The Effect of Changing Targeted Layers of the Deep Dream Technique Using VGG-16 Model

https://doi.org/10.3991/ijoe.v19i03.37235

Lafta R. Al-Khazraji[1,2(✉)], Ayad R. Abbas[1], Abeer S. Jamil[3]
[1]Department of Computer Science, University of Technology, Baghdad, Iraq
[2]General Directorate of Education of Salahuddin Governorate, Iraq
[3]Department of Computer Technology Engineering, Al-Mansour University College, Baghdad, Iraq
l.alkhazraji@gmail.com

Abstract—The deep dream is one of the most recent techniques in deep learning. It is used in many applications, such as decorating and modifying images with motifs and simulating the patients' hallucinations. This study presents a deep dream model that generates deep dream images using a convolutional neural network (CNN). Firstly, we survey the layers of each block in the network, then choose the required layers, and extract their features to maximize it. This process repeats several iterations as needed, computes the total loss, and extracts the final deep dream images. We apply this operation on different layers two times; the former is on the low-level layers, and the latter is on the high-level layers. The results of applying this operation are different, where the resulting image from applying deep dream on the high-level layers are clearer than those resulting from low-level layers. Also, the loss of the images of low-level layers ranges between 31.1435 and 31.1435, while the loss of the images of upper layers ranges between 20.0704 and 32.1625.

Keywords—deep dream, VGG-16, CNN, gradient ascent, normalization

1 Introduction

In recent years, the deep dream technique emerged as a hot topic in deep learning, which provides the possibility of simulating hallucinations experienced by schizophrenic patients and drug addicts, where the drugs and narcotic drugs have a very significant effect on users, causing hallucinations, blurred vision, and delusional perceptions [1]. The deep dream is created through the repeating feeds of the image to the CNN model, where the first layer's low-level features (i.e., edges and lines) are detected. After that, high-level features like faces and trees appeared. In the end, all those features are collected for configuring combined effects such as trees or the whole structure [2]. The deep dream enhances the images' visual attributes [3].

Deep dream visualization was utilized for the determination of whether a CNN had correctly learned correct image features. Which is why, a deep dream has been created through the increasing feeding of an image to the network where the first layers are responsible for the detection of the first low-level features (in other words, the edges). After that, high-level features (such as the trees and faces) appear, which go deeper into the network. Finally, the rare final layers collect all these to configure the combined effects (for example, the trees or whole structures) [4].

There are many deep CNN architectures such as GoogleNet, AlexNet, residual neural network (ResNet), and visual geometry group (VGGNet) [5].

Normalization is a general transformation, ensuring that transformed data has some specific statistical characteristics [6].

In this study, we propose a deep dream system based on the deep learning technique by focusing on low-level and high-level features of the network layers. VGG-16 pre-trained network is used to build the model, and based on the convolution operation in the convolution layer, the features of the targeted layers are extracted from the images. Then the gradient ascent is used to maximize the loss function of those layers. Finally, the deep dream images are generated with the final loss for each image. So, the resulting images are different based on the targeted layers. Also, the loss values change depending on which layers maximized their loss.

2 Related work

Graeme McCaig et al. (2016) [7] have proposed 2 DEEP DREAM algorithms that were successful in achieving the visual blending in the CNNs; the first one was the Google Deep Dream, whereas the second one was the algorithm that was proposed by (Gatys and his colleagues) that took random images as the inputs, and after that, they have split and recombined its style and content with the use of the neural networks for the creation of the artistic images that are thus referred to as the deep style DS. Researchers have implemented applied GoogLeNet [8] for deep dream and VGG for DS algorithms, the network has been trained on Cars and ImageNet data-sets. By their study of the subject of the deep dream, they have noticed 2 clichéd aspects that should be declared, which are: first, a designed DEEP DREAM represents a bottom-up discrimination network, which is why, GoogLeNet ignores a considerable amount of the data about tonic color of the regions with retaining the color contrast around the edges. Second, ImageNet training data has bias towards the animal types due to the fact that it represents an important portion of the 1,000 labeled classes, with a specific focus upon accurate dog breed distinctions. Therefore, tending to treat the patterns as pertinent to dog features results in the emergence of the dog features. Whereas the car features emerge as a result of the network that was trained on the car data-set.

H. Yin et al. (2020) [9] have suggested a novel method for the synthesis of the images from the image distribution for the training of the deep neural networks. Their method that has been referred to as the DeepInversion is made up of 2 parts, namely; teacher and student logits. The teacher represents the inverted trained network which begins from the random noise without the use of any deep dream additional information on training data-set. They have utilized the Deep Dream for building DeepInversion through the improvement of DEEP DREAM's image quality through extending the image regularization with a new term of feature distribution regularization. Their model was trained on the CIFAR-10 and on the ImageNet data-sets.

T. J. Kiran (2021) [10] have proposed a computer vision algorithm that has been referred to as the Deep Inceptionism learning or the DEEP DREAM algorithm. The training phase of this algorithm begins in the case where an image enters the network, which causes the neurons to fire then generate activations. The concept of their proposed algorithm is to make some neurons fire more through the modification of input image (through the boosting or activation of the neurons). DEEP DREAM provides an ability for choosing certain neurons in certain layers they're willing to, which makes them fire more conspicuously. Such process would be frequently repeated to the point where an input image includes all of the features that are required by a certain layer. They continuously feed those images to network, and the more they feed it to network, the more they will have the ability for the extraction or seeing of all those strange elements in an actual image. Which is why, the steps of their algorithm begin by sending an image to the trained CNN, ANN, ResNet, and so on. After that, a layer is chosen (top layer captures the edges, whereas the deeper ones capture the whole shapes like faces) and specify activations (i.e., output) that are produced by layer of interest. After that, calculation of activation gradient concerning input image and altering image to boost those activations, improving patterns that have been detected by network, which results in the production a trippy hallucinated image, and finally, repeating the iteratively across multiple scales. They have maximized loss function through performing gradient ascent at every one of the layers, then they deep dreamed up all losses from every layer and passed them all, which has been the same parameter plotting here or printing and returning it from the function of gradient ascent. They could result in the optimization of their results by the execution of the gradient ascent and running their dream algorithm.

B. Abd El-Rahiem et al. (2022) [11] have utilized the DEEP DREAM for the presentation of a multi-biometric cancellable scheme (MBCS) for the creation of cancellable patterns of the finger-prints which are effective as well as secure, the biometric modalities based upon finger veins and eye iris. They have harnessed the power of the DL models for the creation of a multi-exposure deep fusion module which generates a fused biometrical template, which is collected after that as deep dream module's final cancellable template. They have utilized the Inception v3 as pre-trained network. Loss value has to be maximized throughout gradient-ascent phase for deep dream to work. The objective of the filter visualization is maximizing the value of a certain filter in a certain layer, including the increase of the number of the filter activations, 3 main parameters control gradient ascent process meters, including: gradient step *S*, maximum loss *Lmax*, and several iterations *I*, which are applied to the convent layers' loss gradients.

3 Convolutional neural network (CNN)

Machine learning is widely used to build and evaluate the AI models, and it has many AI algorithms that used for this purpose such as (ANN, SVM, and Random Forest) [12], [13]. Deep learning DL has been considered as a sub-set of the machine learning field and it is an artificial intelligence (AI) area which mimics the human mind work in the area of data processing [14], [15]. CNN is one of the most commonly utilized deep neural networks [16].

CNNs are deep neural models that have been created for handling the image data, and they may be considered as feed-forward ANNs with a variety of the subsampling and convolutional layers [17], [18].

The advanced deep CNN approaches were utilized in order learn the extraction of the representative features from the images [19].

The features layers receive the input from the neighboring layers that can perform the extraction of the basic visual features, which include the edges, ending points, corners, and so on, which are pooled after that via upper layer. It's standard to have image/pattern recognition as featured extractor for collecting the relevant data and removing the unrelated ones, followed by trained classifier for the purpose of sorting feature vectors to classes. The convolutional layer plays feature extractor role in the CNN for the extraction of local features of a model [20]. Figure 1 illustrates the diagram of the CNN [21].

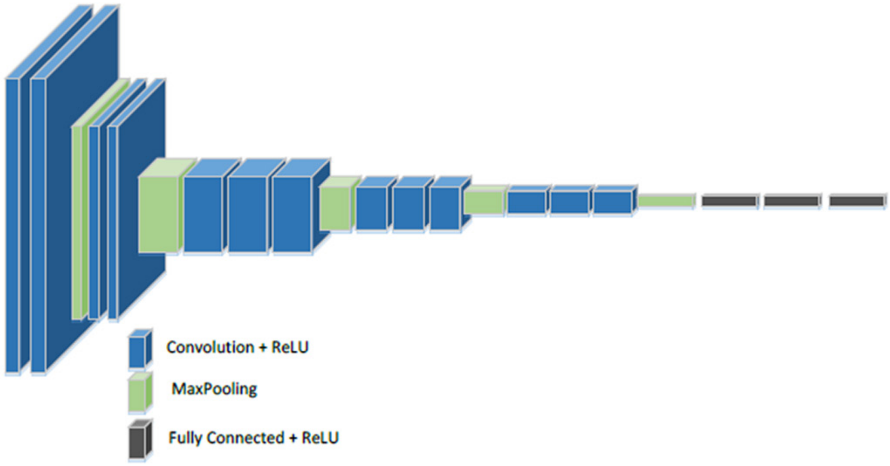

Fig. 1. CNN block diagram

VGG16 (Visual Geometry Group) is a 16-layer network. It includes 5 convolution layer blocks and one max-pooling layer that is followed by 3 fully connected (FC) layers. The convolutional layers are dependent upon a 3x3 kernel with a value of 1 for the padding as well as the stride for ensuring that resultant map has the same dimension length as the activation map of the previous layer. In addition to that, for guaranteeing the activation map's spatial dimension from previous layer is halved, the max-pooling layer utilizes a 2x2 kernel size, no padding and a stride of 2. This differentiates a CNN when it's compared to the backpropagation network size of a similar number of the layers. At the end of every convolution and max pooling block, a rectified linear unit activation would be utilized for the reduction of spatial dimension. Finally, 3 FC layers have been utilized in the VGG-16 model for the final classification [22]. Table 1 shows the ConvNet configurations for the VGG16 network [23].

Table 1. ConvNet configurations

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

As shown in Table 1, the feature maps' number ranging from 64 and 512. Where the input to the first convolution layers block is an image with 224 × 224 × 3 and the feature maps is 64. And here, the process of feature extraction starts by moving the filters from the high left corner in the directions (left-right) the when reach to the right end of the image it back to the left and move down and continue the same way. The filter size in the convolution layers is 3 × 3, and the stride is 2 with no padding. Each convolutional layer is followed by activation function layer, and here ReLU activation function is used. Each convolutional block is followed by maxpooling layer with a kernel size of 2 × 2, the network contains five maxpooling layers and thirteen convolutional layers and after all these layers there are three fully connected layers each one of the first two is 1 × 1 × 4096, where the last number is the depth, while the third fully connected layer is 1 × 1 × 1000, which achieve the classification process, where the number 1000 refers to the number of classes. SoftMax layer is the final layer in this network. Figure 2 illustrate this process [24].

VGG-16 network is pre-trained on a subset of the ImageNet dataset, which is a dataset of images. This dataset contains more than fourteen million images divided into 21,841 classes. The subset that the VGG-16 network pre-trained on consists of 1000 classes, and each class contains more than 1000 images [25].

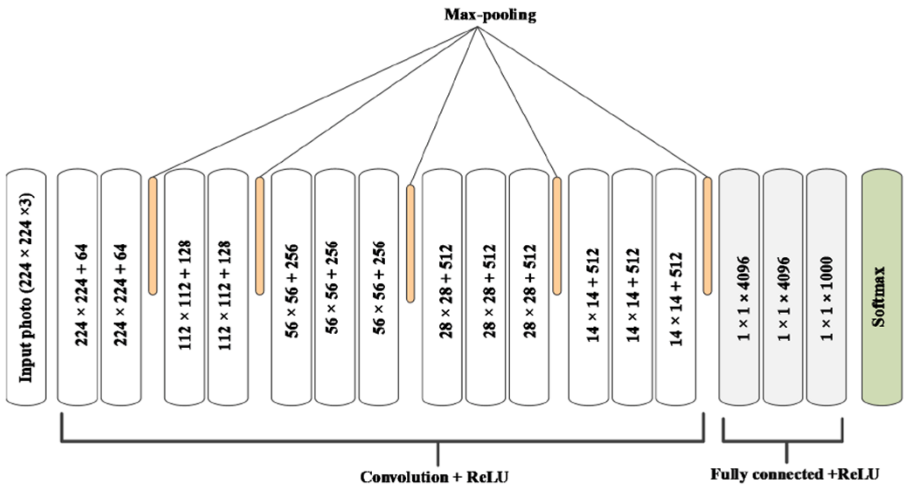

Fig. 2. VGG-16 architecture

The gradient descent is the most famous optimization approach that has been utilized for minimizing the cost function. Equation (1) calculates the gradient descent [26].

$$w = w - \mu \cdot \nabla E(W) \tag{1}$$

Where, $\nabla E(W)$ Represents gradient of the error loss, μ is the learning rate.

4 Methodology

The system was built using the deep CNN method, where the VGG-16 network is used as a pre-trained model to generate deep dream images. The diagram of our proposed model has been illustrated in Figure 3.

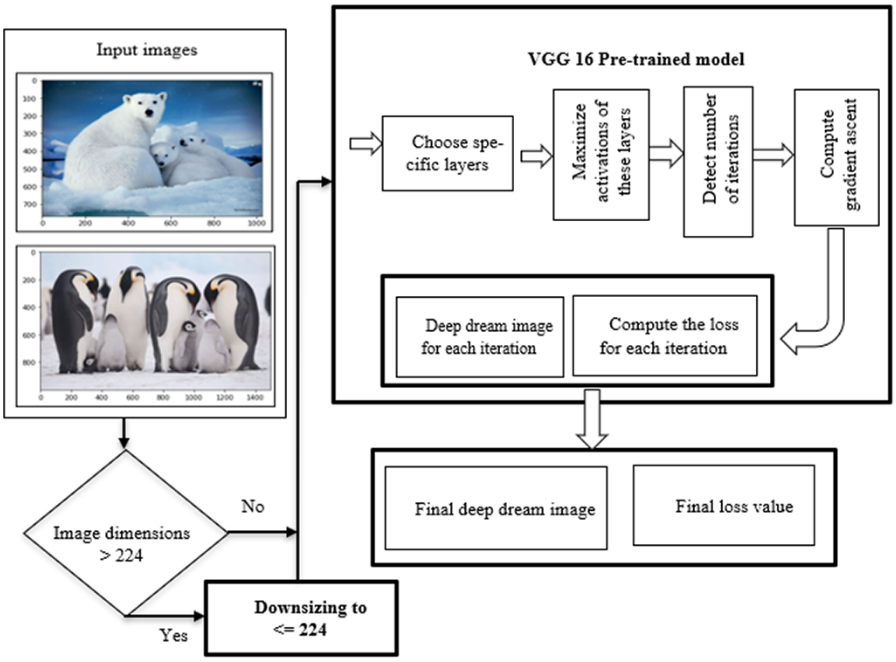

Fig. 3. Diagram of the proposed model

At first, the images are input into the system. Figure 4 shows the real size of the input images.

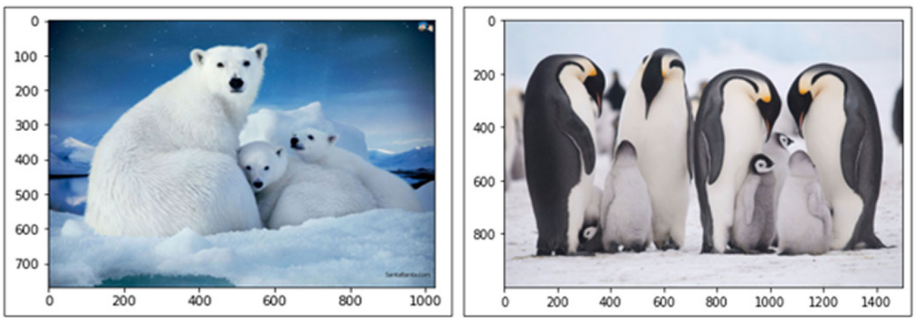

Fig. 4. Input images with real size

These images are passed to a condition to check their size, where each image that has any dimension greater than 224 is resized by downsizing to minimize their size to equal or less than 224 according to the ratio of the its width to height, this makes image manipulation is much faster, otherwise, the images are passed directly to the VGG-16 network model. Equation (2) shows the image downsizing.

$$I(m,n)=\begin{cases} I(x,y), & x \text{ or } y \le 224 \\ I(x',y'), & x \text{ or } y > 224 \end{cases} \tag{2}$$

Where $I(m, n)$ is the final size of the image before entering the VGG-16 model, $I(x, y)$ is the original size of the image, $I(x', y')$ is the new size of the image after downsizing. Where the proportion of the image's width to height can be preserved by applying equation (3).

$$\frac{x'}{y'}=\frac{x}{y}, \quad x',y' \le 224 \tag{3}$$

x', y' represent the new dimensions of the images, while x, y are the original dimensions.

Then normalization process is applied to the input images. Equation (4) shows how the normalization is computed pixel by pixel [27].

$$x_n=\frac{x_r-x_{min}}{x_{max}-x_{min}} \tag{4}$$

where x_r denotes the intensity value of a pixel, x_n is the normalized intensity value, x_{max} and x_{min} denotes the minimum and maximum intensity values of an image, respectively.

In the pre-trained model, we survey the layers inside the VGG-16 network and select specific layers to maximize the activations of these layers. We choose some layers from different blocks to maximize their activation functions. The process of choosing and maximizing the activations of layers is applied twice; at the first one, we applied it on the following layers from the indicated blocks: (block2_conv2, block3_conv1, block3_conv2). After that, we changed some of these layers and applied the same process. These layers are: (block4_conv2, block5_conv1, and block3_conv2).

After maximization of the loss of the targeted layers, we extract the most important features in the pre-trained model. The features in all CNN architectures are extracted by applying convolution, activation function, and pooling process. So, firstly we apply the convolution process in the convolutional layer. Activation function layers follow the convolutional layers, and max pooling layers extract the features from the network. The convolutional layer applies the convolution operation as in equation (5) to extract the features from the image [4].

$$F(i,j)=(A*K)(i,j)=\sum_m\sum_n A(i-m,j-n)K(m,n) \tag{5}$$

Where A represents input image, 2D filter of a size $m \times n$ is represented by K, and 2D feature map has been denoted by F. Here, an image A has been convolved with filter K and produces feature map F. This convolutional operation can be represented by $A * K$.

Each time we apply a convolution operation, it is followed by an activation function layer. Sometimes pooling layer may also be followed by an activation function. ReLU calculates the activation through the thresholding of input at zero. In other words, a rectified linear unit has output 0 in the case where the input is <0 and the raw output otherwise.

The activation function layer follows the convolutional layer; ReLU is one of the most popular activation functions, which is a hidden layer activation in a deep neural network [28]. It provides non-linearity to the algorithm and removes the negative values. ReLU is computed from equation (6).

$$f(x) = max\ (0, x) \tag{6}$$

Then pooling operation is applied. Here, we use the max pooling layer for the reduction of input's spatial size and thereby reduce the number of the parameters in the network. The layers of the Max pooling take the output of the feature map from convolutional layer and down-sample it. Max pooling only selects the strongest activation in the region of the pooling, where it chooses the maximum value in every one of the pooling regions and thereby can avoid the effects of the unwanted background features. Equation (7) shows how to compute max-pooling [29].

$$f_{max} = max\{x_i\}_{i=1}^{N} \tag{7}$$

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

Now we reach the deep dream stage. The gradient ascent that is concerned with the image is used to compute the loss and used to maximize the loss function; we compute the gradient ascent for the images by using equation (8), where it is known that the gradient ascent is the reverse of gradient descent [30], so, the equation of the gradient ascent is the same as gradient descent with reverse the sign, the gradient ascent equation is:

$$w = w + \mu \cdot \nabla E(W) \tag{8}$$

The gradients of the image pixels are added directly to the image. Then we detect the number of iterations we apply the deep dream to the image. So, the image is updated continuously until it reaches the previously detected iterations. So, the gradient ascent maximizes the loss function to excite the layers of the input images increasingly. In this study, the number of iterations is detected by 1000 iterations.

5 Results and discussion

This section explains the final results and losses, and the reasons that led to those results are mentioned.

At first, the input images are resized to minimize it by downsizing their dimensions to be equal or less than 224 for each dimension (width and height). Figure 5 shows the images after downsizing.

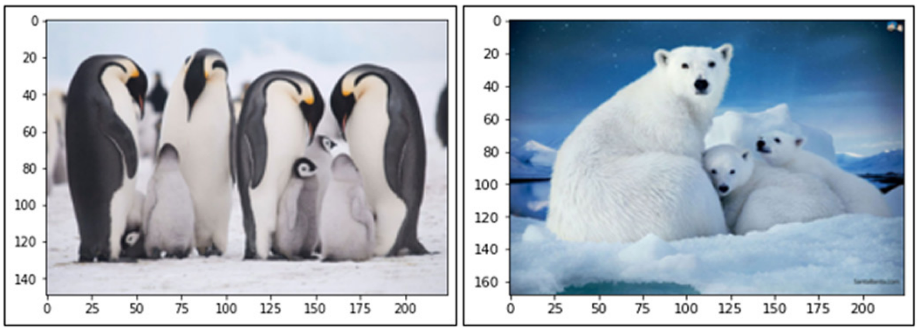

Fig. 5. The images after downsizing

The normalization process is applied to the resized images. Then, we review the existing layers, as shown in Table 1. Then firstly, we choose the detected layers from the required blocks (block2_conv2, block3_conv1, and block3_conv2). The result of the output deep dream images will be as in Figure 6, and the loss is illustrated in Table 2.

Fig. 6. The output images resulting from maximizing the layers (block2_conv2, block3_conv1, and block3_conv2)

Table 2. The loss over the steps during maximizing the layers (block2_conv2, block3_conv1, block3_conv2)

Image	Step	Loss
Bears	100	31.14353116181912
Bears	500	51.25360107421875
Bears	1000	53.97077941894531
Penguins	100	34.01990509033203
Penguins	500	51.80802917480469
Penguins	1000	54.61342620849609

When we choose to maximize the activation function of the upper layers as the layers (block4_conv2, block5_conv1, block3_conv2), we notice that the results are different in both loss and output deep dream images. Figure 7 shows the output of deep dream images when the detected upper layers are maximized.

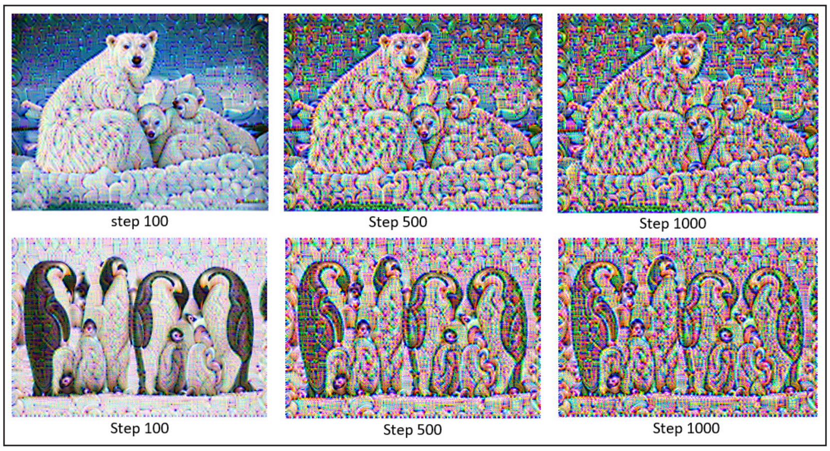

Fig. 7. The output images resulting from maximizing the layers (block4_conv2, block5_conv1, and block3_conv2)

While the loss of these images is illustrated in Table 3.

Table 3. The loss over the steps while maximizing the layers (block4_conv2, block5_conv1, and block3_conv2)

Image	Step	Loss
Bears	100	20.070411682128906
Bears	500	29.900951385498047
Bears	1000	31.95791244506836
Penguins	100	20.634506225585938
Penguins	500	30.03536605834961
Penguins	1000	32.1625862121582

6 Conclusion

This study presents a new method that contributes to generating different deep dream images from the same input image by changing the selected layers to maximize its loss function. Choosing the upper layers gives different results from the lower layers. The loss function for the same input image can take various values depending on the selected layers, where the loss value of the lower layers is greater than its value in the upper layers. Also, the resulting deep dream images depend on the number of iterations or steps we apply gradient ascent to maximize its loss function, where the images are changed each iteration and sound different.

7 References

[1] C. Rastelli, A. Greco, Y. N. Kenett, C. Finocchiaro, and N. De Pisapia, "Simulated visual hallucinations in virtual reality enhance cognitive flexibility," Nature Publishing Group UK, 2022, https://doi.org/10.1101/2021.12.14.472556

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

[3] A. Mordvintsev, C. Olah, and M. Tyka, "Inceptionism: Going deeper into neural networks," *Google Research Blog*, 2015. [Online]. Available: http://googleresearch.blogspot.co.uk/2015/06/inceptionism-going-deeper-into-neural.html [Accessed: 01-Dec-2022].

[4] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, "Basics of supervised deep learning," in *Advances in Deep Learning*, vol. 57, Springer Nature, 2020, pp. 13–30, https://doi.org/10.1007/978-981-13-6794-6_2

[5] B. K. O. C. Alwawi and A. F. Y. Althabhawee, "Towards more accurate and efficient human iris recognition model using deep learning technology," *TELKOMNIKA Telecommun. Comput. Electron. Control*, vol. 20, no. 4, pp. 817–824, 2022, https://doi.org/10.12928/telkomnika.v20i4.23759

[6] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization techniques in training DNNs: Methodology, analysis and application," 2020, pp. 1–20. arxiv.org/abs/2009.12836.

[7] G. McCaig, S. DiPaola, and L. Gabora, "Deep convolutional networks as models of generalization and blending within visual creativity," in *Proceedings of the 7th International Conference on Computational Creativity, ICCC 2016*, 2016, pp. 156–163.

[8] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9, https://doi.org/10.1002/jctb.4820

[9] H. Yin *et al.*, "Dreaming to distill: Data-free knowledge transfer via deepinversion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8715–8724, https://doi.org/10.1109/CVPR42600.2020.00874

[10] 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, https://doi.org/10.21276/ijircst.2020.8.4.13

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

[12] O. M. Al-Hazaimeh, M. Al-Nawashi, and M. Saraee, "Geometrical-based approach for robust human image detection," *Multimed. Tools Appl.*, vol. 78, no. 6, pp. 7029–7053, 2019, https://doi.org/10.1007/s11042-018-6401-y

[13] M. Al-Nawashi, O. M. Al-Hazaimeh, and M. Saraee, "A novel framework for intelligent surveillance system based on abnormal human activity detection in academic environments," *Neural Comput. Appl.*, vol. 28, no. s1, pp. 565–572, 2017, https://doi.org/10.1007/s00521-016-2363-z

[14] A. Aboulmira, H. Hrimech, and M. Lachgar, "Comparative study of multiple CNN models for classification of 23 skin diseases," *Int. J. online Biomed. Eng.*, vol. 18, no. 11, pp. 127–142, 2022, https://doi.org/10.3991/ijoe.v18i11.32517

[15] M. Alghieth, "Skin disease detection for kids at school using deep learning techniques," *Int. J. online Biomed. Eng.*, vol. 18, no. 10, pp. 114–128, 2022, https://doi.org/10.3991/ijoe.v18i10.31879

[16] F. M. Bachay and M. H. Abdulameer, "Hybrid deep learning model based on autoencoder and CNN for palmprint authentication," *Int. J. Intell. Eng. Syst.*, vol. 15, no. 3, pp. 488–499, 2022, https://doi.org/10.22266/ijies2022.0630.41

[17] R. Lateef and A. R. 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, 2022, https://doi.org/10.30684/etj.v40i4.2054

[18] A. Z. Mohammed and L. E. George, "Osteoporosis detection using convolutional neural network based on dual-energy X-ray absorptiometry images," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 29, no. 1, p. 315, 2022, https://doi.org/10.11591/ijeecs.v29.i1.pp315-321

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

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

[21] W. M. Salih Abedi, I. Nadher, and A. T. Sadiq, "Modification of deep learning technique for face expressions and body postures recognitions," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 3 Special Issue, pp. 313–320, 2020.

[22] W. J. Jameel, S. M. Kadhem, and A. R. Abbas, "Detecting deepfakes with deep learning and Gabor filters," *Aro-the Sci. J. Koya Univ.*, vol. 10, no. 1, pp. 18–22, 2022, https://doi.org/10.14500/aro.10917

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations. San Diego, CA, USA.*, 2015.

[24] T. Ameen, L. Chen, Z. Xu, D. Lyu, and H. Shi, "A convolutional neural network and matrix factorization-based travel location recommendation method using community-contributed geotagged photos," *ISPRS Int. J. Geo-Information*, vol. 9, no. 8, 2020, https://doi.org/10.3390/ijgi9080464

[25] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, https://doi.org/10.1007/s11263-015-0816-y

[26] 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, 2022, https://doi.org/10.3991/ijoe.v18i15.35761

[27] Z. Yin, B. Wan, F. Yuan, X. Xia, and J. Shi, "A deep normalization and convolutional neural network for image smoke detection," *IEEE Access*, vol. 5, pp. 18429–18438, 2017, https://doi.org/10.1109/ACCESS.2017.2747399

[28] Z. Mohammed and S. Kadhem, "A study about e-commerce based on customer behaviors," *Eng. Technol. J.*, vol. 39, no. 7, pp. 1060–1068, 2021, https://doi.org/10.30684/etj.v39i7.1631

[29] 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, 2022, https://doi.org/10.1007/s00521-022-06953-8

[30] H. S. Seung, "Convergence of gradient descent-ascent analyzed as a Newtonian dynamical system with dissipation," pp. 1–13. arxiv.org/abs/1903.02536, 2019.

8 Authors

Lafta R. Al-Khazraji received a BSc degree in computer science from Al Rafidain University College, and he received the degree of higher diploma in Information Technology and an MSc degree in Computer Science in 2013 and 2018, respectively from Iraqi Commission for Computers & Informatics, Iraq. Now, he is a Ph.D. candidate in the department of computer science, at the University of Technology, Iraq. His research interests are in Artificial Intelligence, Image Processing, Pattern Recognition, and Computer Networks.

Ayad R. Abbas is a professor at the Computer Science Department, University of Technology, Iraq. He has a PhD in Applied Computer Technology in 2009 from Wuhan University, School of Computer Science, China. Ayad R. Abbas received BSc and MSc degrees in computer science in 2003 and 2005, respectively, from the University of Technology, Computer Science Department, Iraq. 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 MSc. and PhD. in Computer Science from University of Technology, Iraq, 2004 and 2015 respectively. She has around 25 years of teaching experience, 12 years' instructor in Cisco Network Academic (CISCO), and Instructor AWS Academic. Her areas of interests are Digital Image Processing, Video Processing, Security software Engineering, Networking and artificial intelligence applications. She can be contacted at email: abeer.salim@muc.edu.iq.

Article submitted 2022-12-06. Resubmitted 2023-01-11. Final acceptance 2023-01-13. Final version published as submitted by the authors.