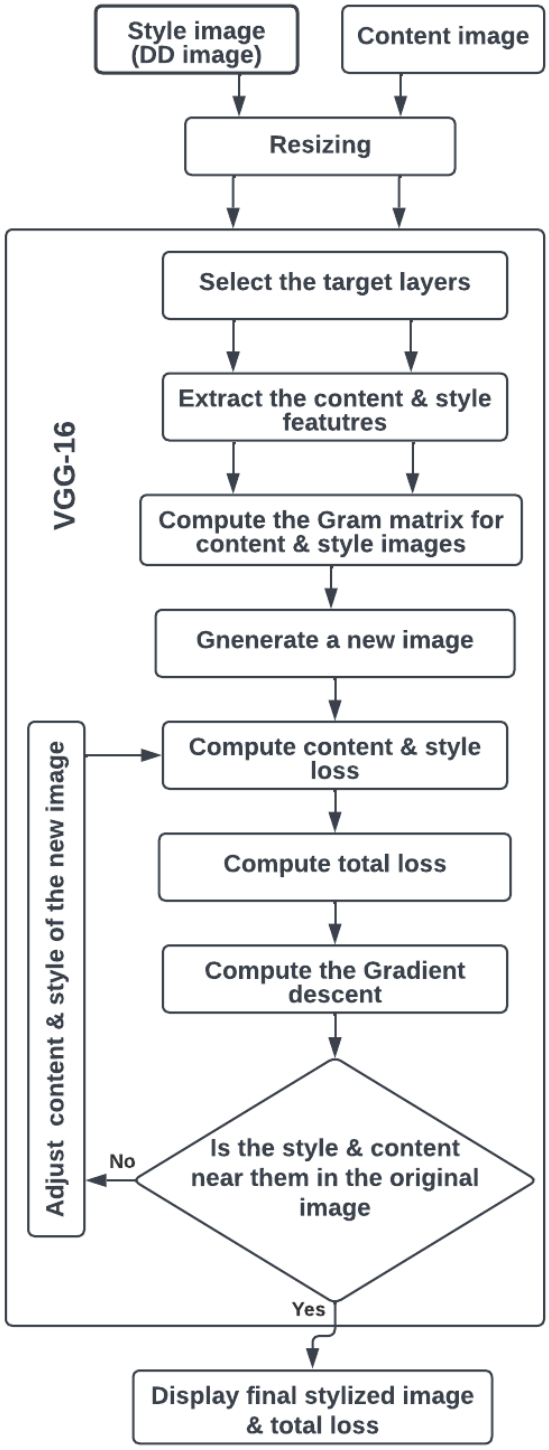

FIGURE 4. Diagram of the proposed NST model.

In order to optimize the output image and allow it to match the input image content and the reference image style, the total loss is calculated. The weighted content and style losses are added to determine the overall loss. The total loss is obtained by applying Equation 7.

$$L_{total}(\vec{p}, \vec{\alpha}, \vec{x}) = \alpha L_{content}(\vec{p}, \vec{x}) + \beta L_{style}(\vec{\alpha}, \vec{x}) \quad (7)$$

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

Gradient descent is used to reduce overall loss by modifying the generated image's pixel values. By reducing the total loss, this process is repeated until the generated image has a style that is similar to or convergent with the style of the style image and content that is similar to or convergent with the content image.

The gradient descent is computed using the Adam optimizer. The Adam optimizer is calculated using Equation 8 [49].

$$w_{t+1} = w_t - \frac{\mu}{\sqrt{\hat{v}_t}}. \quad (8)$$

where w_{t+1} is the updated weight, w_t is the previous weight, μ is the learning rate, $\hat{\mu}_t$ and $\hat{v}_t$ represent the bias-corrected estimate of the first and second moments of the gradient at time step t, respectively.

V. RESULTS

The input images are resized, as indicated previously. The input images were downloaded from Google Images. Figure 5 shows the images after resizing.

FIGURE 5. Resizing A and B images.

The resized images are blended to obtain one blended image. Figure 6 shows the blended image.

Table 3 shows the values of hyperparameters used in our experiments to generate Deep Dream images.

FIGURE 6. The blended image.

TABLE 3. The values of hyperparameters that are used in the deep dream experiments.

HYPERPARAMETER	value
Learning rate	0.01
Number of iterations	1000
Steps per iteration	50
Targeted layers' name	As mentioned in Table II

The blended image in Figure 6 is the input to the Deep Dream model. Figure 7 (a and b) shows the resulting Deep Dream images when using Xception and Inception v3, respectively. Figure 8 shows the resulting Deep Dream image when using Inception-ResNet-v2. While Figure 9 (a and b) shows the resulting Deep Dream images when using VGG16 and VGG19, respectively. Finally, Figure 10 shows the resulting Deep Dream image when applying our proposed deepy-dream model.

Table 4 shows the loss values that results based on each used model.

TABLE 4. The loss of the CNN architectures used to generate deep dream images.

Pre-trained network	Loss value
Xception	1.3086
Inception v3	4.4899
VGG16	7.8752
VGG19	16.1402
Inception-ResNet-v2	4.6213
The Our proposed deepy-dream	23.2599

Further experiments were conducted in which we tested our proposed deepy-dream on blurred images, in two cases; in the first case, a pure image of Franklin and blurring the same image as shown in Figure 11. While in the second case, the same blended image is blurred as in Figure 12.

Franklin's image is tested to show the difference between the blended image and the single image when our proposed

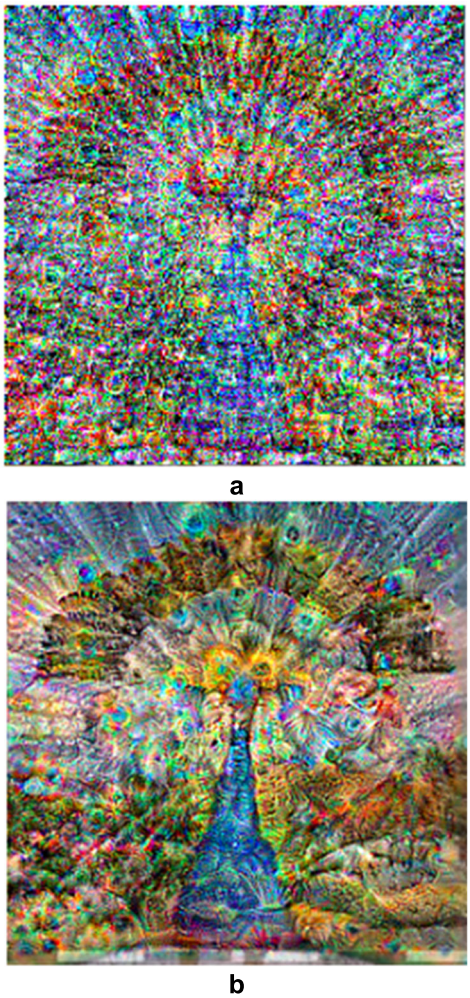

FIGURE 7. The Deep Dream images that result when applying: a. Xception network, b. Inception v3 network.

FIGURE 8. The Deep Dream image results when applying the Inception-ResNet-v2 network.

deepy-dream model is applied. The resulting images are shown in Figures 13 and 14. While Figure 15 shows the resulting Deep Dream image after applying our proposed model to a single pure Franklin's image.

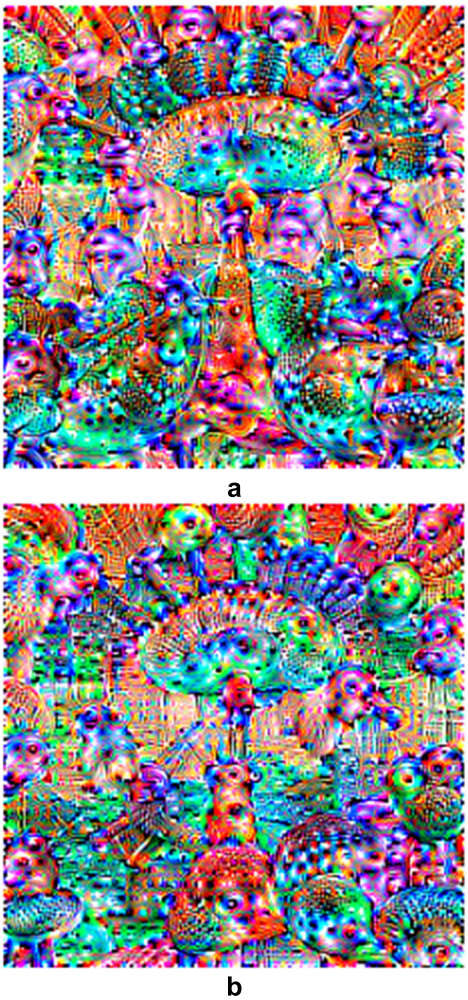

FIGURE 9. The Deep Dream images that result when applying: a. VGG-16 network, b. VGG-19 network.

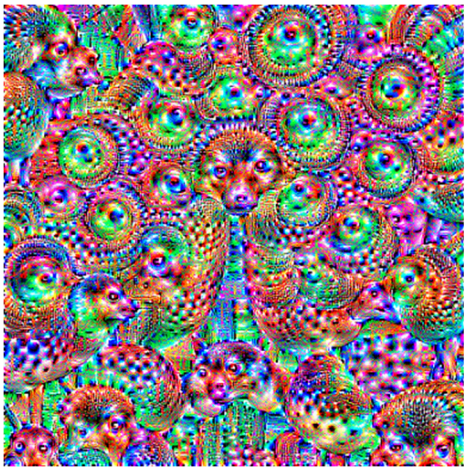

FIGURE 10. The Deep Dream image results when applying our proposed deepy-dream model.

FIGURE 11. The blurring of Franklin's image, A) the original image, B) the blurred image.

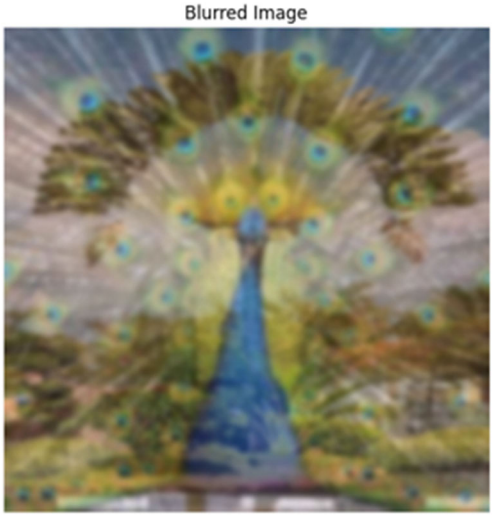

FIGURE 12. Blurring the blended image.

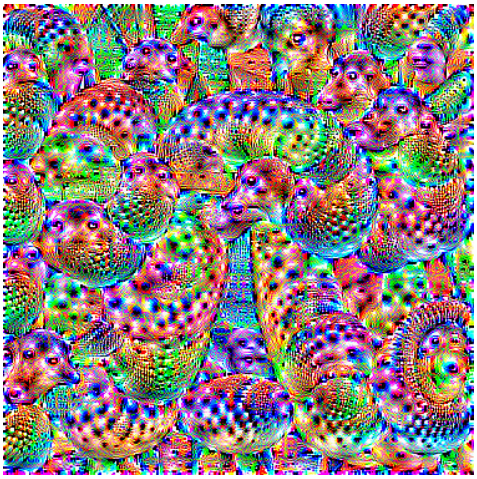

FIGURE 13. The Deep Dream image results when applying our proposed deepy-dream model on a blurred blended image.

The loss of the Deep Dream images results from applying our proposed deepy-dream model on the blurred and pure images are shown in Table 5.

In the case of implementing Deep Dream with autoencoder, after 1000 iterations, we got the Deep Dream image shown in Figure 16, with a loss of (0.5007293820381165).

FIGURE 14. The Deep Dream image results when applying our proposed deepy-dream model on a blurred Franklin's image.

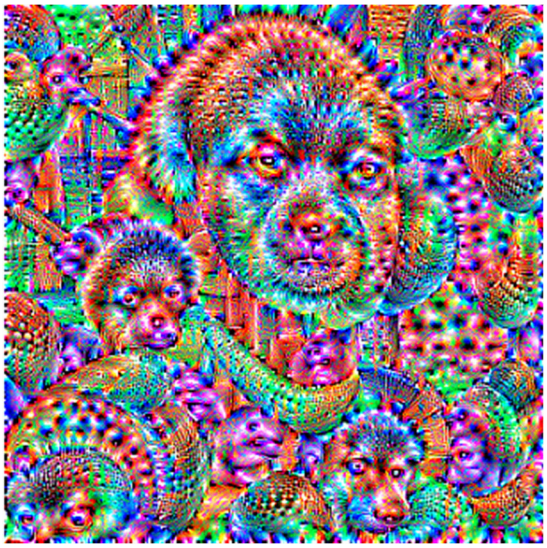

FIGURE 15. The Deep Dream image results when applying our proposed deepy-dream model on a pure Franklin's image.

TABLE 5. The loss values of our proposed model on blurred images.

Name of the image	Loss value
Pure blended image	23.2599
Blurred blended image	23.9267
Pure Franklin's image	22.9816
Blurred Franklin's image	23.4761

We implement the neural style transfer. NST takes two images as input, one of them represents the content image, while the other is the style image. The Deep Dream image is taken as the style image. VGG16 network is used as a pre-trained model in NST. The NST model is applied five times, each time one of the resulting Deep Dream images is used as a style image, while the content image is the same for all five times. Figure 17 shows the content image that is used in the NST model.

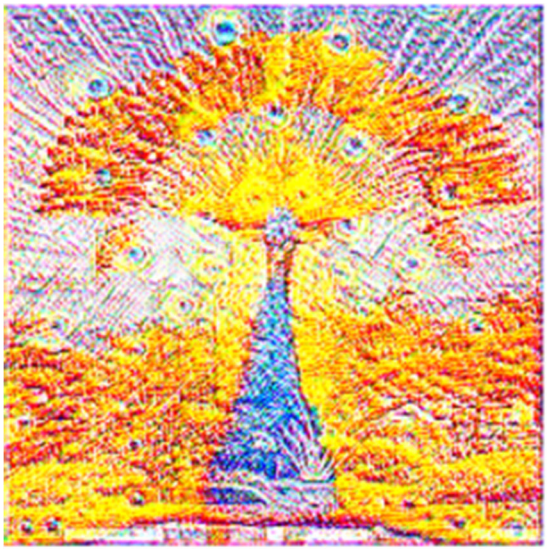

FIGURE 16. The Deep Dream image results when applying an autoencoder model.

FIGURE 17. The content image.

Figure 18 (a and b) shows the stylized images that result from using the Deep Dream images of Xception and Inception v3, respectively as style reference images. While Figure 19 shows the stylized images that result from using Deep Dream image of Inception-ResNet-v2 as a style reference image, and Figure 20 (a and b) shows the stylized images that result from using Deep Dream images of VGG16 and VGG19, respectively as style reference images. Figure 21 shows the stylized images that result from using Deep Dream image of our proposed deepy-dream model as a style reference image. Finally, the stylized images resulting from using the image generated by the autoencoder Deep Dream model as a style image are shown in Figure 22.

The loss of the NST model of each stylized image is computed, as shown in Table 6. This process is computed for each one of the five pre-trained models.

VI. EVALUATION METRICS

Deep Dream and NST models are typically evaluated using the SSIM, PSNR, and Normalized Cross-Correlation (NCC) metrics. The similarity in the structure between the two images (the generated and target images) is computed by

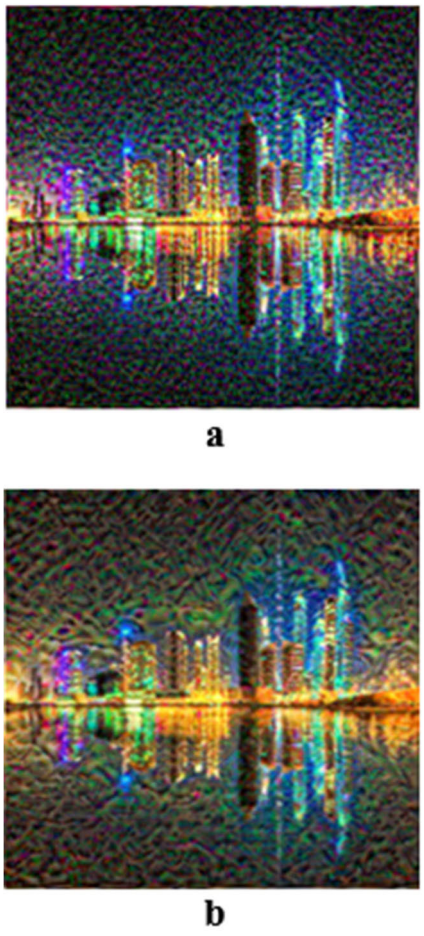

FIGURE 18. The resulting stylized images from applying style reference images from a. DD Xception network, b. DD Inception v3 network.

FIGURE 19. The resulting stylized image from applying style reference images from the DD Inception-ResNet-V2 network.

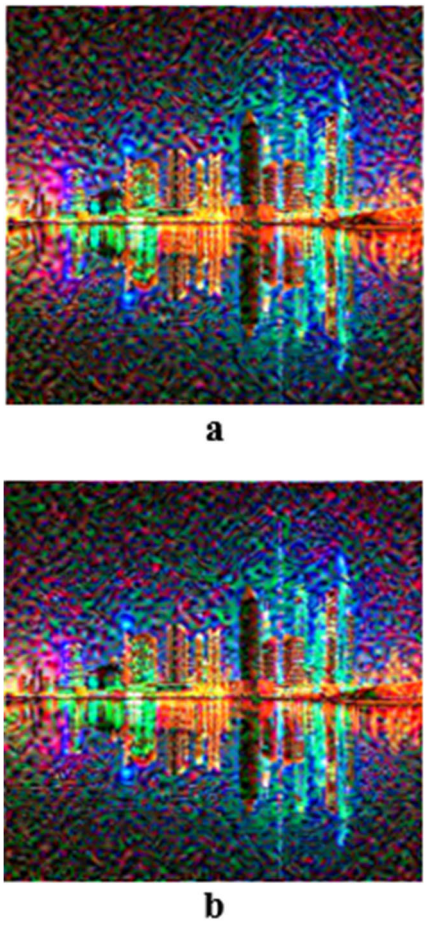

FIGURE 20. The resulting stylized images from applying style reference images from a. DD VGG-16 network, b. DD VGG-19 network.

FIGURE 21. The resulting stylized image from applying style reference image from our proposed deepy-dream model.

SSIM metric which takes the structure, brightness, and contrast into consideration. Higher values of SSIM, indicate that the matching between these images is better [50]. A greater PSNR indicates a smaller disparity between the generated and target images in terms of pixel values [37]. While the NCC metric is used for measuring the correlation between pixels in the source and generated images [15], [51]. These criteria can be used to assess how well Deep Dream and neural style transfer models do at creating images that reflect the desired style or content. In determining the quality of generated images, it is crucial to consider the subjective evaluation and visual inspection as equally significant. Equation 9 calculates

FIGURE 22. The resulting stylized image from applying style reference image from an autoencoder deep dream model.

TABLE 6. The loss values of the NST model.

Pre-trained network	Loss value
Xception	9042.4072
Inception v3	1889.0823
VGG16	10937.8564
VGG19	12055.8174
Inception-ResNet-v2	5392.1963
The Our proposed deepy-dream	49232.36
An autoencoder Deep Dream	37897.71

the SSIM [37].

$$SSIM\ (x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y + C_2)}{(\mu_x^2\mu_y^2 + C_1)(\sigma_x^2\sigma_y^2 + C_2)} \quad (9)$$

where μ_x and μ_y are the local means, σ_x and σ_y are the standard deviations for images x and y respectively, C1 and C2 are small constants to avoid instability.

Equation 10 is used to calculate PSNR [14].

$$PSNR = 10\log_{10}(peakval^2)/MSE \quad (10)$$

where *peakval* is the maximum possible pixel value, and MSE is the mean squared error between the color Deep Dream image and the reference image in the case of NST, and between the generated Deep Dream and the original input image.

Equation 11 is used to calculate NCC [51].

$$NCC\ (m, n) = \frac{\sum_{m,n} r\ (i, j)\ s(i + m, j + n)}{\sqrt{\sum_{m,n} r(i, j)^2 \sum_{m,n} s(i + m, j + n)^2}} \quad (11)$$

where I and j are the indices of the pixels in image r. while m and n represent the displacement that is applied to the indices of the pixels in image s.

In the case of NST, the measurements (PSNR, SSIM, and NCC) are computed two times; one is based on the content image and the other is based on the style image.

Equation 12 is used to calculate the MSE value [14].

$$MSE = \frac{1}{MN} \sum_{n=0}^{M} \sum_{m=1}^{N} [\hat{g}(n, m) - g(n, m)]^2 \quad (12)$$

where $g(n, m)$ is the first image and $\hat{g}(n, m)$ is the generated image.

By applying the SSIM, PSNR, and NCC metrics on the generated Deep Dream images, we get the results shown in Table 7.

TABLE 7. The SSIM, PSNR, AND NCC values based on the images generated from each Deep Dream pre-trained model.

The DD images based on the pre-trained models	SSIM value	PSNR value	NCC value
Xception	0.1495	27.2024	0.3684
Inception v3	0.4207	27.5370	0.7311
VGG16	0.1118	27.1759	0.4384
VGG19	0.1117	27.1863	0.4502
Inception-ResNet-v2	0.2347	27.3182	0.5083
Our proposed deepy-dream	0.0856	27.1919	0.1863
Autoencoder Deep Dream	0.4336	26.5813	0.6785

The SSIM, PSNR, and NCC values of the Deep Dream blurred images are shown in Table 8.

TABLE 8. The values of SSIM, PSNR, AND NCC values of the deep dream blurred images.

The image name	PSNR	SSIM	NCC
Pure Franklin's image	27.8592	0.0680	0.1399
Blurred Franklin's image	27.8314	0.0262	0.1293
Pure blended image	27.1919	0.0856	0.1863
Blurred blended images	27.2545	0.0561	0.1361

In the case of NST the SSIM and PSNR are computed twice; one is based on the content images and the other is based on the style images. There is only one content image, and while there are five style images; one Deep Dream image is generated from every pre-trained model.

Tables 9 and 10 contain the values of SSIM and PSNR of the stylized images based on content and style images, respectively, where the NST images are resulted by taking the Deep Dream images as style images.

TABLE 9. The values of SSIM and PSNR of the stylized images are based on the content image.

The style image of the pre-trained model	PSNR value	SSIM value	NCC value
Xception	28.1523	0.1110	0.6989
Inception v3	27.9137	0.1339	0.6732
VGG16	28.1344	0.0885	0.6625
VGG19	28.1627	0.0889	0.6693
Inception-ResNet-v2	27.9854	0.1243	0.7043
our proposed deepy-dream	29.1082	0.3097	0.8119
An autoencoder Deep Dream	30.1347	0.5399	0.8900

In the case of augmentation, Figure 23 depicts the Deep Dream image that is generated after 50 iterations. So, all

TABLE 10. The values of SSIM and PSNR of the stylized images are based on the style image.

The style image of the pre-trained model	PSNR value	SSIM value	NCC value
Xception	28.0279	0.0119	0.0013
Inception v3	27.9532	0.0297	0.0793
VGG16	28.0288	0.0103	0.0183
VGG19	28.0344	0.0089	0.0261
Inception-ResNet-v2	27.9479	0.0170	0.0628
our proposed deepy-dream	27.9427	0.0073	0.0007
An autoencoder Deep Dream	27.9600	0.0155	0.1514

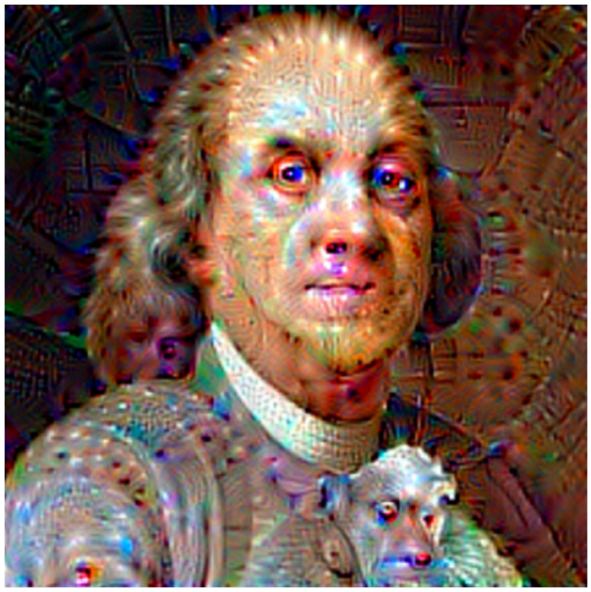

FIGURE 23. The resulting deep dream image from the proposed deepy-dream model using 50 iterations.

the previous images can be saved in the dataset to avoid overfitting problems and increase the system experience.

This experiment is conducted according to the hyper-parameters' values listed in Table 3, and by a trial-and-error method, we found that the optimal result is met at iteration 550.

VII. DISCUSSION

Deep Dream and neural style transfer are the most recent techniques which are used for multiple applications such as simulating the cases of schizophrenic patients and drug addicts, the entertainment industry, and generating images like art, in addition to increasing the size of the dataset in the case of saving the resulting images in the same dataset as every image has small variation from the previous one. In this study, a hybrid model that combines both a Deep Dream and NST models was developed. For the Deep Dream model, at first, five CNN architectures are used to generate Deep Dream images, which are VGG16, Inception v3, VGG19, Inception-ResNet-v2, and Xception. Then we proposed a new deepy-dream model to generate Deep Dream images by combining the features that are extracted from the above-mentioned CNN architecture models. While only VGG16 is used for NST.

All models produced various Deep Dream images, which are different visually and have different evaluation metrics values like loss, PSNR, and ISSM.

For the VGG-16, VGG-19, Inception v3, Inception-ResNet-v2, and Xception models, the differences in the resulting Deep Dream images and loss values are due to some factors including the different architectures that are used to build CNN variants such as residual connections, Inception modules, and depthwise separable convolutions, in the case of Inception-ResNet-v2, Inception v3, and Xception, respectively. While VGG16 and VGG19 structures start with a convolutional layer followed by pooling and the fully connected layers producing various images and different loss values. The second factor is the trained parameters, where each one of those models is developed for particular tasks and also trained on a specified dataset in a special pattern. The third factor is choosing the layers to extract their features, where the ability of each model to extract the features is different from the others. For example, the VGG16 and VGG19 are deeper than the others and can capture further spatial hierarchies. The fourth factor is the type of pooling layer, where some architectures use max pooling, while others use average pooling. Finally, the type of activation function, where there are many types of activation functions like ReLU, Leaky ReLU, ELU, etc.

In the case of our novel deepy-dream Deep Dream model, using this model led to generating dream-like images with visually high robustness, creativity, and diversity, where each one of those mentioned models has its specific learned features, and when those features are combined by using our deepy-dream method resulting in different and interesting visualization that shows wide features by exploiting the strong points of each model.

Our proposed deepy-dream Deep Dream model generates dream-like images that have a high simulation degree to the imaginary images that schizophrenic patients and drug addicts may imagine.

The same factors that caused the variations of the loss when using the VGG-16, VGG-19, Inception v3, Inception-ResNet-v2, and Xception models, caused the PSNR values of the images generated by the VGG19 model to be higher than the others, and the values of SSIM for VGG19 images are lower than the others because this network has more layers than VGG-16, while the design of Inception v3, Inception-ResNet-v2, and Xception models have different designs that are focused on the image classification and are less effective in generating dream-like images.

The loss of our proposed deepy-dream model has a higher value compared to the five CNN models because of the diverse representation of each model, where each model is trained differently and learned with a various number of parameters. So, our proposed deepy-dream Deep Dream

results from all these different models, causing its loss value to be higher than the others' values. While the loss value of the autoencoder Deep Dream model is the lowest value with a very wide variation compared to others. The PSNR value of our proposed deepy-dream Deep Dream model makes a compromise between the VGGs' and inceptions' families, where its value is higher than VGGs' values and less than Inceptions' values. The PSNR of the autoencoder Deep Dream model is the lowest among all models. While the SSIM value of our proposed model is the smallest compared to the values of the other five CNN models, and its highest value is with the autoencoder Deep Dream model because of the high similarity between the original and generated Deep Dream image. The variation in the SSIM value is back to the same reasons that make the variation in the loss values.

In the case of the NCC, we found that its value is the lowest on our proposed deepy-dream model, while it is the highest on the Inception v3 Deep Dream model, and this reflects how the generated Deep Dream image is modified in our proposed model, while the change is small in the case of the Inception v3 Deep Dream model.

The proposed deepy-dream model is implemented with blurred images for both a single Franklin's image and blended images and proves that the loss value of both the single and blended blurred images are higher than their values for the same pure images.

For Neural Style Transfer (NST), when using the stylized images generated by employing images resulting from the Deep Dream based on the Inception family (Inception v3, Inception-ResNet-v2, and Xception) as style reference images, higher loss values are observed compared to using images resulting from the Deep Dream based on the VGG family (VGG-16 and VGG-19). This is because the VGG networks are deeper, and as the image passes through deeper layers, the loss value is minimized.

The loss values of the stylized images generated using our proposed Deep Dream model are the highest loss value among all other values of the loss of other models.

The PSNR, SSIM, and NCC values for the stylized images are computed twice; the former is based on the content images, while the latter is based on the style reference images. When computing the values of the metrics based on the content image, we found that all the metrics are the highest in the case of the stylized image is generated by employing style images resulting from autoencoder Deep Dream model then those generated from our proposed deepy-dream Deep Dream model, since their stylized images is more complex in case of style and/or overlapping colors. When relying on style reference images to compute the metrics values, it is obvious that the values of SSIM and NCC are the lowest in the case of stylized image is generated by employing style images resulting from our proposed deepy-dream Deep Dream model, while PSNR is the highest in the case of the autoencoder Deep Dream model.

In summary, the proposed deepy-dream Deep Dream model proved creativity and stability in addition to the diversity of the visualized features in the dreamed images, but it still has limitations such as time-consuming and large space complexity of the model. Where mixing the features from many networks adds high complexity and consumes more time.

VIII. CONCLUSION

Recently, Deep Dream and neural style transfer emerged and developed as important techniques in the computer vision field. This study presents a hybrid model which integrates those two techniques. First, the input images are blended into particular weights. Deep Dream model is built and implemented using five pre-trained models, which are VGG16, Inception v3, VGG19, Inception-ResNet-v2, and Xception. A novel deepy-dream Deep Dream model is developed by combining those five CNN architectures, taking advantage of their properties which results in higher stability, creativity, and diversity. The generated images from this model mimic the imaginations of schizophrenic patients and drug addicts.

A new autoencoder Deep Dream model is developed based on VGG-19 features to generate Deep Dream images, the images generated from this model are less effective and have the smallest loss values from all other Deep Dream models.

The proposed deepy-dream model is also tested against blurred images for both blended and Franklin's images, and the results show that the loss of the pure Franklin's Deep Dream image is higher than the blurred Deep Dream of the same image, while in the case of blended Deep Dream image, we got the opposite.

The NST model is implemented by using VGG16 pre-trained model. The loss in the Deep Dream is computed using gradient ascent; which works to maximize the loss value. In NST the loss is computed using gradient descent; which works to minimize the loss value using, Adam optimizing algorithm. Xception pre-trained network gives the lowest loss value in a deep dream, while our proposed deepy-dream Deep Dream model gives the highest value. In NST, the stylized images that take the output of our proposed deepy-dream Deep Dream images as style images are the highest loss value and the stylized images resulting from the style images of the Inception v3 have the lowest loss values. The evaluation metrics used in this study are SSIM, PSNR, and NCC. These metrics indicated that our proposed deepy-dream Deep Dream model gives the lowest values in the case of using SSIM and NCC metrics, and the generated Deep Dream images have the highest difference from the original image that is input to this model. PSNR is used to measure the quality of the generated Deep Dream images, where the inception v3 provided the highest value and VGG-16 had the lowest PSNR value. In the case of the NST, the NCC, SSIM, and PSNR of the stylized image resulting from our proposed deepy-dream Deep Dream model have the highest values when benchmarked with the content image and the lowest values when compared with the style image. Furthermore, as shown in Figure 23, we have observed that images generated during the initial 50 iterations can be strongly considered

as an augmentation technique. Where these images are clear, preserving the most important features while introducing alterations that aid in improving the classification process across various scenarios.

Future direction will involve the use of average bagging ensembles to reduce the space complexity of the model.

REFERENCES

[1] N. Ketkar and E. Santana, *Deep Learning With Python*, vol. 1. Berkeley, CA, USA: Apress, 2017.

[2] C. Rastelli, A. Greco, Y. N. Kenett, C. Finocchiaro, and N. De Pisapia, "Simulated visual hallucinations in virtual reality enhance cognitive flexibility," *Sci. Rep.*, vol. 12, Mar. 2022, Art. no. 4027.

[3] S. Tammina, "Transfer learning using VGG-16 with deep convolutional neural network for classifying images," *Int. J. Sci. Res. Publications*, vol. 9, no. 10, p. 9420, Oct. 2019, doi: 10.29322/ijsrp.9.10.2019.p9420.

[4] C. Lin, L. Li, W. Luo, K. C. P. Wang, and J. Guo, "Transfer learning based traffic sign recognition using inception-v3 model," *Periodica Polytechnica Transp. Eng.*, vol. 47, no. 3, pp. 242–250, Aug. 2018, doi: 10.3311/PPtr.11480.

[5] L. Wen, X. Li, X. Li, and L. Gao, "A new transfer learning based on VGG-19 network for fault diagnosis," in *Proc. IEEE 23rd Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2019, pp. 205–209, doi: 10.1109/CSCWD.2019.8791884.

[6] J. Wang, X. He, S. Faming, G. Lu, H. Cong, and Q. Jiang, "A real-time bridge crack detection method based on an improved inception-ResNet-v2 structure," *IEEE Access*, vol. 9, pp. 93209–93223, 2021, doi: 10.1109/ACCESS.2021.3093210.

[7] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258, doi: 10.1109/CVPR.2017.195.

[8] A. Mordvintsev, C. Olah, M. Tyka, and E. Al. (2015). Inceptionism: Going deeper into neural networks. Google Research Blog. Accessed: Dec. 1, 2022. [Online]. Available: http://googleresearch.blogspot.co.uk/2015/06/inceptionism-going-deeper-into-neural.html

[9] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," 2015, *arXiv:1508.06576*.

[10] A. Singh, V. Jaiswal, G. Joshi, A. Sanjeeve, S. Gite, and K. Kotecha, "Neural style transfer: A critical review," *IEEE Access*, vol. 9, pp. 131583–131613, 2021, doi: 10.1109/ACCESS.2021.3112996.

[11] H. Li, "A literature review of neural style transfer," Princeton Univ., Princeton, NJ, USA, Tech. Rep. 085442019, 2018.

[12] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization techniques in training DNNs: Methodology, analysis and application," 2020, *arXiv:2009.12836*.

[13] L. Zhang, T. Wen, and J. Shi, "Deep image blending," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 231–240.

[14] U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through FSIM, SSIM, MSE and PSNR—A comparative study," *J. Comput. Commun.*, vol. 7, no. 3, pp. 8–18, 2019, doi: 10.4236/jcc.2019.73002.

[15] A. Kaso, "Computation of the normalized cross-correlation by fast Fourier transform," *PLoS ONE*, vol. 13, no. 9, pp. 1–16, 2018, doi: 10.1371/journal.pone.0203434.

[16] X. Chen, Y. Zhang, Y. Wang, H. Shu, C. Xu, and C. Xu, "Optical flow distillation: Towards efficient and stable video style transfer," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 614–630.

[17] H. C. Choi, "Toward exploiting second-order feature statistics for arbitrary image style transfer," *Sensors*, vol. 22, no. 7, p. 2611, 2022, doi: 10.3390/s22072611.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis.*, Zurich, Switzerland, 2014, pp. 740–755, doi: 10.1007/978-3-319-10602-1_48.

[19] L. R. Al-Khazraji, A. R. Abbas, and A. S. Jamil, "Employing neural style transfer for generating deep dream images," *ARO, Sci. J. Koya Univ.*, vol. 10, no. 2, pp. 134–141, Dec. 2022, doi: 10.14500/aro.11051.

[20] H. Yin, P. Molchanov, J. M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz, "Dreaming to distill: Data-free knowledge transfer via deepinversion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8715–8724, doi: 10.1109/CVPR42600.2020.00874.

[21] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 0, 2009.

[22] R. Arthi, A. R. Kishan, A. Abraham, and A. Sattenapalli, "Centralized intelligent authentication system using deep learning with deep dream image algorithm," in *Proc. ETAEERE*, 2021, pp. 169–178, doi: 10.1007/978-981-15-7504-4_18.

[23] T. T. J. Kiran, "Deep inceptionism learning performance analysis using TensorFlow with GPU—Deep dream algorithm," *J. Emerg. Technol. Innov. Res.*, vol. 8, no. 5, pp. 322–328, 2021.

[24] B. A. El-Rahiem, M. Amin, A. Sedik, F. E. A. E. Samie, and A. M. Iliyasu, "An efficient multi-biometric cancellable biometric scheme based on deep fusion and deep dream," *J. Ambient Intell. Hum. Comput.*, vol. 13, no. 4, pp. 2177–2189, Apr. 2022, doi: 10.1007/s12652-021-03513-1.

[25] L. R. Al-Khazraji, A. R. Abbas, and A. S. Jamil, "Effect of changing targeted layers of the deep dream technique using VGG-16 model," *Int. J. Online Biomed. Eng.*, vol. 19, no. 03, pp. 34–47, Mar. 2023.

[26] P. Sahu, A. Chug, A. P. Singh, and D. Singh, "Classification of crop leaf diseases using image to image translation with deep-dream," *Multimedia Tools Appl.*, pp. 1–35, Mar. 2023, doi: 10.1007/s11042-023-14994-x.

[27] L. R. Ali, B. N. Shaker, and S. A. Jebur, "An extensive study of sentiment analysis techniques: A survey," *AIP Conf. Proc.*, vol. 2591, no. 1, 2023, Art. no. 030022, doi: 10.1063/5.0119604.

[28] T. A. Jaber, "Artificial intelligence in computer networks," *Periodicals Eng. Natural Sci.*, vol. 10, no. 1, pp. 309–322, 2022, doi: 10.21533/pen.v10i1.2616.

[29] J. Zhang, Y. Li, W. Xiao, and Z. Zhang, "Non-iterative and fast deep learning: Multilayer extreme learning machines," *J. Franklin Inst.*, vol. 357, no. 13, pp. 8925–8955, Sep. 2020, doi: 10.1016/j.jfranklin.2020.04.033.

[30] R. Lateef and A. Abbas, "Tuning the hyperparameters of the 1D CNN model to improve the performance of human activity recognition," *Eng. Technol. J.*, vol. 40, no. 4, pp. 547–554, Apr. 2022, doi: 10.30684/etj.v40i4.2054.

[31] S. A. Jebur, K. A. Hussein, H. K. Hoomod, and L. Alzubaidi, "Review on deep learning approaches for anomaly event detection in video surveillance," *Electronics*, vol. 12, no. 1, pp. 1–22, 2023, doi: 10.3390/electronics12010029.

[32] L. R. Ali, H. K. Homood, and A. S. Elameer, "Facial expression recognition by using modified convolutional neural network (MCNN) and modified Gabor filter," *Int. J. Develop. Res.*, vol. 7, no. 11, pp. 16960–16967, 2017, doi: 10.13140/RG.2.2.31549.56805.

[33] S. A. Sabeeh and S. H. Ameen, "Detection and classification of leaf disease using deep learning for a greenhouses' robot," *Iraqi J. Comput. Commun. Control Syst. Eng.*, vol. 21, no. 4, pp. 15–28, 2021, doi: 10.33103/uot.ijccce.21.4.2.

[34] J. Zhang, Y. Zhao, F. Shone, Z. Li, A. F. Frangi, S. Q. Xie, and Z.-Q. Zhang, "Physics-informed deep learning for musculoskeletal modeling: Predicting muscle forces and joint kinematics from surface EMG," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 484–493, 2023, doi: 10.1109/TNSRE.2022.3226860.

[35] S. M. E. Abyad, M. M. Soliman, and K. M. E. Sayed, "Deep video hashing using 3DCNN with BERT," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 5, pp. 113–127, 2022, doi: 10.22266/ijies2022.1031.11.

[36] J. Chaki and M. Woźniak, "A deep learning based four-fold approach to classify brain MRI: BTSCNet," *Biomed. Signal Process. Control*, vol. 85, Aug. 2023, Art. no. 104902, doi: 10.1016/j.bspc.2023.104902.

[37] O. M. Al-Hazaimeh, A. Abu-Ein, N. Tahat, M. Al-Smadi, and M. Al-Nawashi, "Combining artificial intelligence and image processing for diagnosing diabetic retinopathy in retinal fundus images," *Int. J. Online Biomed. Eng. (iJOE)*, vol. 18, no. 13, pp. 131–151, Oct. 2022, doi: 10.3991/ijoe.v18i13.33985.

[38] Q. Ji, J. Huang, W. He, and Y. Sun, "Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images," *Algorithms*, vol. 12, no. 3, pp. 1–12, 2019, doi: 10.3390/a12030051.

[39] T.-H. Nguyen, T.-N. Nguyen, and B.-V. Ngo, "A VGG-19 model with transfer learning and image segmentation for classification of tomato leaf disease," *AgriEngineering*, vol. 4, no. 4, pp. 871–887, Oct. 2022, doi: 10.3390/agriengineering4040056.

[40] M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, and Y. Zhang, "Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery," *Remote Sens.*, vol. 10, no. 7, pp. 1–21, 2018, doi: 10.3390/rs10071119.

[41] B. Jabir, N. Falih, and K. Rahmani, "Accuracy and efficiency comparison of object detection open-source models," *Int. J. Online Biomed. Eng.*, vol. 17, no. 5, pp. 165–184, 2021, doi: 10.3991/ijoe.v17i05.21833.
[42] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, Dec. 2019, doi: 10.1186/s40537-019-0197-0.
[43] M. Attia, M. Hossny, S. Nahavandi, A. Yazdabadi, and H. Asadi, "Realistic hair simulation using image blending," 2019, *arXiv:1904.09169*.
[44] M. Woźniak, M. Wieczorek, and J. Siłka, "Deep neural network with transfer learning in remote object detection from drone," in *Proc. 5th Int. ACM Mobicom Workshop Drone Assist. Wireless Commun. for 5G Beyond*, Oct. 2022, pp. 121–126, doi: 10.1145/3555661.3560875.
[45] W. Siłka, M. Wieczorek, J. Siłka, and M. Woźniak, "Malaria detection using advanced deep learning architecture," *Sensors*, vol. 23, no. 3, p. 1501, Jan. 2023, doi: 10.3390/s23031501.
[46] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, "Basics of supervised deep learning," in *Advances in Deep Learning*, vol. 57. Singapore: Springer, 2020, pp. 13–30.
[47] Y. Bai, "RELU-function and derived function review," in *Proc. SHS Web Conf.*, vol. 144, 2022, p. 2006, doi: 10.1051/shsconf/202214402006.
[48] R. Nirthika, S. Manivannan, A. Ramanan, and R. Wang, "Pooling in convolutional neural networks for medical image analysis: A survey and an empirical study," *Neural Comput. Appl.*, vol. 34, no. 7, pp. 5321–5347, Apr. 2022, doi: 10.1007/s00521-022-06953-8.
[49] L. R. Ali, S. A. Jebur, M. M. Jahefer, and B. N. Shaker, "Employing transfer learning for diagnosing COVID-19 disease," *Int. J. Online Biomed. Eng.*, vol. 18, no. 15, pp. 31–42, Dec. 2022, doi: 10.3991/ijoe.v18i15.35761.
[50] I. Bakurov, M. Buzzelli, R. Schettini, M. Castelli, and L. Vanneschi, "Structural similarity index (SSIM) revisited: A data-driven approach," *Expert Syst. Appl.*, vol. 189, Mar. 2022, Art. no. 116087, doi: 10.1016/j.eswa.2021.116087.
[51] N. Dematteis and D. Giordan, "Comparison of digital image correlation methods and the impact of noise in geoscience applications," *Remote Sens.*, vol. 13, no. 2, pp. 1–25, 2021, doi: 10.3390/rs13020327.

LAFTA R. AL-KHAZRAJI received the B.Sc. degree in computer science from Al Rafidain University College, Baghdad, Iraq, and the higher Diploma degree in information technology and the M.Sc. degree in computer science from Iraqi Commission for Computers and Informatics, Iraq, in 2013 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science, University of Technology, Iraq. His research interests include machine learning, deep learning, NLP, image processing, pattern recognition, and computer networks.

AYAD R. ABBAS received the B.Sc. and M.Sc. degrees in computer science from the Computer Science Department, University of Technology, Iraq, in 2003 and 2005, respectively, and the Ph.D. degree in applied computer technology from the School of Computer Science, Wuhan University, China, in 2009. He is a Professor with the Computer Science Department, University of Technology. His research interests include artificial intelligent, machine learning, natural language processing, deep learning, data mining, web mining, information retrieval, soft computing, e-learning, e-commerce, and recommended systems.

ABEER S. JAMIL received the M.Sc. and Ph.D. degrees in computer science from the University of Technology, Iraq, in 2004 and 2015, respectively. She has around 25 years of teaching experience. She was an Instructor with the Cisco Network Academic (CISCO), for 13 years. Her research interests include digital image processing, video processing, security software engineering, and networking and artificial intelligence applications.

ABIR JAAFAR HUSSAIN (Senior Member, IEEE) received the Ph.D. degree from The University of Manchester (UMIST), U.K., in 2000. Her Ph.D. thesis entitled "Polynomial Neural Networks for Image and Signal Processing." She is a Visiting Professor of machine learning with Liverpool John Moores University, U.K. She has worked with higher-order and recurrent neural networks and their applications to e-health and medical image compression techniques. She and her research students have developed several recurrent neural network architectures. She is a Ph.D. supervisor and an external examiner of research degrees, including the Ph.D. and the M.Phil. She is one of the initiators and chairs of the Development in e-Systems Engineering (DeSE) series. She has published numerous refereed research papers in conferences and journals in the research areas of neural networks, signal prediction, telecommunication fraud detection, and image compression.