Assignment 3

Deep Learning And Its Applications

Group 14

Members:

- Prashant Kumar (B19101)
- Lahari Talla (B19197)
- Ravi Kumar (B19191)

Abstract

This report is based on the assignment 3 of Deep Learning course which includes the Convolutional neural network. The primary task is of building a convolutional neural network for image classification which is broken down into smaller hierarchical tasks. We have been given the subset of Caltech-101 dataset for the same. This dataset consists of 3 classes. Given dataset is train-validation-test separated. The Caltech-101 dataset consists of color images with varying sizes.

Data Set

We are given Training, Validation and Test Data. Each contains the subset of the grad_piano, starfish and trilobite

We have to resize each image into 224 x 224 size.

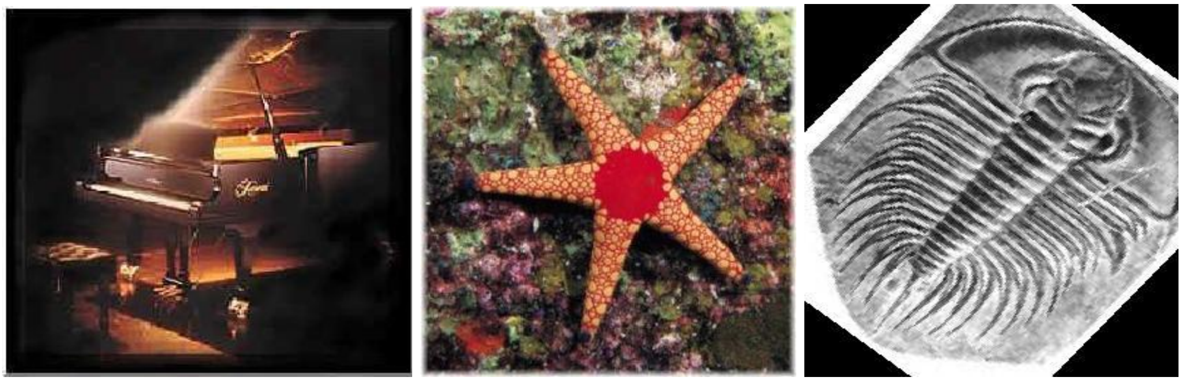
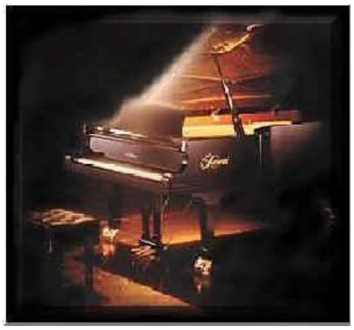


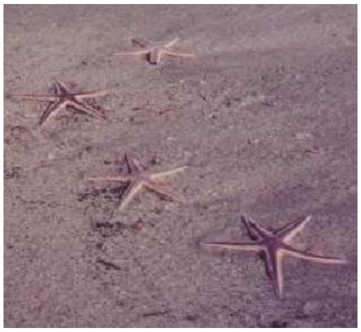
Fig. 1: Images showing Classes of the Data

Task 1

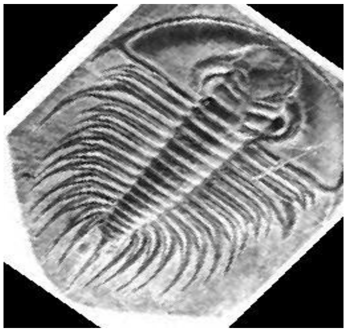
We have used these images for task 1:



Class 1 Image



Class 2 image



Class 3 image

Fig. 2 Image from different classes

Now we convert these images into single channel grayscale images and initialize the 3 x 3 filter using kaiming initialisation. We used 3 such filters for each image.

For class 1 image, single channel grayscale image is as follows:

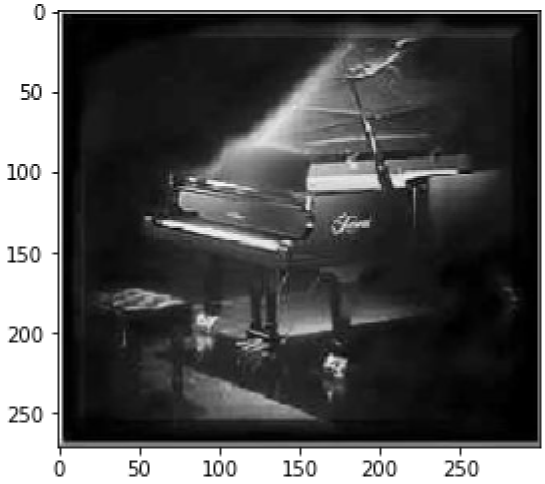


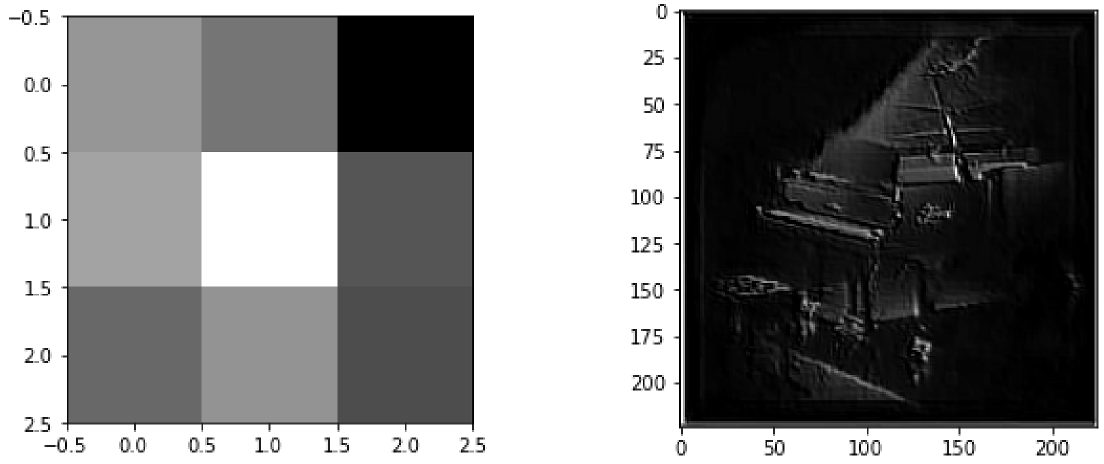
Fig. 3 Image form class 1 on which task 1 is to be performed

Below matrix shows the convolution filter 1 which we traversed over the image to get the feature map:

0.236080938	-0.008822056	-0.882205632
0.330134281	0.330134281	-0.229474052
-0.229474052	-0.229474052	-0.289226019

Table 1: Filter 1

In the below two figures we have plotted filter 1 as image and feature image obtained after convoluting image with filter 1.



Filter 1 as image

Feature map obtained after from filter 1

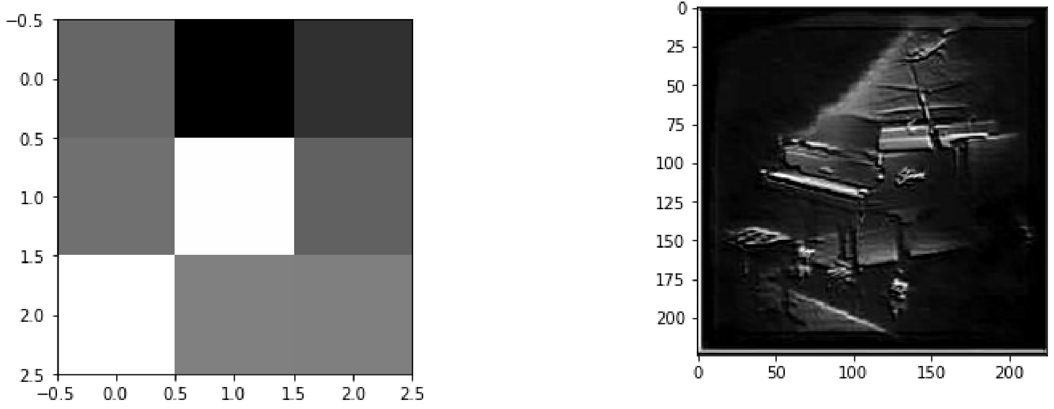
Fig. 4 Images showing filter 1 as image and feature map obtained from that filter

We have also made filter 2. Below matrix shows the convolution filter 2 which we traversed over the image for class 1 to get the feature map:

-0.0840303	-0.74223136	-0.42638484
-0.01870875	0.90304297	-0.114215
0.90952438	0.08935444	0.08306197

Table 2: Matrix of 3 x 3 showing Filter 2

Plot of filter 1 as image and feature image obtained from filter 2:



Filter 2 as image

Feature map obtained after from filter 2

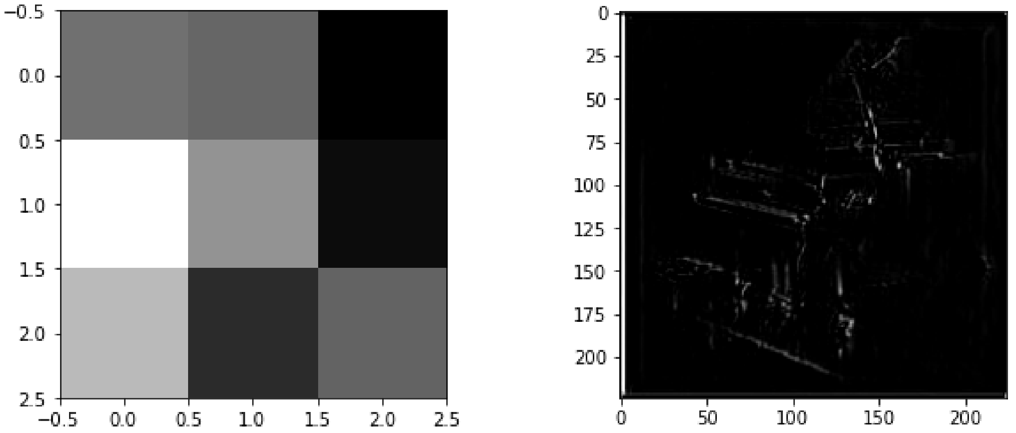
Fig. 5. Images showing filter 2 as image and feature map obtained from that filter

Apart from filter 1 and filter 2, we have also initialized the filter 2 matrix and obtained the feature image using filter 3. Below matrix shows the convolution filter 3 which we traversed over the image for class 1 to get the feature map:

0.02419115	-0.03181858	-0.58793797
0.80076632	0.20999494	-0.52280228
0.42222579	-0.35430824	-0.04973137

Table 3: Matrix of 3 x 3 showing Filter 3

In the below two figures we have plotted filter 1 as image and feature image obtained after convoluting image with filter 1.



Filter 3 as image

Feature map obtained after from filter 3

Fig. 6. Images showing filter 3 as image and feature map obtained from that filter

For class 2 image, single channel grayscale image is as follows:

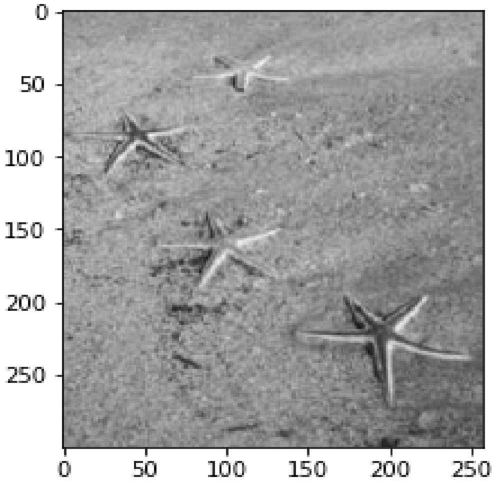


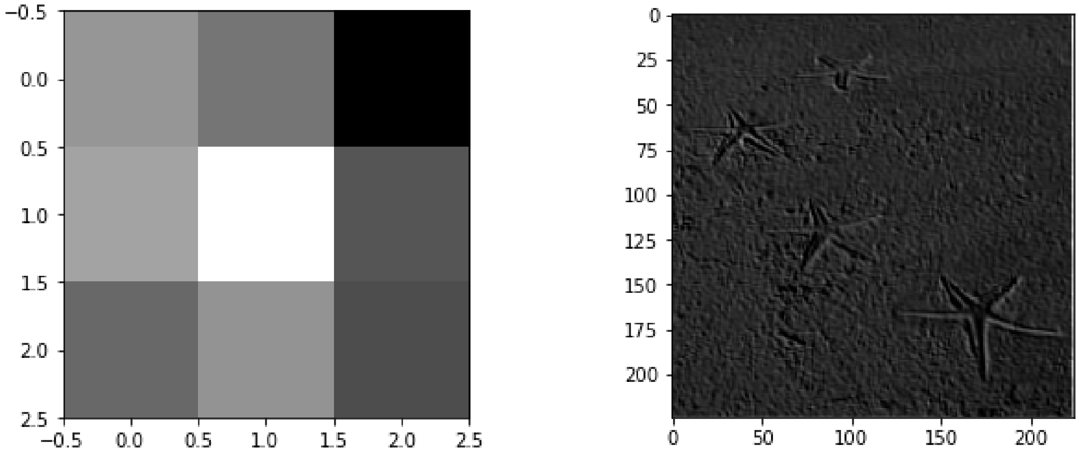
Fig. 7 Image form class 2 on which task 1 is to be performed

Below matrix shows the convolution filter 1 which we traversed over the image to get the feature map:

0.236080938	-0.008822056	-0.882205632
0.330134281	0.330134281	-0.229474052
-0.229474052	-0.229474052	-0.289226019

Table 4: Matrix of 3 x 3 showing filter 1

In the below two figures we have plotted filter 1 as image and feature image obtained after convoluting image with filter 1.



Filter 1 as image

Feature map obtained from filter 1

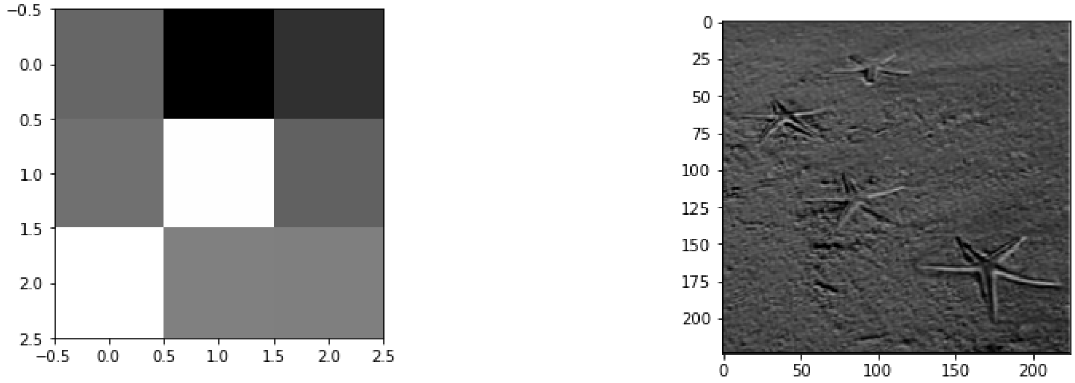
Fig. 8. Images showing filter 3 as image and feature map obtained from that filter

We have also made filter 2. Below matrix shows the convolution filter 2 which we traversed over the image for class 1 to get the feature map:

-0.0840303	-0.74223136	-0.42638484
-0.01870875	0.90304297	-0.114215
0.90952438	0.08935444	0.08306197

Table 5: Matrix of 3 x 3 showing Filter 2

Plot of filter 1 as image and feature image obtained from filter 2:



Filter 2 as image

Feature map obtained from filter 2

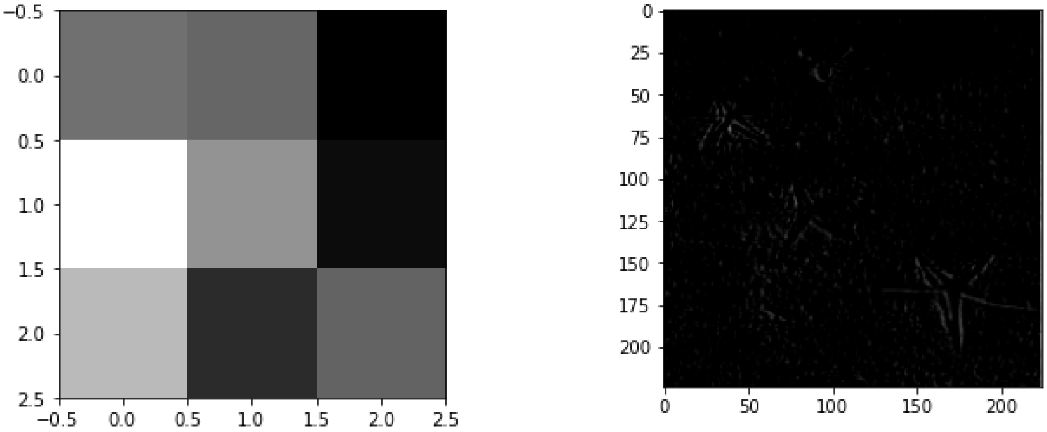
Fig. 9. Images showing filter 3 as image and feature map obtained from that filter

Apart from filter 1 and filter 2, we have also initialized the filter 2 matrix and obtained the feature image using filter 3. Below matrix shows the convolution filter 3 which we traversed over the image for class 1 to get the feature map:

0.02419115	-0.03181858	-0.58793797
0.80076632	0.20999494	-0.52280228
0.42222579	-0.35430824	-0.04973137

Table 6: Matrix of 3 x 3 showing Filter 3

In the below two figures we have plotted filter 1 as image and feature image obtained after convoluting image with filter 1.



Filter 3 as image

Feature map obtained after from filter 3

Fig. 10. Images showing filter 3 as image and feature map obtained from that filter

For class 3 image, single channel grayscale image is as follows:

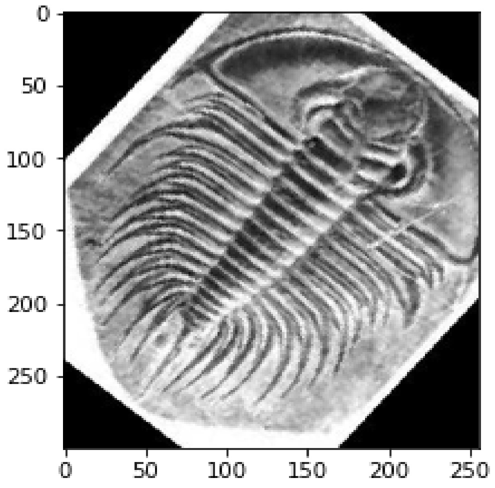


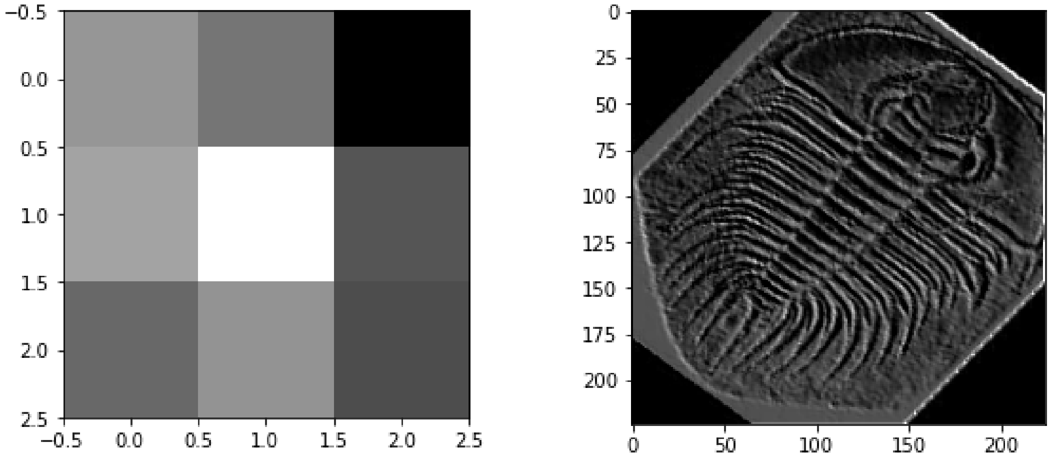
Fig. 11 Image form class 3 on which task 1 is to be performed

Below matrix shows the convolution filter 1 which we traversed over the image to get the feature map:

0.236080938	-0.008822056	-0.882205632
0.330134281	0.330134281	-0.229474052
-0.229474052	-0.229474052	-0.289226019

Table 7: Matrix of 3 x 3 showing filter 1

In the below two figures we have plotted filter 1 as image and feature image obtained after convoluting image with filter 1.



Filter 1 as image

Feature map obtained from filter 1

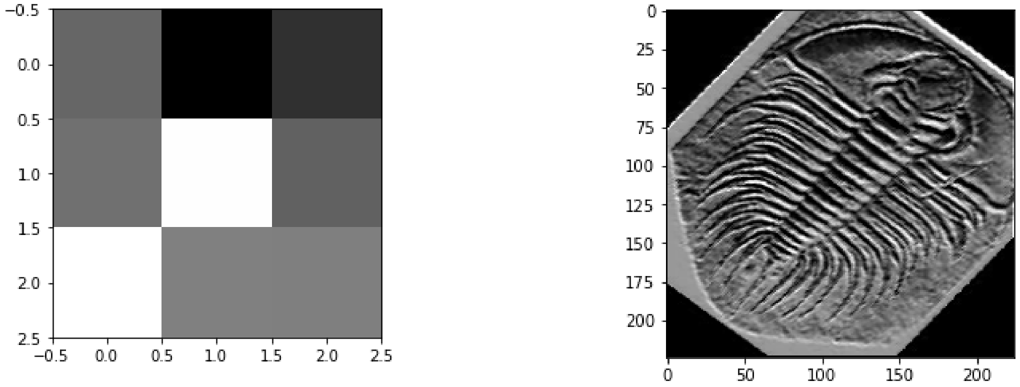
Fig. 12. Images showing filter 3 as image and feature map obtained from that filter

We have also made filter 2. Below matrix shows the convolution filter 2 which we traversed over the image for class 1 to get the feature map:

-0.0840303	-0.74223136	-0.42638484
-0.01870875	0.90304297	-0.114215
0.90952438	0.08935444	0.08306197

Table 8: Matrix of 3 x 3 showing Filter 2

Plot of filter 1 as image and feature image obtained from filter 2:



Filter 2 as image

Feature map obtained from filter 2

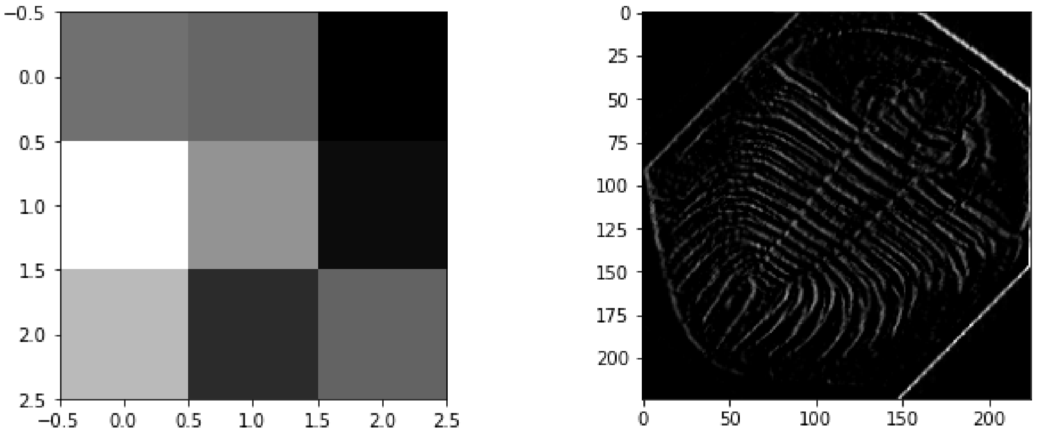
Fig. 13. Images showing filter 3 as image and feature map obtained from that filter

Apart from filter 1 and filter 2, we have also initialised the filter 2 matrix and obtained the feature image using filter 3. Below matrix shows the convolution filter 3 which we traversed over the image for class 1 to get the feature map:

0.02419115	-0.03181858	-0.58793797
0.80076632	0.20999494	-0.52280228
0.42222579	-0.35430824	-0.04973137

Table 9: Matrix of 3 x 3 showing Filter 3

In the below two figures we have plotted filter 1 as image and feature image obtained after convoluting image with filter 1.



Filter 3 as image

Feature map obtained after from filter 3

Fig. 14. Images showing filter 3 as image and feature map obtained from that filter

Expected dimensions of the feature image obtained using mathematical formula :

$$W_2 = \frac{W_1 - F + 2P}{S} + 1$$

$$H_2 = \frac{H_1 - F + 2P}{S} + 1$$

W2 = (224 - 3 + 0)/1 + 1 = 222

H2 = (224 - 3 + 0)/1 + 1 = 222

Hence, Mathematically obtained dimension of the feature map is 222 x 222.

Dimension of the feature map obtained from the code is 222 x 222.

Both dimensions are the same Hence verified.

Task 2

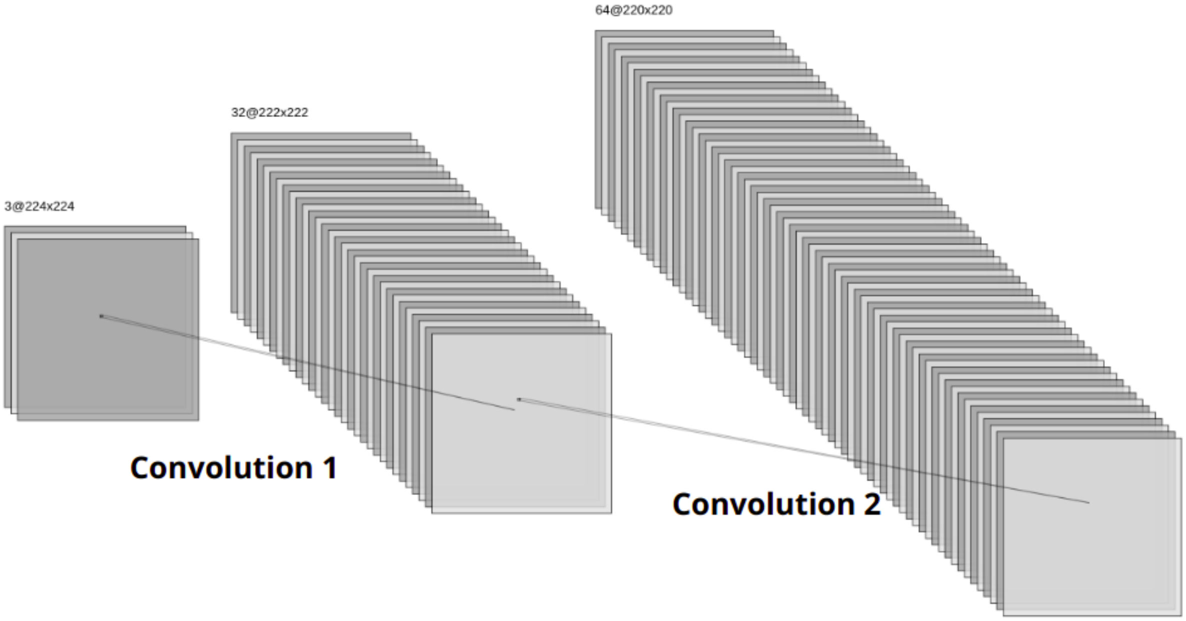


Fig.15 : Convolution

Implementing the convolutional layer with Relu activation function on images of each class of size **224 X 224 X 3**. Initialize two convolutional layers having 32 and 64 filters. First convolve layer initializes **32 filters of 3X3X3** with Kaiming initialization.

Images Considered from each class:

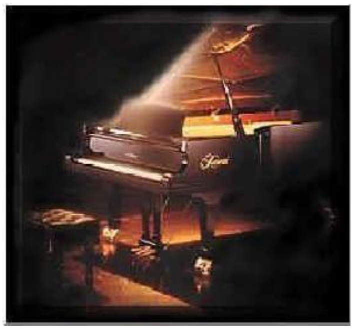


Fig.16 : Grand Piano

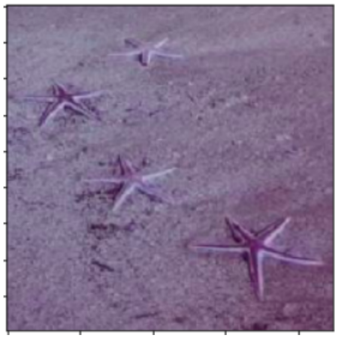


Fig.17 : StarFish

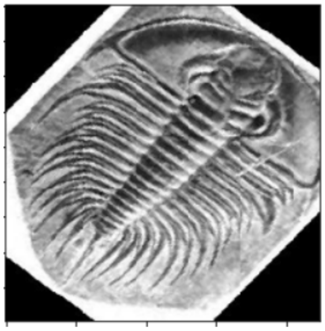
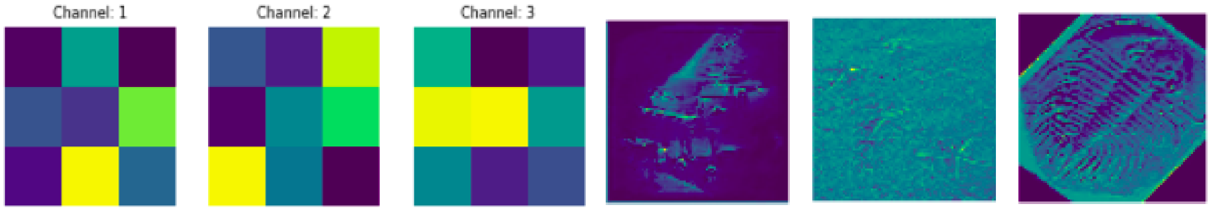


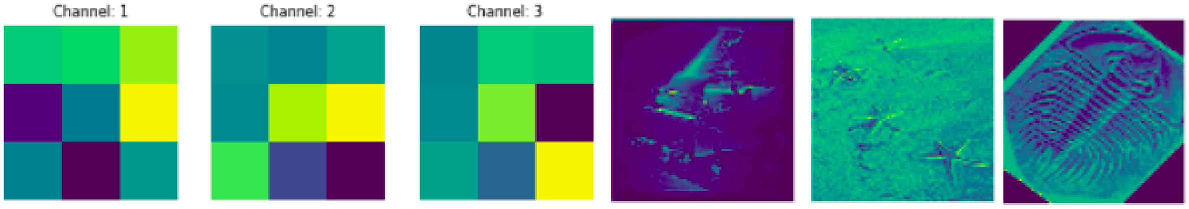
Fig.18 : Trilobite

Any 6 filters of the first convolutional layers and there correspondent feature map on each image are shown below:

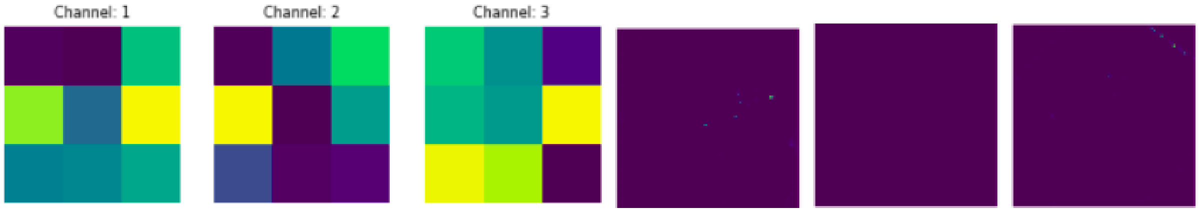
Filter 15:



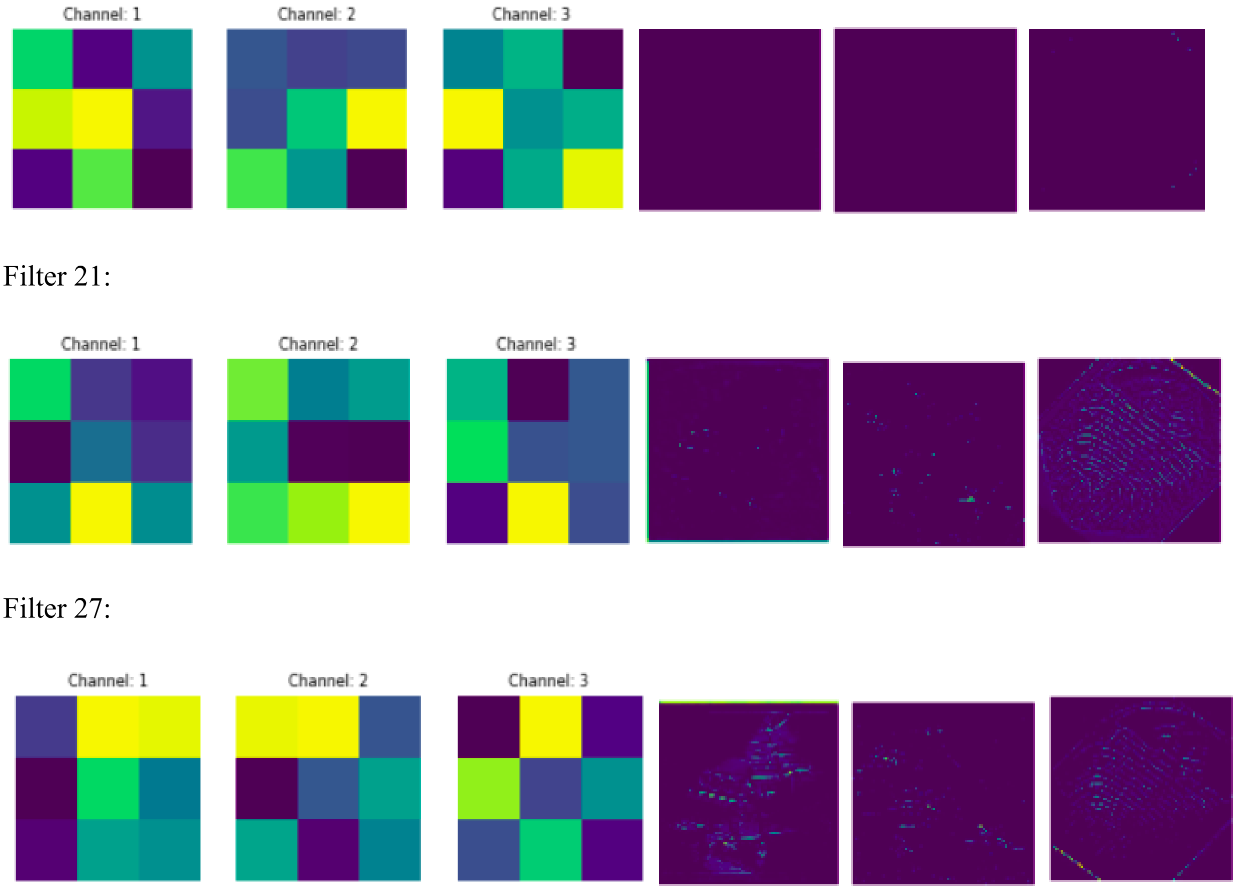
Filter 24:



Filter 7:



Filter 11:



Filter 21:

Filter 27:

Fig. 19: Plot of filters of first convolution layer as image and there corresponding feature map on each class image

$$W_2 = \frac{W_1 - F + 2P}{S} + 1$$

$$H_2 = \frac{H_1 - F + 2P}{S} + 1$$

W2 = (224 - 3 + 0)/1 + 1 = 222

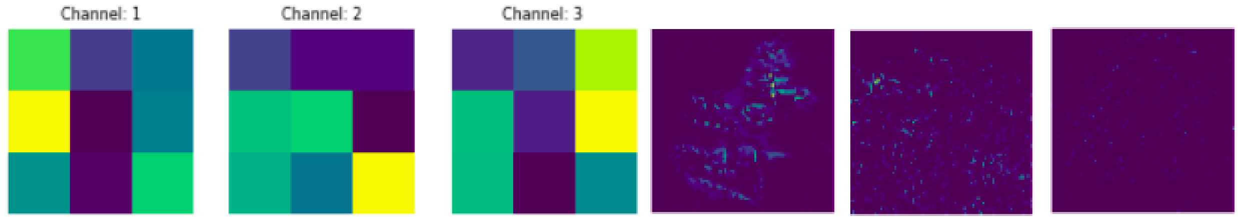
H2 = (224 - 3 + 0)/1 + 1 = 222

D = 32

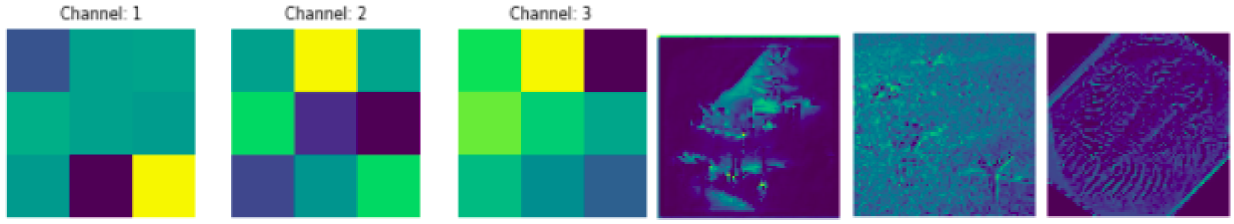
Dimension of the feature map that is obtained after the first convolutional layer is **222 X 222 X 32.**

Now second layer convolution with **64 filters of 3 X 3 X 32** on 222 X 222 X 32. Any 8 filters of the second convolutional layers and there correspondent feature map on each image are shown below:

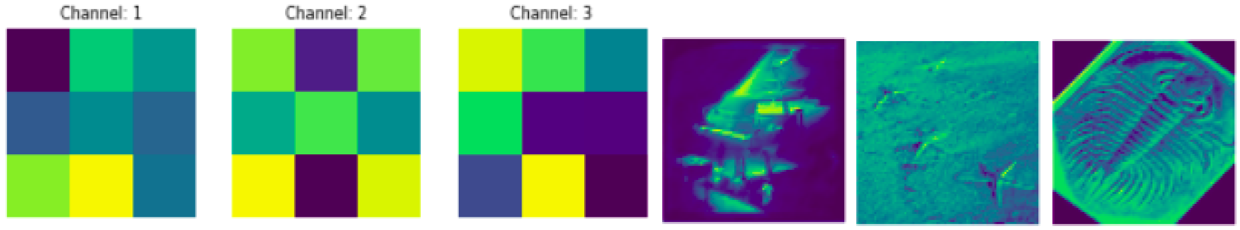
Filter 21:



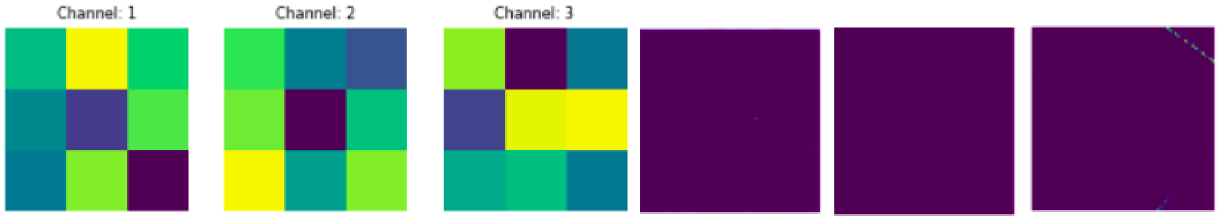
Filter 22:



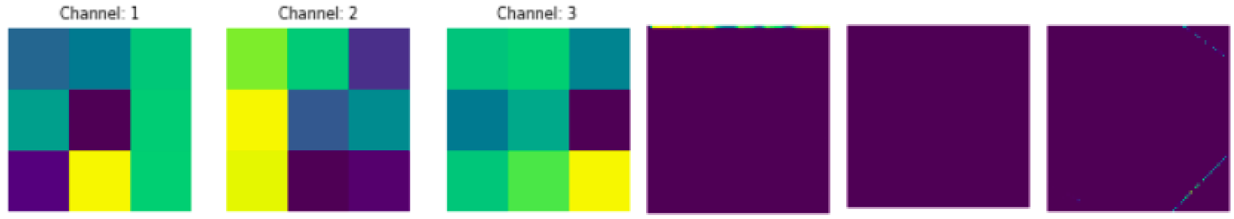
Filter 38:



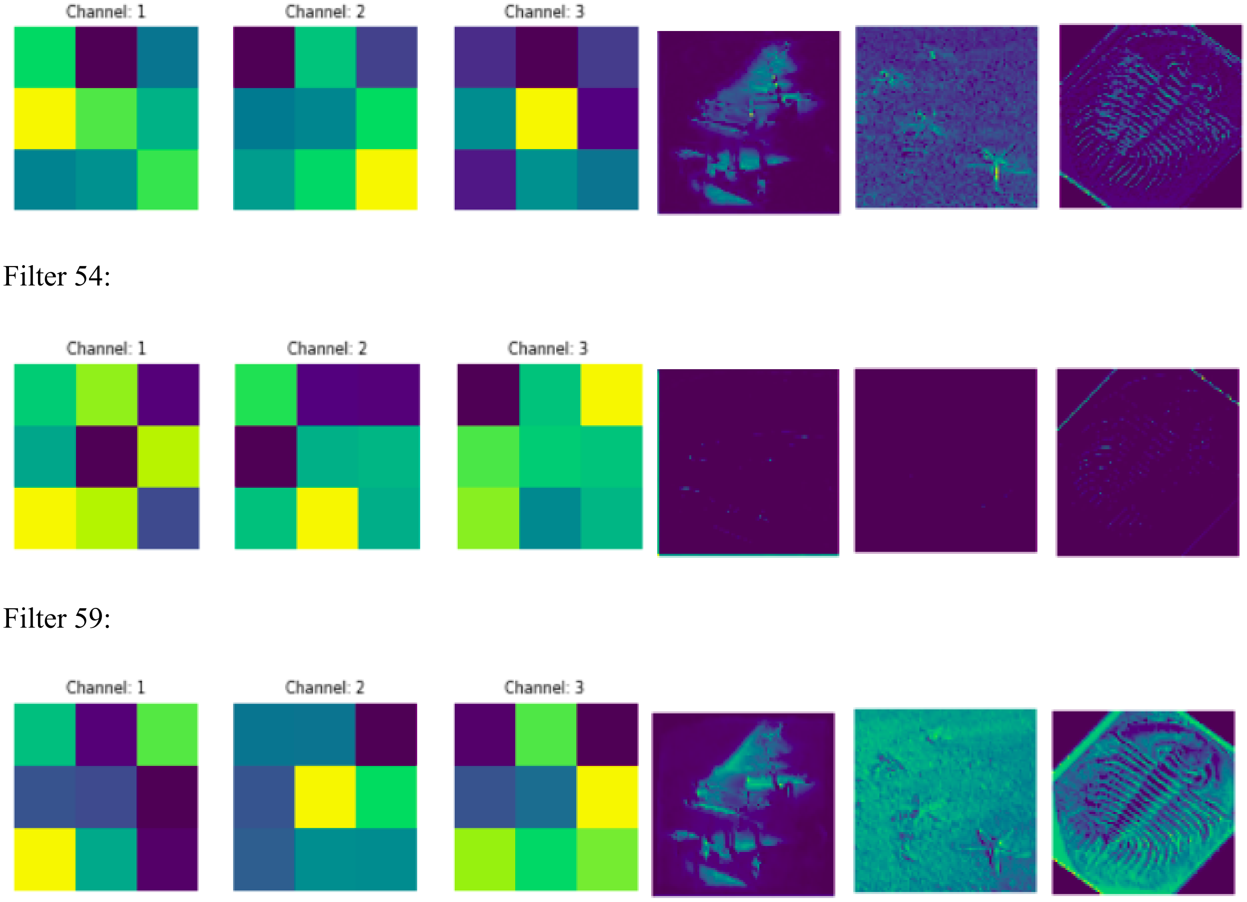
Filter 40:



Filter 49:



Filter 51:



Filter 54:

Filter 59:

Fig.20 : Plot of filters of second convolution layer as image and there corresponding feature map on each class image

$$W_2 = \frac{W_1 - F + 2P}{S} + 1$$

$$H_2 = \frac{H_1 - F + 2P}{S} + 1$$

W2 = (222 - 3 + 0)/1 + 1 = 220

H2 = (222 - 3 + 0)/1 + 1 = 220

D = 64

Dimension of the feature map that is obtained after the second convolutional layer is **220 X 200 X 64.**

Task 3: Implementing CNN

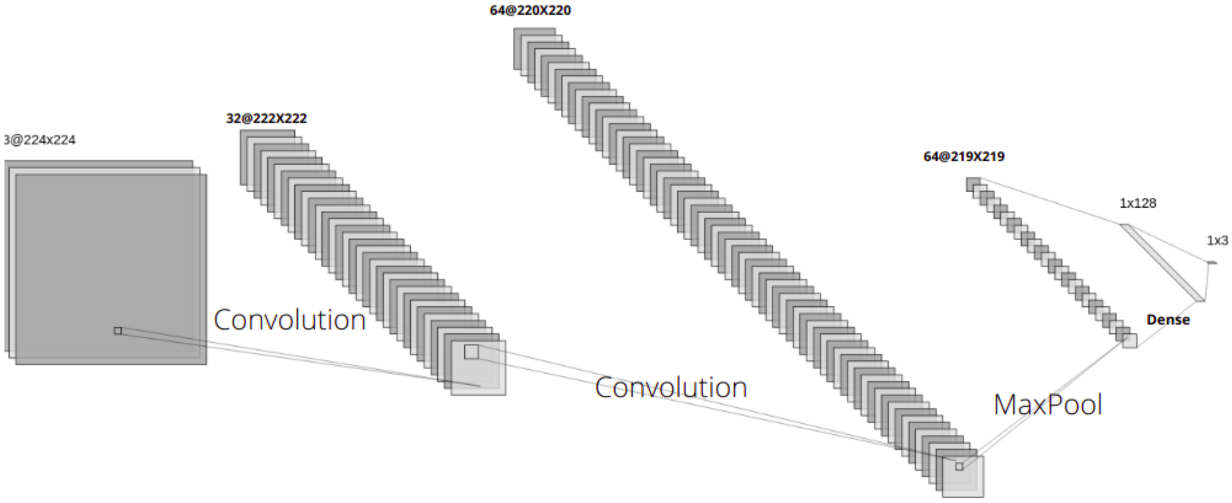


Fig.21 : Convolution Neural Network

Implement a convolutional neural network. The input layer take 224 x 224 3 channel image, stack two convolutional layers with 32 and 64 3x3 filters one after the other, down-sample the feature map using the 2x2 max pooling operation (stride 1), and flatten it. For convolution in both the layers, stride = 1, padding = 0. Next, two fully connected layers, 128 hidden nodes with rectified linear activation function in the first hidden layer and 3 neurons with a softmax activation function in the second layer (output layer).

Convolutional filters are initialized using Kaiming initialization and fully-connected layers with weights randomly sampled from a zero mean normal distribution with a standard deviation inversely proportional to the square root of the number of units. The weights are updated using a backpropagation algorithm with stochastic gradient descent.

Training model on training data. Below are the accuracy and the confusion matrix of CNN on training, validation and test data.

Accuracy:

- Train Data: 99.33 %
- Validation Data: 83.33 %
- Test Data: 88.33 %

Confusion Matrix :

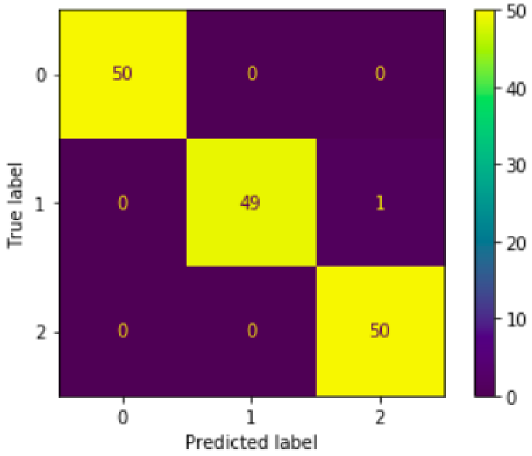


Fig.22 : Confusion Matrix of Training Data

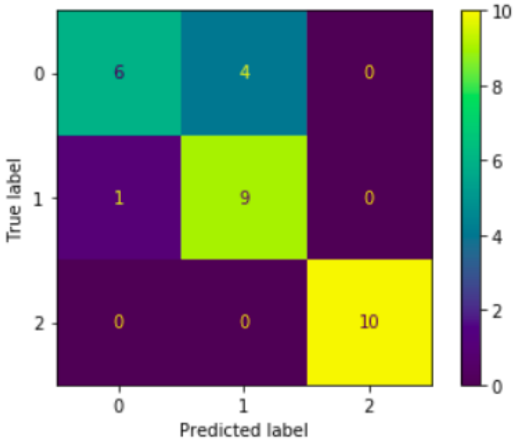


Fig.23 : Confusion Matrix of Validation Data

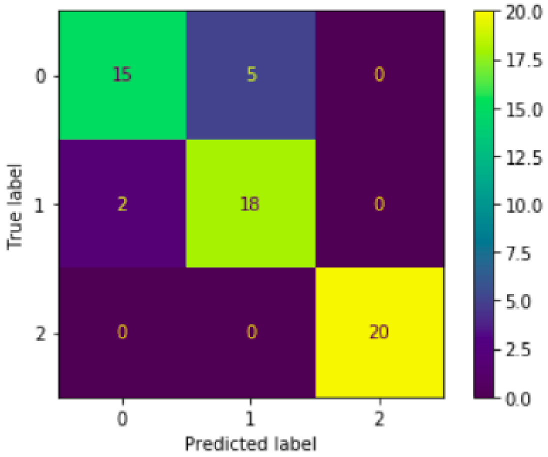


Fig.24 : Confusion Matrix of Test Data

Now considering the same images as before:

Images Considered from each class:

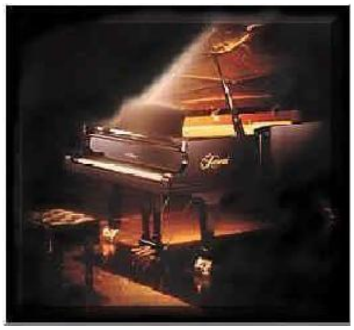


Fig.25 : Grand Piano

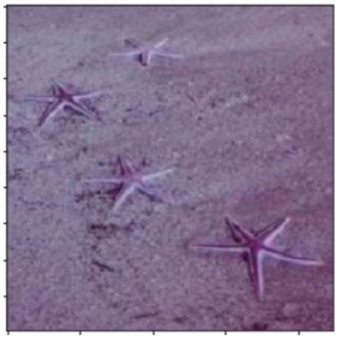


Fig.26 : StarFish

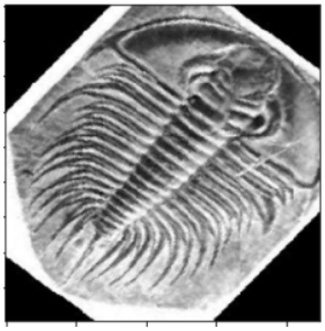
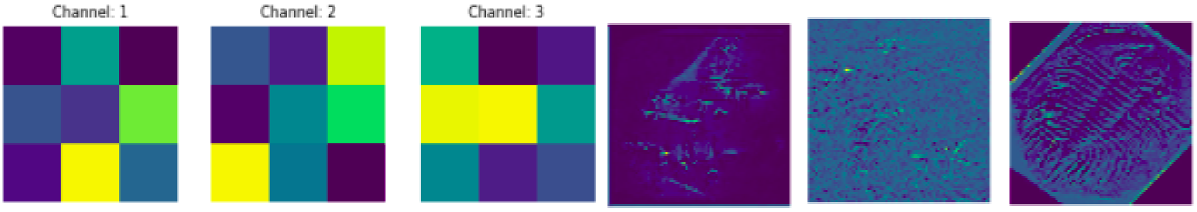


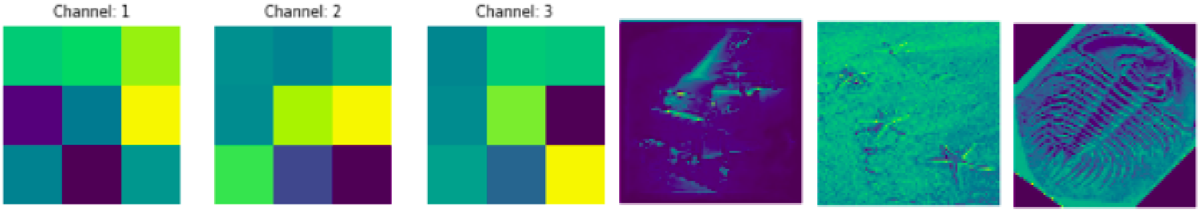
Fig.27 : Trilobite

Plotting same 6 filters (that we used above in task 2) as image from first convolution layers with corresponding filter maps:

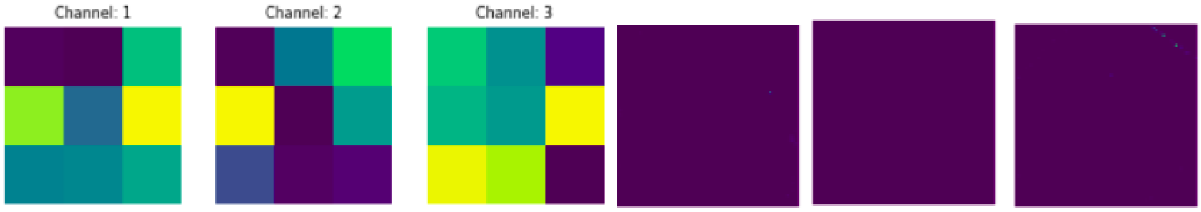
Filter 15:



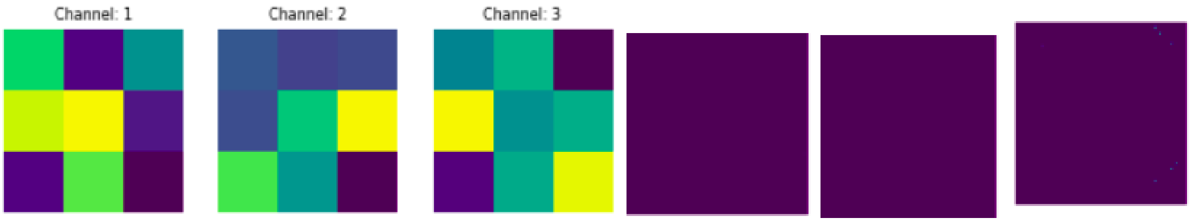
Filter 24:



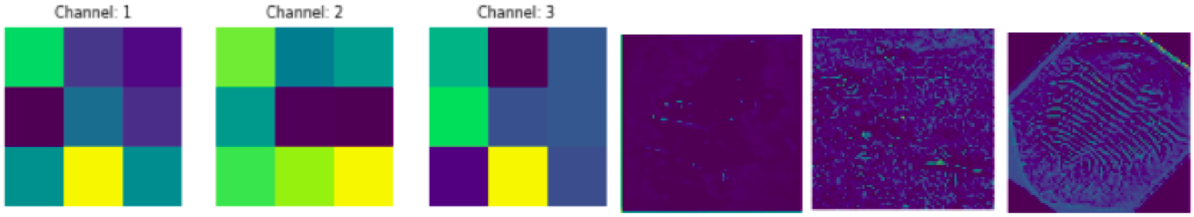
Filter 7:



Filter 11:



Filter 21:



Filter 27:

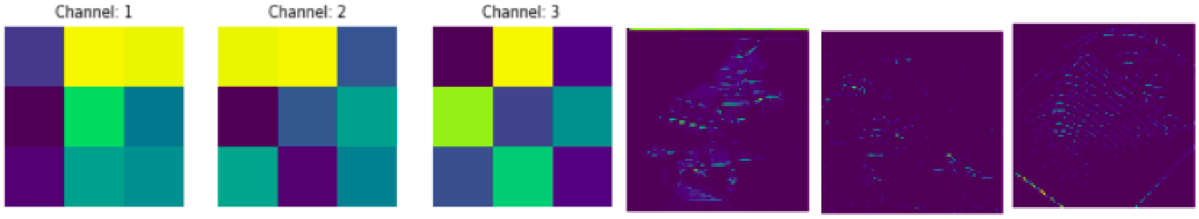


Fig.28 : Plot of filters of first convolution layer as image and there corresponding feature map on each class image

$$W_2 = \frac{W_1 - F + 2P}{S} + 1$$

$$H_2 = \frac{H_1 - F + 2P}{S} + 1$$

W2 = (224 - 3 + 0)/1 + 1 = 222

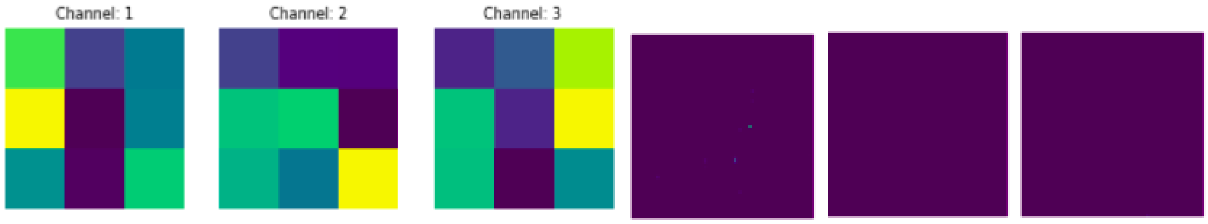
H2 = (224 - 3 + 0)/1 + 1 = 222

D = 32

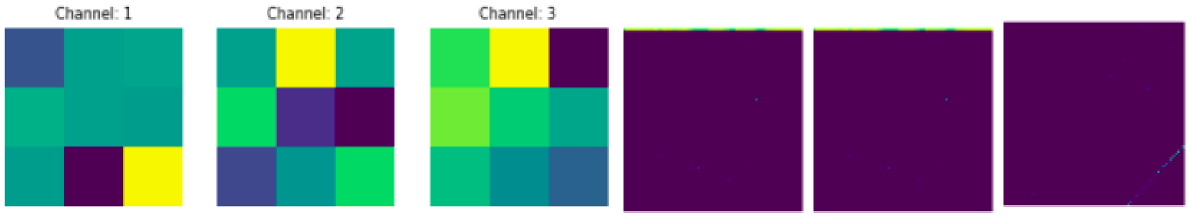
Dimension of the feature map that is obtained after the first convolutional layer is **222 X 222 X 32.**

Plotting same 8 filters (that we used above in task 2) as image from second convolution layers with corresponding filter maps:

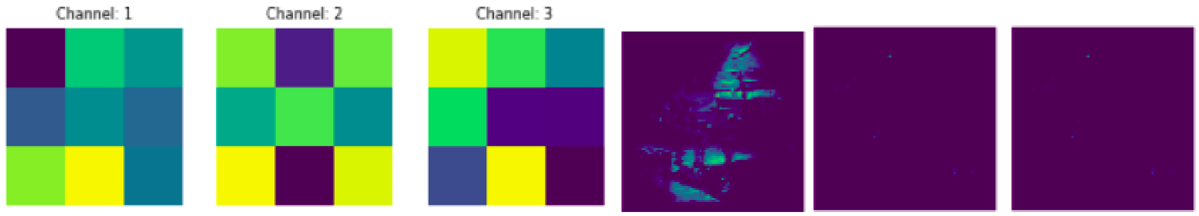
Filter 21:



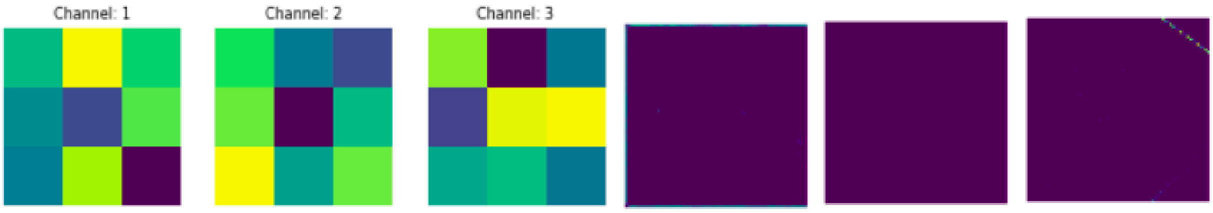
Filter 22:



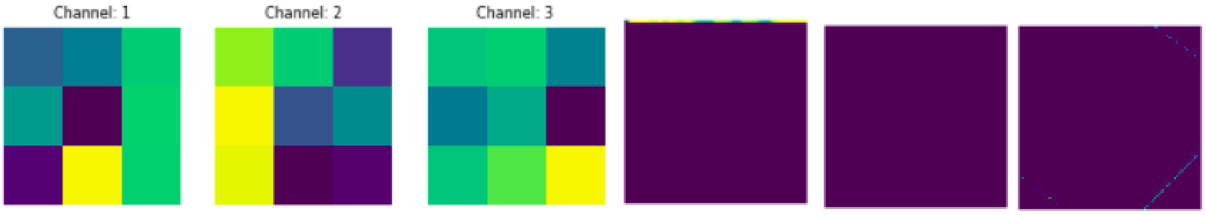
Filter 38:



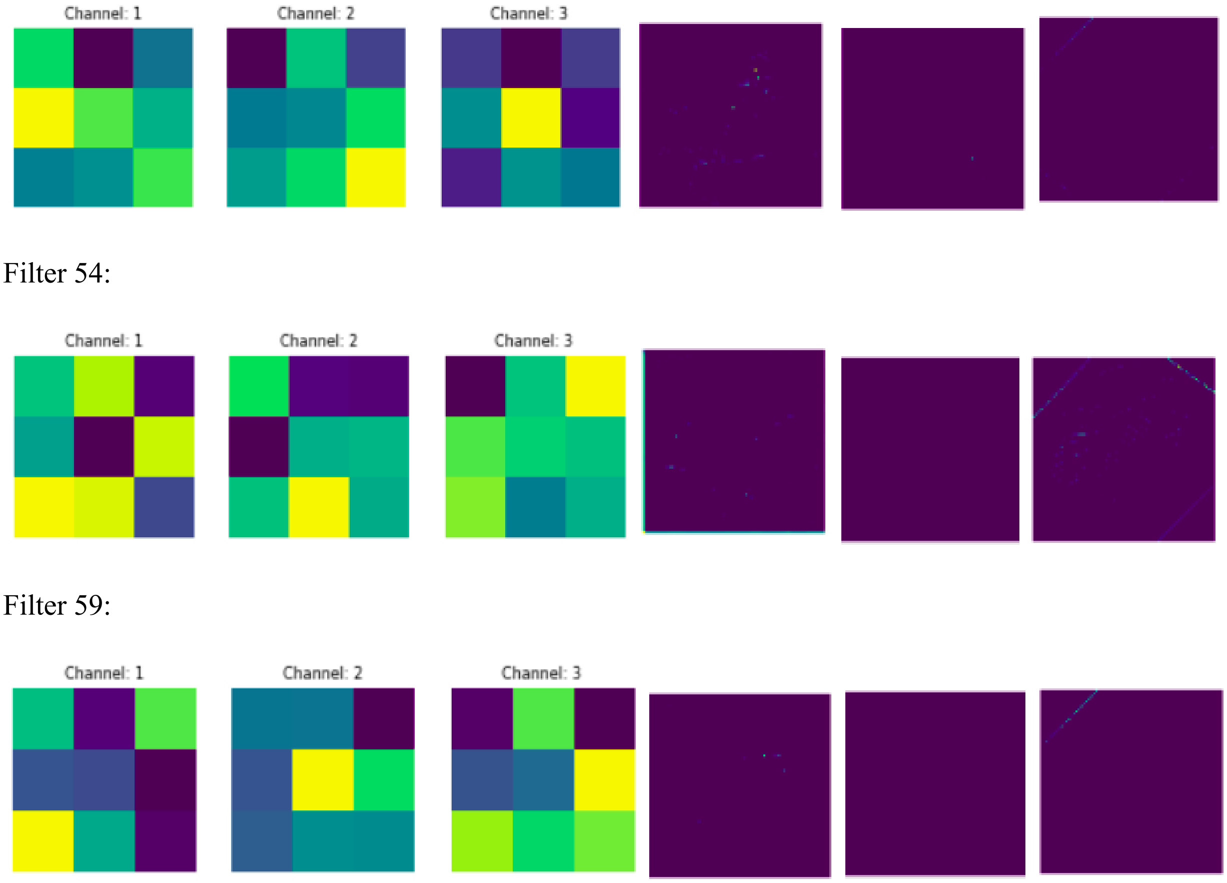
Filter 40:



Filter 49:



Filter 51:



Filter 54:

Filter 59:

Fig.29 : Plot of filters of second convolution layer as image and there corresponding feature map on each class image

$$W_2 = \frac{W_1 - F + 2P}{S} + 1$$

$$H_2 = \frac{H_1 - F + 2P}{S} + 1$$

W2 = (222 - 3 + 0)/1 + 1 = 220

H2 = (222 - 3 + 0)/1 + 1 = 220

D = 64

Dimension of the feature map that is obtained after the second convolutional layer is **220 X 220 X 64.**

After this maxpool layer is applied with **(2 X 2)** with stride = 1.

Hence dimension that we obtained = **219 X 219 X64**

Task 4

Part 1)

Accuracy:

- Train Data: 98.37 %
- Validation Data: 96.68 %
- Test Data: 96.34 %

Confusion Matrix :

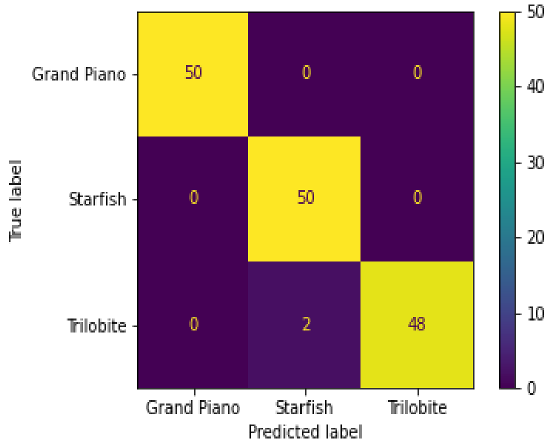


Fig.30 : Confusion Matrix of Training Data

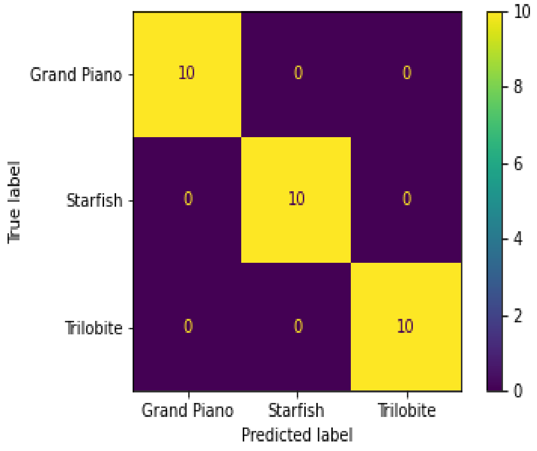


Fig.31 : Confusion Matrix of Validation Data

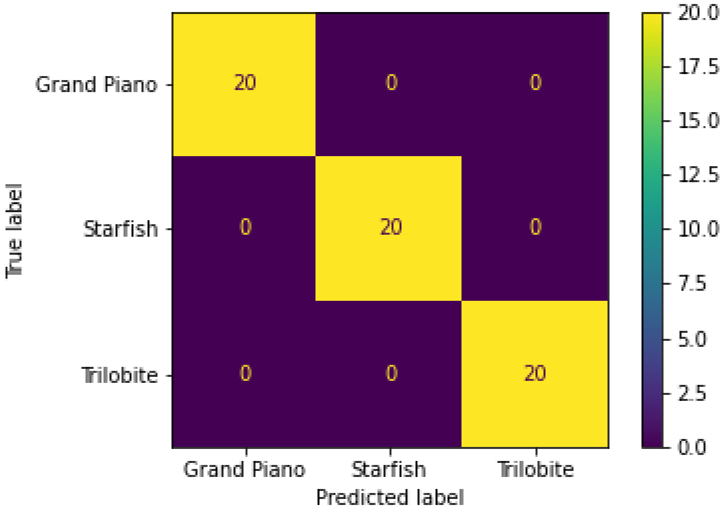


Fig.32 : Confusion Matrix of Test Data

S.No.	Accuracy of CNN implement with two conv layer	Accuracy after using VGG19 pre trained on ImageNet
Train Data	99.33 %	98.67 %
Validation Data	83.33 %	100 %
Test Data	88.33 %	100 %

Table 10: Accuracy Comparison

Part 2)

5 neurons in the last convolutional layer that are maximally activated:

For Class 1 image:

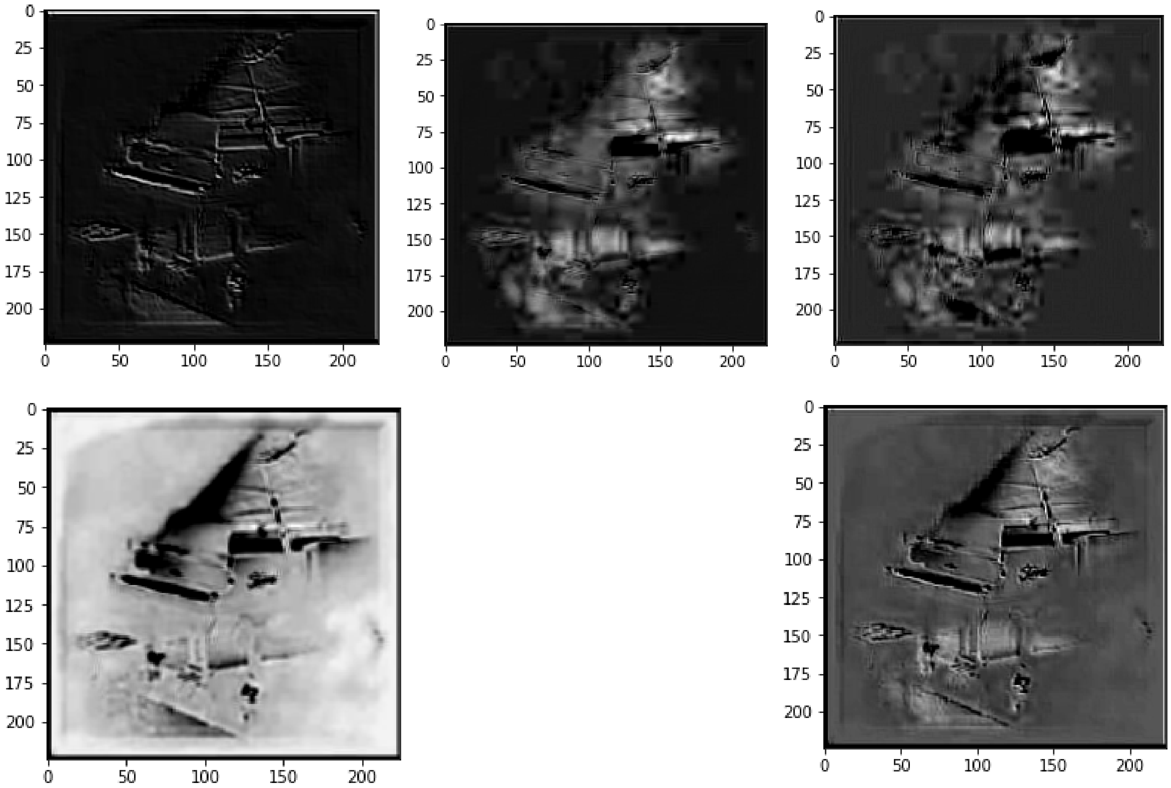
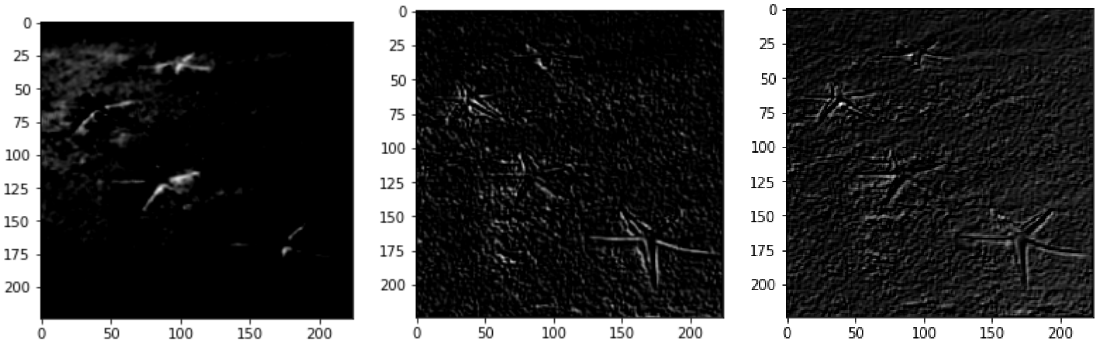


Fig.33 Visualizing patch of Image 1

For Class 2 Image:



0
25
50
75
100
125
150
175
200
0
50
100
150
200
0
25
50
75
100
125
150
175
200
0
50
100
150
200

Fig.34 : Visualizing patch of Image 2

For Classs 3 image :

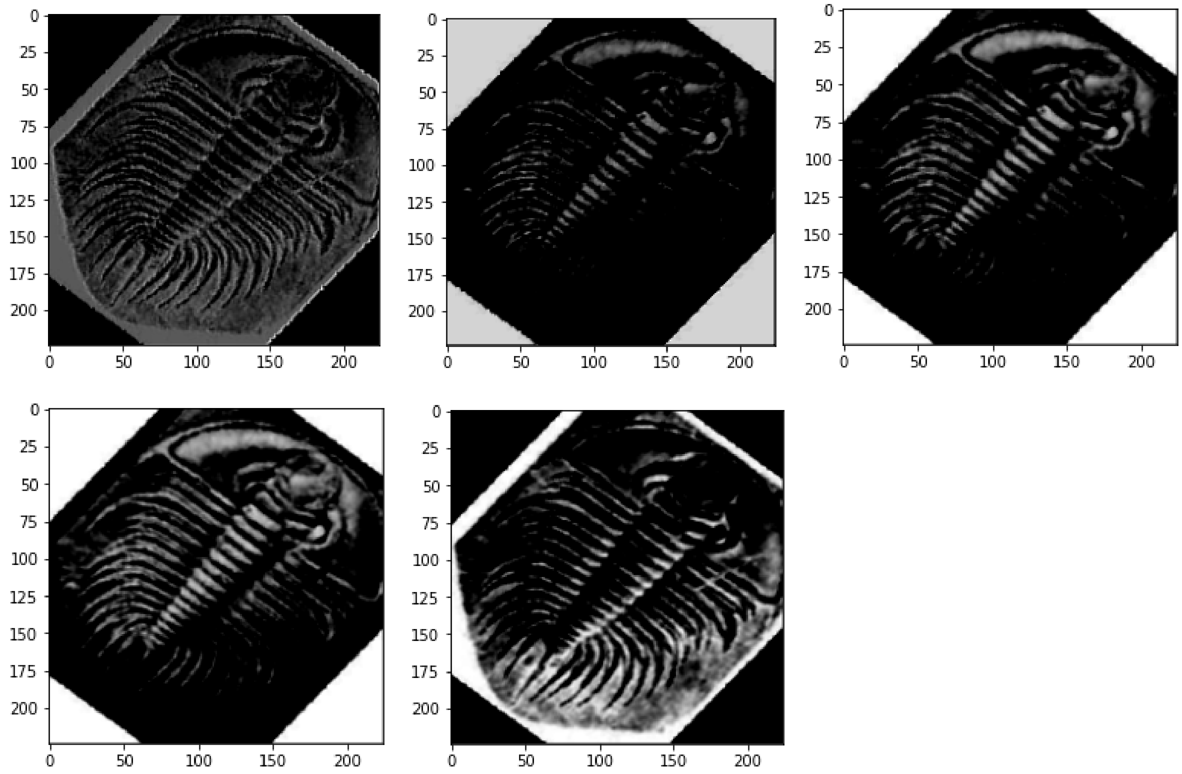


Fig.35 Visualizing patch of Image 3

Part 3)

For Class 1 Image

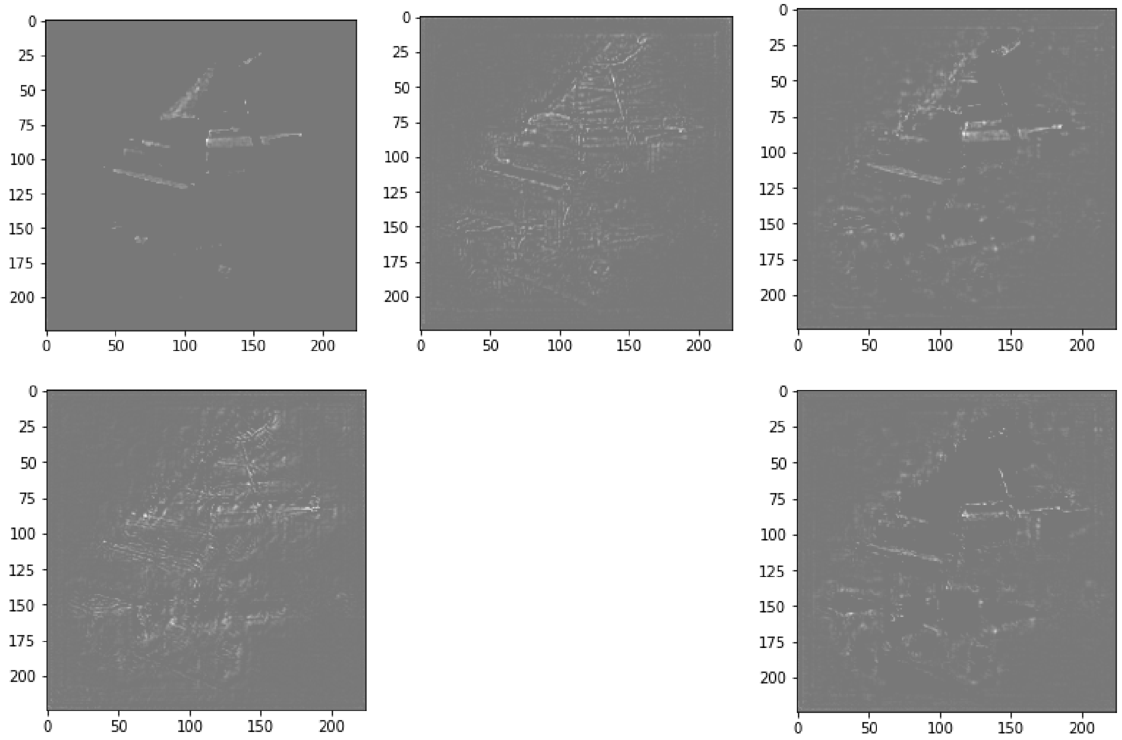


Fig.36 Resultant gradient images after guided backpropagation on Image 1

For Class 2 Image

0
25
50
75
100
125
150
175
200
0
50
100
150
200

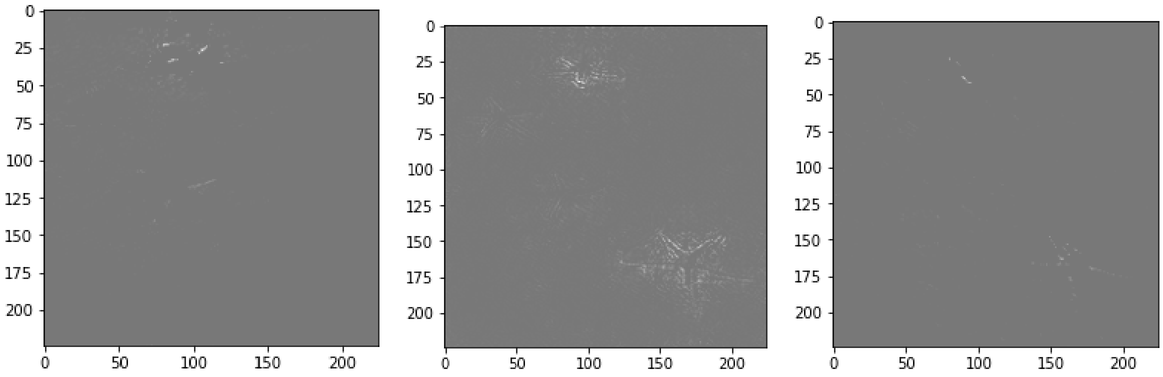


Fig.37 : Resultant gradient images after guided backpropagation on Image 2

For Class 3 Image:

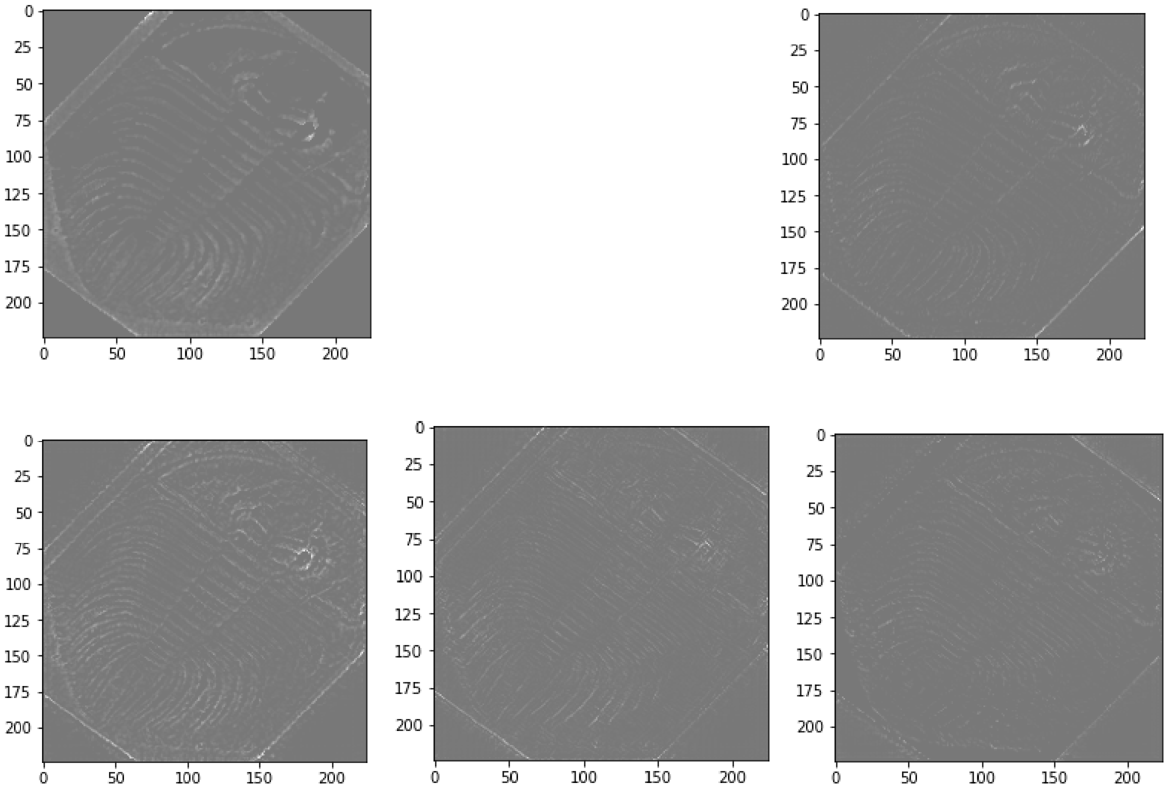


Fig.38 : Resultant gradient images after guided backpropagation on Image 3