End Evaluation Report for SMAI

Visualizing and Understanding Convolutional Networks

Alefiah Mubeen (20172146)
Ankita Maity (2021701017)
Seshadri Mazumder (2021801002)
Vivek Pareek (2021701001)

COLAB NOTEBOOK LINKS:

Code for VGG-16
Code for VGG-19

December 6, 2021

Contents

1 Introduction

CNNs have been used for a wide variety of tasks, and using them leads to excellent performance in many benchmarks. However, there is little understanding of why these models perform as well as they do. The paper we implement provides visualization techniques to understand why large CNNs perform well and how to improve them. Multi-layer deconvnet provides a visualization technique that reveals the input stimuli that excite individual feature maps at any layer in the model. It allows us to observe the evolution of features during training and diagnose potential problems. Sensitivity analysis is performed by occluding portions of the image to understand which parts are important for classification. The ImageNet trained model is also shown to generalize well to other data sets.

Section 2 outlines the various objectives of our work. We plan to use two architectures from two related papers [1,2], as described in the methodology section 3. The experiments performed using various models like VGG-16 [3], VGG-19 [4] and Alexnet [5] and the experiments have been described in section 4, and the results of these experiments will be analysed. Section 7 gives details on the deliverables and the tentative timeline.

2 Problem Statement

In this project, we aim to visualise the internal working of the CNNs, and the features learned by the different layers of the architectures. We also augment the data with different variations and report the results. The different experiments to be performed are discussed in section 4.

2.1 Objectives

- First we will complete the two end-to-end pipelines discussed in Section 3
- The pipelines we will create for VGG-16 [3], VGG-19 [4] and Alexnet [5].
- As part of feature inversion reconstructing from different layers of VGG-16 [3], VGG-19 [4] and Alexnet [5].
- As part of occlusion, we will experiment by masking a part of the image before feeding it to the CNN, and we will visualise the features after inversion in the image space.
- We perform experiments with different data augmentation based on affine transformations like scaling, rotation, and translation to study feature invariance.

3 Architecture

This section discusses the architecture using which we will experiment with different variances and visualise the feature maps represented by the corresponding layers.

3.1 Architecture 1

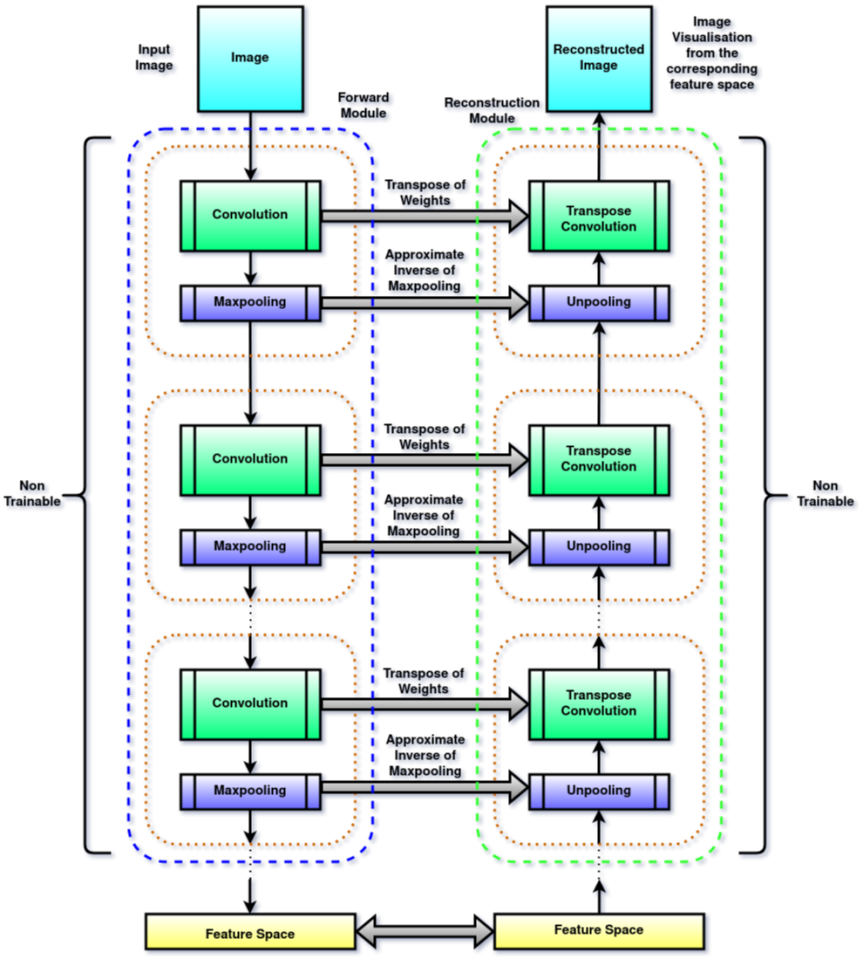

Figure 1: This architecture is the schematic representation of the end-to-end pipeline inspired from [2].

3.1.1 Description :

This architecture is for visualising the internal layers of the CNN. We proceed by feeding the input image through a series of convolution and max-pooling blocks. Ideally, these blocks will contain pretrained weights of either of VGG-16, VGG-19 and AlexNet models.

In the image the blue box encloses the modules that correspond to the conversion of an image to its corresponding feature space, and the green box has the modules that take the high dimensional feature space values as an input and bring it down to image space for the the purpose of visualisation. The brown dotted modules inside the blue and green are opposite to each other in term of functionalities. By means of thick black arrows the image shows the opposite functionalities. Maxpooling is substituted by an Unpooling block which is an approximate inverse and works by interpolating values. Convolution is substituted by Transposed Convolution which loads the transposed weights of the actual kernels into it.

4 Experiments

4.1 Dataset and Framework

We are using Mini Imagenet [6] dataset for our experiments for comparison and analysis, and may use other datasets in the future. We are using Pytorch as our framework for these experiments.

4.2 Feature Visualisation

Here we have visualised the features of different layers from the first architecture discussed above.

4.3 Feature Invariance

Here we have experimented with an image by performing some affine transformations on it like translation, rotation and scaling by varying degrees, while comparing the untransformed feature to the changes in the feature vectors from the top and bottom layers of the model.

4.4 Occlusion Sensitivity

Here, we have checked if the model is correctly detecting the location of the object in the image or if it is simply relying on the surrounding context. We have address this question by systematically occluding various areas of the input image (both center and corner) with a grey square and analysing the classifier's output at various levels.

5 Results and Discussions

5.1 Quantitative Analysis

5.1.1 VGG-16

Animal	Method	VGG-16	Predicted As
Cat	Normal Feature Visualisation	0.4848	Tabby Cat
	Feature Visualisation after Rotation (90 degree clockwise)	0.4213	Tabby Cat
	Feature Visualisation after Corner Occlusion	0.4861	Egyptian Cat
	Feature Visualisation after Center Occlusion	0.2347	Egyptian Cat
Ostrich	Normal Feature Visualisation	0.9998	Ostrich
	Feature Visualisation after Rotation (90 degree anti-clockwise)	0.2031	Siamang
	Feature Visualisation after Corner Occlusion	0.9995	Ostrich
	Feature Visualisation after Center Occlusion	0.9991	Ostrich
Baboon	Normal Feature Visualisation	0.9334	Baboon
	Feature Visualisation after Rotation (180 degree clockwise)	0.3598	Spider monkey
	Feature Visualisation after Corner Occlusion	0.9284	Baboon
	Feature Visualisation after Center Occlusion	0.6603	Baboon
Dog	Normal Feature Visualisation	0.6498	German Shepherd
	Feature Visualisation after Rotation (180 degree anti-clockwise)	0.3816	Guenon
	Feature Visualisation after Corner Occlusion	0.6639	German Shepherd
	Feature Visualisation after Center Occlusion	0.1969	German Shepherd

5.1.2 VGG-19

Animal	Method	Accuracy-VGG-19	Predicted As
Cat	Normal Feature Visualisation	0.5260	**Tabby Cat**
	Feature Visualisation after Rotation (90 degree clockwise)	0.6875	**Tabby Cat**
	Feature Visualisation after Corner Occlusion	0.5990	**Tabby Cat**
	Feature Visualisation after Center Occlusion	0.2217	**Egyptian Cat**
Ostrich	Normal Feature Visualisation	0.9995	**Ostrich**
	Feature Visualisation after Rotation (90 degree anti-clockwise)	0.4087	**Magpie**
	Feature Visualisation after Corner Occlusion	0.9994	**Ostrich**
	Feature Visualisation after Center Occlusion	0.9669	**Ostrich**
Baboon	Normal Feature Visualisation	0.9764	**Baboon**
	Feature Visualisation after Rotation (180 degree clockwise)	0.6981	**Ostrich**
	Feature Visualisation after Corner Occlusion	0.9689	**Baboon**
	Feature Visualisation after Center Occlusion	0.4295	**Baboon**
Dog	Normal Feature Visualisation	0.5789	**German Shepherd**
	Feature Visualisation after Rotation (180 degree anti-clockwise)	0.2197	**Marmoset**
	Feature Visualisation after Corner Occlusion	0.6017	**German Shepherd**
	Feature Visualisation after Center Occlusion	0.2836	**German Shepherd**

5.2 Qualitative Analysis

5.2.1 Normal Feature Visualisations

(A) **VGG-16 Model for Architecture-1**
CAT :

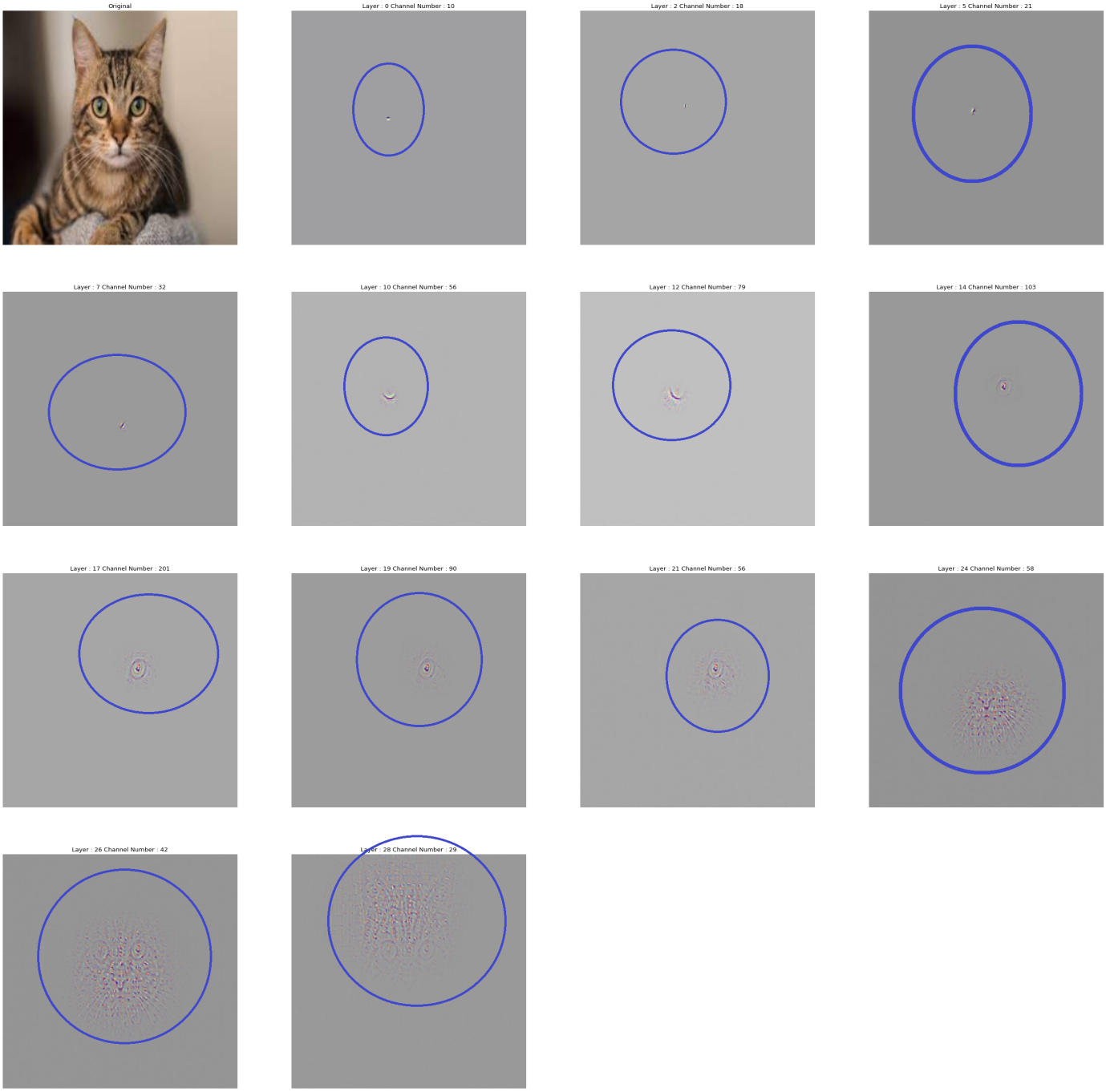

Figure 2: Cat feature Visualization

Discussion :

For the cat image it is evident that the lower layers are learning the low level features i.e the eye edges and the higher layers are learning the high level features that include the textures visualising the mouth regions of the cat and the furs. The attended feature in each layers are highlighted in blue circle. It has also predicted the image as **'tabby'** with an accuracy of **48.4%**.

OSTRICH :

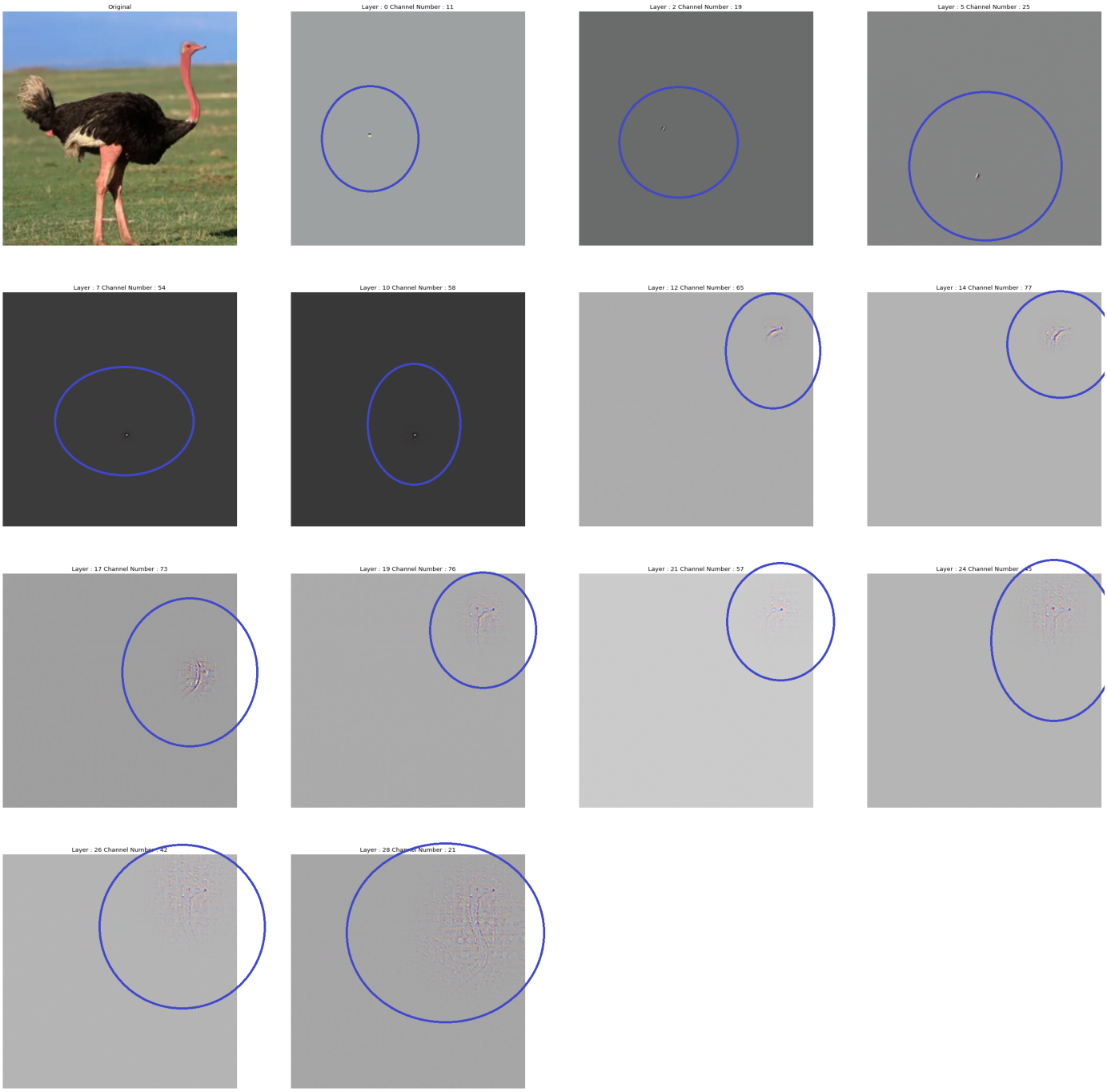

Figure 3: Ostrich feature Visualization

Discussion : For the ostrich image it can be evidently seen that the layers 12, 14 are learning the low level features i.e the head, beak edges and the higher layers are learning the high level features that include the textures visualising the the neck and body of the ostrich. The attended feature in each layers are highlighted in blue circle. It has predicted **'ostrich'** correctly with **99.9%** accuracy.

BABOON :

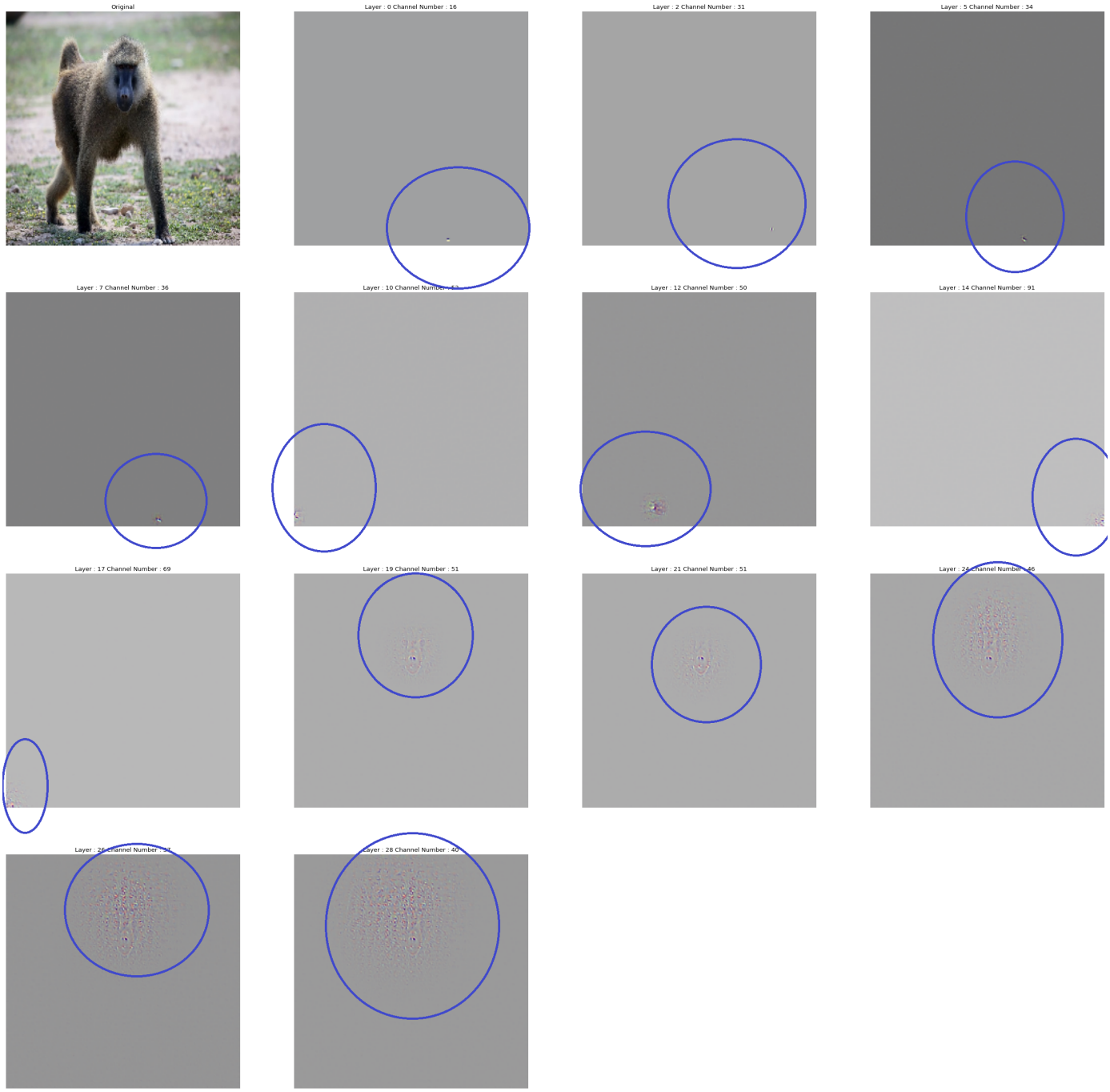

Figure 4: Baboon feature Visualization

Discussion : For the Baboon image it can be seen that the lower layers are learning the low level features i.e feet and the higher layers are learning the high level features that include the textures visualising the face of the baboon. The attended feature in each layers are highlighted in blue circle. It has predicted correctly **'baboon'** with **93.3%** accuracy.

DOG :

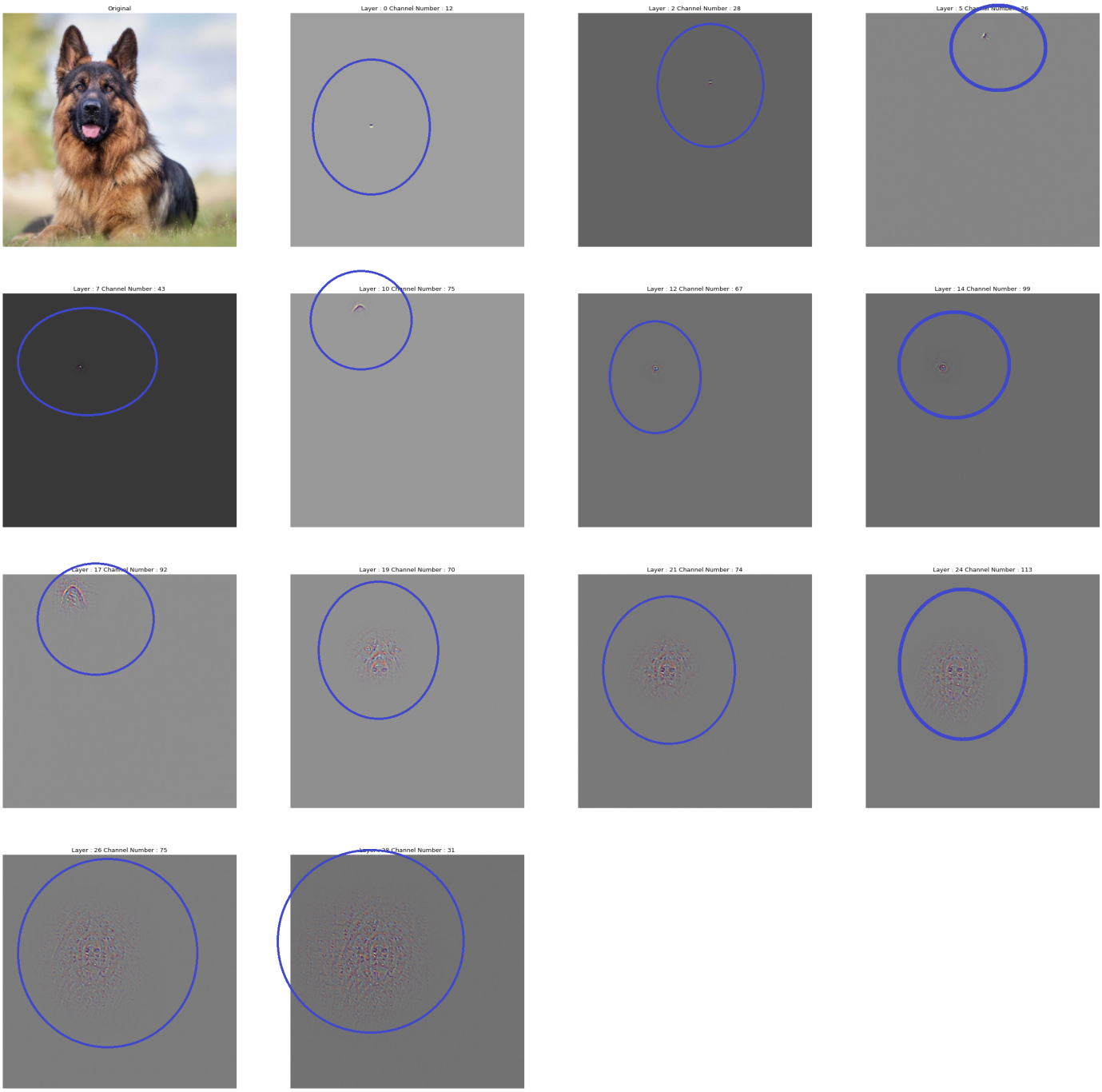

Figure 5: Dog feature Visualization

Discussion : For the Dog image it can be seen that the lower layers are learning the low level features i.e eyes and ears and the higher layers are learning the high level features that include the textures visualising the face of the dog. The attended feature in each layers are highlighted in blue circle. It has predicted correctly as **'German shepherd'** with **64.9%** accuracy.

(B) **VGG-19 Model for Architecture-1**
CAT :

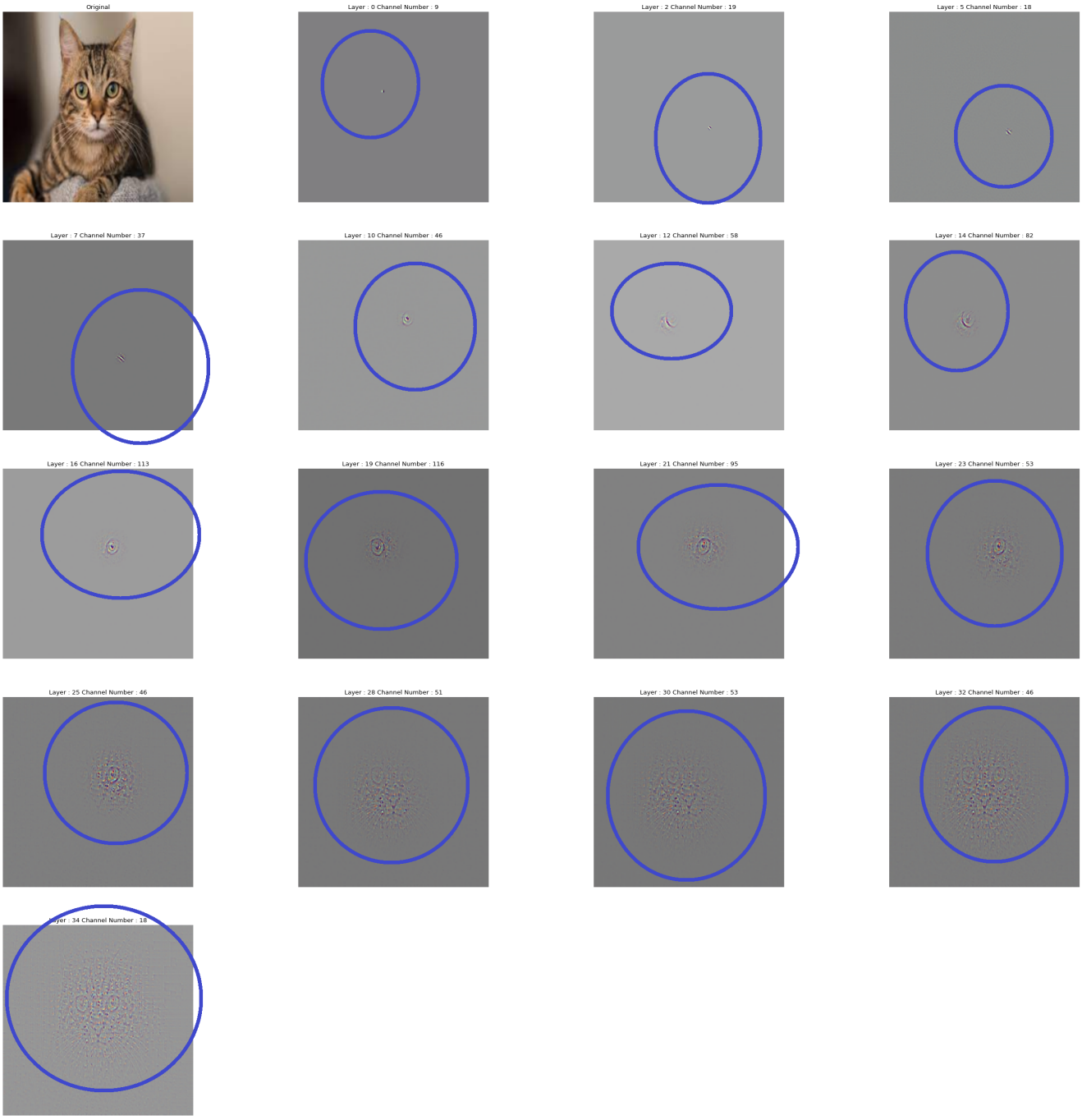

Figure 6: Cat feature Visualization

Discussion :
For the cat image it is evident that the lower layers are learning the low level features i.e the eye edges and the higher layers are learning the high level features that include the textures visualising the mouth regions of the cat and the furs from

layer 46 on wards. The attended feature in each layers are highlighted in blue circle. It has also predicted the image as **'tabby'** with an accuracy of **52.6%**.

OSTRICH :

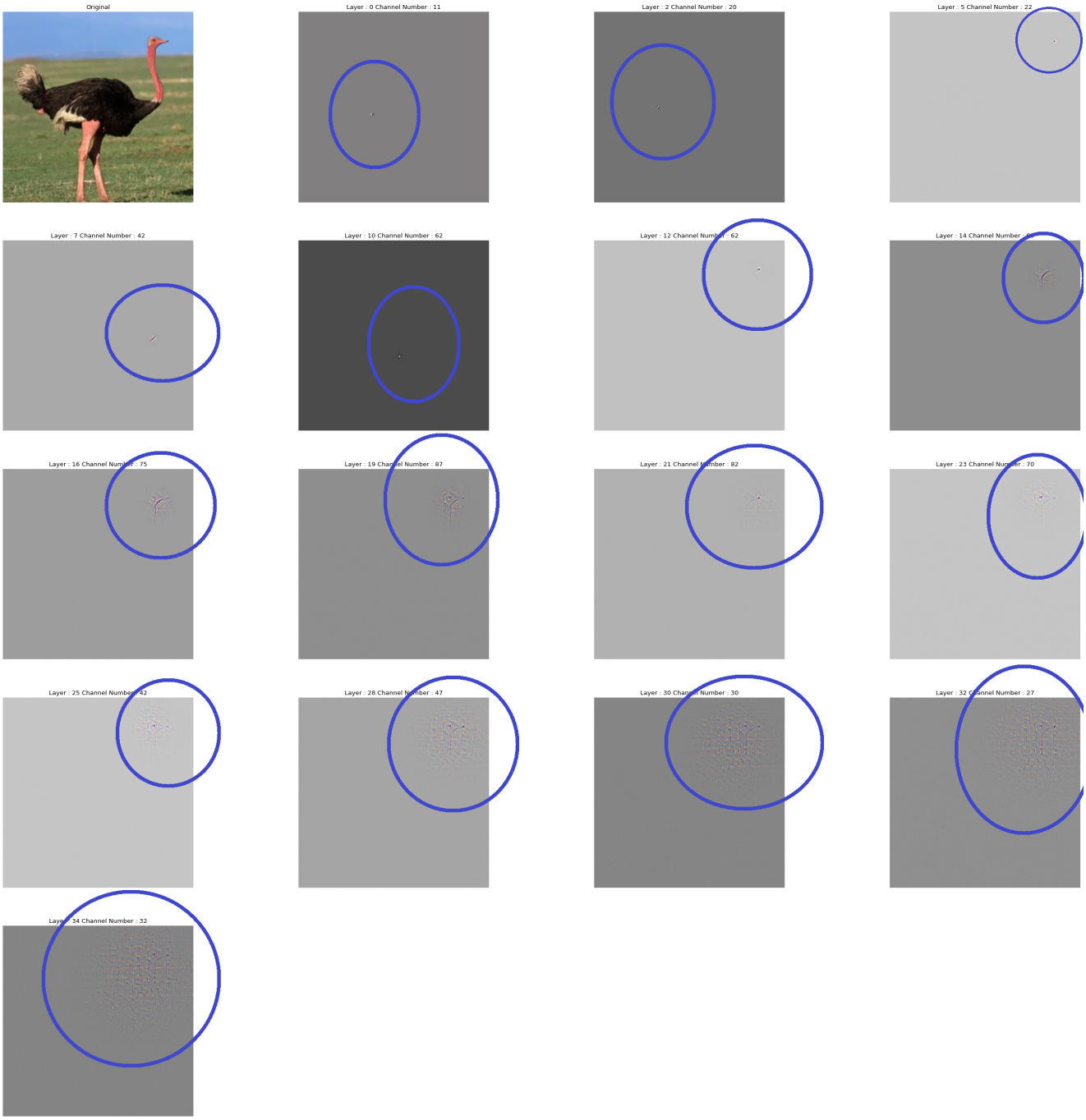

Figure 7: Ostrich feature Visualization

Discussion : For the ostrich image it can be evidently seen that the layers 12, 14 are learning the low level features i.e the eye, beak edges and the higher layers are learning the high level features that include the textures visualising the the neck and body of the ostrich. The attended feature in each layers are highlighted in blue circle. It has predicted **'ostrich'** correctly with **99.9%** accuracy.

BABOON :

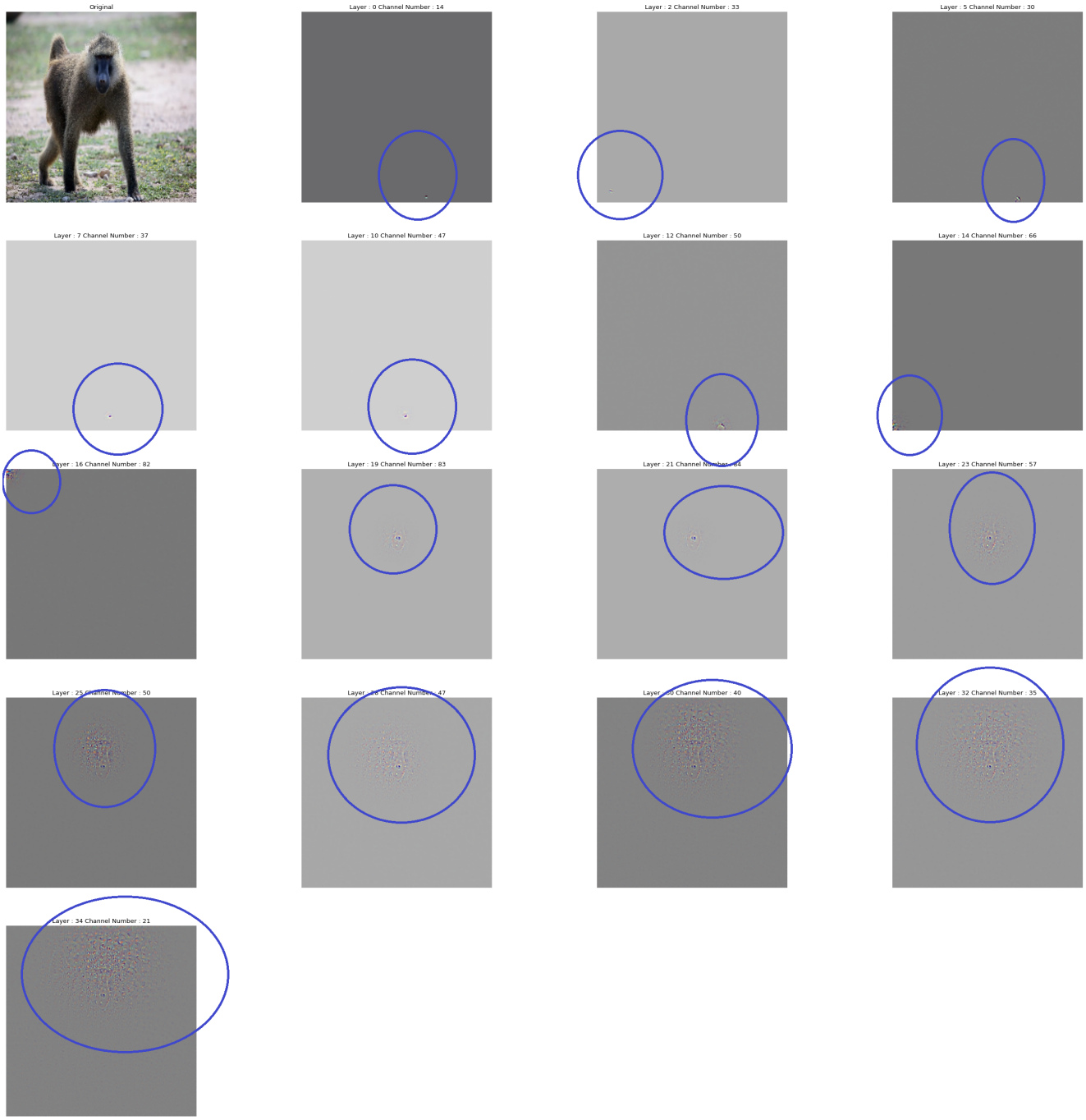

Figure 8: Baboon feature Visualization

Discussion : For the Baboon image it can be seen that the lower layers are trying to learn the low level features like nose from layer 14 and the higher layers are learning the features like face from layer 25 and finally visualising the face of the baboon. The attended feature in each layers are highlighted in blue circle. It has predicted correctly **'baboon'** with **97.6%** accuracy.

DOG :

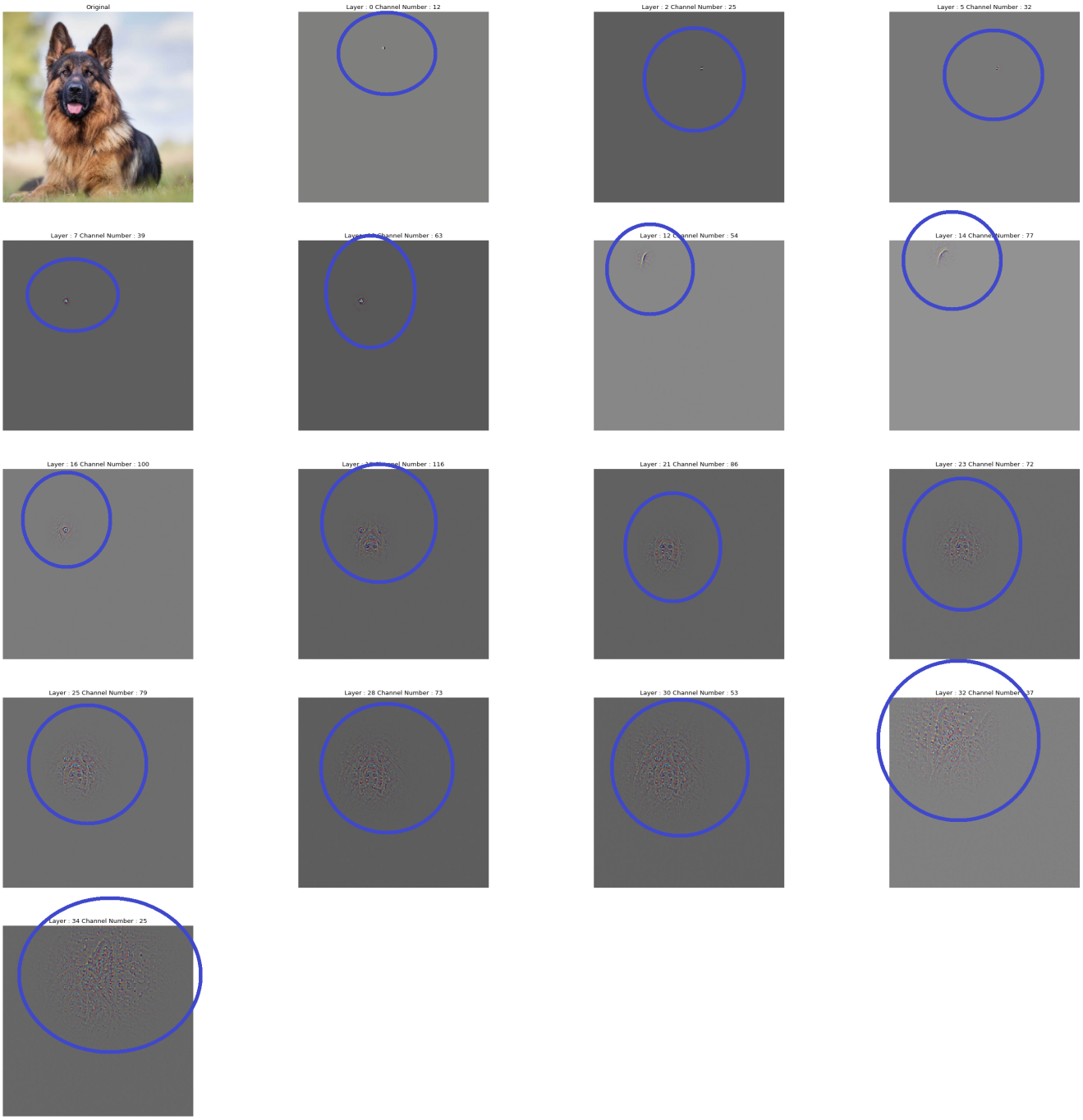

Figure 9: Dog feature Visualization

Discussion : For the Dog image it can be seen that the lower layers are learning the low level features i.e eyes and ears and the higher layers are learning the high level features that include the textures visualising the face of the dog. The attended feature in each layers are highlighted in blue circle. It has predicted as **'German Shepherd'** with a low accuracy of **57.8%**.

5.2.2 Feature Visualisation for Image under Affine Transformations

(A) **VGG-16 Model for Architecture-1**
CAT : 90 degree clockwise

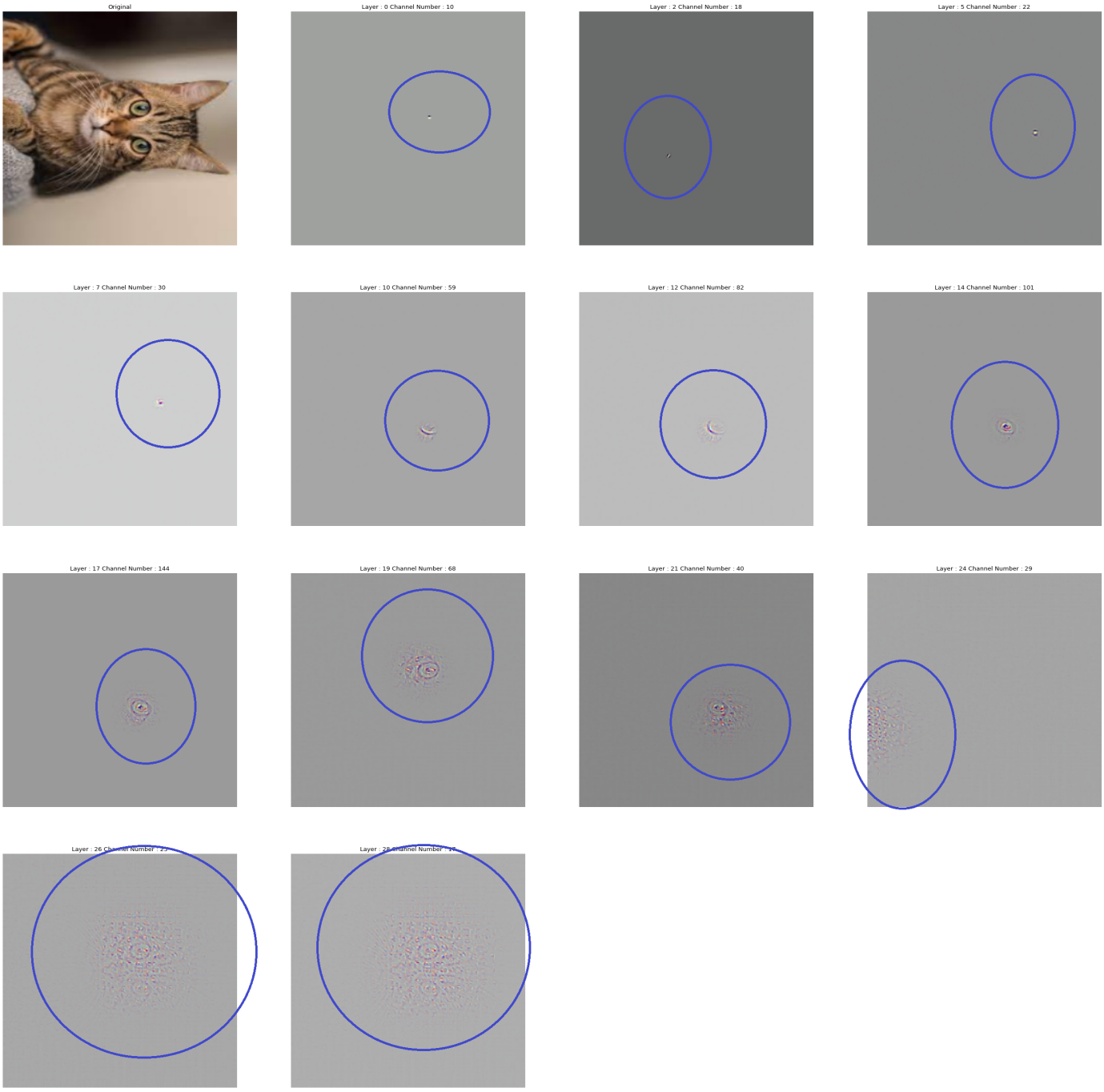

Figure 10: Feature Visualization for Cat image rotated by 90 degree

Discussion : The cat image has been rotated at 90 degree clockwise image it can be seen that the lower layers are learning the low level features i.e eyes and whiskers and the higher layers layer 25 are learning the high level features that include the textures visualising the face of the cat. The attended feature in each

layers are highlighted in blue circle. It has predicted correctly as **'tabby'** but with a low accuracy of **42.1%** .

OSTRICH : 90 degree anti clock wise

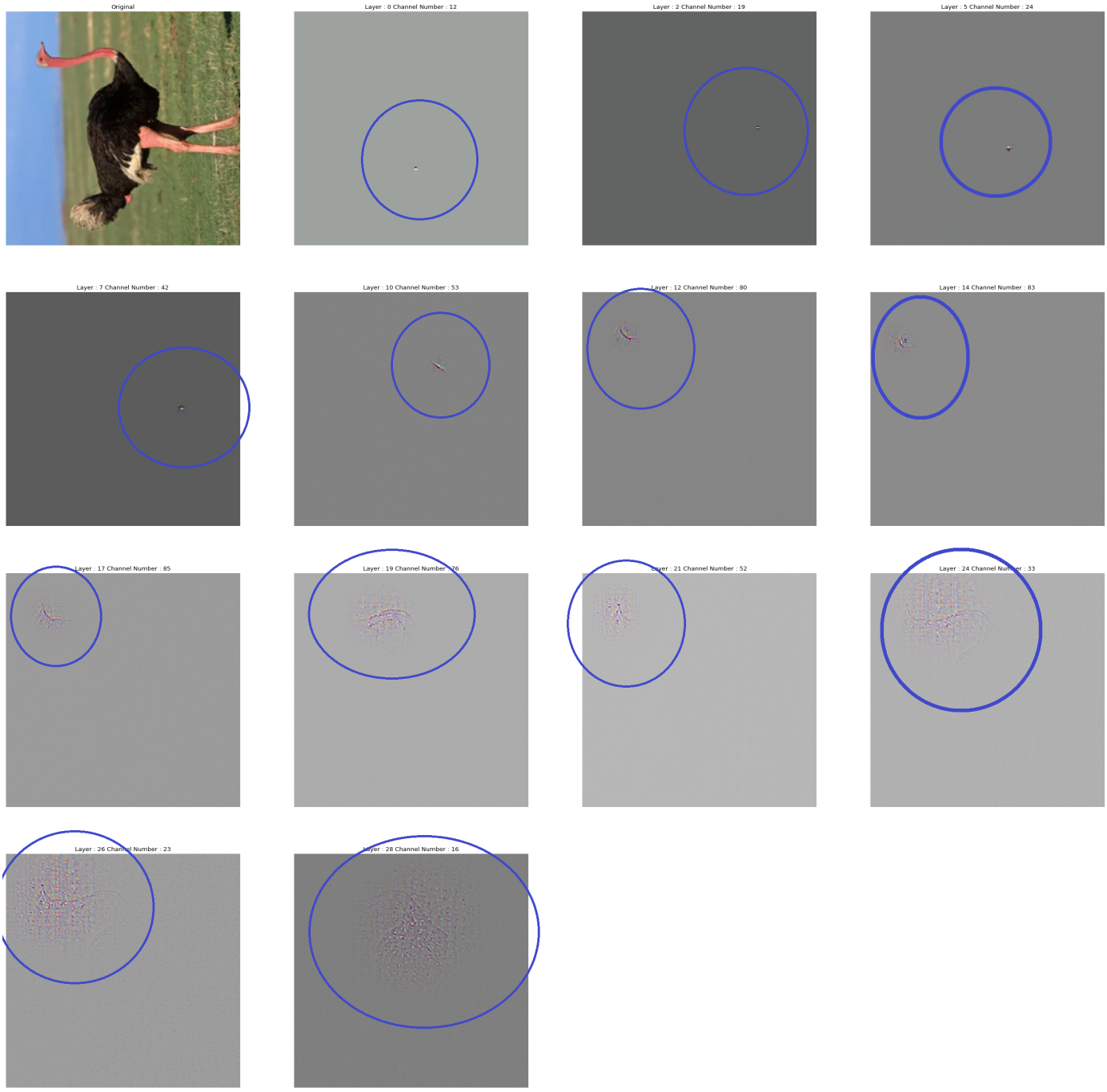

Figure 11: Feature Visualization for Ostrich image rotated by 90 anticlockwise degree

Discussion : The ostrich image has been rotated at 90 degree anti clockwise image it can be seen that the lower layers are learning the low level features i.e beak neck and the higher layers are learning the high level features that include the textures visualising the body of the ostrich, yet it has wrongly predicted as **'siamang'** which is from the family of monkeys with a low **20.3%** accuracy. The attended feature in each layers are highlighted in blue circle.

BABOON : 180 degree Clockwise

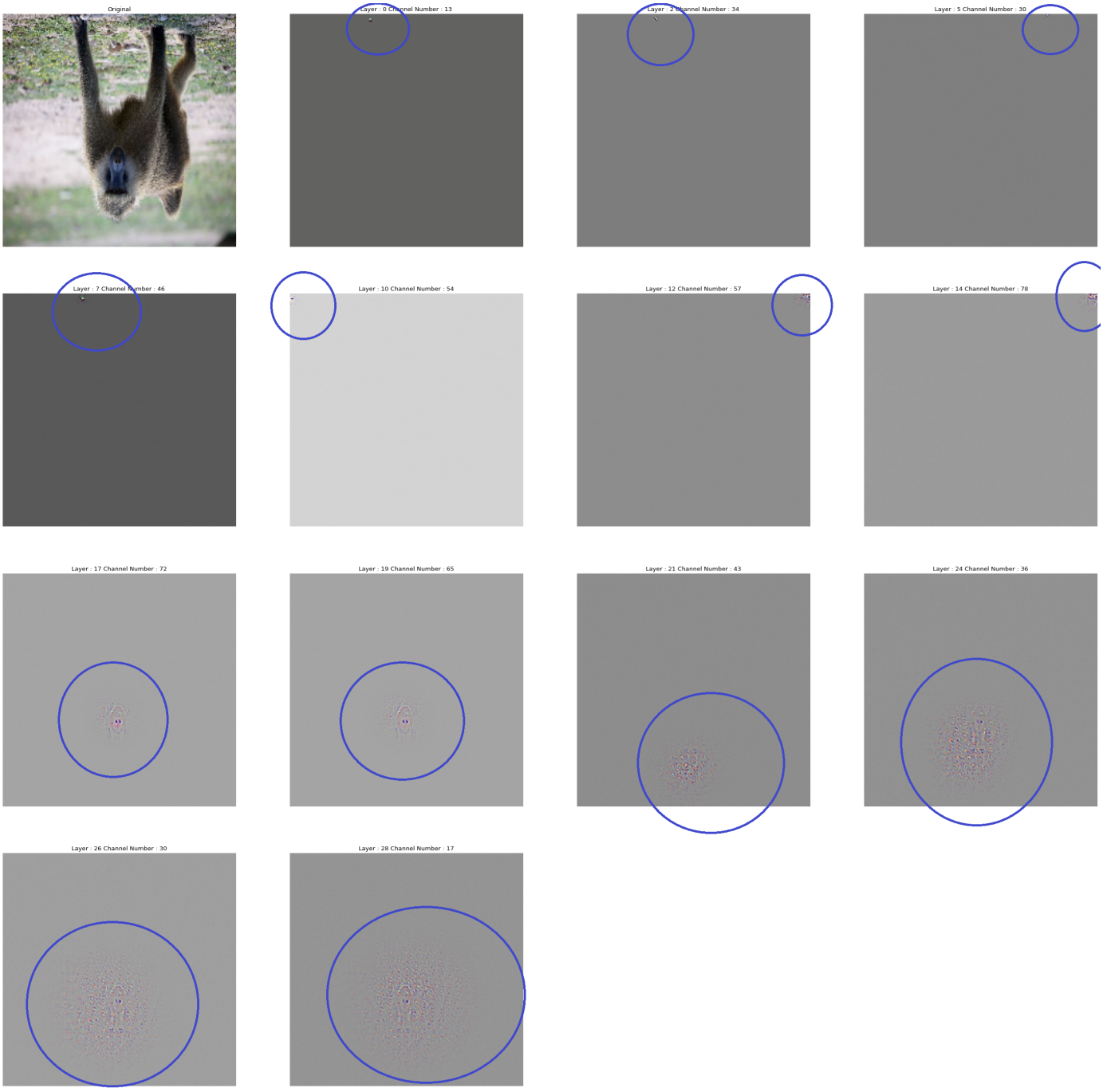

Figure 12: Feature Visualization for Baboon image rotated by 180 degree

Discussion : The baboon image has been rotated at 180 degree clockwise image it can be seen that the lower layers are not learning the low level features very well but the higher layers are learning the features like nose of baboon and gradually learning the face, yet it has wrongly predicted as **'spider monkey'** with **35.9%** accuracy. The attended feature in each layers are highlighted in blue circle.

DOG : 180 degree anti clockwise

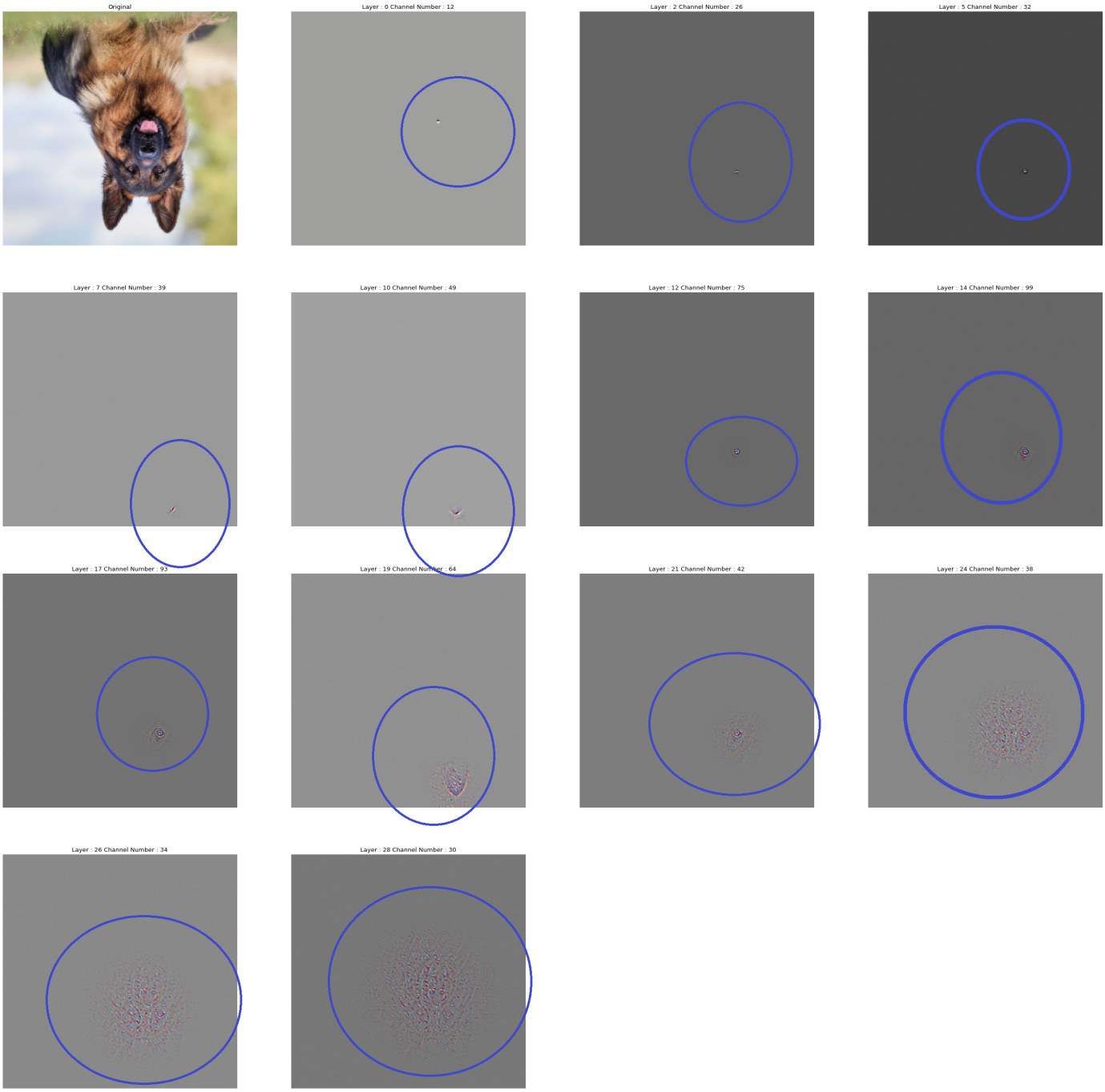

Figure 13: Feature Visualization for Dog image rotated by 180 degree anticlockwise

Discussion : The dog image has been rotated at 180 degree anti clockwise image it can be seen that the lower layers are learning the low level features like ears and eyes and the higher layers are learning the features like learning the face, yet it has wrongly predicted as **'guenon'** which is also from the family of monkeys with **38.1%** accuracy. The attended feature in each layers are highlighted in blue circle.

(B) **VGG-19 Model for Architecture-1**
CAT : 90 degree clockwise

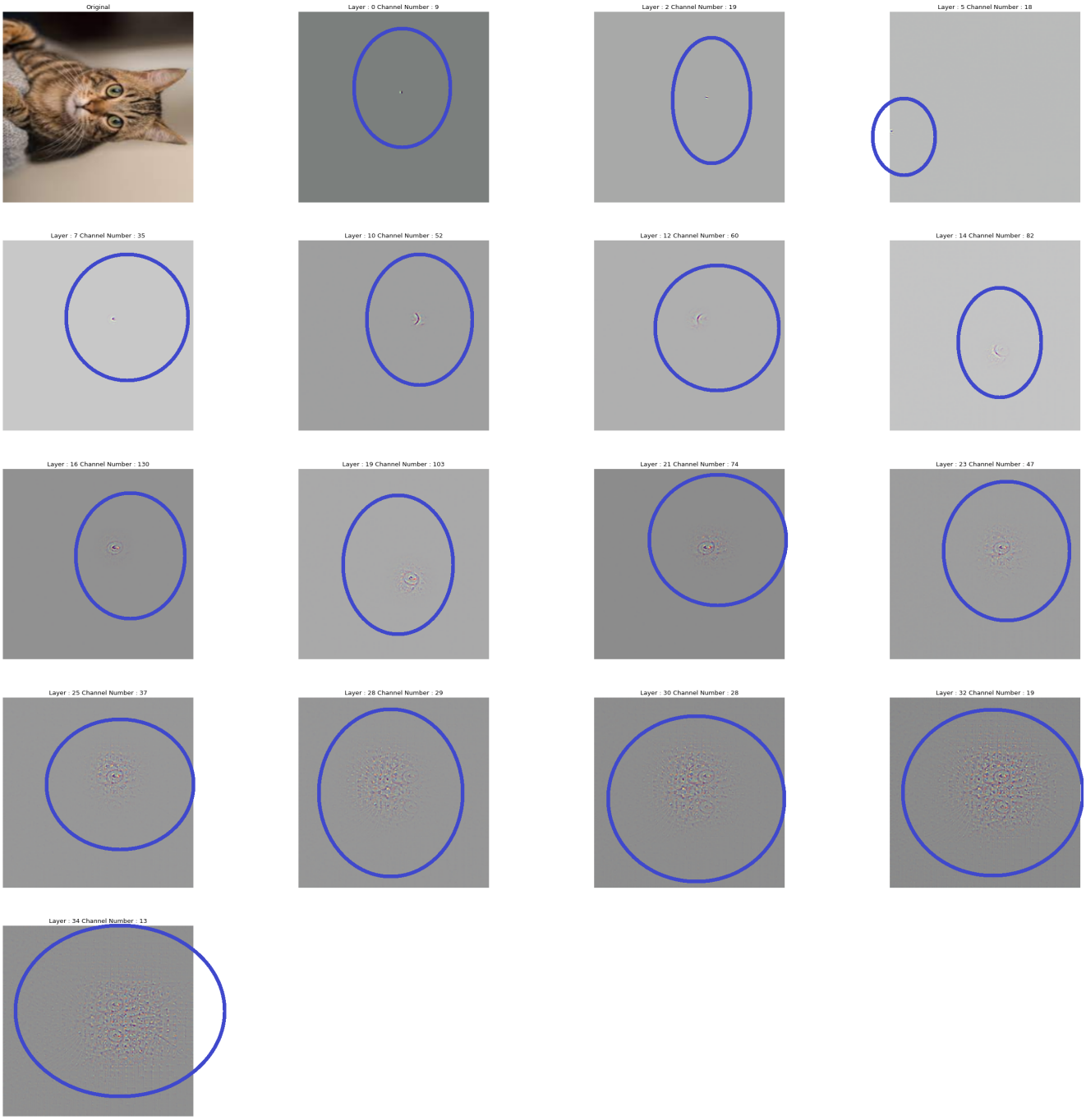

Figure 14: Feature Visualization for cat image rotated by 90 degree clockwise

Discussion : The cat image has been rotated at 90 degree clockwise image it can be seen that the lower layers are learning the low level features i.e eyes and the higher layers are learning the high level features that include the textures visualising the face of the cat. The attended feature in each layers are highlighted in blue circle. It has predicted correctly as **'tabby'** but with a low accuracy of **68.7%** .

OSTRICH : 90 degree anti clock wise

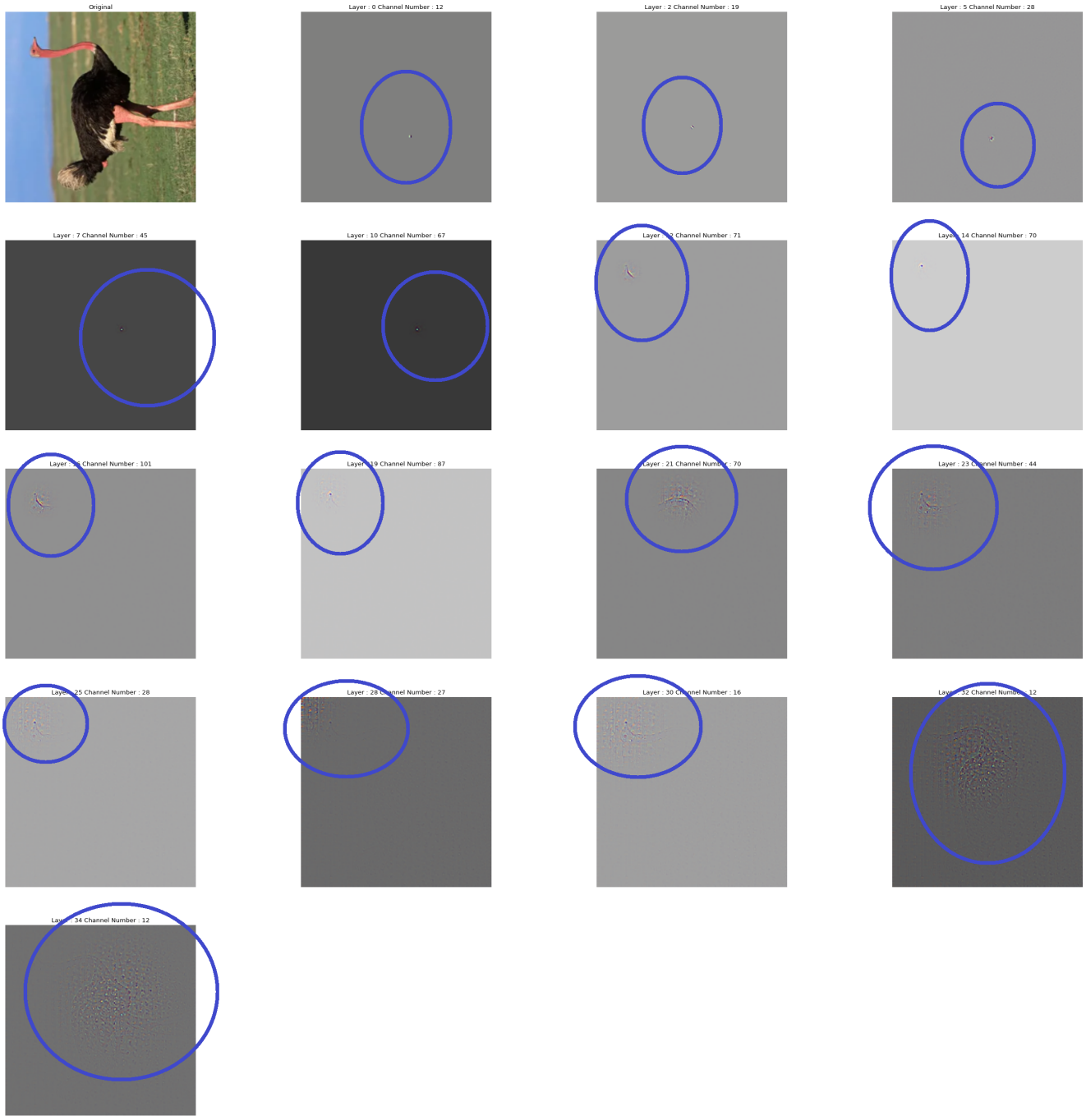

Figure 15: Feature Visualization for Ostrich image rotated by 90 degree anticlockwise

Discussion : The ostrich image has been rotated at 90 degree anti clockwise image it can be seen that the lower layers are learning the low level features i.e beak,neck and the higher layers are learning the high level features that include the textures visualising the body of the ostrich, yet it has wrongly predicted as **'magpie'** which is from the family of birds with **40.8%** accuracy. The attended feature in each layers are highlighted in blue circle.

BABOON : 180 degree Clockwise

Figure 16: Feature Visualization for Baboon image rotated by 180 degree clockwise

Discussion : The baboon image has been rotated at 180 degree clockwise image it can be seen that the lower layers are not learning the low level features very well but the higher layers are learning the features like nose of baboon and gradually learning the face, yet it has wrongly predicted as **'ostrich'** which is from the family of birds with **69.8%** accuracy. The attended feature in each layers are highlighted in blue circle.

DOG : 180 degree anti clockwise

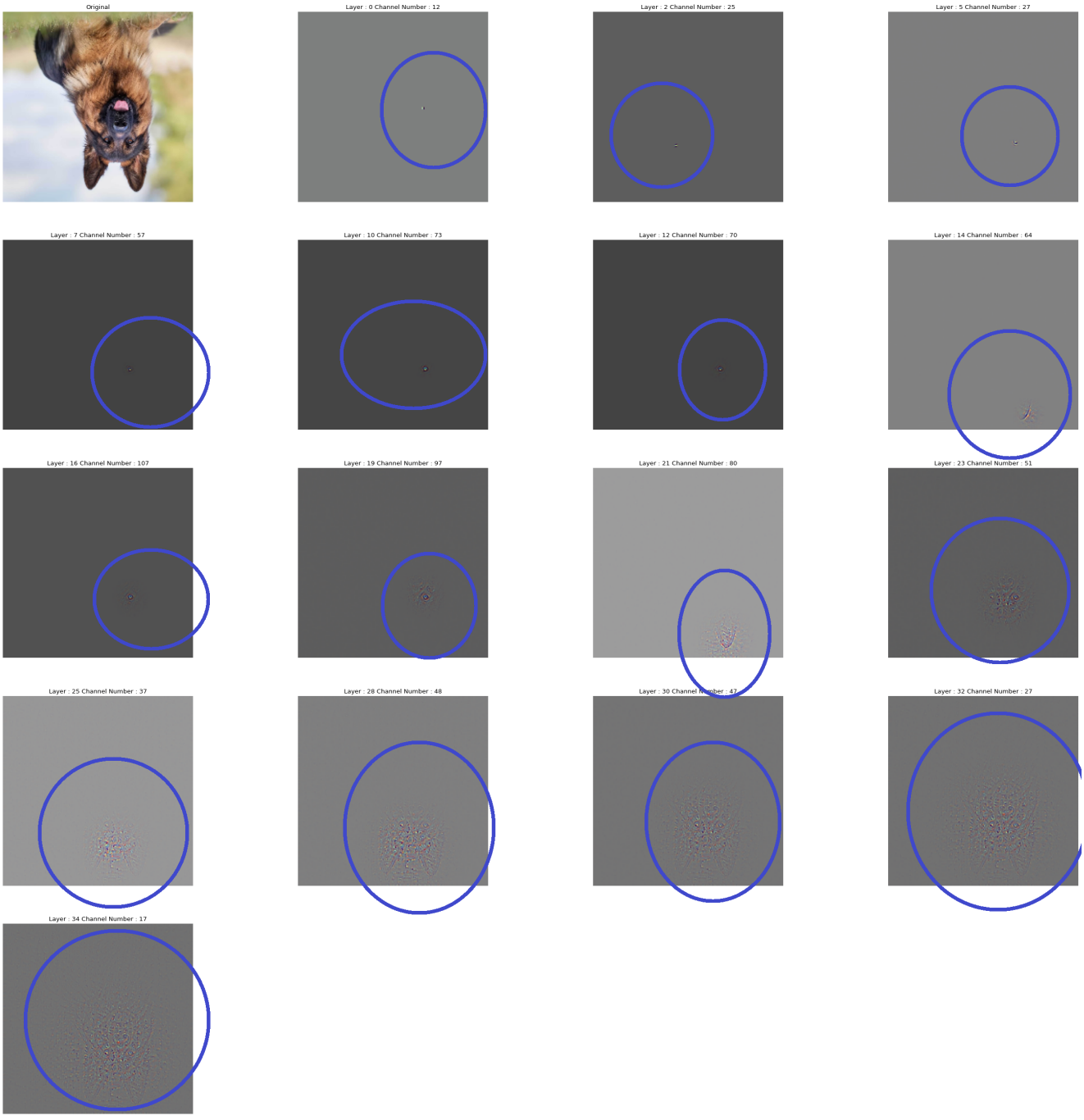

Figure 17: Feature Visualization for Dog image rotated by 180 degree anticlockwise

Discussion : For the Dog image it can be seen that the lower layers are learning the low level features i.e eyes and ears and the higher layers are learning the high level features that include the textures visualising the face of the dog. The attended feature in each layers are highlighted in blue circle. It has predicted wrongly as **'marmoset'** with a low accuracy of **21.9%**.

5.2.3 Feature Visualisation for Image under Occlusional Sensitivity

CORNER-OCCLUSION

(A) **VGG-16 Model for Architecture-1**
CAT :

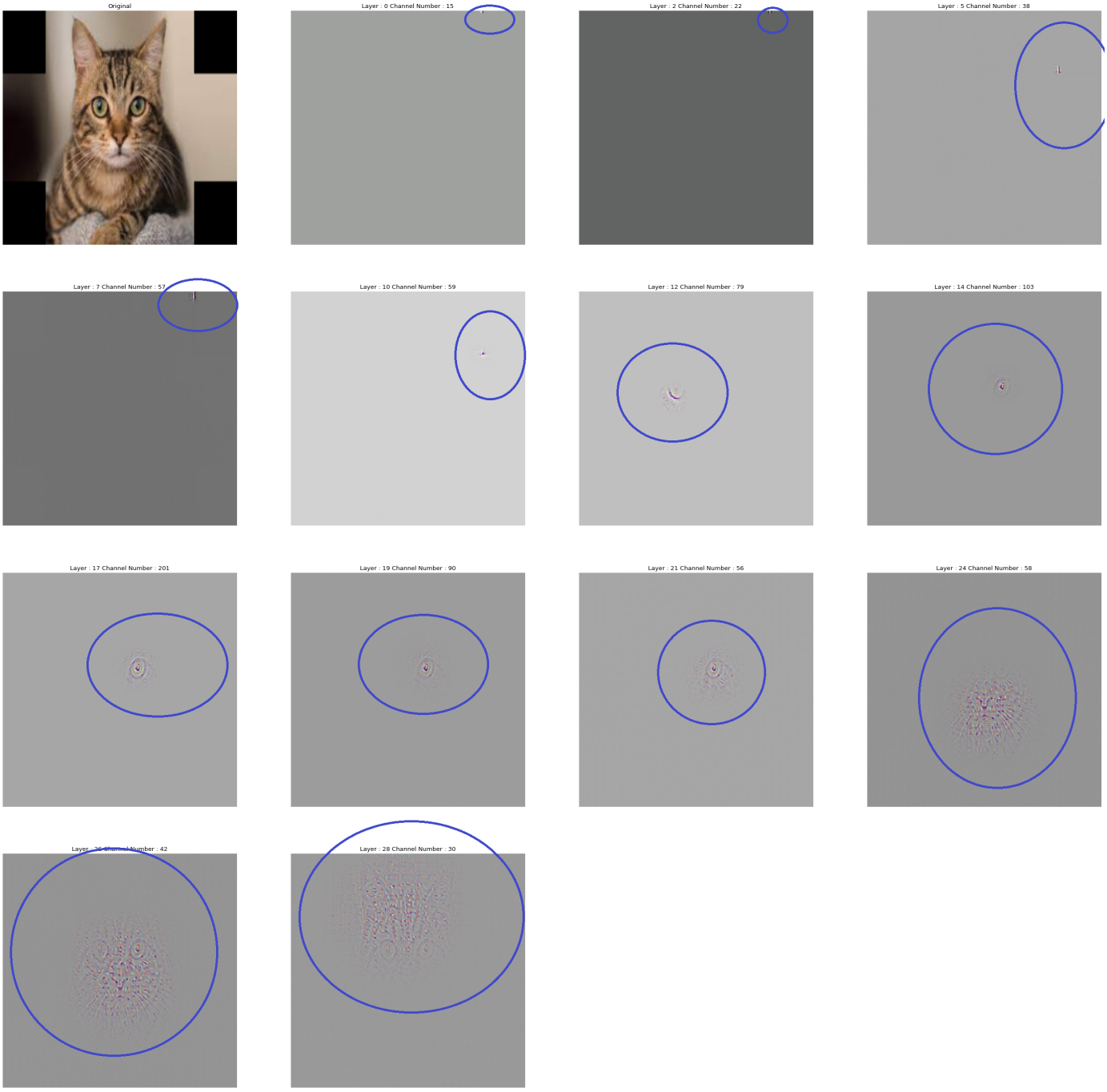

Figure 18: Feature Visualization for Cat image Corner occluded

Discussion : The cat image was corner occluded it can be seen that the lower layers are learning the low level features like eyes and the higher layers are learning

the features like learning the face, whiskers and visualises the cat partially yet it has wrongly predicted the breed as **'Egyptian cat'** with **48.6%** accuracy. The attended feature in each layers are highlighted in blue circle.

OSTRICH :

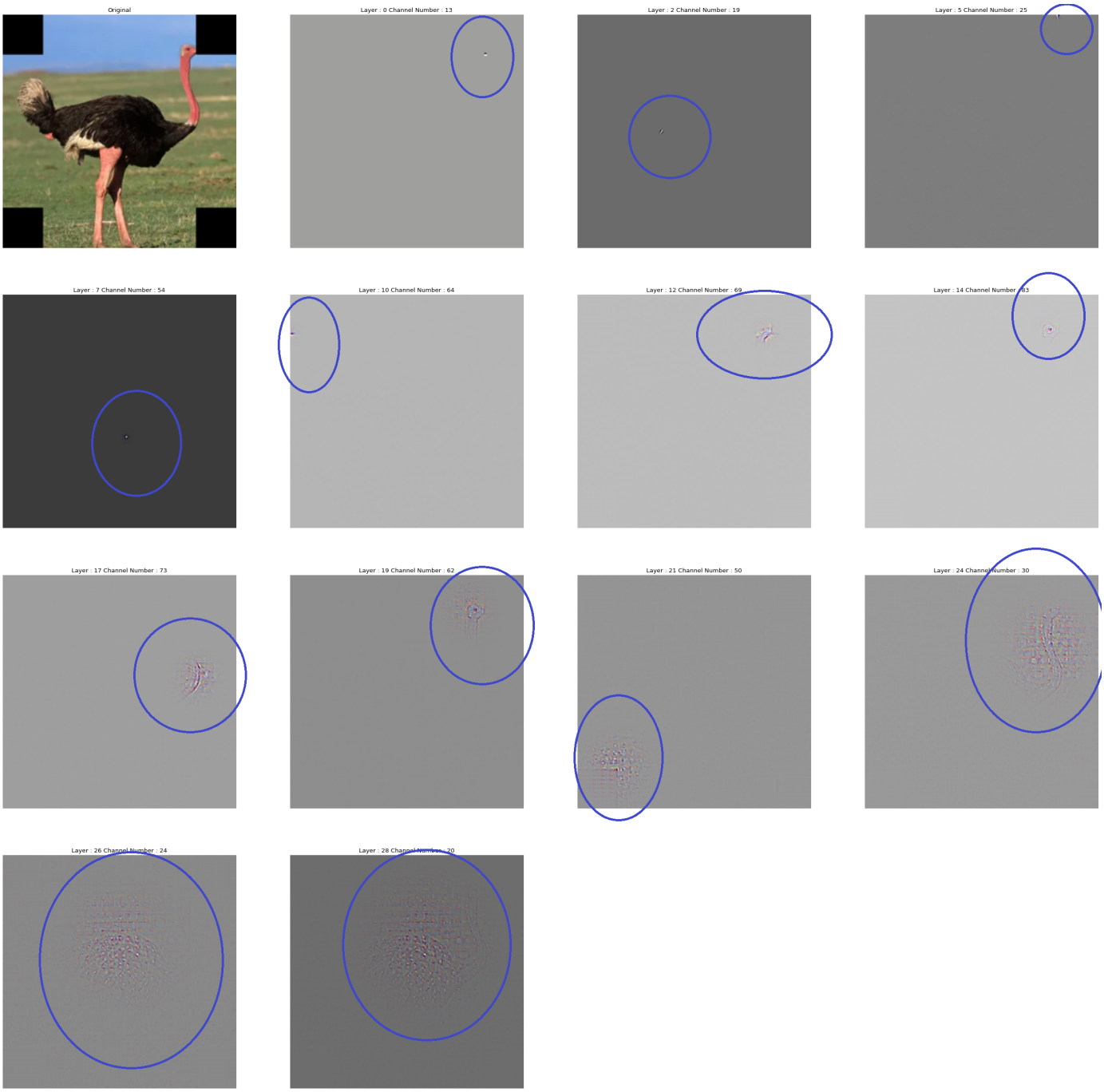

Figure 19: Feature Visualization for Ostrich image Corner occluded

Discussion : The ostrich image was corner occluded it can be seen that the lower layers are learning the low level features like eyes and the higher layers are learning the features like learning the neck and body of the bird, yet it has correctly predicted as **'ostrich'** with **99.9%** accuracy. The attended feature in each layers are highlighted in blue circle.

BABOON :

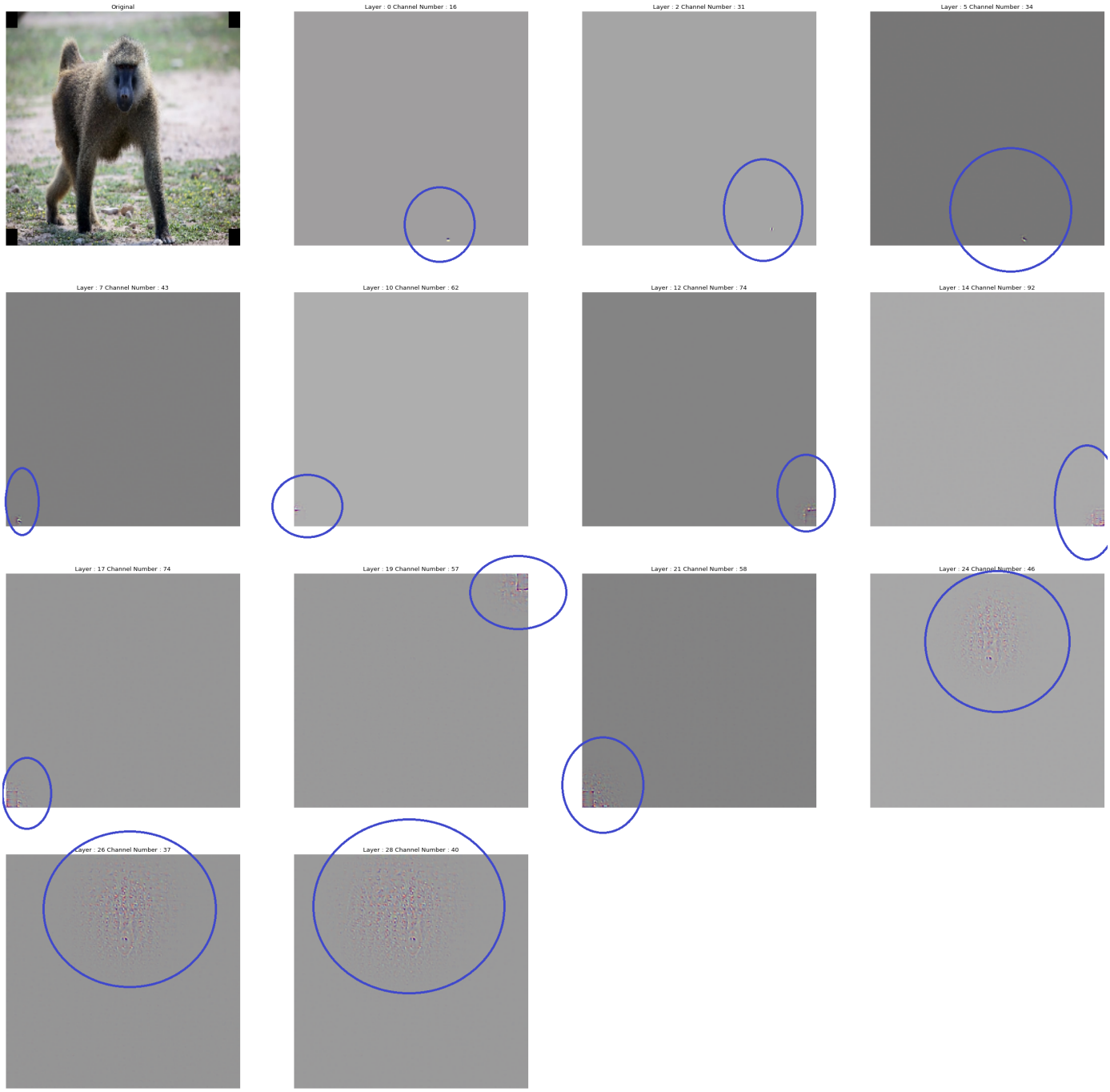

Figure 20: Feature Visualization for Baboon image Corner occluded

Discussion : The baboon image was corner occluded it can be seen that the lower layers did not learn the features of baboon but the surrounding features of the image like grass etc and the higher layers from 24 directly learned the face of the baboon, and it has correctly predicted as **'baboon'** with **92.8%** accuracy. The attended feature in each layers are highlighted in blue circle.

DOG :

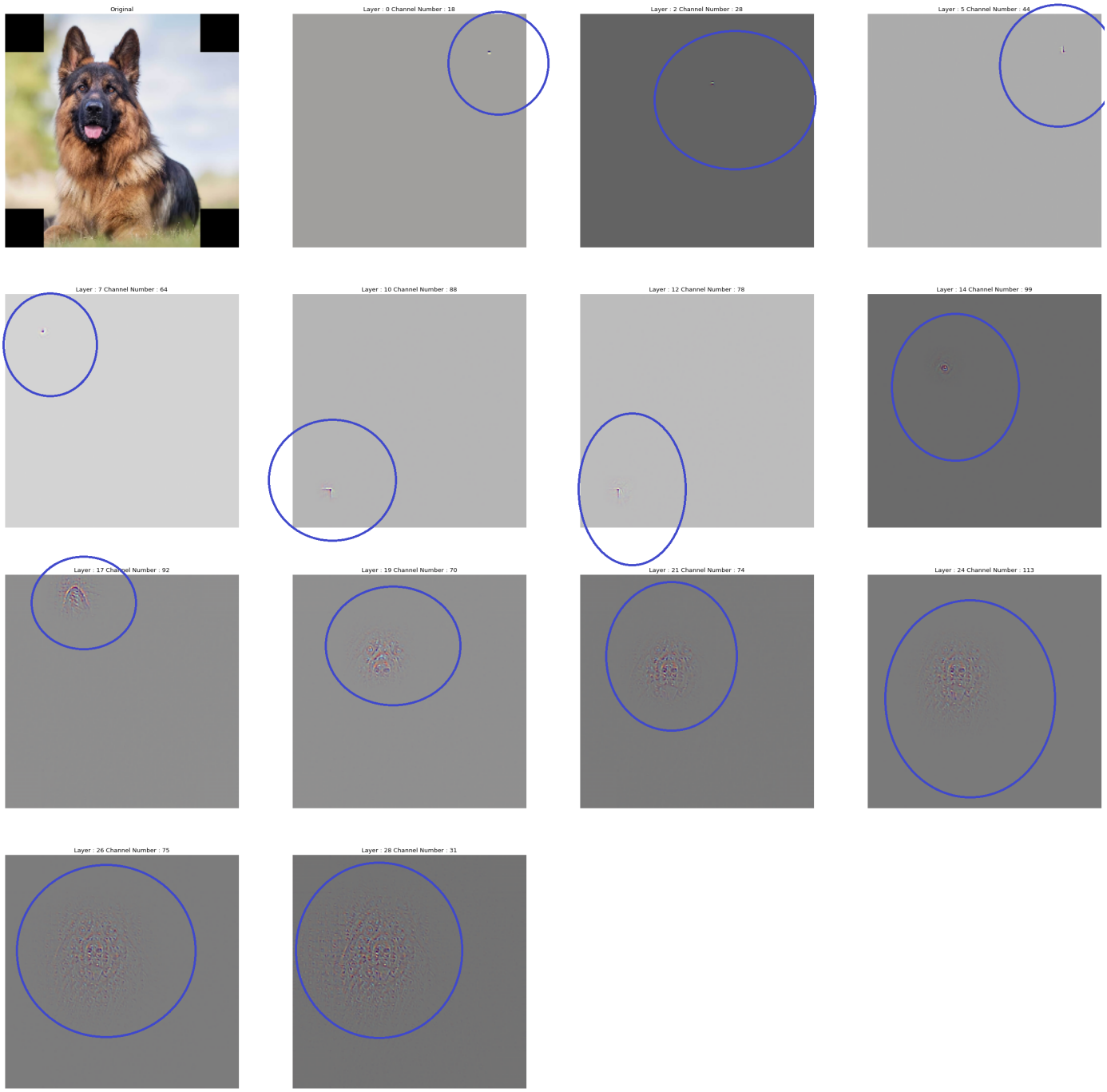

Figure 21: Feature Visualization for Dog image Corner occluded

Discussion : The dog image was corner occluded it can be seen that the lower layers learnt the features of background and as it was approaching higher layers it started learning eyes, ears and finally the visualization of the face and it has correctly predicted as **'German Shepherd'** but with **66.3%** accuracy. The attended feature in each layers are highlighted in blue circle.

(B) **VGG-19 Model for Architecture-1**
CAT :

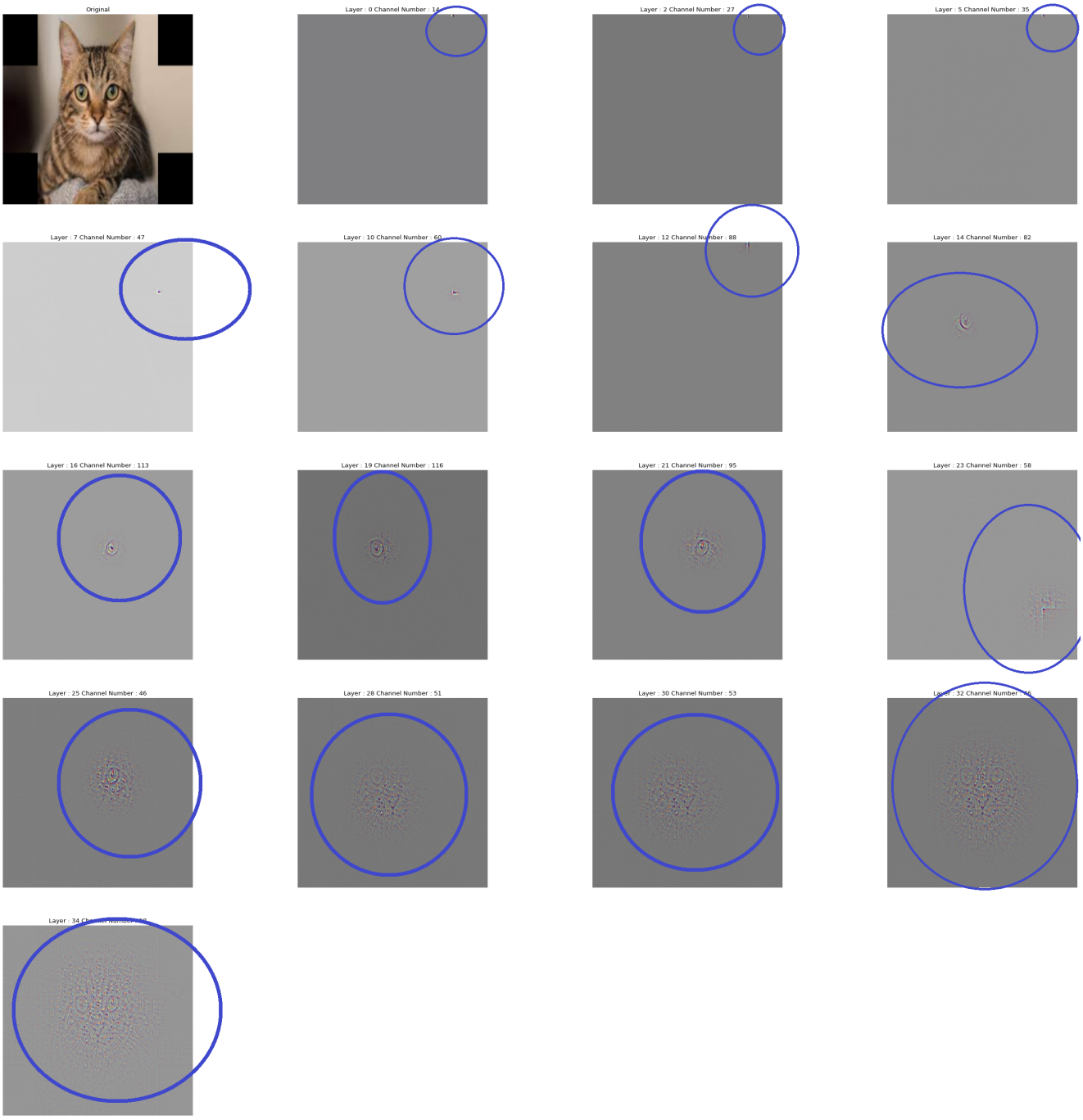

Figure 22: Feature Visualization for Cat image Corner occluded

Discussion : The cat image was corner occluded it can be seen that the lower layers are learning the low level features like eyes and the higher layers are learning the features like learning the face, whiskers and visualises the cat.The model correctly predicted as **'tabby'** with **59.9%** accuracy. The attended feature in each layers are highlighted in blue circle.

OSTRICH :

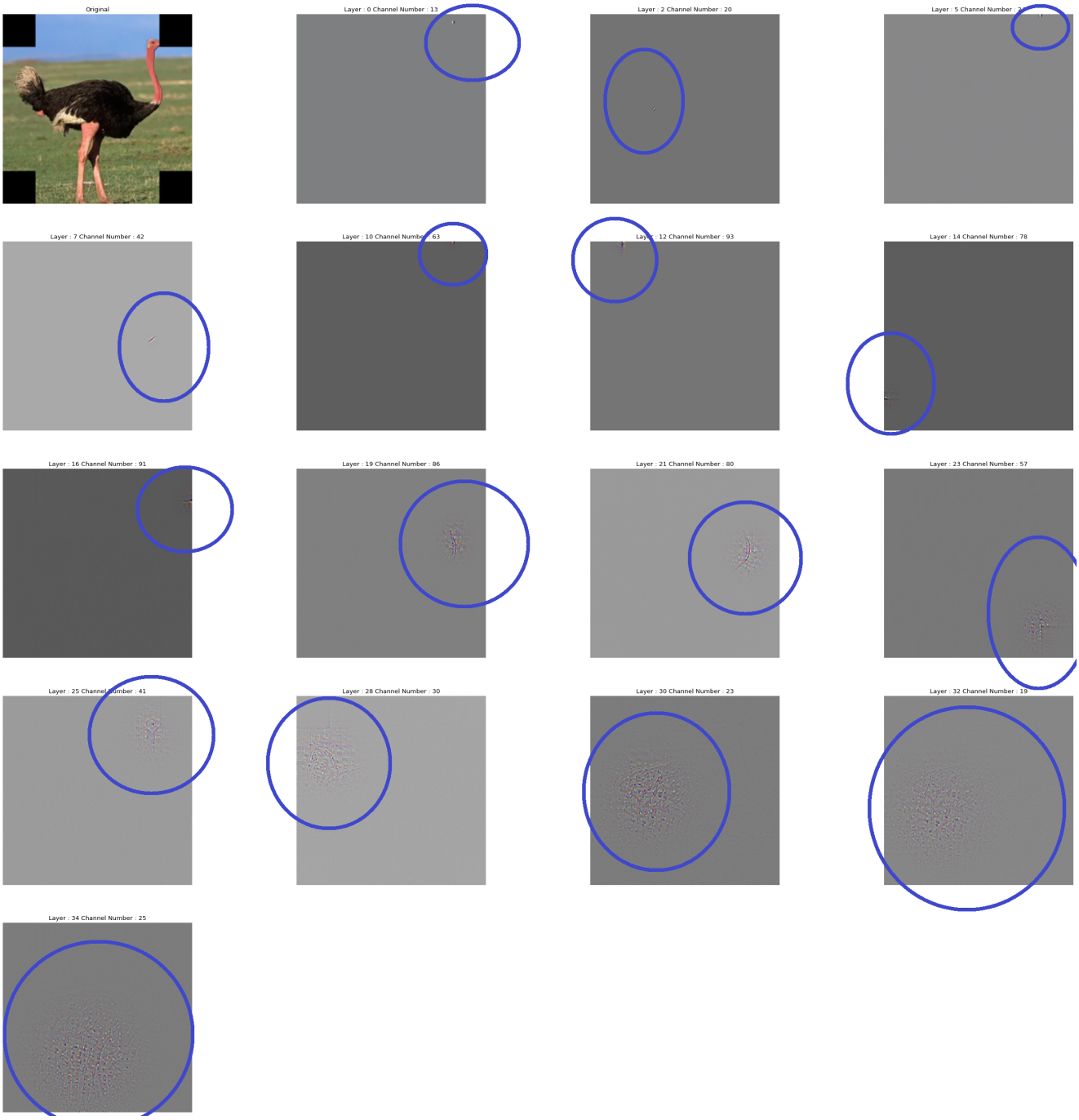

Figure 23: Feature Visualization for Ostrich image Corner occluded

Discussion : The ostrich image was corner occluded it can be seen that the lower layers are learning the low level features like eyes and some background features and the higher layers are learning the features like learning the body,feet of the bird, yet it has correctly predicted as **'ostrich'** with **99.9%** accuracy. The attended feature in each layers are highlighted in blue circle.

BABOON :

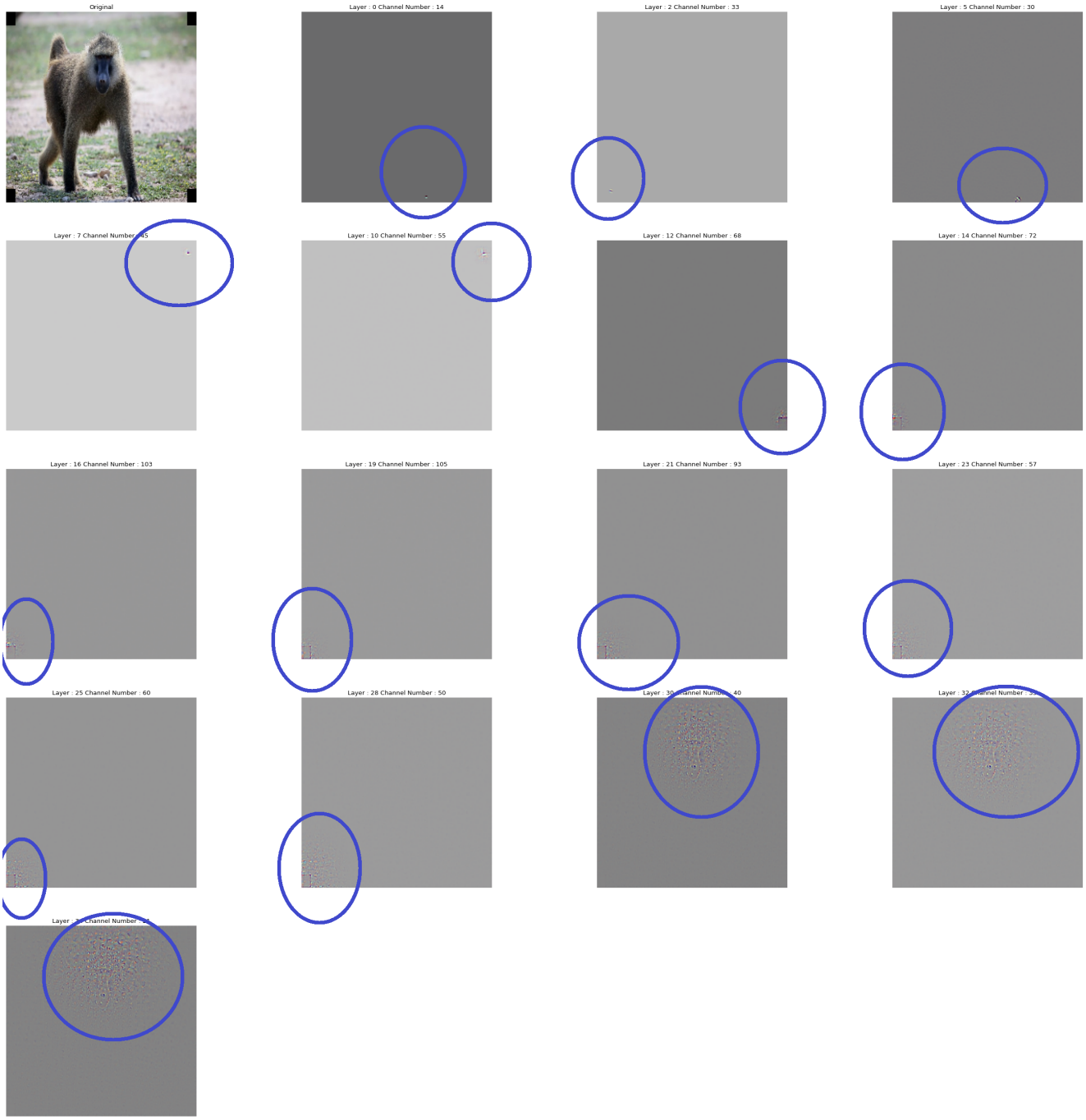

Figure 24: Feature Visualization for Baboon image Corner occluded

Discussion : The baboon image was corner occluded it can be seen that the lower layers did not learn the features of baboon but the surrounding features of the image like grass etc and the higher layers from 30 directly learned the face of the baboon, and it has correctly predicted as **'baboon'** with **96.8%** accuracy. The attended feature in each layers are highlighted in blue rectangle.

DOG :

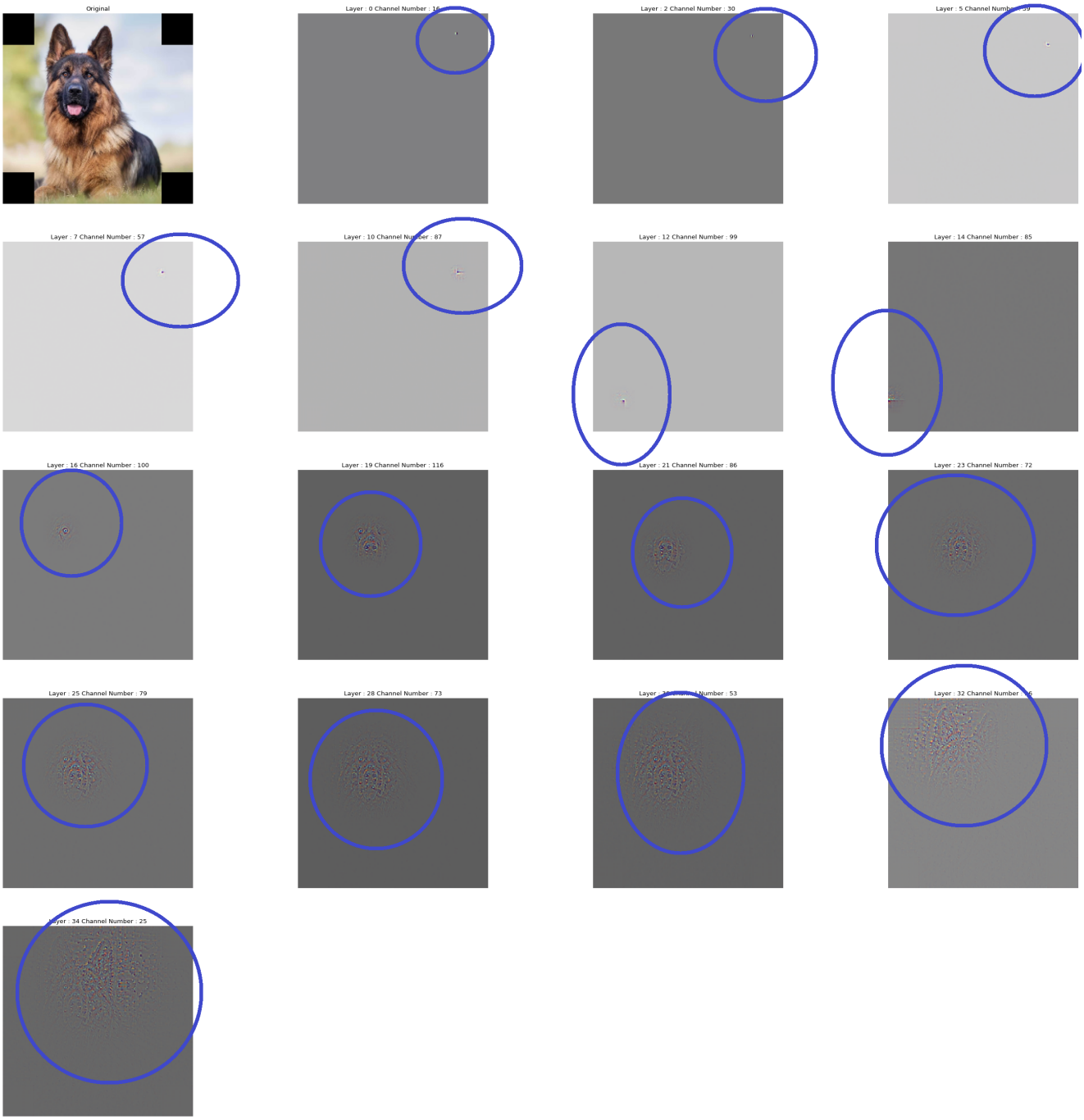

Figure 25: Feature Visualization for Dog image Corner occluded

Discussion : The dog image was corner occluded it can be seen that the lower layers learnt the features of background and as it was approaching higher layers it started learning eyes, ears and nose with a partial visualization of the face although it has correctly predicted as **'German Shepherd'** but with **60.1%** accuracy. The attended feature in each layers are highlighted in blue rectangle.

CENTER-OCCLUSION

(A) **VGG-16 Model for Architecture-1**
CAT :

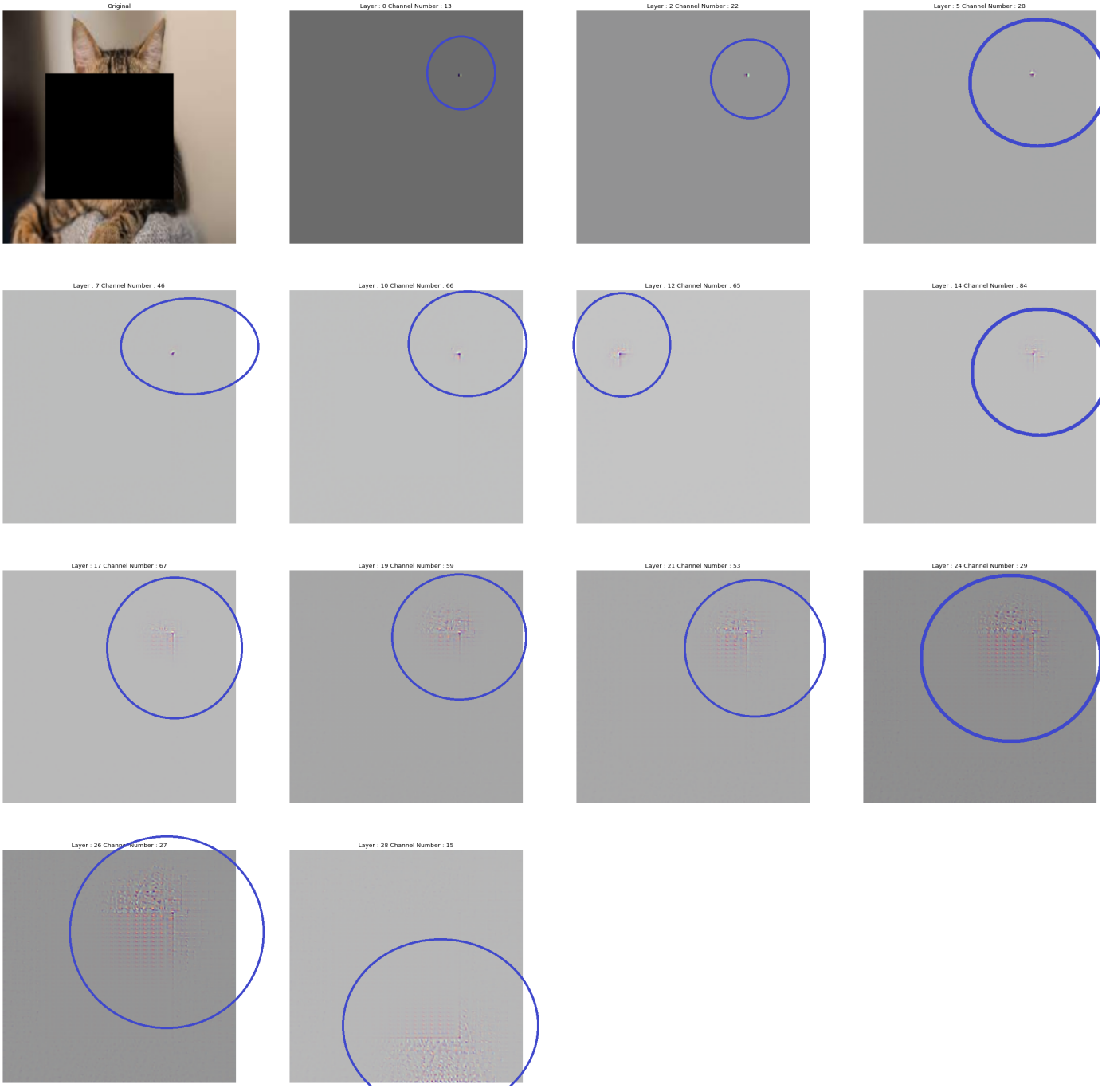

Figure 26: Feature Visualization for Cat image Center occluded

Discussion : The cat image was center occluded it can be seen that the lower layers are learning the low level features like eyes and the higher layers are learning the features like learning the face, yet it has wrongly predicted the breed of cat

as **'Egyptian cat'** with **23.4%** accuracy. The attended feature in each layers are highlighted in blue rectangle.

OSTRICH :

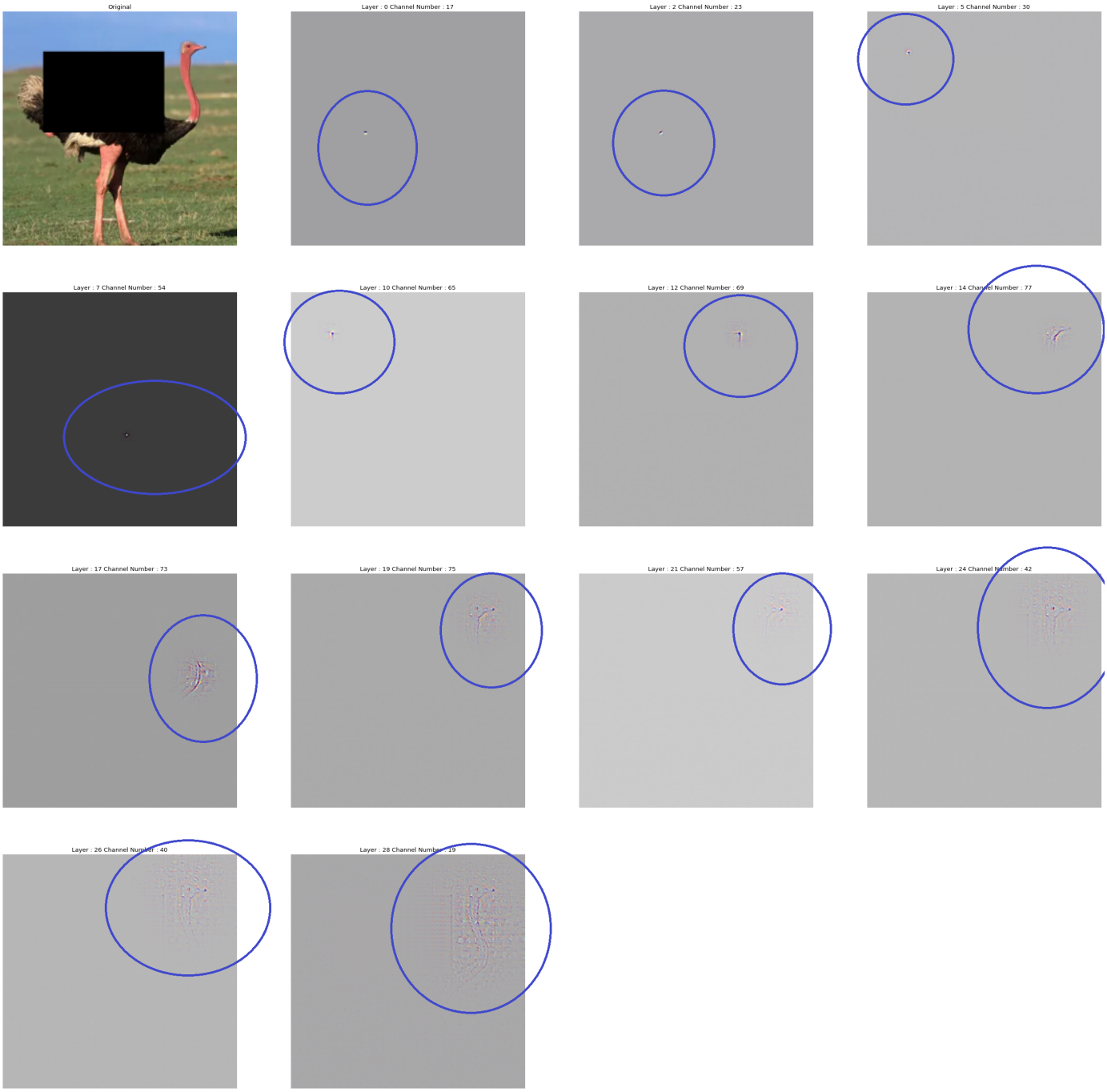

Figure 27: Feature Visualization for Ostrich image Center occluded

Discussion : The ostrich image was center occluded it can be seen that the lower layers are learning the low level features like eyes and the higher layers are learning the features like learning the neck and body of the bird, yet it has correctly predicted as **'ostrich'** with **99.9%** accuracy. The attended feature in each layers are highlighted in blue circle.

BABOON :

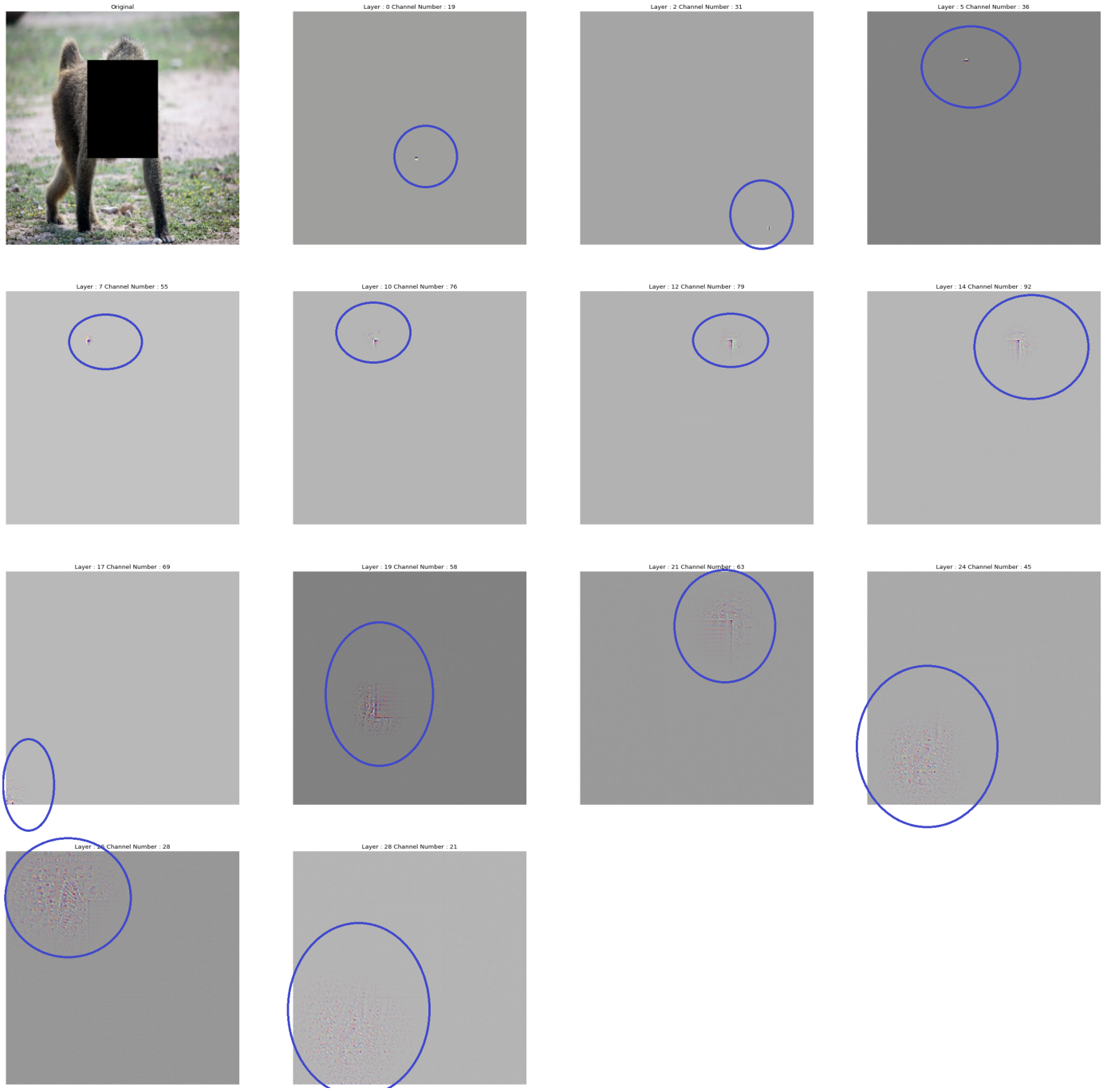

Figure 28: Feature Visualization for Baboon image Center occluded

Discussion : The baboon image was center occluded it can be seen that the lower layers did not learn the features of baboon but background corners of the image however, the higher layers from 21 directly learned the face of the baboon, and it has correctly predicted as **'baboon'** with **92.8%** accuracy. The attended feature in each layers are highlighted in blue circle.

DOG :

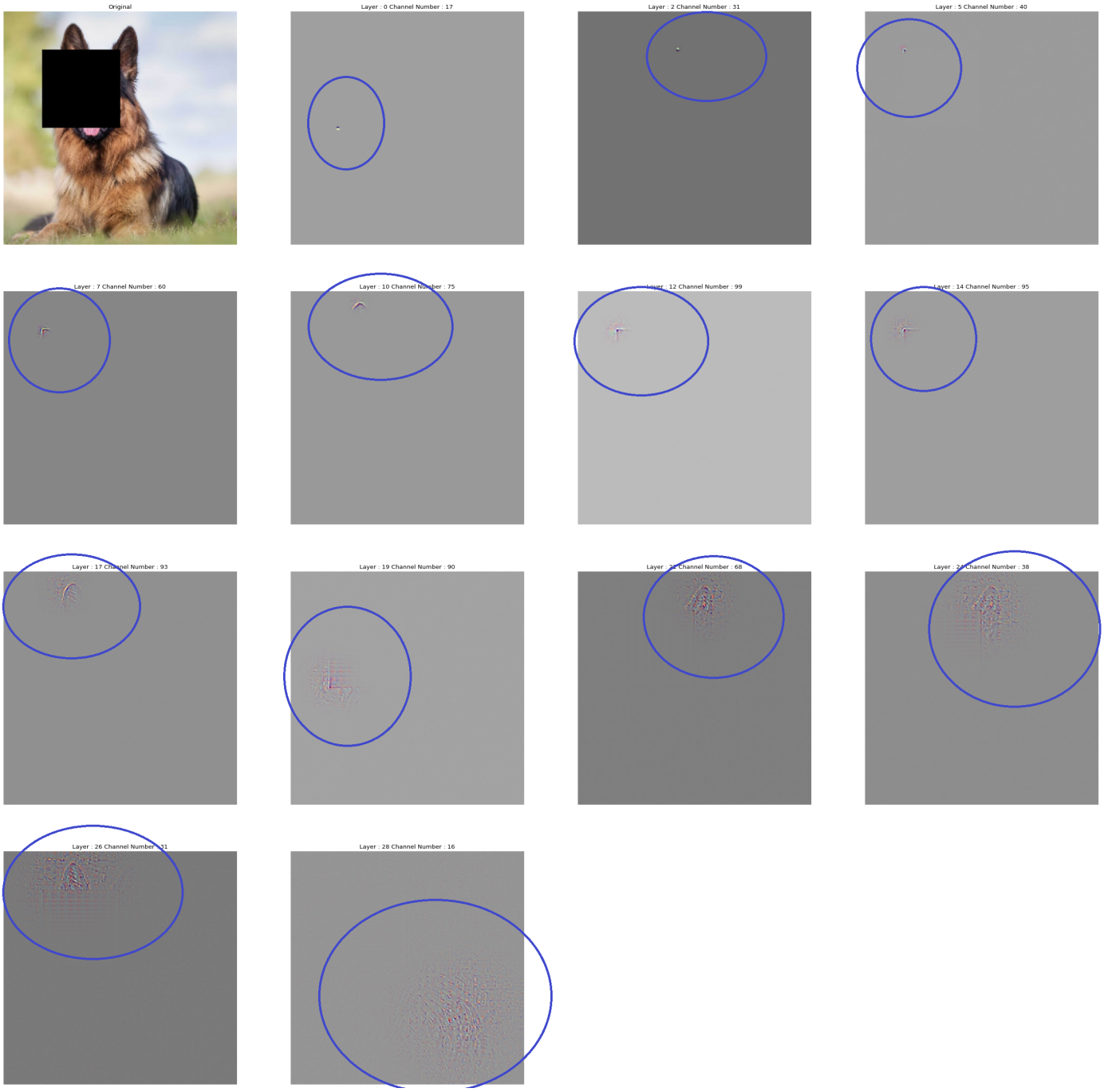

Figure 29: Feature Visualization for Dog image Center occluded

Discussion : The dog image was center occluded it can be seen that the lower layers learnt the features like eye, ears, tongue and as it was approaching higher layers it was not able to visualize the dog although it has correctly predicted as **'German Shepherd'** but with a very low accuracy of **19.6%**. The attended feature in each layers are highlighted in blue circle.

(B) **VGG-19 Model for Architecture-1**
CAT :

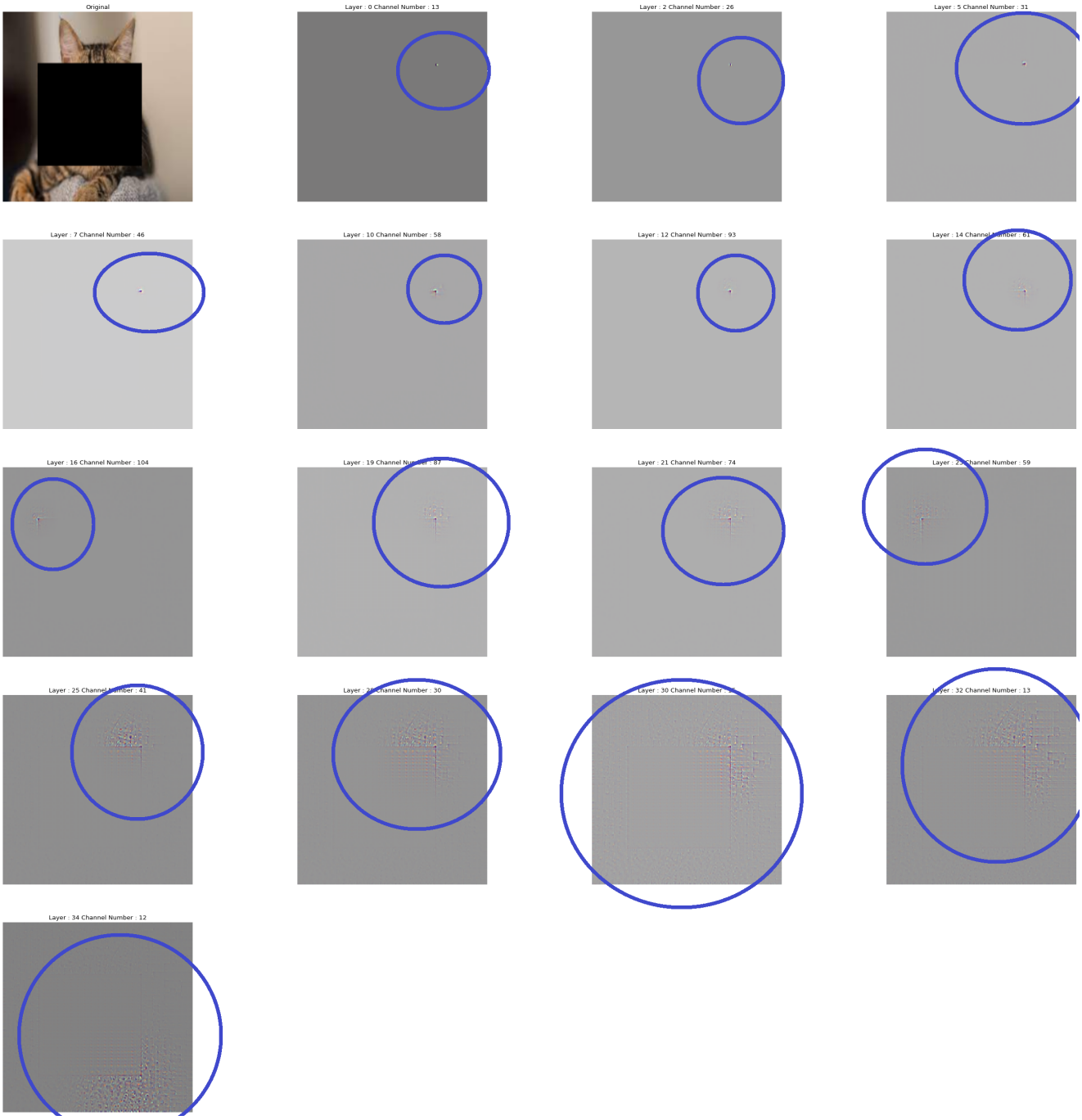

Figure 30: Feature Visualization for Cat image Center occluded

Discussion : The cat image was center occluded it can be seen that the lower layers are learning the low level features like eyes and the higher layers are learning the features like ear and it has wrongly predicted the breed of cat as **'Egyptian cat'** with **22.1%** accuracy. The attended feature in each layers are highlighted in blue circle.

OSTRICH :

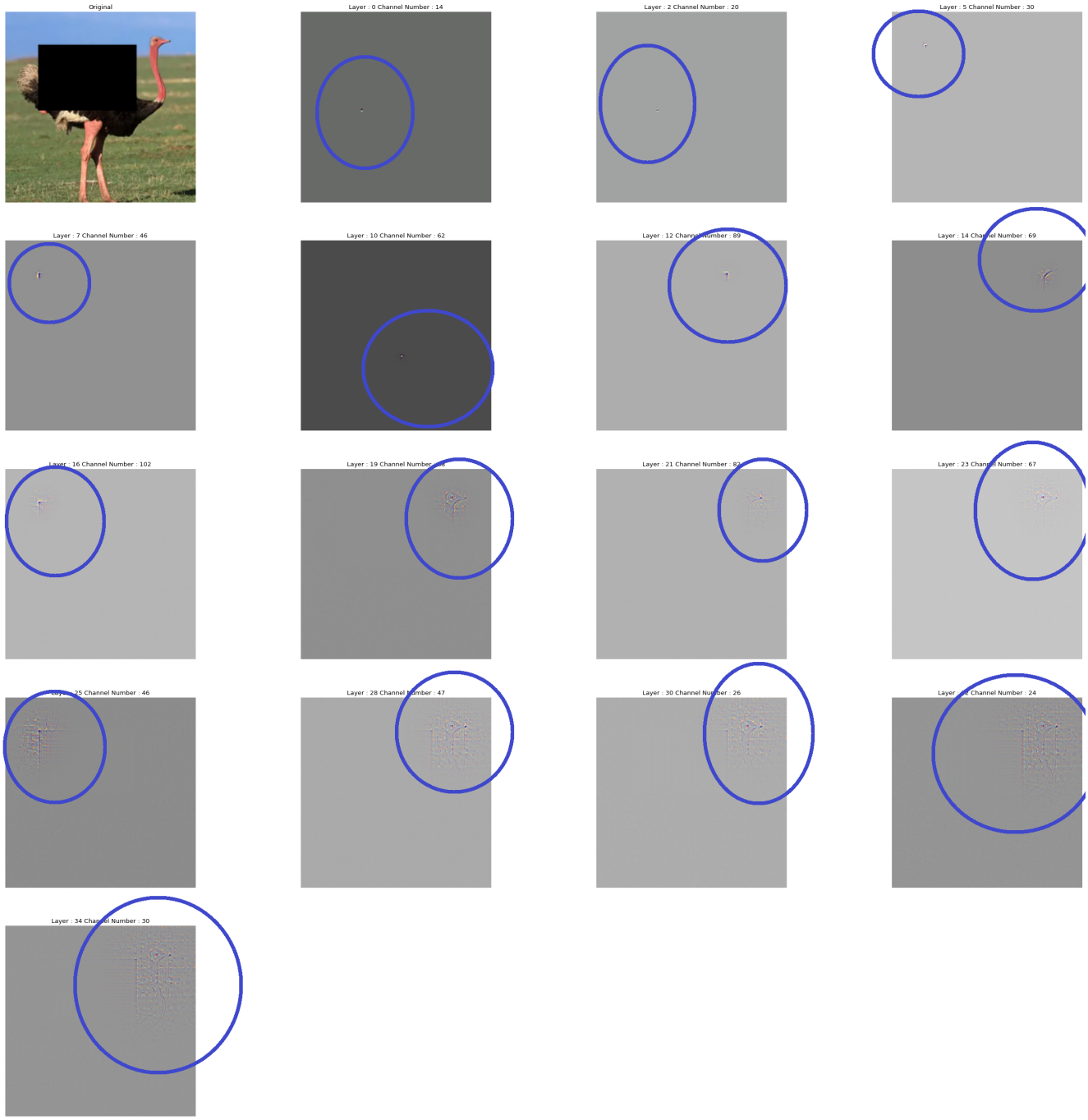

Figure 31: Feature Visualization for Ostrich image Center occluded

Discussion : The ostrich image was center occluded it can be seen that the lower layers are learning the low level features like eyes and the higher layers are learning the features like learning the neck and face of the bird, yet it has correctly predicted as **'ostrich'** with **96.6%** accuracy. The attended feature in each layers are highlighted in blue circle.

BABOON :

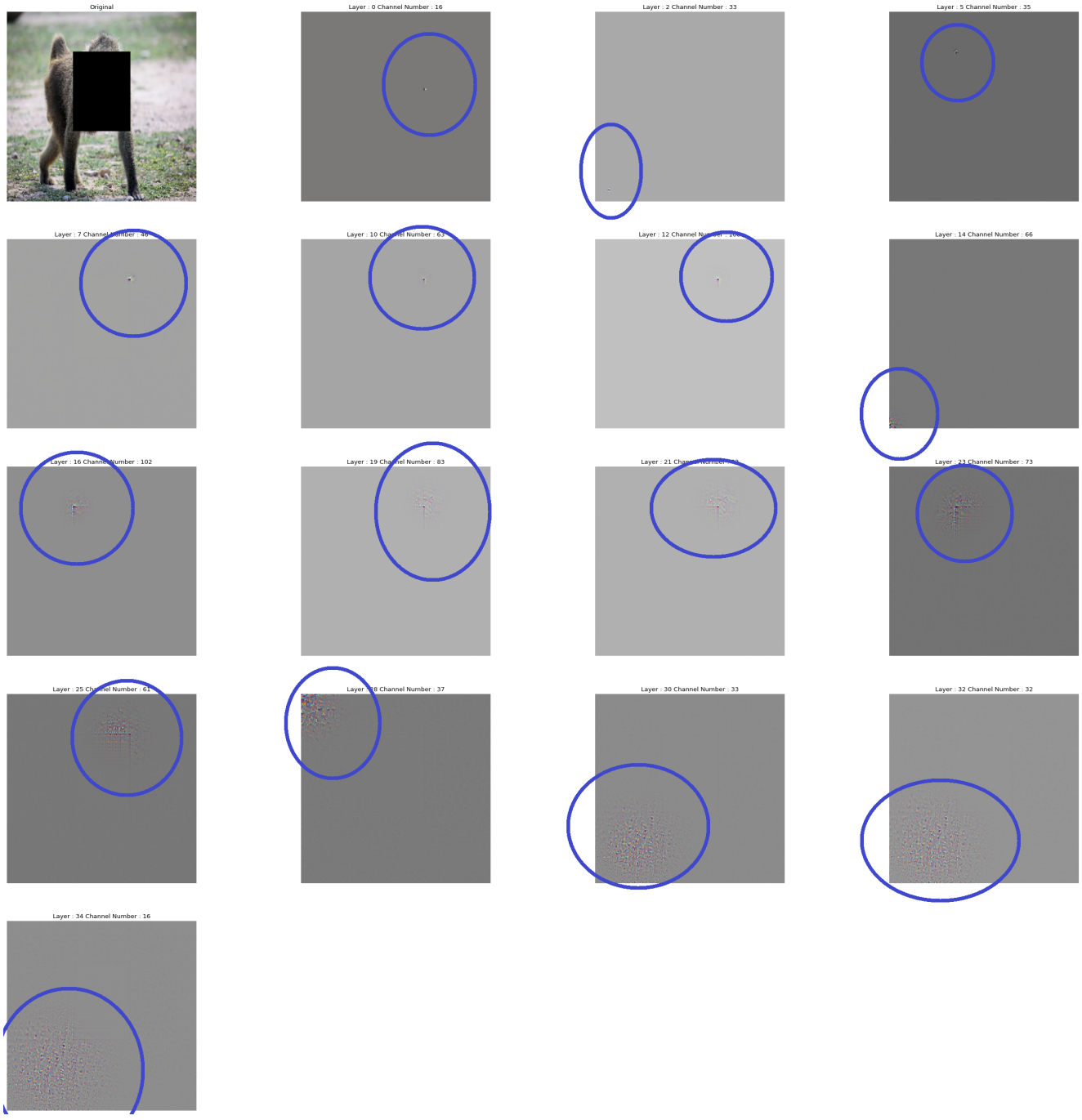

Figure 32: Feature Visualization for Baboon image Center occluded

Discussion : The baboon image was center occluded it can be seen that the lower layers did not learn the features of baboon but background corners of the image even in the higher layers it just got the background of image yet it has correctly predicted as **'baboon'** although with low accuracy of **42.9%**. The attended feature in each layers are highlighted in blue circle.

DOG :

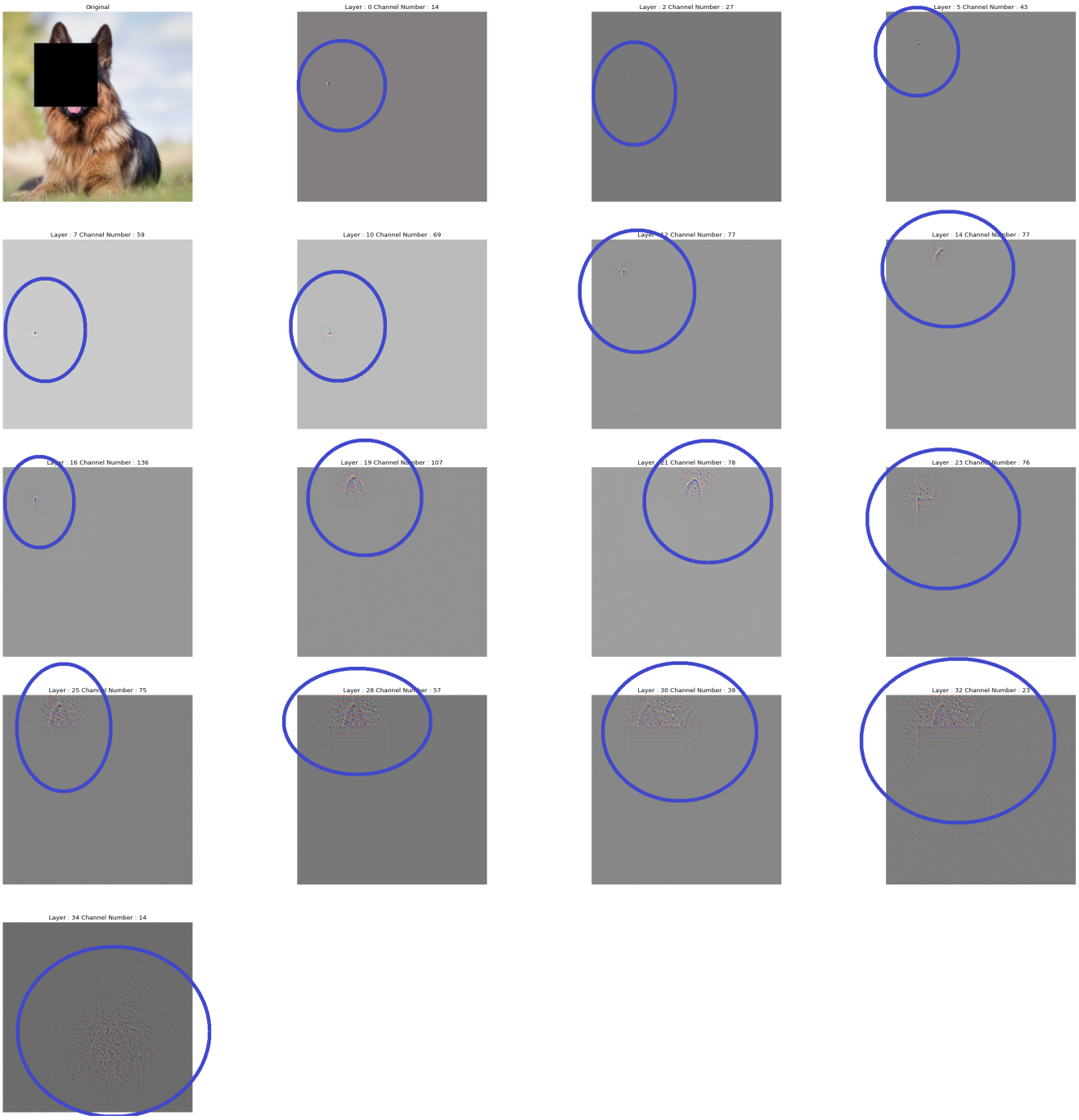

Figure 33: Feature Visualization for Dog image Center occluded

Discussion : The dog image was center occluded it can be seen that the lower layers did not learn any features and as it was approaching higher layers it was able to visualize only the ears of the dog although it has correctly predicted as **'German Shepherd'** but with a very low accuracy of **28.3%**. The attended feature in each layers are highlighted in blue circle.

6 Architecture 2

6.1 Architecture

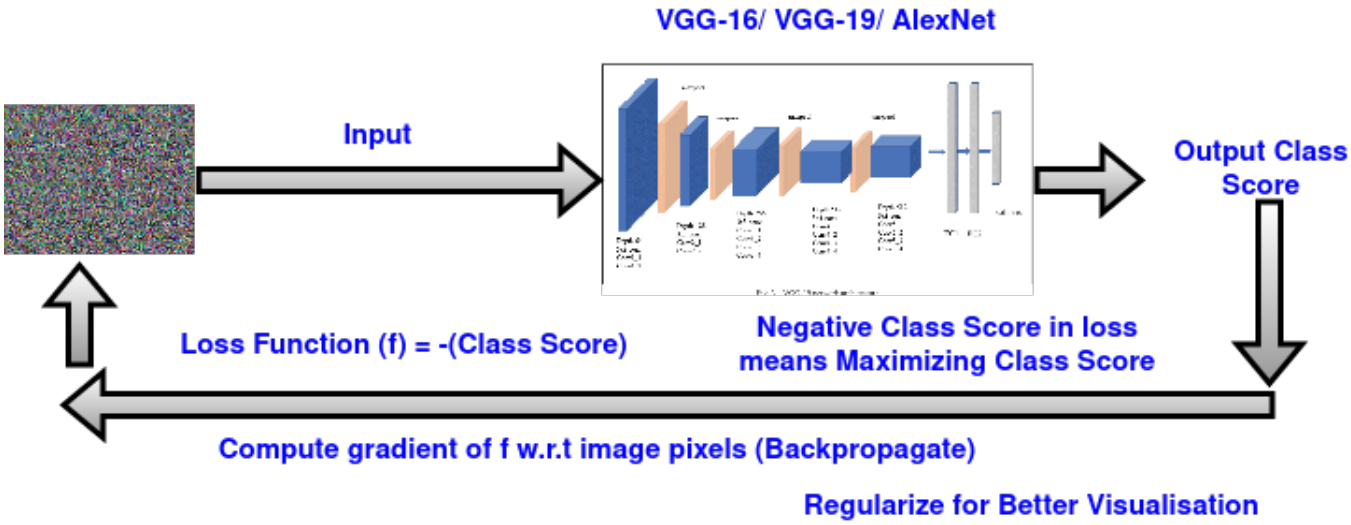

Figure 34: This architecture is the schematic representation of the end to end pipeline inspired from [7].

6.1.1 Description :

In this architecture, we will start with a random trainable image and provide it as input to a CNN model(ex: Vgg-16/19/Alexnet). Our aim is to synthesize an image that would maximize a specific class score.

The network we use will be a pre-trained one. We will also perform some regularization techniques to better visualize the image to belong to a specific class.

We will also generate heatmap using occlusion sensitivity. Our aim will be to localize the object the model focuses on, to classify an image as belonging to a particular class.

6.2 Comparing various regularization's

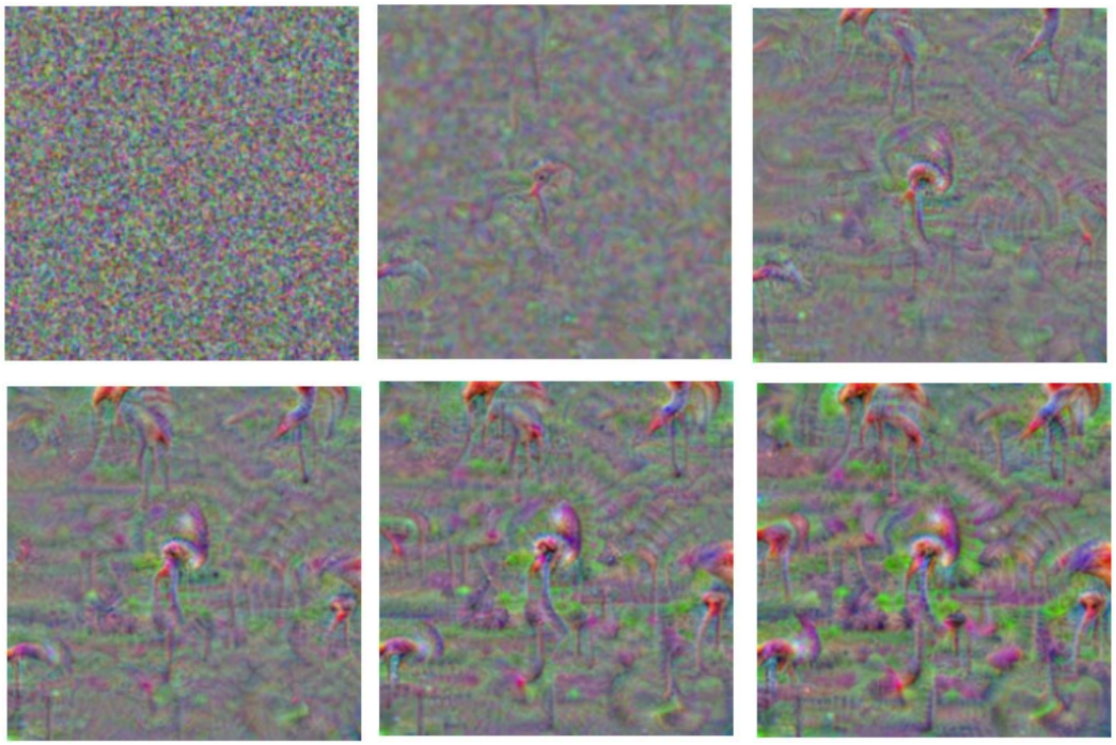

Figure 35: a, b, c, d, e, f - maximizing the class score progressively

6.2.1 Analysis

Starting with a random image, we use stochastic gradient descent to try to maximize the class score for images of a particular class. A picture of flamingos has been used, and although figure 35(a) maximized the class score for flamingos, it is very difficult to actually make out where the flamingos are. We use L2 regularization to help with the over-fitting problem to get the image shown in figure 35(b) [7]. Gaussian blur also helps because we can blur out the higher frequency ones, so we use this on every alternate iteration. This makes it progressively easier to visualize the flamingos (35(c), 35(d), 35(e)). Finally, we use clipping for normalization to clip pixels which go beyond the range in 35(f). Pretrained VGG-16 model was used for this, but we can generalize it to other models as well. We have repeated the same procedure for VGG-19 and AlexNet to get similar results.

6.3 Heat map generation using occlusion sensitivity

In this section we compute the heat-map followed by Gaussian blur, as shown in the following figure.

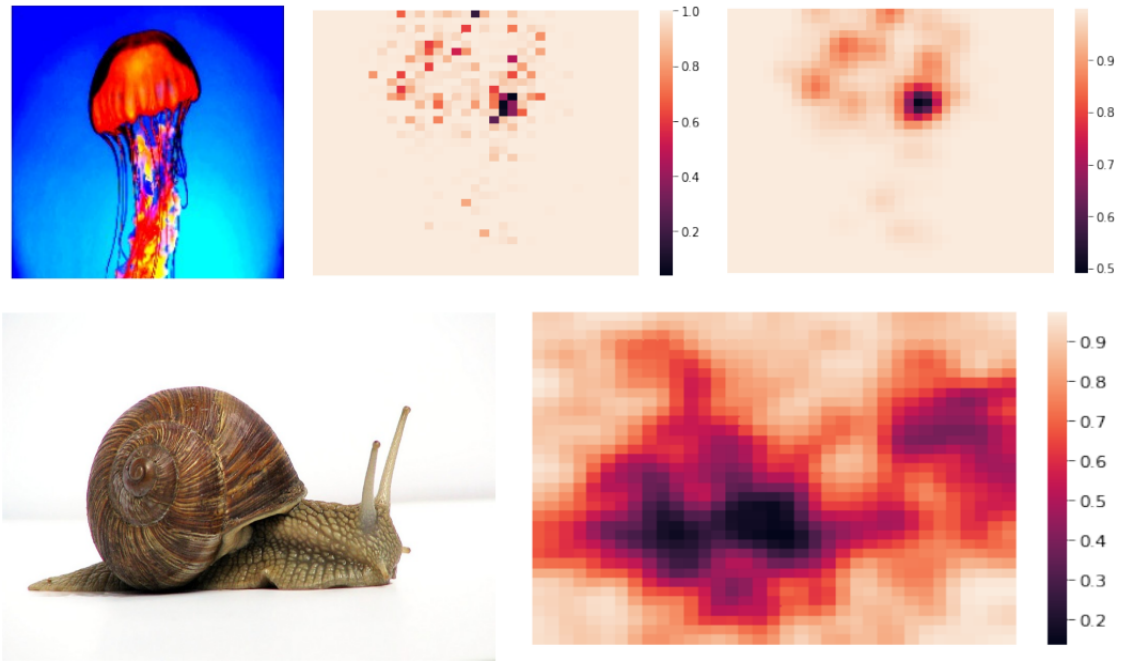

Figure 36: a,b,c,d,e - generating heat maps

6.3.1 Analysis

Similar to how kernels stride across the input image in CNNs, we take a black patch and stride it across the image to see how the probability of correct classification varies with respect to the area occluded. Using an image of a jellyfish (figure 36(a)), we show that the probability of correct classification remains high when the background water is occluded, but it drops significantly when the head is occluded.

If the patch size is larger, then it takes lesser time for the patch to stride across the image, and hence less time to generate the heat maps, but the heat maps look less fine grained. A smaller patch size leads to better localization, as shown in figure 36(b) This is followed by Gaussian blur to make the heat map look smoother, as shown in figure 36(c). Covering significant portions of the image leads to a sharp drop in probability values (for the correct class) as shown by the scale in the figure. However, covering the background maintains the probability values at about 0.9-1. From this, we can get an idea of what part of the image is being focused on when trying to classify the image. Similarly, we can repeat this process for a snail (figure 36(d), 36(e)).

Suppose the patch size is of 50*50 size and the image is of 224*224 size. With a stride of length 5, we will have the height of the heat-map to be around $\left(\frac{224-50}{5}\right) \approx 35$. So the size of heat-map will be around 35*35 which means we will have to classify

the image using the model 1225 times. This takes a bit of time (around 15 minutes on Google Colab CPU run-time). If we take smaller strides or smaller size of patch, then the process becomes slower but we will have better localization. If we take larger patch size and bigger strides, then we will have results faster but the localization will suffer. Based on this trade off we have generally seen decent results for 50*50 size of patch with a stride length of 5 and we resize the image to 224*224.

To further check that the correct areas are being highlighted on the heat map, we can project the heat map back onto the image and see if they align. This is done by simple matrix multiplication as shown in the code, and we find that the two images align.

7 Work Divisions

7.1 Deliverable & Timelines

Deliverables	**Date**
Goals and objective setting	3 to 4-Nov
Report Refinement	5 to 6-Nov
Proposal Report Submission	7-Nov-2021
Vgg16 Model for Architecture 1	8 to 10-Nov
Vgg19 model for Architecture 1 Alexnet model for Architecture 1	11 to 13 Nov
Report, Results, PPT for Mid Evaluation	14 to 16-Nov
Vgg16 model for Architecture 2	17 to 20 Nov
Vgg19 model for Architecture 2 Alexnet model for Architecture 2	21 to 24 Nov
Result Analysis of all models	25 to 26-Nov
Final PPT and Report	27 to 30-Nov

Deliverable with Timeline

Name	**Task1**	**Task2**	**Task3 (as per Task2)**
Alefiah	Project proposal formation	Design (Architecture 1) Vgg16(Architecture 1) Vgg19(Architecture 1) Alexnet(Architecture 1)	PPT, Coding, Final Report
Ankita	Project proposal formation	Design (Architecture 2) Vgg16(Architecture 2) Vgg19(Architecture 2) Alexnet(Architecture 2)	PPT, Coding, Final Report
Seshadri	Project proposal formation	Design (Architecture 1) Vgg16(Architecture 1) Vgg19(Architecture 1) Alexnet(Architecture 1)	PPT, Coding, Final Report
Vivek	Project proposal formation	Design(Architecture 2) Vgg16(Architecture 2) Vgg19(Architecture 2) Alexnet(Architecture 2)	PPT, Coding, Final Report
Ablation studies(optional) if time permits			

Work Division

References

[1] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," pp. 5188–5196, 06 2015.

[2] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional neural networks," vol. 8689, 11 2013.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[4] B. Bermeitinger, S. Donig, M. Christoforaki, A. Freitas, and S. Handschuh, "Vgg19," 10 2017.

[5] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[7] A. N. T. F. H. L. Jason Yosinski, Jeff Clune, "Understanding neural networks through deep visualization," 06 2015.