ISSN: 0067-2904

Generating Various Deep Dream Images Through Maximizing the Loss Function of Particular Layers Using Inception-v3 and Inception-ResNet-V2 Models

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*

Received: 25/1/2023 Accepted: 5/6/2023 Published: 30/6/2024

Abstract

Recently, Deep Learning (DL) has been used in a new technology known as the Deep Dream (DD) to produce images that resemble dreams. It is utilized to mimic hallucinations that drug users or people with schizophrenia experience. Additionally, DD is sometimes incorporated into the images as decoration. This study produces DD images using two deep-CNN model architectures (Inception-ResNet-V2 and Inception-v3). It starts by choosing particular layers in each model (from both lower and upper layers) to maximize their activation function, then detect several iterations. In each iteration, the gradient is computed and then used to compute loss and present the resulting images. Finally, the total loss is presented, and the final deep dream image is visualized. The output of the two models is different, and even for the same model there are some variations, the lower layers' loss values in the Inception-v3 model are significantly higher in comparison to the upper levels' values. In the case of Inception-ResNet-V2, the loss values are convergent.

Keywords: Deep dream; Inception-v3; Inception-ResNet-V2; gradient ascent; Convolutional neural network (CNN)

إنشاء العديد من صور الأحلام العميقة من خلال تعظيم وظيفة الخسارة لطبقات معينة باستخدام طرازي Inception-ResNet-V2 و Inception-v3

لفتة رحيم علي الخزرجي[1،2]*، اياد روضان عباس[1] ، عبير سالم جميل[3]

[1] قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق
[2] المديرية العامة لتربية صلاح الدين، العراق
[3] قسم هندسة تقنيات الحاسوب، كلية المنصور الجامعة، بغداد، العراق

الخلاصة

في الآونة الأخيرة ، تم استخدام التعلم العميق في تقنية جديدة تُعرف باسم الحلم العميق لإنتاج صور تشبه الأحلام. يتم استخدامه لمحاكاة الهلوسة التي يعاني منها متعاطو المخدرات أو الأشخاص المصابون بالفصام. يتم أحيانًا دمج الحلم العميق في الصور لتزيينها وزخرفتها. تنتج هذه الدراسة صور الحلم العميق باستخدام معماريتين نموذجيتين عميقتين لـ CNN هما (Inception-ResNet-V2 و Inception-v3). يبدأ

[*] Email: l.alkhazraji@gmail.com

الموديل باختيار طبقات معينة في كل نموذج (من كلا الطبقتين السفلية والعليا) ويقوم بتعظيم وظيفة التنشيط لكل طبقة تم اختيارها، بعد ذلك يتم تحديد عدد من التكرارات، وفي كل تكرار يتم حساب قيمة التدرج والذي بدوره يستخدم لحساب الخسارة، ثم يعرض الصور الناتجة. أخيرًا ، يتم عرض الخسارة الكاملة والصورة النهائية للحلم العميق. يختلف ناتج النموذجين ، بل وحتى بالنسبة لنفس النموذج ، توجد بعض الاختلافات ، حيث تكون قيم خسارة الطبقات السفلية في نموذج Inception-v3 أعلى بشكل ملحوظ مقارنةً بقيم المستويات العليا. اما في حالة Inception-ResNet-V2 ، تكون قيم الخسارة متقاربة.

1. Introduction

Deep learning is a subfield of machine learning that uses visual data [1]–[3]. The DD, a method that aims to enhance the visual qualities of images, was created as the newest DL technology by Google [4]. The CNN model receives repeated feeds of the image, which are used to recreate the DD. The features represent the most important information in the image [5]. So, the first layer begins by detecting the low-level features, such as lines and edges. Then, advanced characteristics like faces and trees visible. All of these features are eventually compiled for use in configuring multiple effects, like trees or the entire structure [6]. The evaluation of whether a CNN has appropriately learned the correct image features involved DD visualization. Due to the network's expanding image stream, in which the earliest layers oversee detecting the first low-level features (put differently, the edges), DD was produced. Then, higher-level features (such as faces and trees) that delve further into the network start to show up. Lastly, to configure the combined effects (for instance, the trees or whole structures), the uncommon final layers gather all of these [7].

In this study, two DD models are proposed based on the CNN architectures. Inception-v3 is used to build the former DD model, while Inception-ResNet-V2 is used in the latter. In each model, the layers that are chosen to maximize their loss function are selected from two places: the lower and upper layers.

Normalization is also used, where it ensures some particular statistical characteristics of the transformed data [8]. The gradient ascent is used to maximize the loss function [9]. Thus, it is an optimization algorithm that iteratively moves in the path of the function's value's steepest increase to maximize the objective function. This is accomplished by calculating the gradient of the function with respect to its parameters and then modifying the parameters correspondingly. Gradient ascent, as opposed to gradient descent, which minimizes a function, looks for the maximum value by moving up the gradient in the direction where the increase is the biggest [10]. Then, the loss function in each layer is computed. After the number of iterations is met, the model produces the final deep dream image with the final loss.

2. Related Works

Graeme McCaig et al. (2016) [4] suggested two DD algorithms that have been effective in reaching visual blending in CNNs. The first one was the Google Deep Dream, and the second was the algorithm suggested by (Gatys and his colleagues). The algorithm took random images as inputs, and then, they split and recombined its content and style by using NNs for the development of artistic images that are therefore indicated as the deep style DS. Researchers used Google's Deep Dream and VGG for DS techniques. The network was trained using the Cars and ImageNet data sets. As a result of their research on the topic of DD, they have identified two overused features that deserve to be acknowledged. First, a developed DD data as a bottom-up discrimination network, which is why Google Net ignores a large portion of the information on the tonic color of the regions while maintaining the color contrast around edges. The fact that the 1,000 labelled classes in ImageNet reflect a second section of the animal types, with a focus on precise dog breeds distinctions, results in a bias towards animal types in the training data. Thus, dog traits tend to arise when the patterns are treated as relevant to dog features. As opposed to this, the network trained on the car dataset produces the car features.

A unique technique for creating images from image distributions for deep neural distribution training was proposed by H. Yin et al. (2020) [11]. Their approach, known as the DeepInversion, consists of two components: student logits and teacher logits. The teacher is a representation of the inverted trained network that starts with random noise and is not using any DD extra training data. Through the enhancement of DD's image quality, by extending the image regularization with a new term of feature distribution regularization, they have used DD for developing Deep Inversion. their model was trained on the CIFAR-10 and ImageNet data sets.

A computer vision algorithm known as Deep Inceptionism learning, or the DD algorithm, was presented by T. J. Kiran (2021) [12]. When an image enters the network, causes the neurons to fire, and produce activations, the algorithm's training phase starts. The idea behind their suggested algorithm is to increase the firing of a few neurons by changing the input image (through the activation or boosting of the neurons). With the help of DD, users can decide which neurons where layers they're willing to fire more noticeably. Such a procedure might be repeatedly carried out until an input image has all the attributes needed by a specific layer. They continuously send these images into the network, and the more they send it into the network, the more they will be able to extract or see all those strange elements in the actual image. Because of this, the first stage of their algorithm is to submit an image to a trained CNN, ResNet, ANN, etc. Following that, a layer is selected and the activations (or output) that are generated by the layer of interest are specified (the top layer captures the edges, whilst the deeper ones capture the full shapes like faces). The next steps involve calculating the activation gradient for the input image, modifying the image to increase those activations, enhancing network-detected patterns that produce a trippy hallucinatory image, and repeating the process continually across various sizes. Gradient ascent was used to maximize the loss function at each layer. After deeply dreaming up all losses from each layer, they passed them all, which was the same parameter plotted or printed and returned from the gradient ascent function. By using their dream algorithm and the gradient ascent, they may improve their output.

A DD model was also presented by Lafta R. Al-Khazraji et al. (2022) [9]. It applies the DD algorithm after receiving the output of NST images as input. To implement the NST depending on the Gram matrix, which serves as the model's core, and gradient descent, which reduces the loss function in the case when the image is cleared and increases with distorted images, we employ the VGG-19 pre-trained network as a deep CNN network. The DD was created using a pre-trained model called the Inception-v3 network. Using gradient ascent, the loss is highest in this situation. The loss of the dreamt images, such as the circular and spiral stylized images, is lower than the loss of the clear images (3D squares and triangles images).

Al-Khazraji et al. (2023) [10]developed a deep dream model that generates deep dream images by using CNNs. They used VGG-16 pre-trained network to build their model. The method includes selecting certain network layers, extracting features, and optimizing them through repeated methods. They applied this technique on both low-level and high-level layers. The authors noticed that images formed with high-level layers were clearer than those generated with low-level layers. Furthermore, they reported that the variation in the loss values for low-level layers is smaller than that for upper-level layers

.

3. Convolutional Neural Network (CNN)

Deep neural models known as CNNs were developed specifically to handle image data, and they can be thought of as feed-forward ANNs with various convolutional and subsampling layers [13]–[18]. It processes incoming data and extracts relevant features by using convolutional, pooling, and fully connected layers. The architecture's hierarchical design makes it capable of learning intricate patterns, which makes it perfect for applications like self-driving cars, facial recognition, and object identification [19]–[24].

3.1 Inception-v3

Inception V3 is a deep CNN architecture developed by Google. A total of 11 Inception modules totaling five different types make up the network. Each module has an activation layer, a convolutional layer, a batch normalization layer, and a pooling layer that are all created by professionals. This network is more precise and computationally efficient compared to the previous Inception models since it employs convolution processes in each inception module to reduce the grid size. In addition to employing numerous parallel of both convolutional and pooling layers to capture a range of features at various scales [25]–[27]. Figure 1 shows a compressed view of the InceptionV3 model.

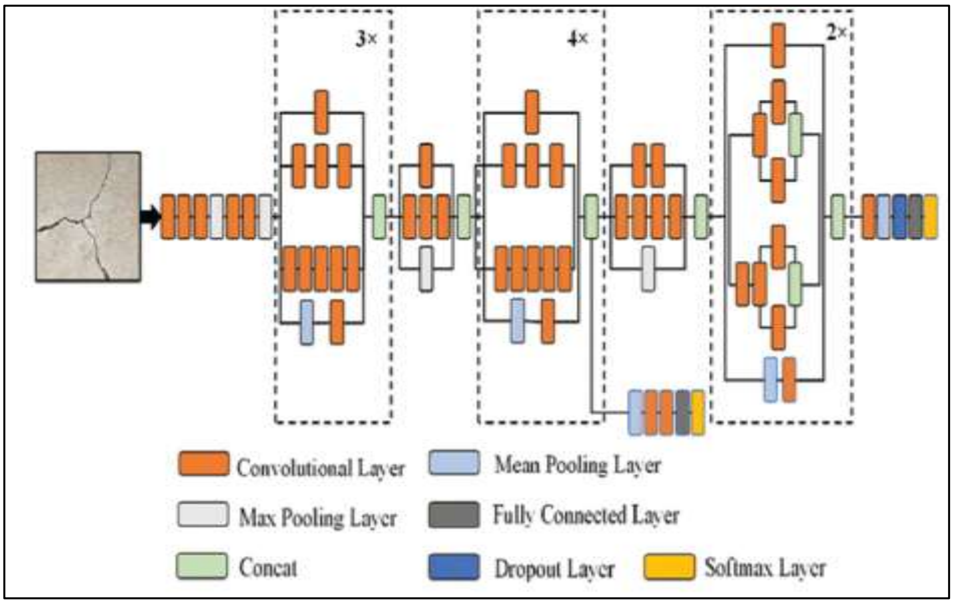

Figure 1: The diagram of a compressed view of the InceptionV3 model [9].

Table 1 shows the Inception-v3 network structure model.

Table 1: Network structure of the Inception-v3 model [28].

Type	Patch Size/ Stride	Input Size
Conv	$3 \times 3/2$	$299 \times 299 \times 3$
Conv	$3 \times 3/1$	$149 \times 149 \times 32$
Conv	$3 \times 3/1$	$147 \times 147 \times 32$
Pool	$3 \times 3/2$	$147 \times 147 \times 64$
Conv	$3 \times 3/1$	$7373 \times 73 \times 64$
Conv	$3 \times 3/2$	$71 \times 71 \times 80$
Conv	$3 \times 3/1$	$35 \times 35 \times 192$
$3 \times Inception$	-------	$35 \times 35 \times 288$
$5 \times Inception$	-------	$17 \times 17 \times 768$
$2 \times Inception$	-------	$8 \times 8 \times 1280$
Pool	8×8	$8 \times 8 \times 2048$
Linear	logits	$1 \times 1 \times 2048$
SoftMax	classifier	$1 \times 1 \times 1000$

3.2 Inception-ResNet-V2

The two most popular deep CNNs, Inception and ResNet are combined to create this network, although batch normalization is just applied on top of the standard layers and not the summations. For enhancing the number of Inception blocks and, thus, the network depth, the residual modules are specifically used [29]. The Inception-ResNet-V2 architecture is depicted in Figure 2.

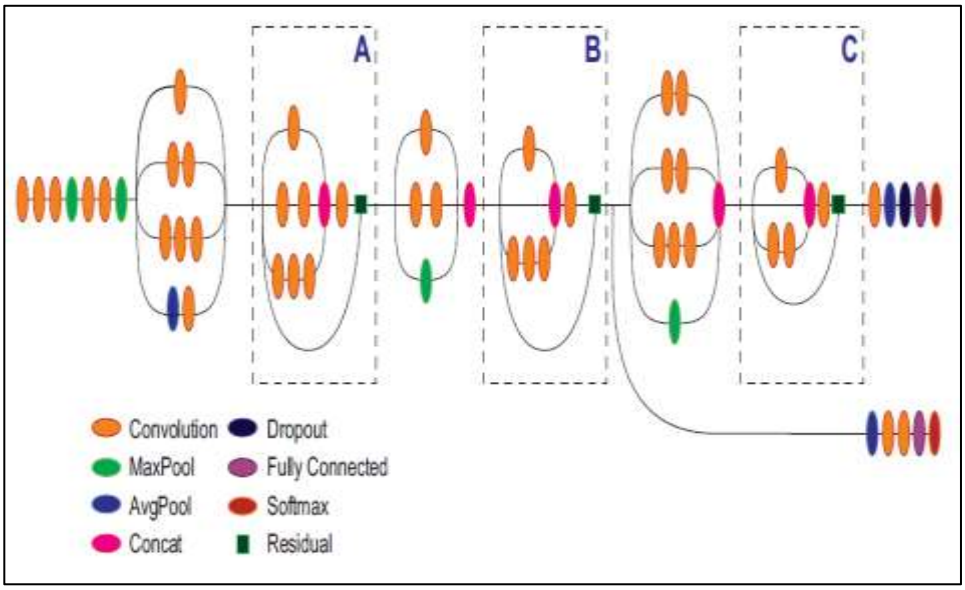

Figure 2: Inception-ResNet-V2 [30].

Figure 3, [31] illustrates with more detail the blocks inside this architecture, where there are several operations in each block.

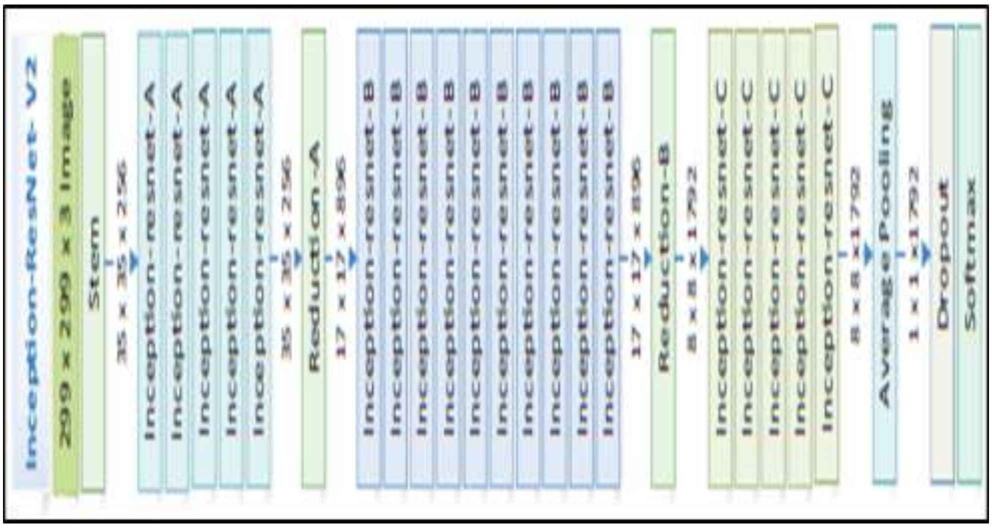

Figure 3: Inception-ResNet-V2 architecture blocks [31].

4. Research Methodology

In this study, a comparison between two pre-trained models was applied to generate different deep dream images according those models. These models are Inception-v3 and Inception-ResNet-V2. Figure 4 shows the diagram of our model.

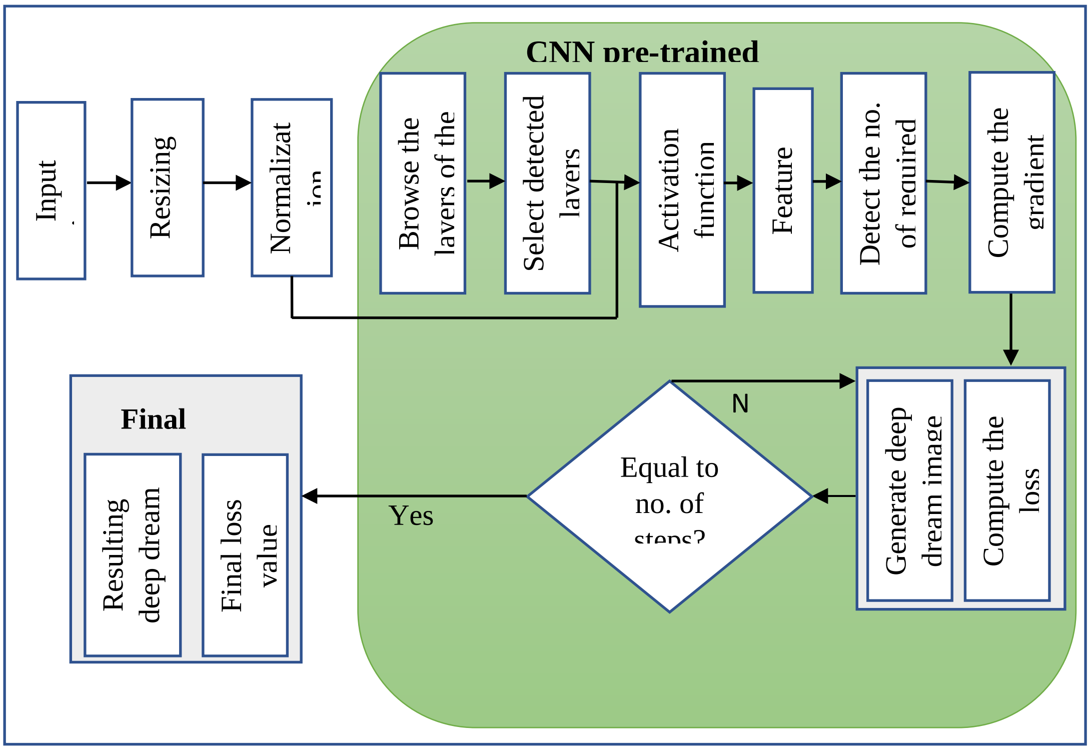

Figure 4: Diagram of the proposed model.

Firstly, we input two images, which are a three-bear image and the penguins' image. Figure 5 shows the input images.

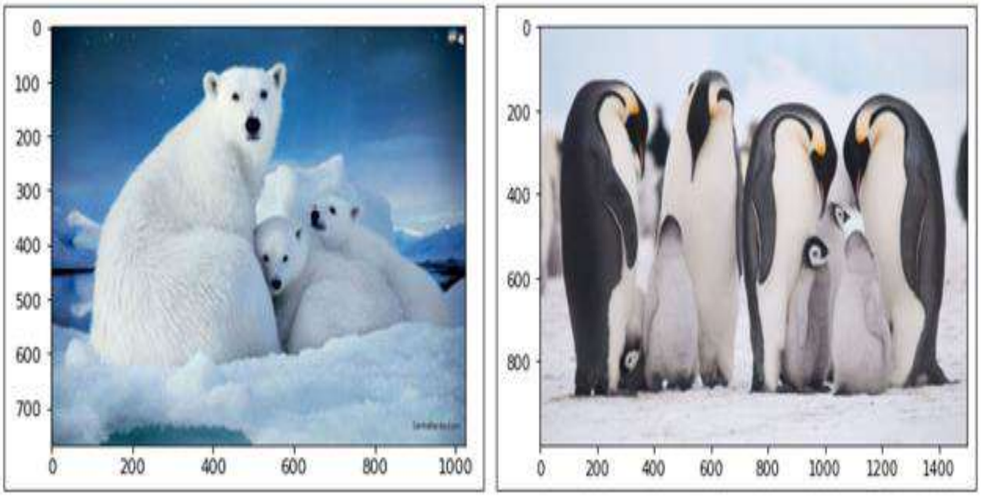

Figure 5: Real-sized input image.

The second step is minimizing their size to accelerate the processing time. Then applying the normalization process. The normalization process is calculated for each pixel by applying equation (1) [32].

$$x_n = \frac{x_r - x_{min}}{x_{max} - x_{min}} \tag{1}$$

where x_r indicates a pixel's intensity value, x_n is the normalized value of x_r, x_{max} and x_{min} are the image's lowest and highest intensity values, respectively.
Then, two CNN pre-training models are used; Inception-v3 and Inception-ResNet-V2, one model each time.

In the case of using Inception-v3, firstly, we survey the layers of the model, then choose the targeted layers for maximizing their loss functions; these layers belong to different blocks. The layered targeting process takes place in two stages: at the first stage we choose blocks and target layers from the low-level layers, which are (mixed0, mixed1, and mixed2), while at the second the layers (mixed8, mixed7, and mixed6) are chosen to maximize their loss functions.
When Inception-ResNet-V2 is used, the same processes are followed. There is only one difference, which is the names of the layers, as the blocks and layers in this model are not the same as in the Inception-v3 model. The process of choosing the targeted layers to maximize them is very important since the selected layers have major effects on the visualized images, since each layer have a unique effect. Maximizing the activation function of the lower

The next step for both models is extracting the important features from the targeted layers. This is achieved by applying some convolution processes, each one followed by an activation function process rectified linear unit (ReLU) is applied. Then one or more max pooling processes are applied. This process is continued to extract more features.

Afterwards, the gradient ascent is computed, and while the gradient descent works to minimize the loss function [33], the gradient ascent is the complete opposite; it works to maximize the loss function. Therefore, the gradient ascent is computed by applying equation 2.

$$w = w + \mu . \nabla E(W) \tag{2}$$

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

For the image pixels, their gradients are added to the image immediately, suppose the required iterations, and apply the DD to the image. As long as the required iterations are not met, the image will be updated. The gradient ascent maximizes the loss function to extract the layers of the input images increasingly. In this work, the number of iterations is 1000.
Lastly, the final deep dream image is generated with the final loss, and this is the final output.

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. Figure 6 shows the images after downsizing.

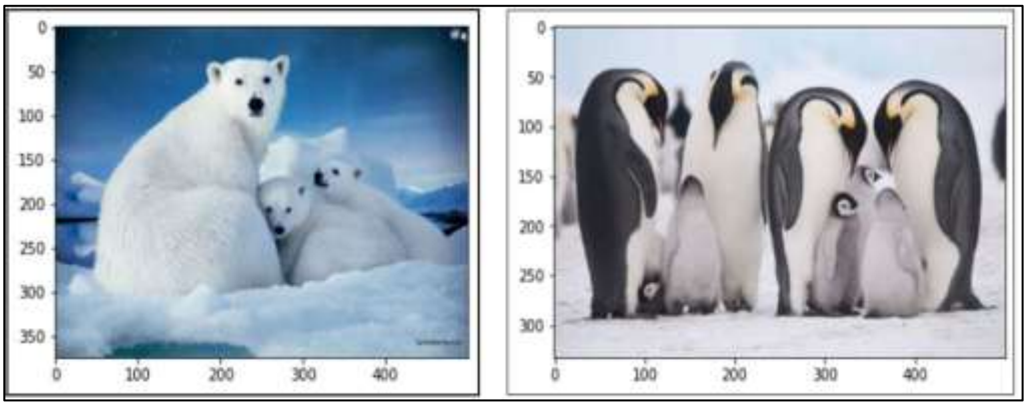

Figure 6: The downsized images.

The normalization technique is used for resizing images; normally, it is used to change an input image into a set of pixel values that are more normal or familiar to the senses.

5.1 Inception-v3 Model

We apply the Inception-v3 model, where we review its blocks and layers. Firstly, we choose the detected layers from the required blocks mixed0, mixed1, and mixed2. The result of the output deep dream images is in Figure 7. The loss is illustrated in Figures 8 and 9, which ranged between (2.720845222 - 5.53691101) for the three bears' image, and (3.17698955535888 - 6.7354884147644) for the penguins' image.

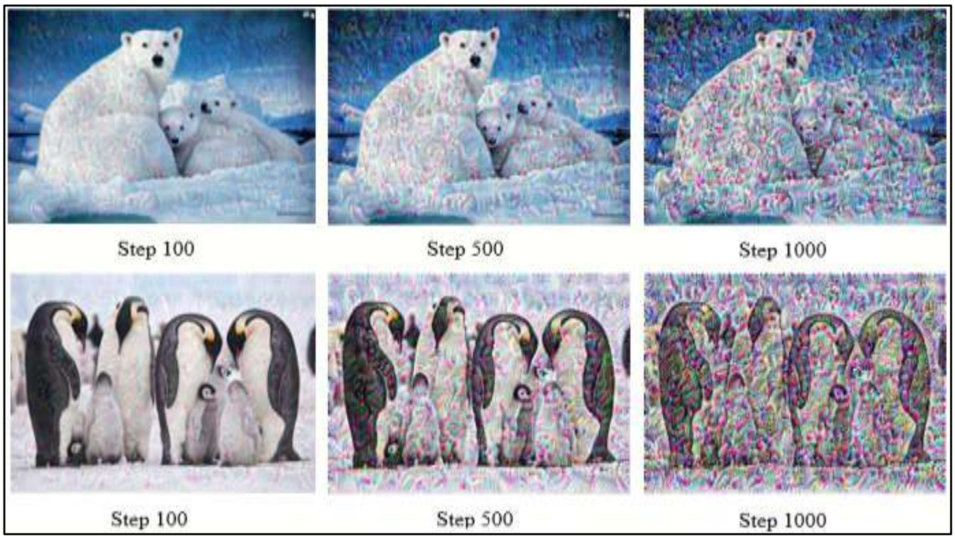

Figure 7: The output images resulting from maximizing the layers (mixed0, mixed1, and mixed2).

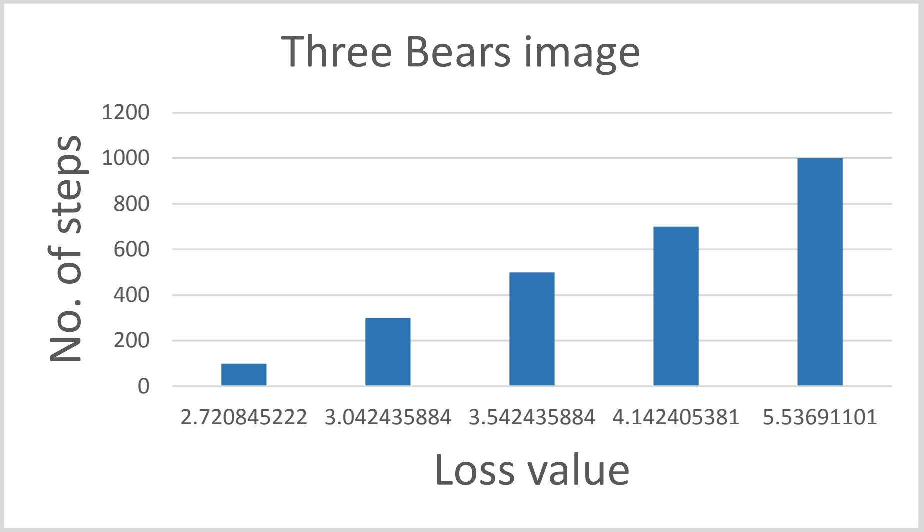

Figure 8: The loss values of three bears image when maximizing the loss function of mixed0, mixed1, and mixed2.

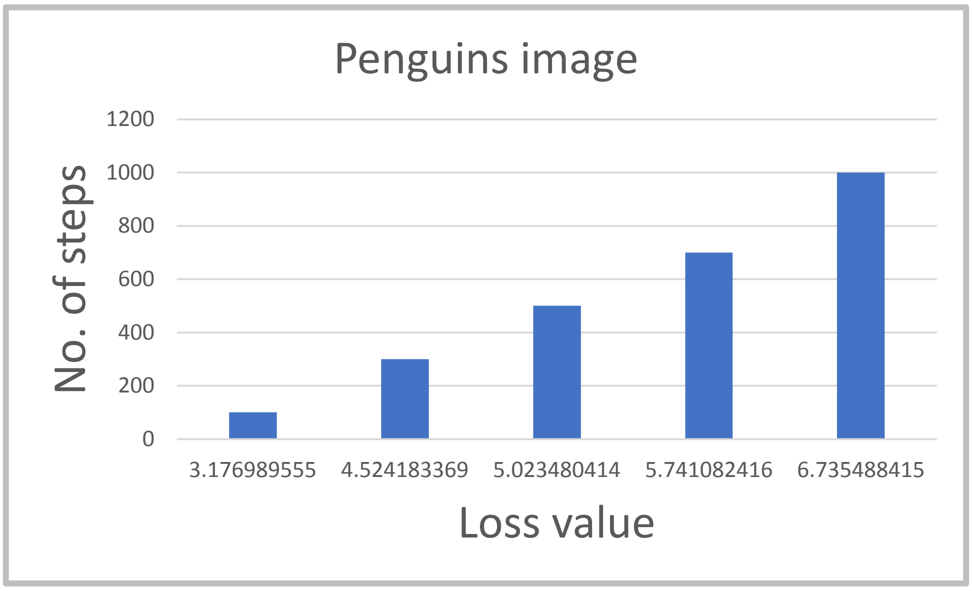

Figure 9: The loss values of penguins' image when maximizing the loss function of mixed0, mixed1, and mixed2.

Lastly, we also continue with the Inception-v3 model and choose the high-level layers to maximize their loss function, these layers are mixed8, mixed9, and mixed10. The result is deep dream images that are shown in Figure 10. And the loss values are in Figures 11 and 12, which are ranged between (1.7353308200836 - 3.5670118331909) for the three bears image, and (1.85638332366943 - 3.48054218292236) for penguins' image.

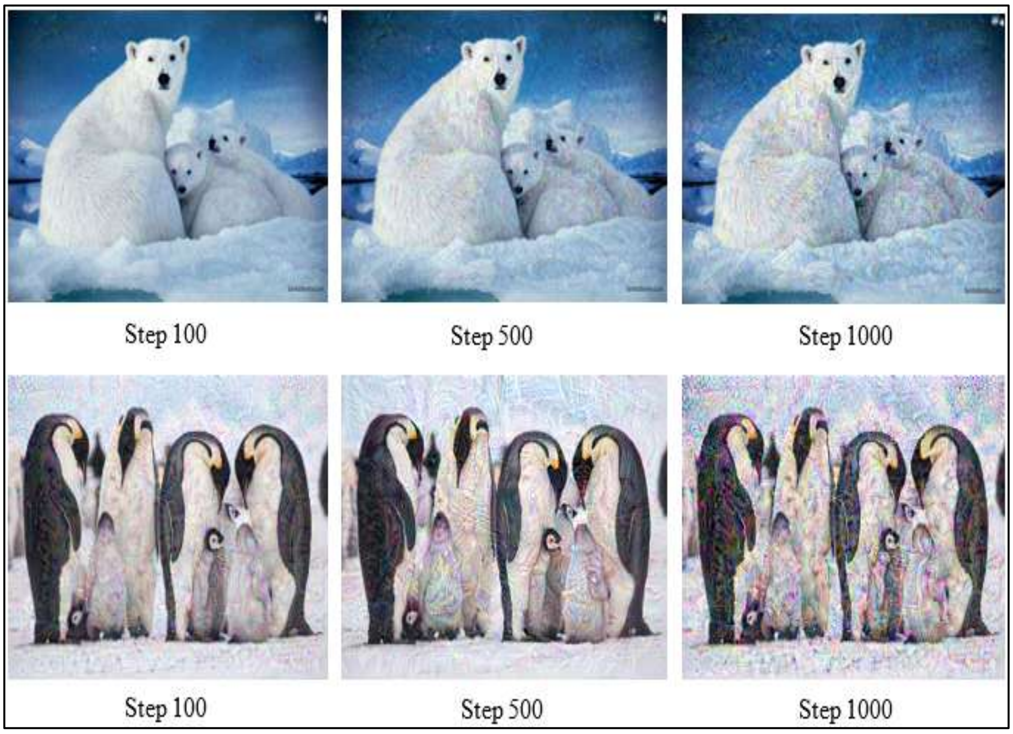

Figure 10: The loss values of the penguins' image when maximizing the loss function of mixed6, mixed7, and mixed8.

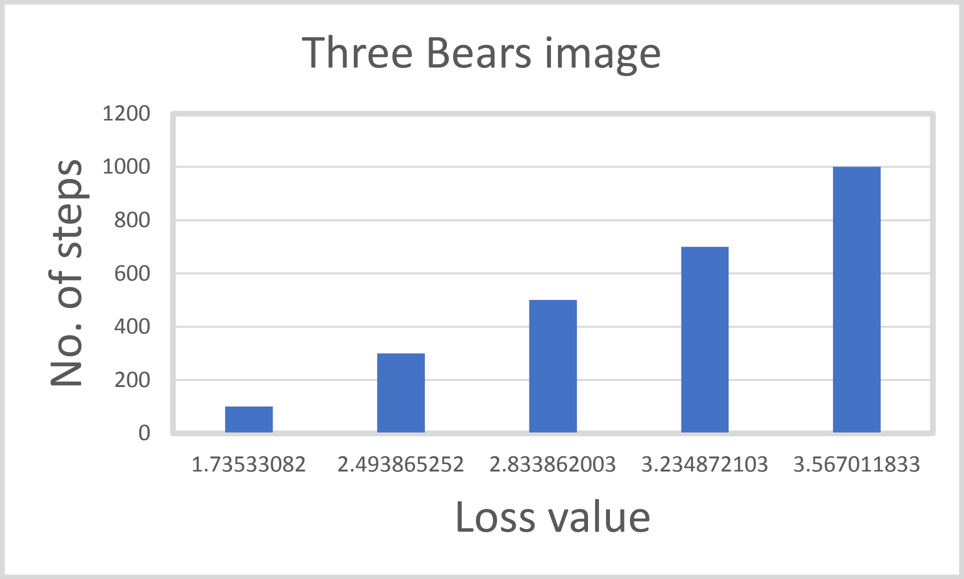

Figure 11: The loss values of the three bears' image when maximizing the loss function of mixed6, mixed7, and mixed8.

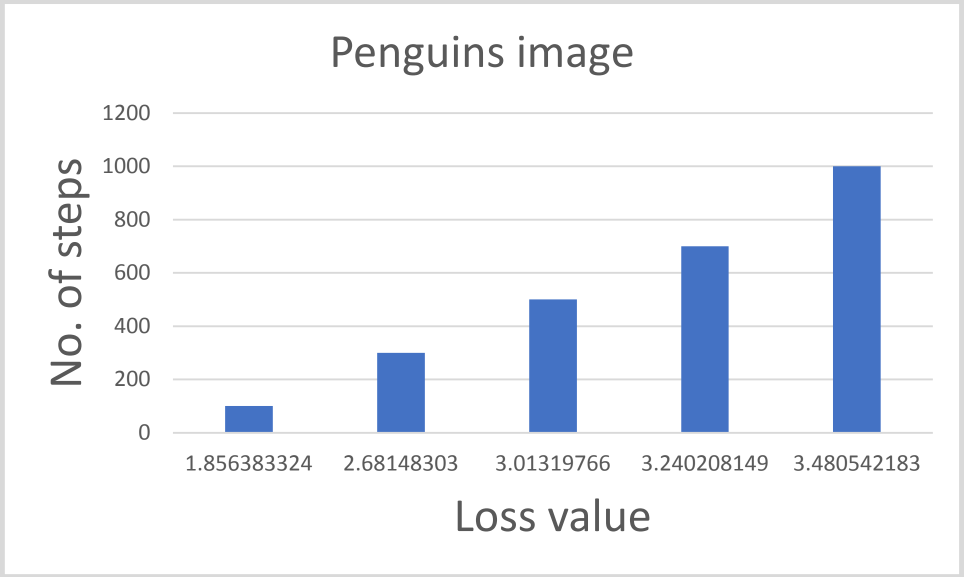

Figure 12: The loss values of the penguins' image when maximizing the loss function of mixed6, mixed7, and mixed8.

5.2 Inception-ResNet-V2

With this model, the maximization of the loss function is applied two times; the former is on the layers (block35_2_mixed and block35_4_mixed). While the latter is on the layers (block35_5_mixed, block35_6_mixed, block35_8_mixed, and block35_9_mixed).

Firstly, we choose to maximize the loss function of the layers (block35_2_mixed and block35_4_mixed). The result is the deep dream images that are shown in Figure 13. The loss values are in Figures 14 and 15, which ranged between (1.1835618019104 - 2.51987242698669) for the three bears image, and (1.85638332366943 - 3.48054218292236) for the penguins' image.

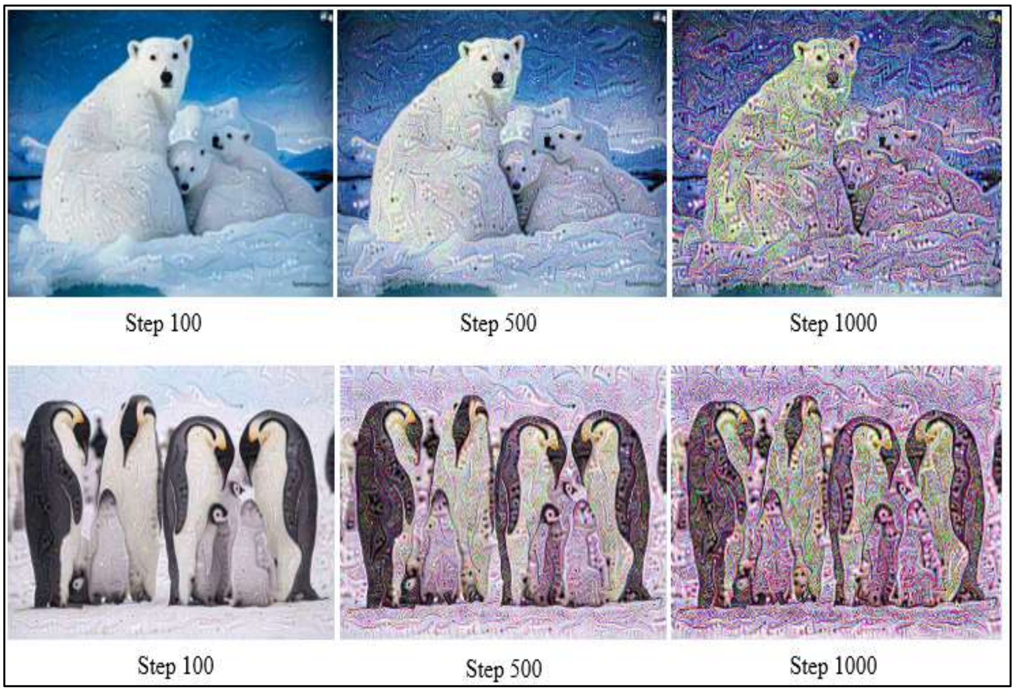

Figure 13: The output images resulting from maximizing the layers (block35_2_mixed and block35_4_mixed).

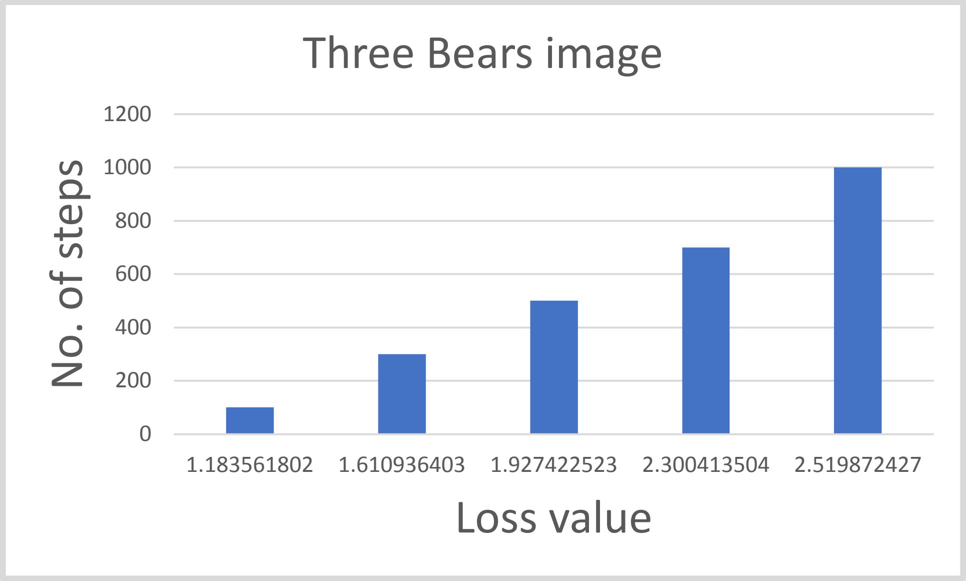

Figure 14: The loss values of the three bears' image when maximizing the loss function of block35_2_mixed and block35_4_mixed.

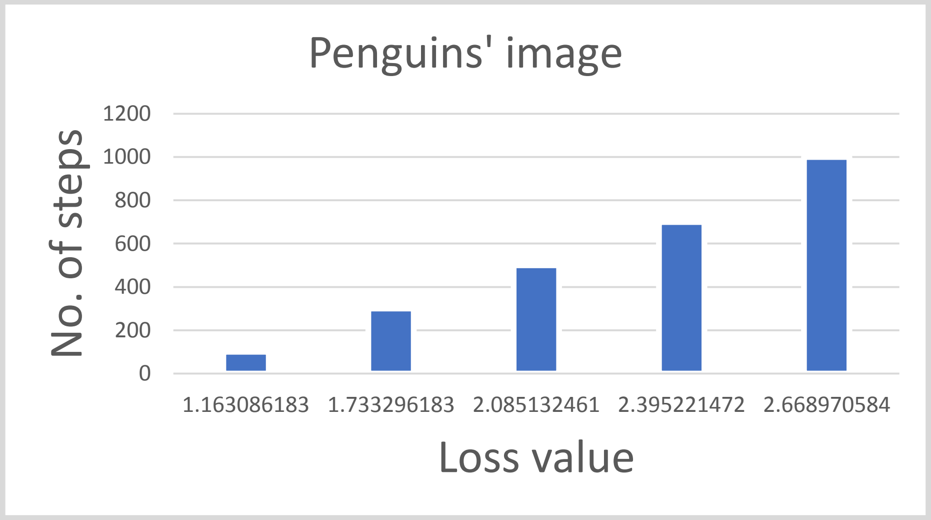

Figure 15: The loss values of the penguins' image when maximizing the loss function of block35_2_mixed and block35_4_mixed.

Then, with the layers (block35_5_mixed, block35_6_mixed, block35_8_mixed, and block35_9_mixed).
The result is deep dream images that are shown in Figure 16. And the loss values are in Figures 17 and 18, which are ranged between (1.7127691507339478- 2.7816869354248047) for the three bears' image, and (1.7327109575271606- 2.651268243789673) for the penguins' image.

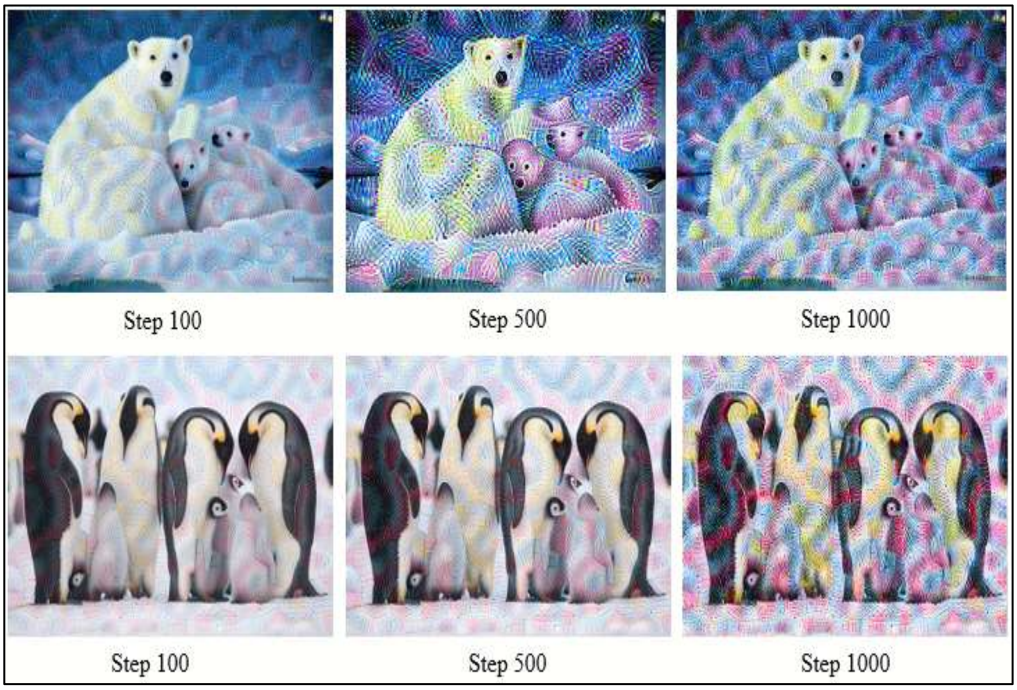

Figure 16: The output images resulting from maximizing the layers (block35_5_mixed, block35_6_mixed, block35_8_mixed, and block35_9_mixed).

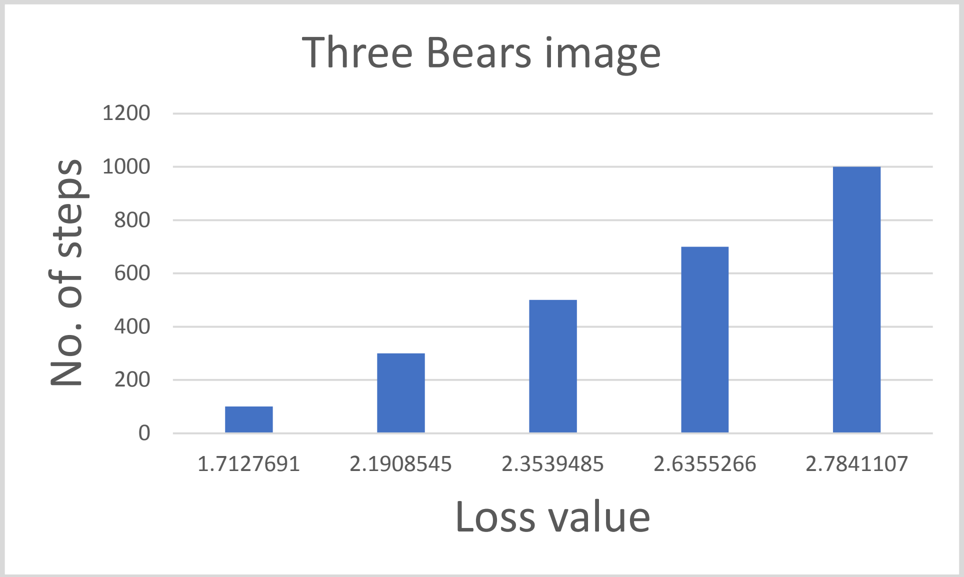

Figure 17: The loss values of the three bears' image when maximizing the loss function of the layers (block35_5_mixed, block35_6_mixed, block35_8_mixed, and block35_9_mixed).

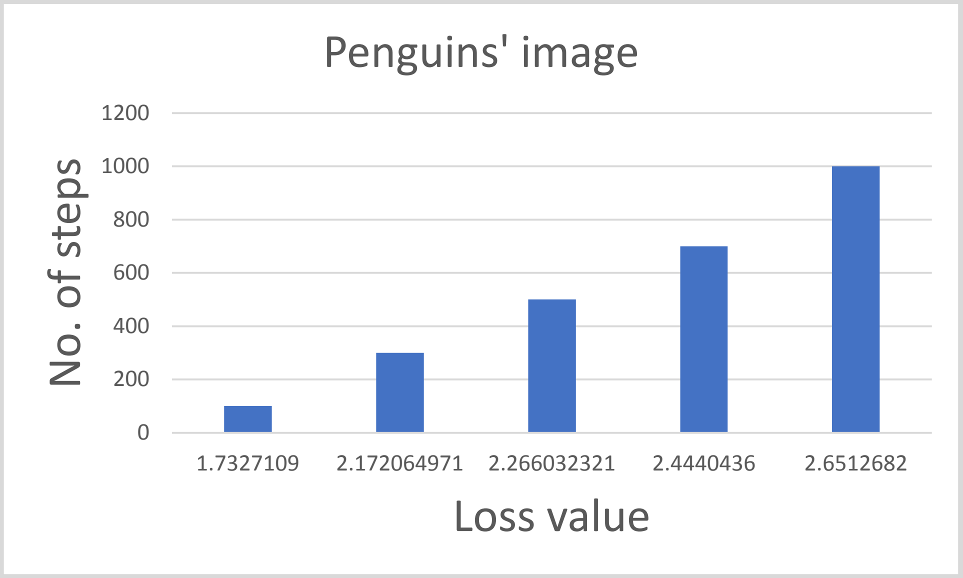

Figure 18: The loss values of the penguins' image when maximizing the loss function of the layers (block35_5_mixed, block35_6_mixed, block35_8_mixed, and block35_9_mixed).

From the above results, it is clear that maximization of the loss function of low-level layers has a higher effect on the loss value than higher layers when using the Inception-v3 model. When using Inception ResNet-V2 this role is changed, where the image of three bears has a higher loss value in the higher layers and this is the opposite in the case of the penguins' image.

Also, Inception-v3 model, the loss value of the penguins' image in most cases is higher than its value in the three-bear image because there are different colors used with the penguins' image. This means that more colors have a higher loss value.

6. Conclusion

Deep Dream is a very recent and promising approach for generating and visualizing images. This research introduces a new method for generating deepdream images based on Inception-v3 and Inception-ResNet-V2 models. The resulting deep dreamed images of these two models are compared depending on their loss values. The activation function of each model is maximized at two different places; the first place is in the upper layers and the other for the lower ones. The resulting deep dream images and the loss value that results from each image are different. In the case of the Inception-v3 model, the loss values of the lower layers are much higher than the values in the upper layers. Whereas their loss values are convergent in the case of Inception-ResNet-V2. The results are obtained after 1000 iterations of applying gradient ascent. The selected layers have a major effect on both the generated images and the resulting loss values.

7. Acknowledgements

This study is supported by the Computer Science Department, University of Technology, Baghdad, Iraq.

References

[1] L. R. Ali, B. N. Shaker, and S. A. Jebur, "An extensive study of sentiment analysis: A survey," in *AIP Conference Proceedings 2591*, 2023, p. 030022, doi: 10.1063/5.0119604.

[2] Zahraa Ch. Oleiwi, E. N. AlShemmary, and S. Al-Augby, "NEW TRENDS IN ARRHYTHMIA CLASSIFICATION BASED ON ARTIFICIAL INTELLIGENT TECHNIQUES :

ANALYTICAL STUDY," *NeuroQuantology*, vol. 20, no. 5, pp. 2083–2113, 2022, doi: 10.14704/nq.2022.20.5.NQ22572.

[3] S. S. Abdul-jabbar, A. K. Farhan, and A. S. Luchinin, "A Comparative Study of Anemia Classification Algorithms for International and Newly CBC Datasets," *Int. J. online Biomed. Eng.*, vol. 19, no. 06, pp. 141–157, 2023.

[4] A. Mordvintsev, C. Olah, M. Tyka, and E. Al., "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].

[5] L. R. Ali, H. K. Homood, and A. S. Elameer, "Feature Extraction Techniques on Facial Images : An Overview," *Int. J. Sci. Res.*, vol. 6, no. 9, pp. 2015–2018, 2017, doi: 10.21275/ART20176682.

[6] M. A. Wani, F. A. Bhat, Khan, S. Afzal, A. Iqbal, and E. Al., "Basics of supervised deep learning," in *Advances in Deep Learning*, vol. 57, Springer Nature., 2020, pp. 13–30.

[7] G. McCaig, S. DiPaola, L. Gabora, and E. Al., "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] 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.

[9] 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. 10, no. 2, pp. 134–141, 2022, doi: doi.org/10.14500/aro.11051.

[10] L. R. Al-Khazraji, A. R. Abbas, and A. S. Jamil, "The Effect of Changing Targeted Layers of The Deep Dream Technique Using VGG-16 Model," *Int. J. online Biomed. Eng.*, vol. 19, no. 3, pp. 34–47, 2022.

[11] 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, doi: 10.1109/CVPR42600.2020.00874.

[12] 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.

[13] 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, doi: 10.30684/etj.v40i4.2054.

[14] 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: https://doi.org/10.3390/electronics12010029.

[15] 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. Dev. Res.*, vol. 07, no. 11, pp. 16960–16967, 2017, doi: 10.13140/RG.2.2.31549.56805.

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

[17] W. J. Hadi, S. M. Kadhem, and A. R. Abbas, "A survey of deepfakes in terms of deep learning and multimedia forensics," *Int. J. Electr. Comput. Eng.*, vol. 12, no. 4, pp. 4408–4414, 2022, doi: 10.11591/ijece.v12i4.pp4408-4414.

[18] H. A. Ahmed and E. A. Mohammed, "Detection and Classification of The Osteoarthritis in Knee Joint Using Transfer Learning with Convolutional Neural Networks (CNNs)," *Iraqi J. Sci.*, vol. 63, no. 11, pp. 5058–5071, 2022, doi: 10.24996/ijs.2022.63.11.40.

[19] 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, doi: 10.14500/aro.10917.

[20] 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.

[21] 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, pp. 315–321, 2022, doi: 10.11591/ijeecs.v29.i1.pp315-321.

[22] Z. C. Oleiwi, E. N. AlShemmary, and S. Al-augby, "Efficient ECG Beats Classification

Techniques for The Cardiac Arrhythmia Detection Based on Wavelet Transformation," *Int. J. Intell. Eng. Syst.*, vol. 16, no. 2, pp. 192–203, 2023, doi: 10.22266/ijies2023.0430.16.

[23] W. M. S. Abedi, I. Nadher, A. T. Sadiq, and E. Al., "Modified deep learning method for body postures recognition," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 2, pp. 3830–3841, 2020.

[24] S. J. A. Al-Atroshi and A. M. Ali, "Facial Expression Recognition Based on Deep Learning: An Overview," *Iraqi J. Sci.*, vol. 64, no. 3, pp. 1401–1425, 2023, doi: 10.24996/ijs.2023.64.3.32.

[25] 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.

[26] S. Arumugaperumal, B. Sivagami, and K. Pazhanikumar, "Inception-V3 Architecture in Dermatoglyphics-Based Temperament Classification," *Philipp. Soc. Sci. J.*, vol. 3, no. 2, pp. 196–200, 2020, doi: 10.1109/ICECTECH.2011.5941886.

[27] M. A. S. Al Husaini, M. H. Habaebi, T. S. Gunawan, M. R. Islam, E. A. A. Elsheikh, and F. M. Suliman, "Thermal-based early breast cancer detection using inception V3, inception V4 and modified inception MV4," *Neural Comput. Appl.*, vol. 34, no. 1, pp. 333–348, 2022, doi: 10.1007/s00521-021-06372-1.

[28] J. Cao, M. Yan, Y. Jia, X. Tian, and Z. Zhang, "Application of a modified Inception-v3 model in the dynasty-based classification of ancient murals," *EURASIP J. Adv. Signal Process.*, vol. 2021, no. 1, pp. 1–25, 2021, doi: 10.1186/s13634-021-00740-8.

[29] 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.

[30] U. Nazir, N. Khurshid, M. A. Bhimra, and M. Taj, "Tiny-Inception-ResNet-v2: Using Deep Learning for Eliminating Bonded Labors of Brick Kilns in South Asia," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 39–43.

[31] A. Saber, M. Sakr, O. M. Abo-Seida, A. Keshk, and H. Chen, "A Novel Deep-Learning Model for Automatic Detection and Classification of Breast Cancer Using the Transfer-Learning Technique," *IEEE Access*, vol. 9, pp. 71194–71209, 2021, doi: 10.1109/ACCESS.2021.3079204.

[32] 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, doi: 10.1109/ACCESS.2017.2747399.

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