CS671: Deep Learning and Applications
Programming Assignment 3
Group 16

Anmol Bishnoi (B19069)
Anshul (B19150)
Rachita (B19258)

Convolution using a 3x3 filter

We’ve changed the dimensions of all images to 224x224.
We have taken one image from each of the three classes and convolved it with a 3x3 convolutional filter initialized using kaiming initialization.

Parameter considered for convolution are -
Stride (S) = 1
Padding (P) = 0
F = 3

Expected dimension of the feature map using formulas :-
W2 = ((W1 - F + 2*P) / S) + 1
H2 = ((H1 - F + 2*P) / S) + 1

W2 = 222, H2 = 222

The dimensions obtained through code were same as above.

Images considered

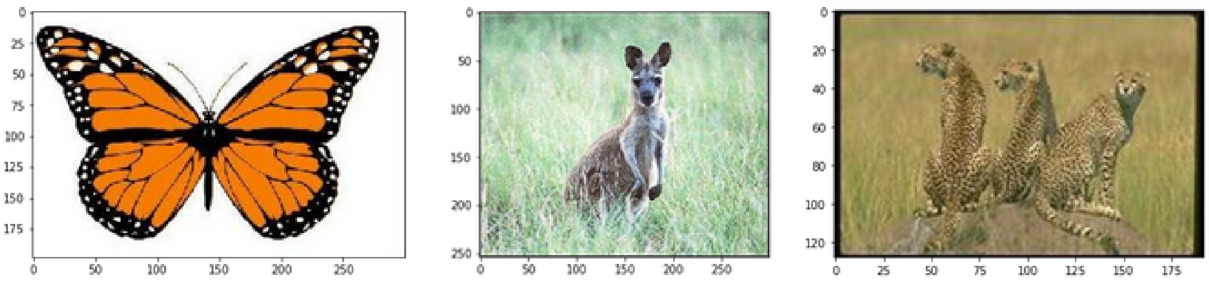
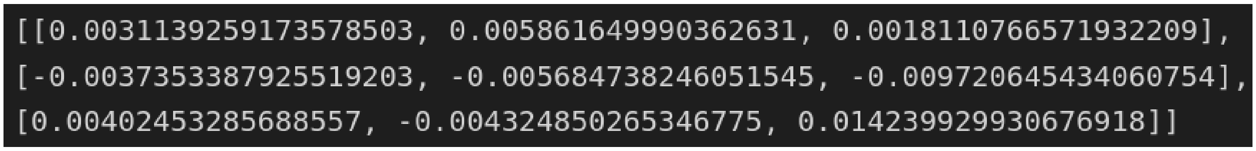


Image1 - Butterfly Image2 - Kangaroo Image3 - Leopard

Fig. 1 Images taken from each of the three classes

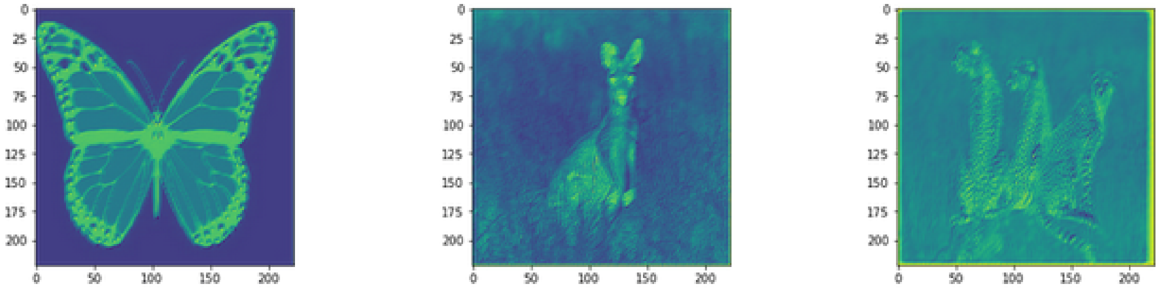
Convolutional filter -



```
[[0.0031139259173578503, 0.005861649990362631, 0.0018110766571932209],
[-0.0037353387925519203, -0.005684738246051545, -0.009720645434060754],
[0.00402453285688557, -0.004324850265346775, 0.014239929930676918]]
```

Fig. 2 Filter values

Filter maps



filtermap1 - Butterfly

filtermap2 - Kangaroo

filtermap3 - Leopard

Fig. 3 Feature maps obtained from each of the three considered images

Convolution using two layers

We have stacked two convolutional layers each with a ReLU activation function.

Layer1 -
Number of filters = 32
Dimension of filters = 3x3
Initialization method = Kaiming initialization

Parameters considered for convolution are -
Stride (S) = 1
Padding (P) = 0
F = 3

Expected dimension of the feature map using formulas :-
W2 = ((W1 - F + 2*P) / S) + 1
H2 = ((H1 - F + 2*P) / S) + 1

W2 = 222, H2 = 222, Depth = 32

Now, since the number of filters used are 32, we will obtain 32 such filter maps each with a dimension of 222x222. Therefore, the input image to the next layer will be of dimension 222x222x32 (Depth will be equal to the number of filters).

Filter images

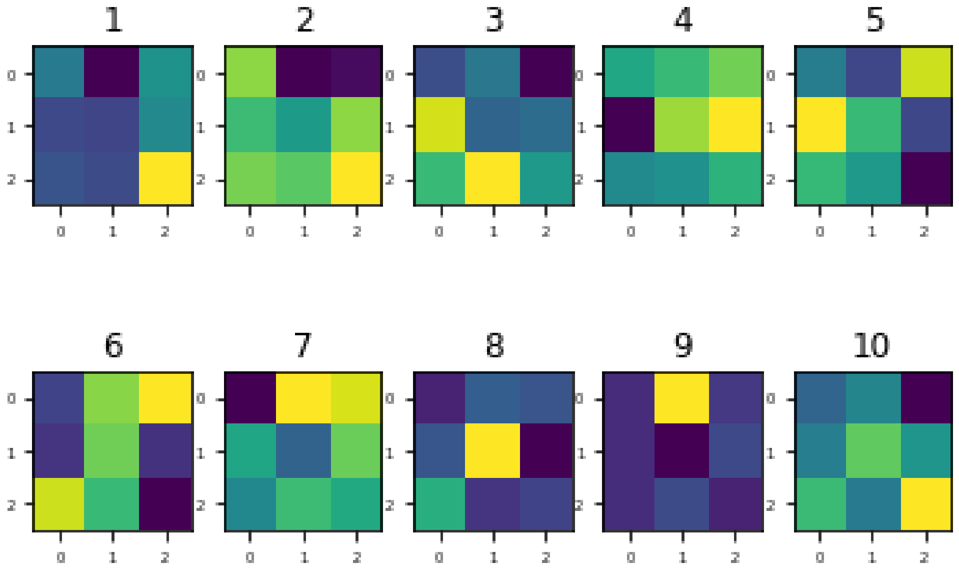


Fig. 4 Images of first 10 filters (out of 32) used for convolution in the first layer

Feature maps

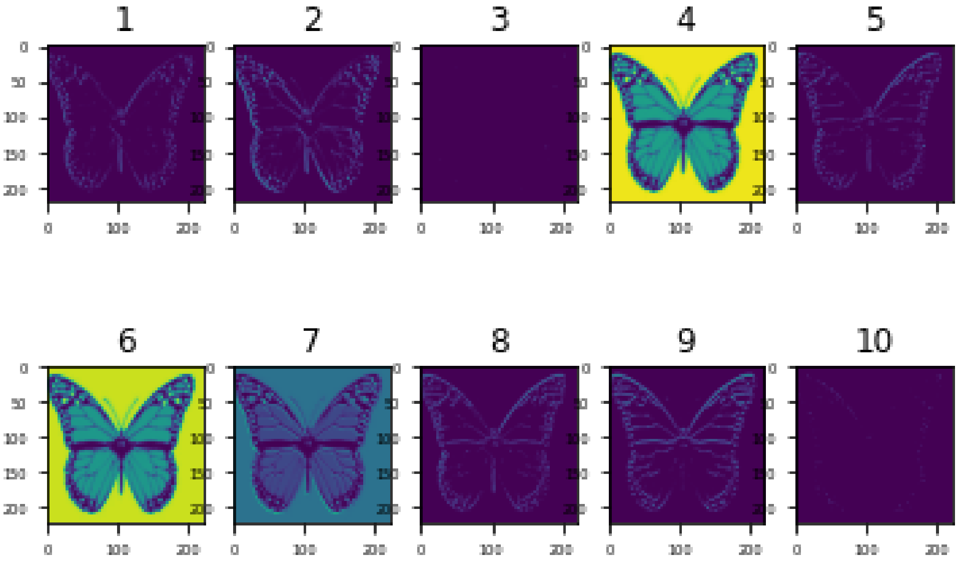


Fig. 5 Images of feature maps corresponding to the first 10 filters (butterfly)

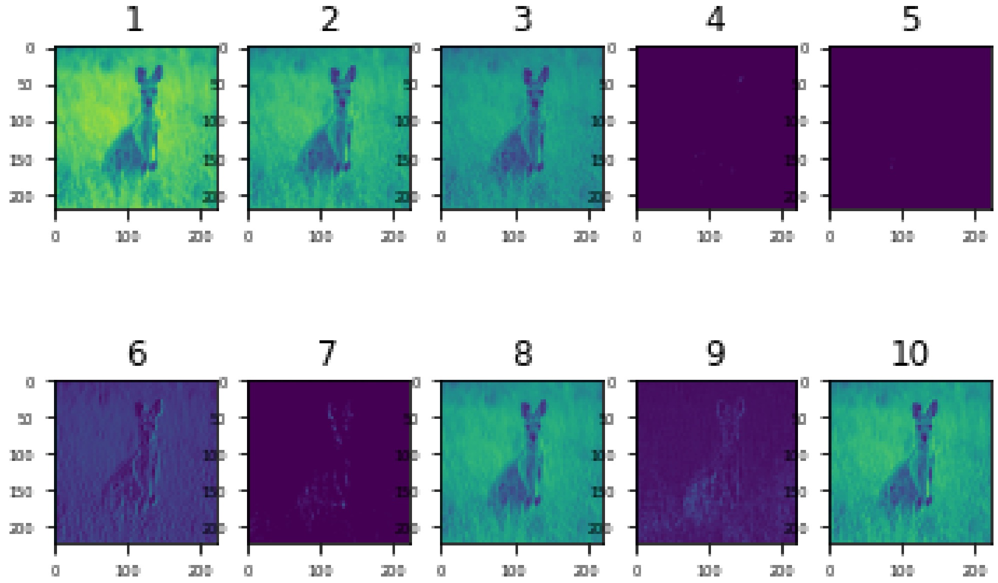


Fig. 6 Images of feature maps corresponding to the first 10 filters (Kangaroo)

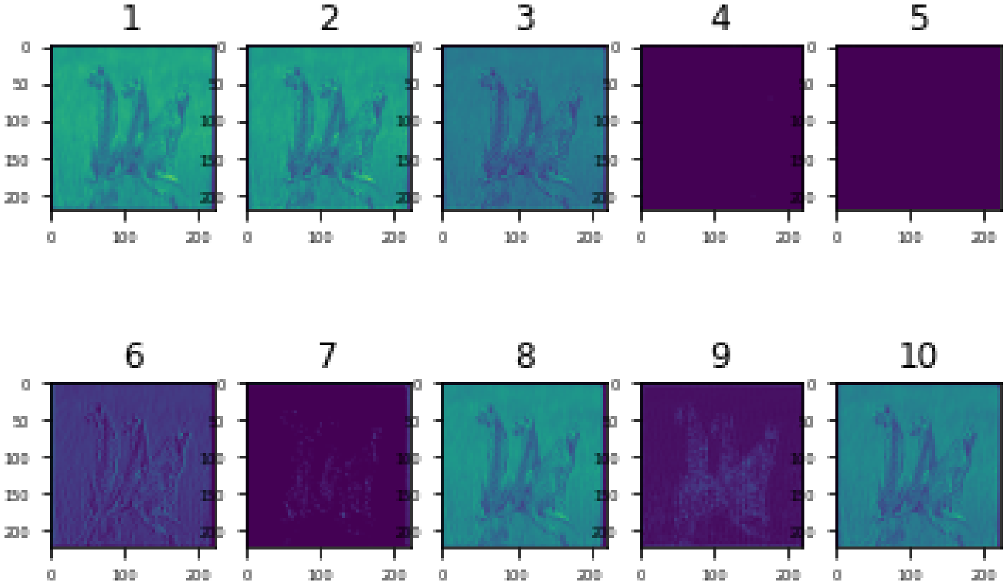


Fig. 7 Images of feature maps corresponding to the first 10 filters (Leopard)

Layer2 -
Number of filters = 64
Dimension of filters = 3x3x32 (The depth of each filter will be equal to the depth of the input image)
Initialization method = Kaiming initialization

Parameters considered for convolution are -
Stride (S) = 1
Padding (P) = 0
F = 3

Expected dimension of the feature map using formulas :-
$W2 = ((W1 - F + 2*P) / S) + 1$
$H2 = ((H1 - F + 2*P) / S) + 1$

W2 = 220, H2 = 220, Depth = 64

Now, since the number of filters used are 64, we will obtain 64 such filter maps each with a dimension of 220x220. Therefore, the input image to the next layer will be of dimension 220x220x64 (Depth will be equal to the number of filters).

Filter images

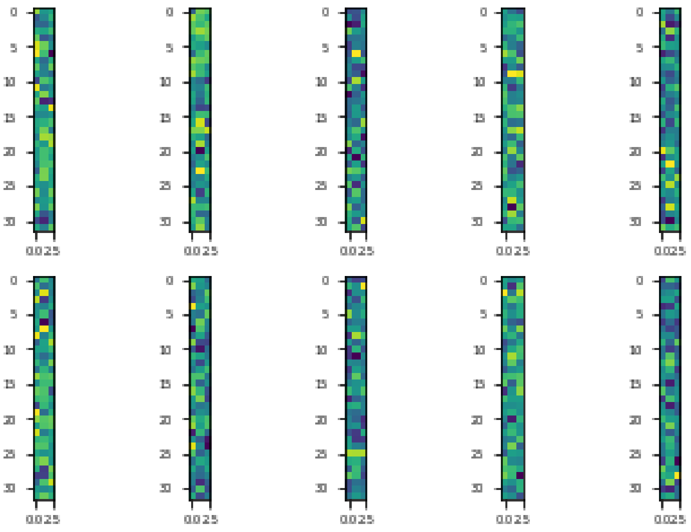


Fig. 8 Images of first 10 filters

Feature maps

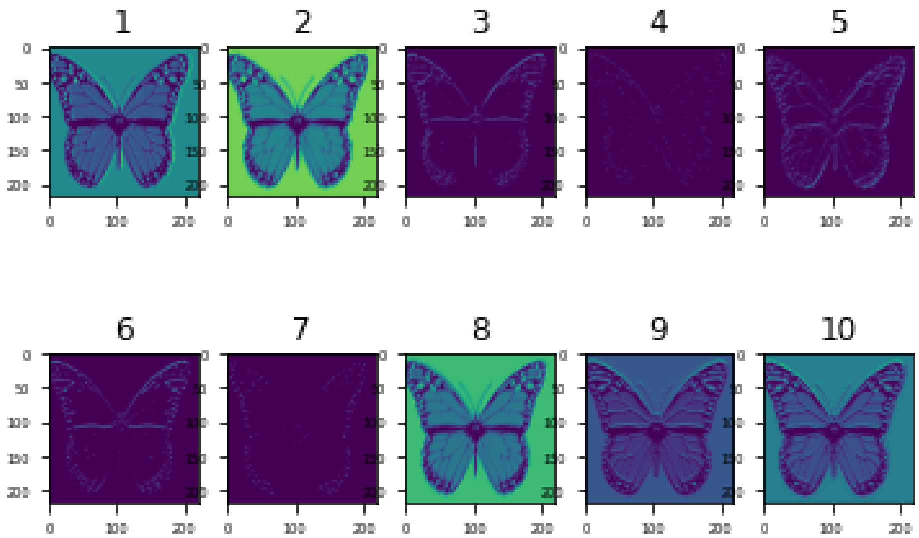


Fig. 9 Images of feature maps corresponding to the first 10 filters (Butterfly)

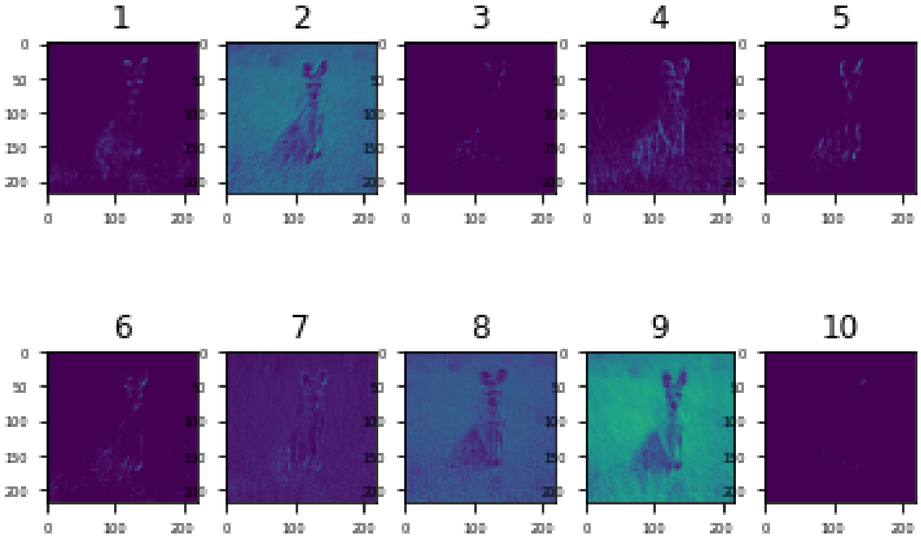


Fig. 10 Images of feature maps corresponding to the first 10 filters (Kangaroo)

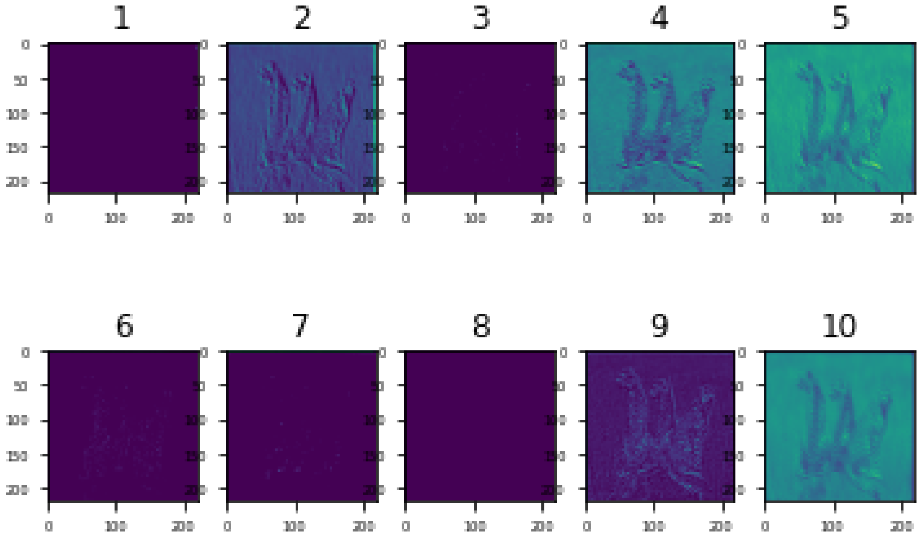


Fig. 11 Images of feature maps corresponding to the first 10 filters (Leopard)

Retrained VGG19

VGG19 pretrained on ImageNet was imported using Keras. The final classification layer in the model with 1000 different classes was dropped to add a new output layer to classify among the three classes of Butterfly, Kangaroo and Leopard.

Only the output layer was trained with the weights of all previous layers, kept unchanged.

The final classification accuracies are as follows:

Train Accuracy - ***1.00***
Val Accuracy - ***1.00***
Test Accuracy - ***0.983***

Test Confusion Matrix:

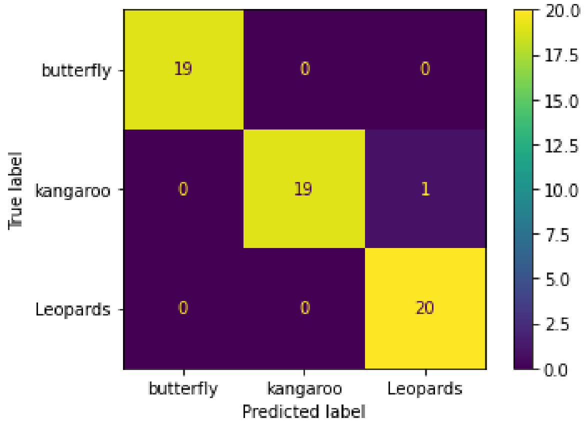


Fig 12. Confusion Matrix

Visualising Patches that maximally activate neurons

The last convolution layer in VGG19 is of dimension 14x14x512

For Butterfly

Input Image:

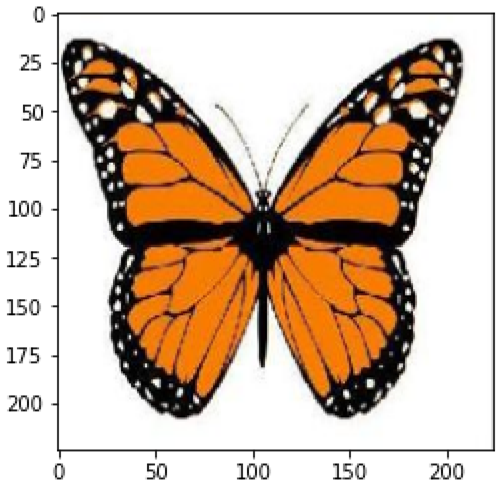


Fig. 13 Butterfly

Maximally Excited Neurons are

```
(4, 6, 103), (3, 6, 103), (2, 2, 373), (4, 7, 103), (5, 6, 359)
```

The patches that these represent are:

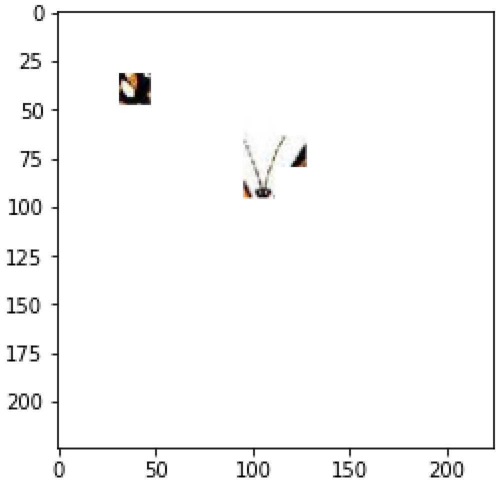


Fig. 14 Maximally activating patches in Butterfly

For Kangaroo

Input Image:

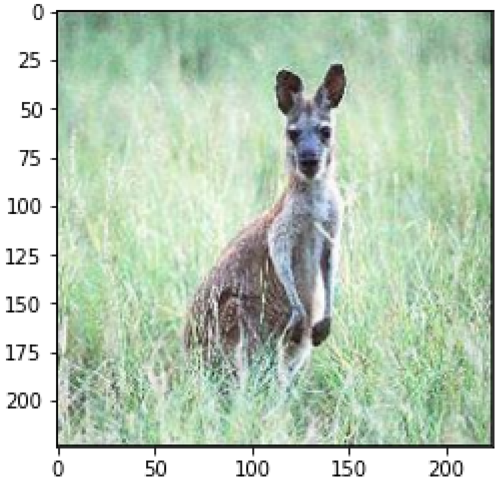


Fig. 15 Kangaroo

Maximally Excited Neurons are

```
(2, 7, 349), (3, 8, 349), (2, 8, 349), (3, 7, 349), (4, 8, 446)
```

The patches that these represent are:

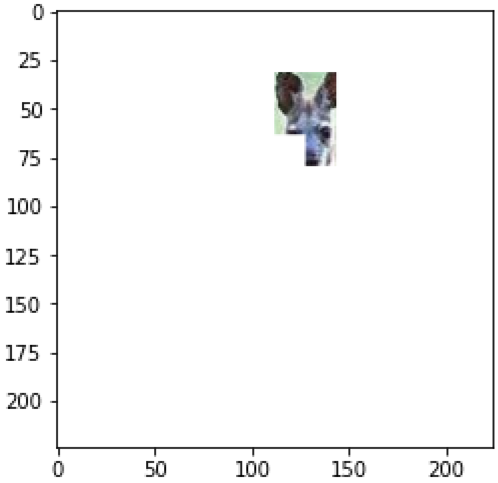


Fig. 16 Maximally activating patches in Kangaroo

For Leopard

Input Image:

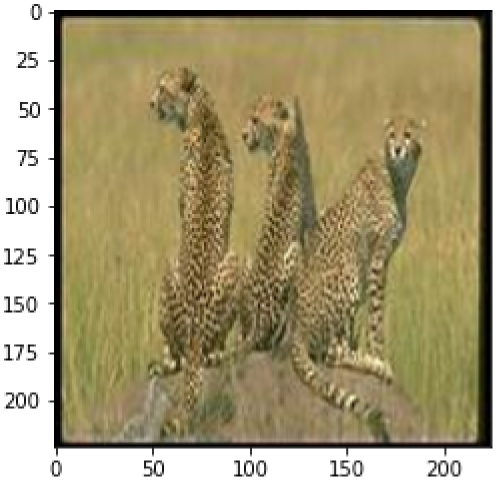


Fig. 17 Leopard

Maximally Excited Neurons are

```
(5, 11, 200), (4, 11, 200), (5, 10, 200), (4, 10, 183), (5, 10, 91)
```

The patches that these represent are:

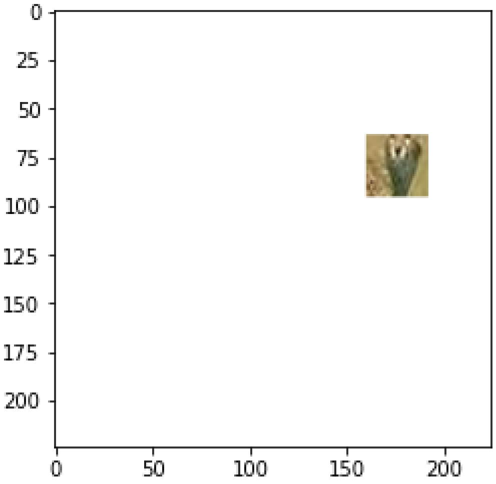


Fig. 18 Maximally activating patches in Leopard

From all three experiments as run above, we can notice very clearly that every single time, the portion of the image that caused the maximum activation of any group of neurons was the face of the animal.

The animals given to us being a butterfly, kangaroo and leopard have drastically different facial structures and can be easily understood as defining features for each individual animal and that is exactly what the CNN has caught upon and highly specialised on.

Guided Backpropagation

For Butterfly

Input Image:

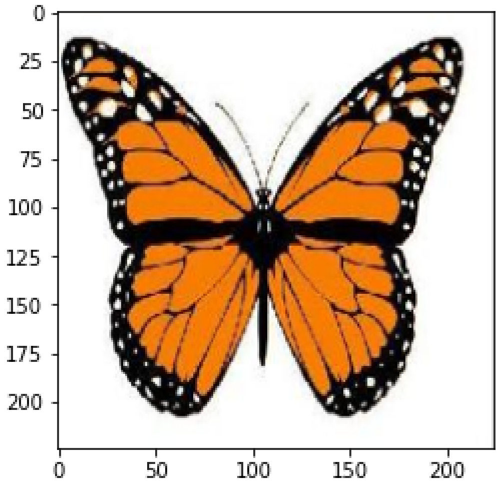


Fig. 19 Butterfly

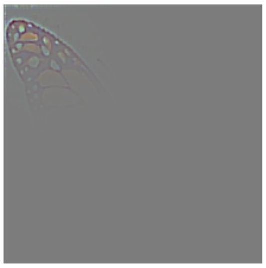
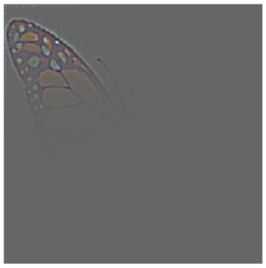
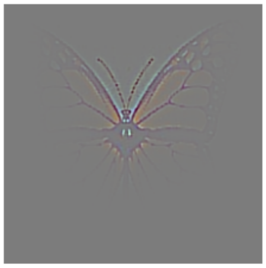
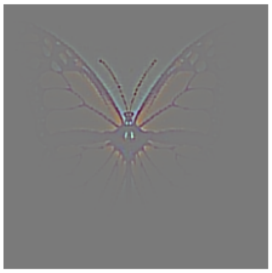
Position of Neuron in last layer	Neuron’s Ouput	Resultant Gradient Image
(0,0,0)	0	
(0,0,510)	26.967373	
(2,2,373)	267.20782	
(4,7,103)	242.79477	
(4,6,103)	306.61386	

Table 1 For Butterfly

For Kangaroo

Input Image:

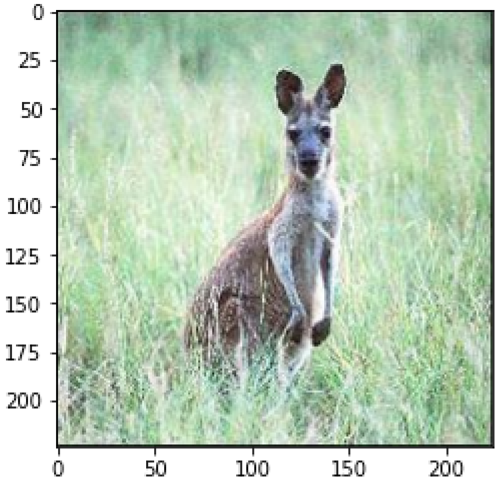
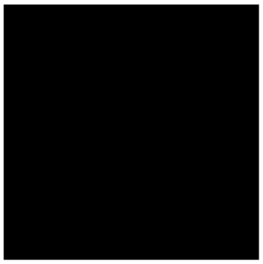


Fig. 20 Kangaroo

Position of Neuron in last layer	Neuron’s Ouput	Resultant Gradient Image
(0,0,0)	0	
(0,0,12)	20.5028	

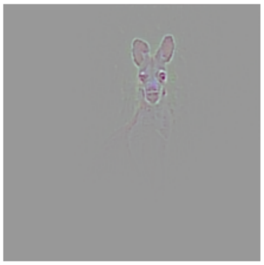
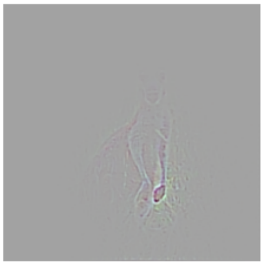
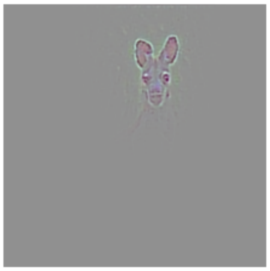
(4,8,446)	396.82816	
(9,8,0)	19.151016	
(2,7,349)	495.85565	

Table 2 For Kangaroo

For Leopard

Input Image:

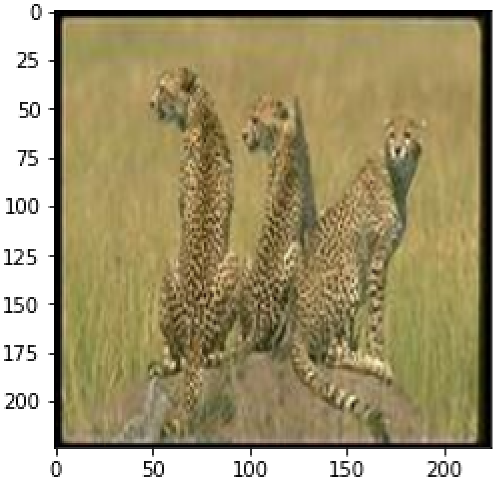


Fig. 21 Leopard

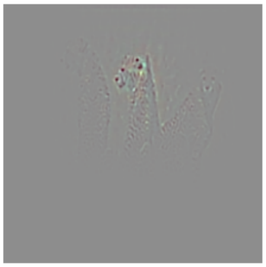
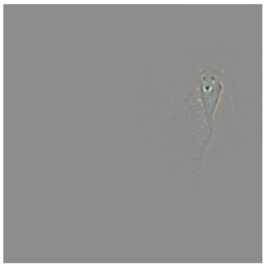
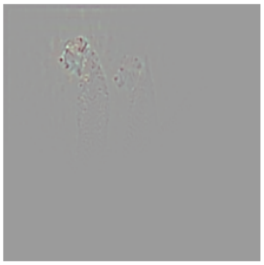
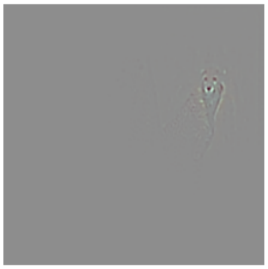
Position of Neuron in last layer	Neuron’s Ouput	Resultant Gradient Image
(0,0,0)	0	
(4,7,0)	41.999374	
(5,11,200)	128.2857	
(3,4,0)	16.86093	
(4,10,183)	105.82069	

Table 3 For Leopard

In the above experiments, we performed guided backpropagation to evaluate the regions that had an influence over what the final layers in the Convoluted Neural Network gave as an output.

The Convolutional Layers use ReLu as their activation function which gives an output of zero for any value less than or equal to zero. Guided Backpropagation considers only positive gradients that too only for neurons with a positive output value.

Thus we can see and confirm in above experiments that wherever a neuron gave output as zero (first neuron for all three animals), our resultant gradient images were always blank.

Further we see that for the images that aren't all black, the gradient images never contained any information about the background which also makes sense since the background doesn't really help with classification at all.

Additionally, we once again notice that the resultant gradient images contain the faces of animals more often than not, thus confirming our hypothesis about facial features being a more important basis for classification for atleast the subset of animals that we have gotten.