CS671 - Group 8

Assignment 03

Akash Karnatak (B19147)
Laishram Ponghtangamba (B19011)
Abhishek Mishra (B19231)

Report

Ques -1 :

The images shown below were used for the question .
The original input image shape in every class case was (224,224,3), and when converted to grayscale shape changed to (224, 224, 1).

The expected dimension of the image =
(((nH_prev - f + 2P) / stride)+1, ((nW_prev-f+2P)/stride)+1, nC_prev).

Putting values in the given formula we get ,
Formula expected shape = (222, 222, 1) .

The exact same shape was indeed obtained through the code as well .
Z.Shape = A.shape = (222, 222, 1).
This was true for every class.

Kangaroo Class :

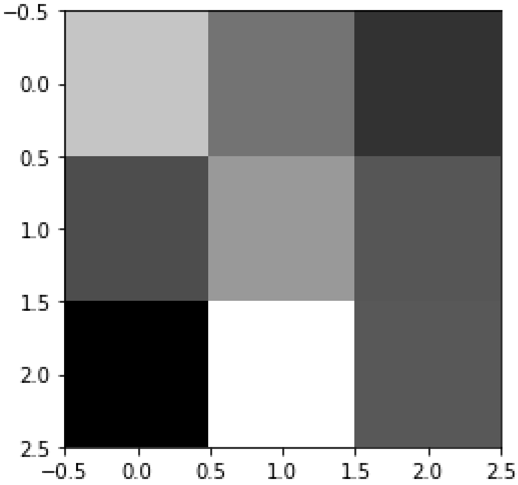

Fig 1 : Initialized filter for kangaroo

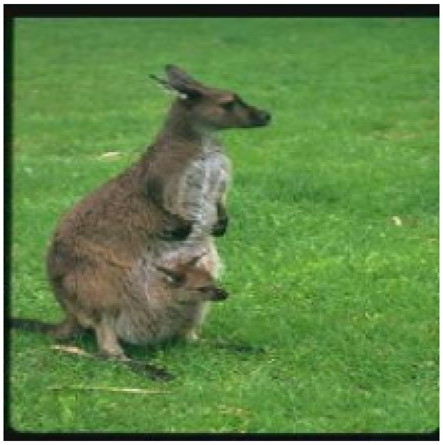

Fig 2 : Kangaroo considered for Q1 (rgb)

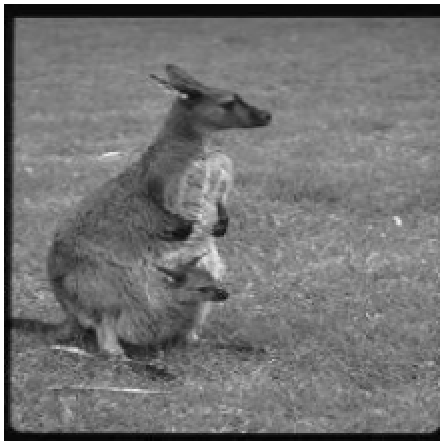

Fig 3 : Kangaroo considered for Q1 (grayscale)

After convolution following feature maps were obtained (before and after applying activation function).

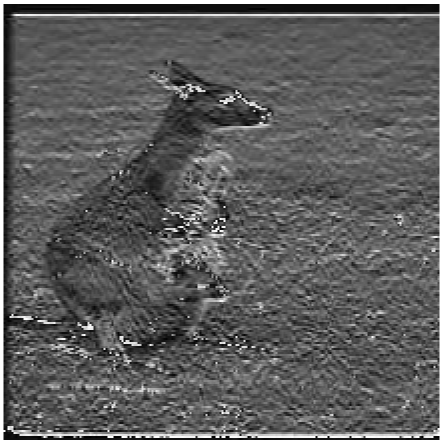

Fig 4 : Conv layer output (before activation)

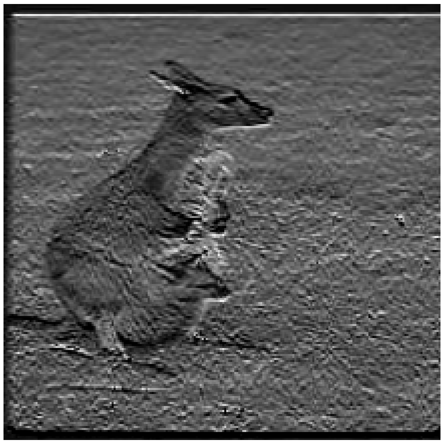

Fig 5 : Conv layer output (after activation)

Brain Class :

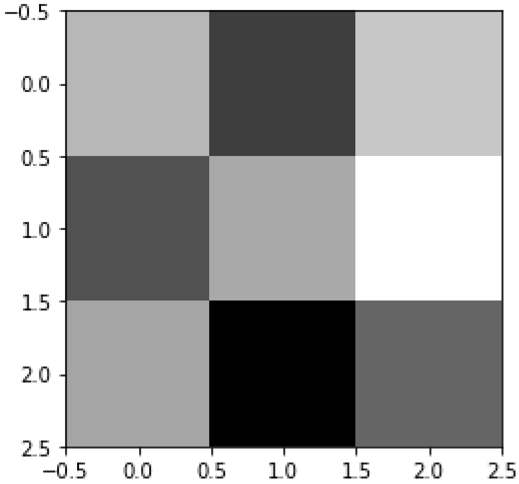

Fig 6 : Initialized filter for Brain

Fig 7 : Brain considered for Q1 (rgb)

Fig 8 : Brain considered for Q1 (grayscale)

After convolution following feature maps were obtained (before and after applying activation function).

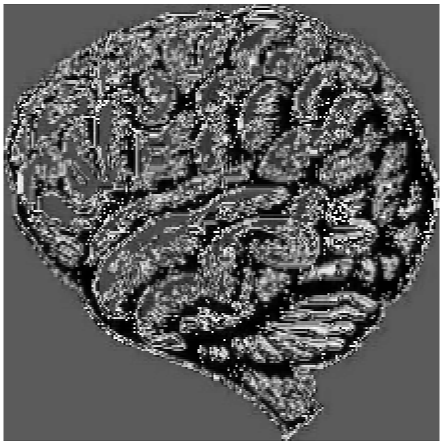

Fig 9 : Conv layer output (before activation)

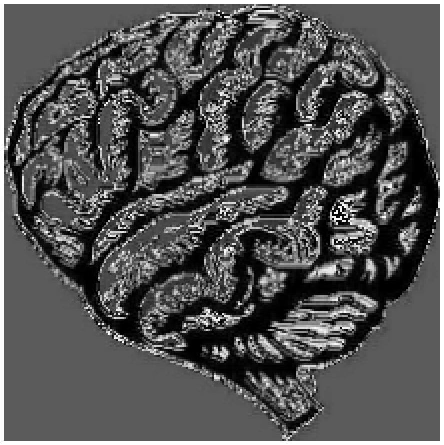

Fig 10 : Conv layer output (after activation)

Bonsai Class :

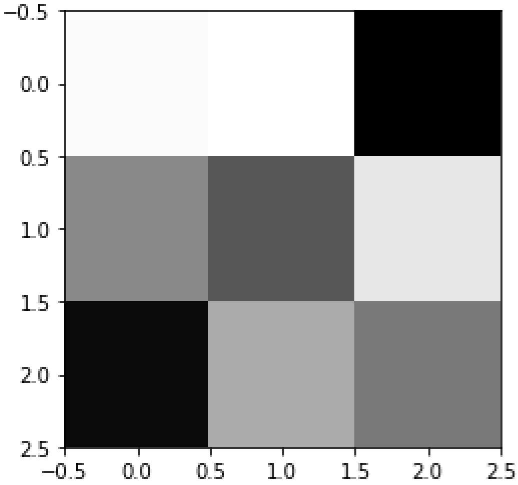

Fig 11 : Initialized filter for Bonsai

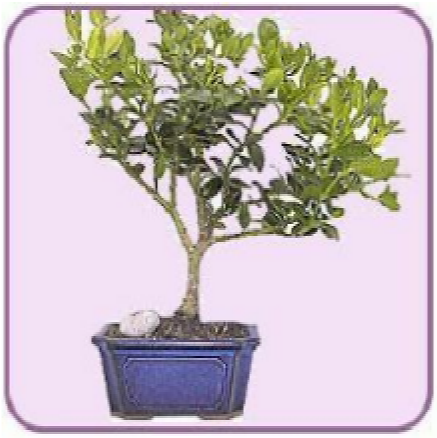

Fig 12 : Bonsai considered for Q1 (rgb)

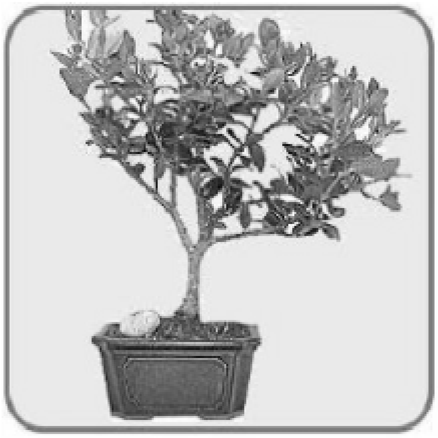

Fig 13 : Bonsai considered for Q1 (grayscale)

After convolution following feature maps were obtained (before and after applying activation function).

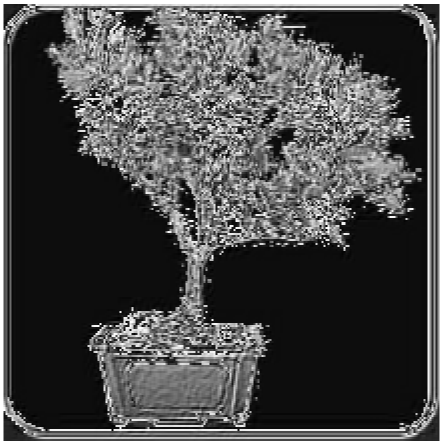

Fig 14 : Conv layer output (before activation)

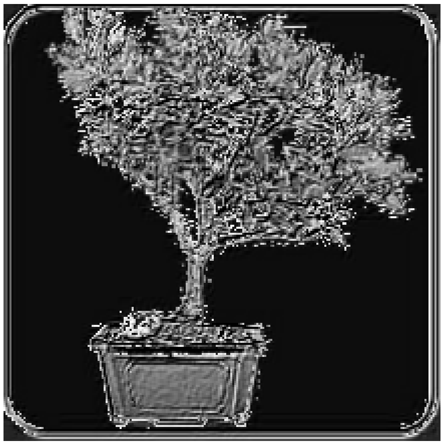

Fig 15 : Conv layer output (after activation)

Ques - 2

We implemented CNN functionality with the help of a self-coded Conv2D class. For simulating convolution a conv_forward function was defined for forward propagation and conv_backward was defined to handle the backward propagation.
Let conv_1 be the output of the 1st convolutional layer and conv_2 be the output of the 2nd convolutional layer (conv_1 is fed as input to this layer).

For all images the shape of conv_1 was observed to be (222, 222, 32) while the shape of conv_2 was observed to be (220, 220, 64). These observations are in agreement with the formula mentioned above.

First 10 neurons have been plotted for each class of the layer conv_1 and conv_2.

Kangaroo Class :

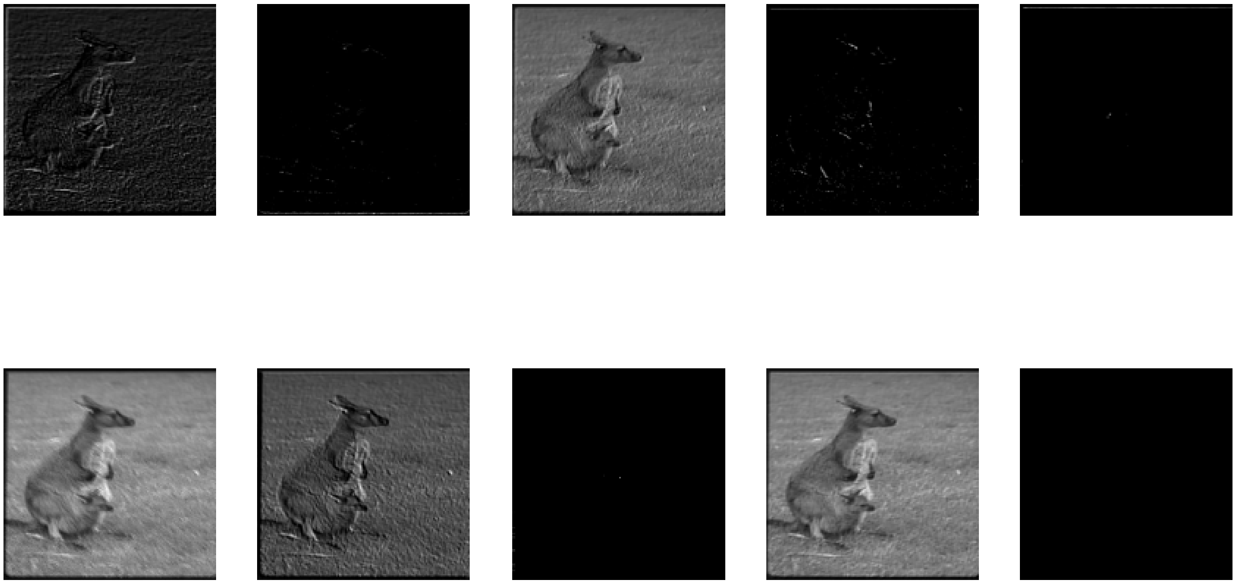

Fig 16: Output of first 10 neurons - conv_1 layer

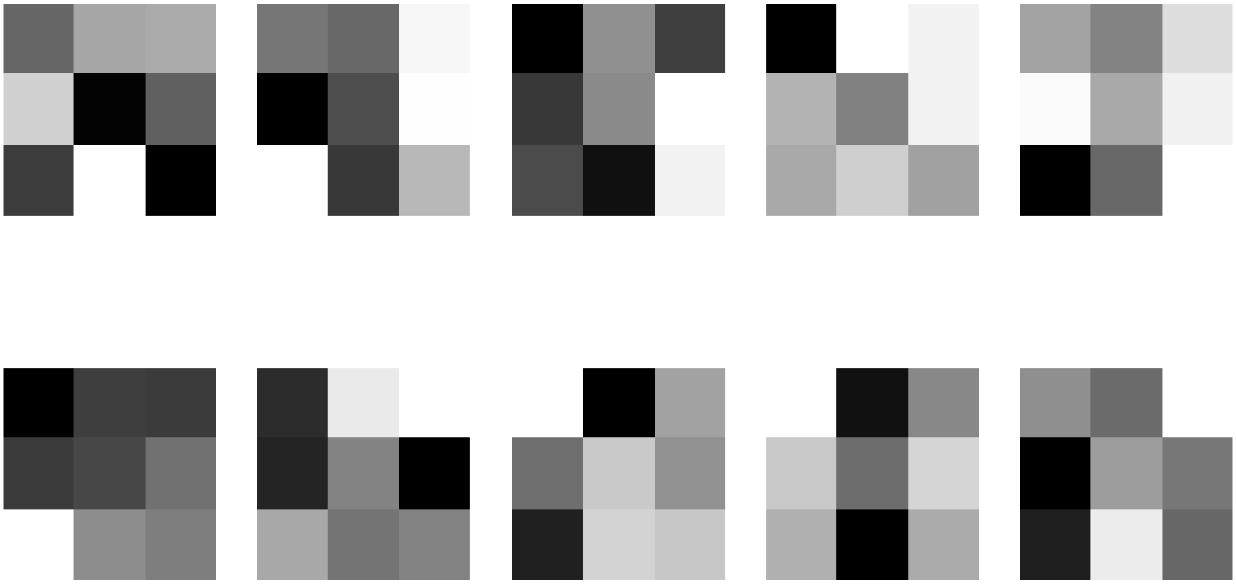

Fig 17: Filter used for conv_1

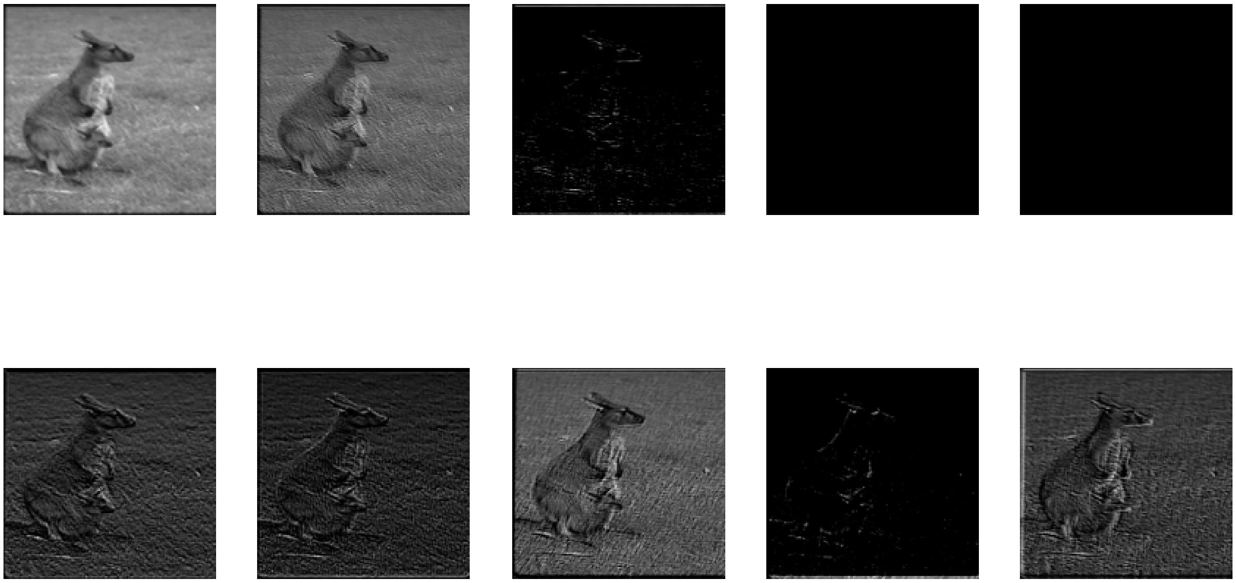

Fig 18 : Output of first 10 neurons - conv_2 layer

Brain Class :

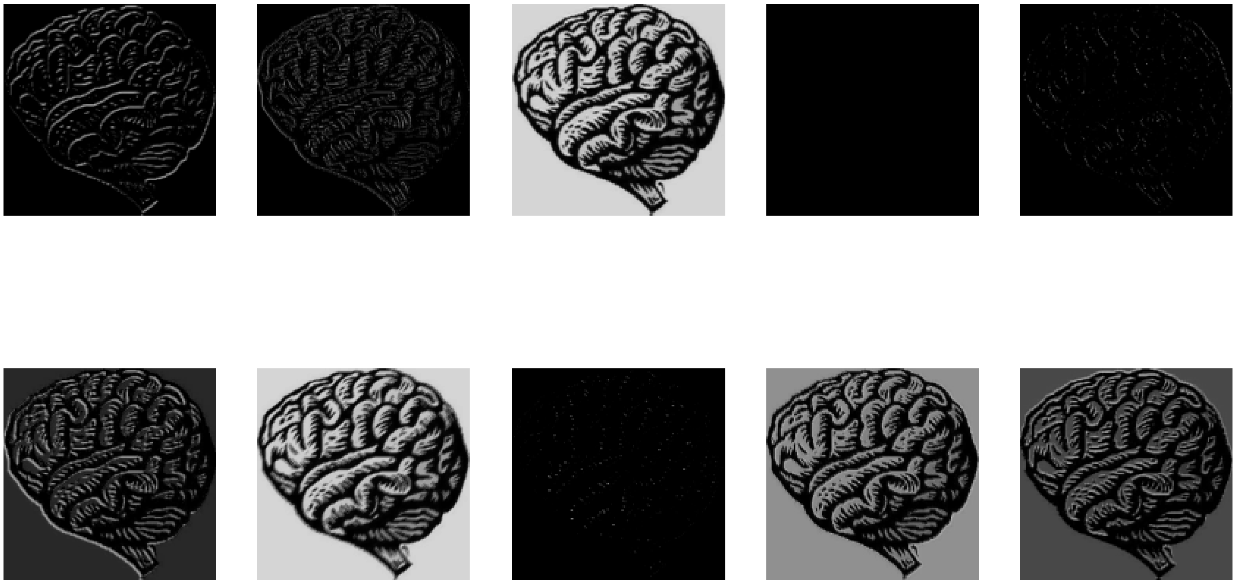

Fig 19 : Output of first 10 neurons - conv_1 layer

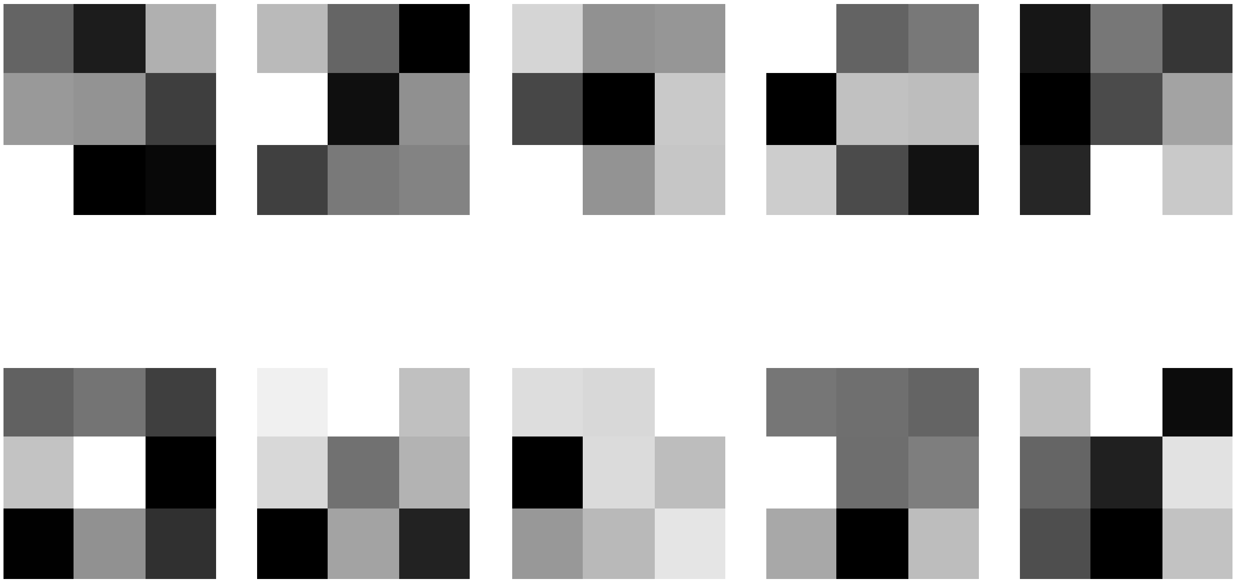

Fig 20: Filter used for conv_1

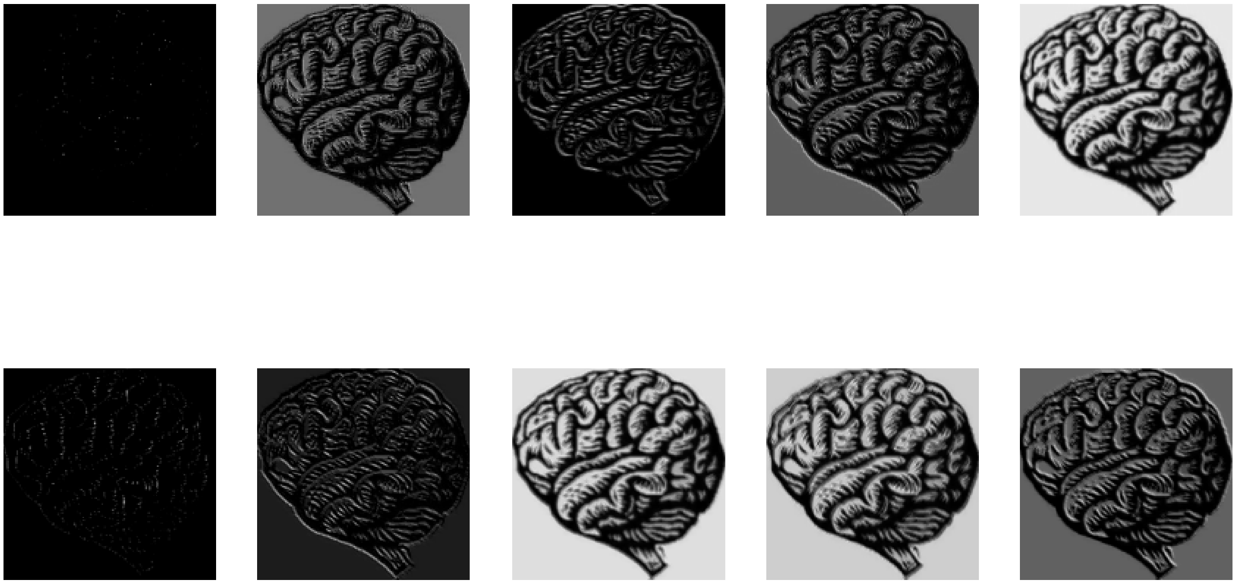

Fig 21 : Output of first 10 neurons - conv_2 layer

Bonsai Class :

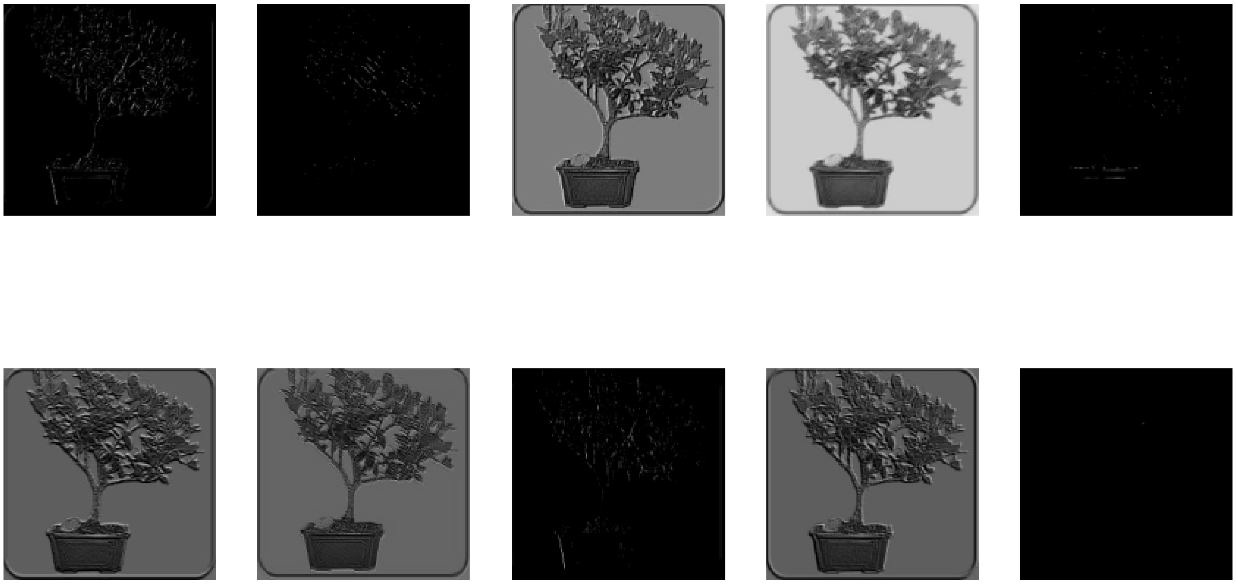

Fig 22 : Output of first 10 neurons - conv_1 layer

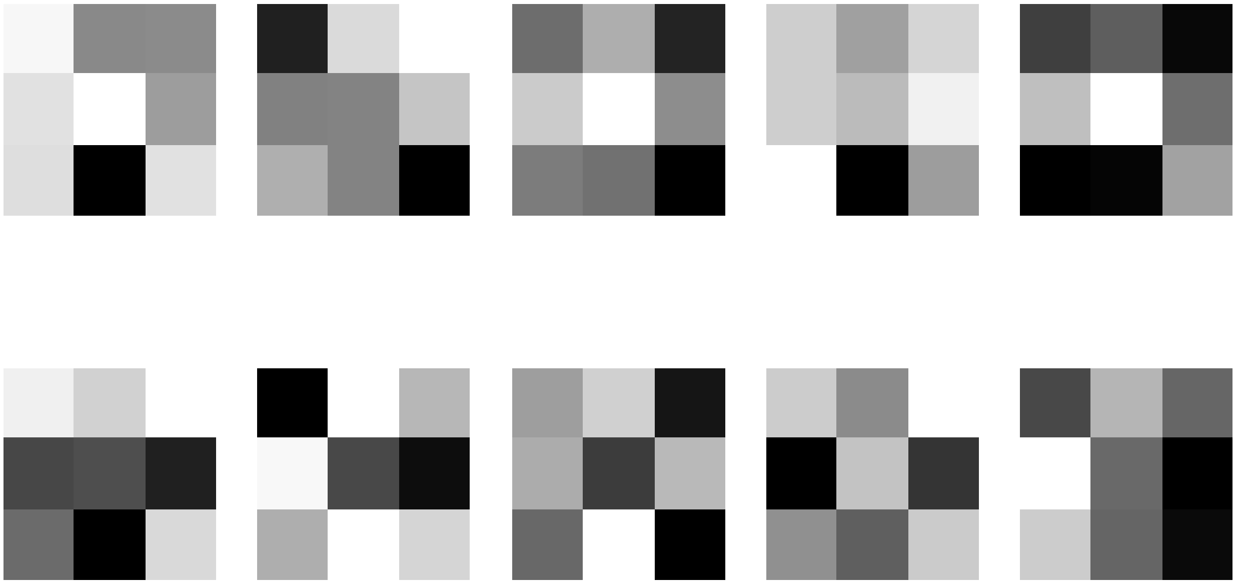

Fig 23 : Filter used for conv_1

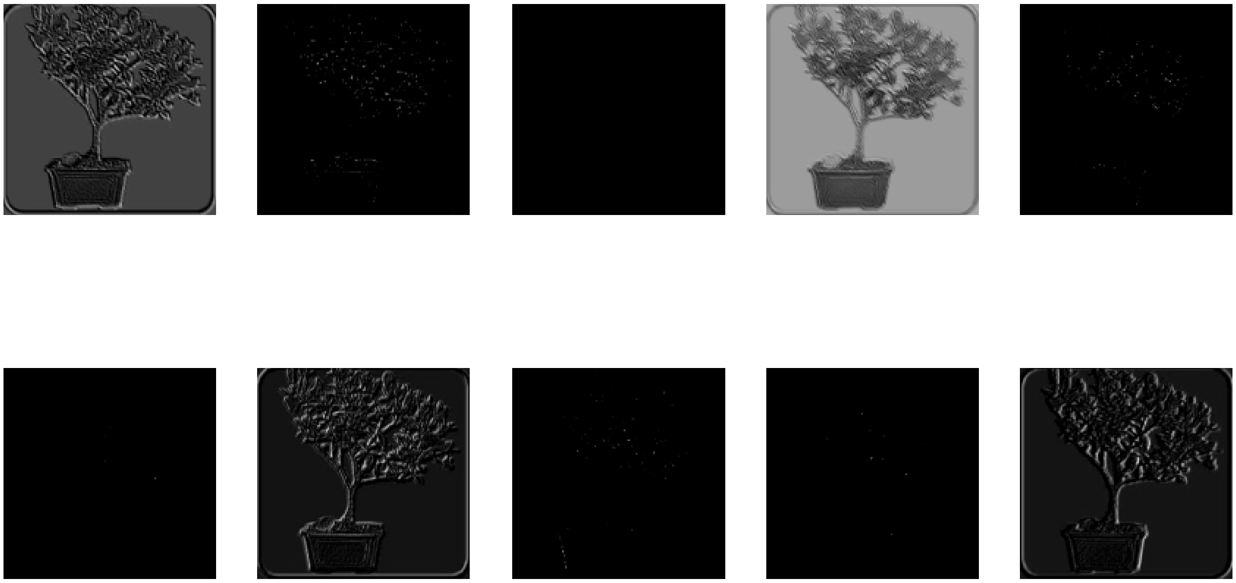

Fig 24 : Output of first 10 neurons - conv_2 layer