Report on

Deep Learning and Applications (CS671)

Assignment 5

Submitted by:

GROUP - 9

Shashank Kapoor (S22022)

Shubham Patwar (T22108)

Syed Rizwan Ali Quadri (T22113)

Assignment Problem Statement

In this assignment, the key objective is to deepen understanding of feature extraction using convolution neural network (CNN) and classification using dense hidden layer. The major task here is classification using this given Architecture network.

Task 1:

Architecture 1

- Image Size
  - Height: 224, width: 224, channel: 3
- Convolution Layer 1
  - Kernel: 8 | Filter Size: 11*11 | Stride: 4 | Padding: 0

    $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 54$

    $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 54$

    $D_2 = K_1 = 8$

- Max Pooling operation
  - Kernel: 8 | Filter Size: 3*3 | Stride: 2| Padding: 0

    $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 26$

    $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 26$

    $D_2 = K_1 = 8$

- Convolution Layer 2
  - Kernel: 16 | Filter Size: 5*5 | Stride: 1 | Padding: 0

    $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 22$

    $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 22$

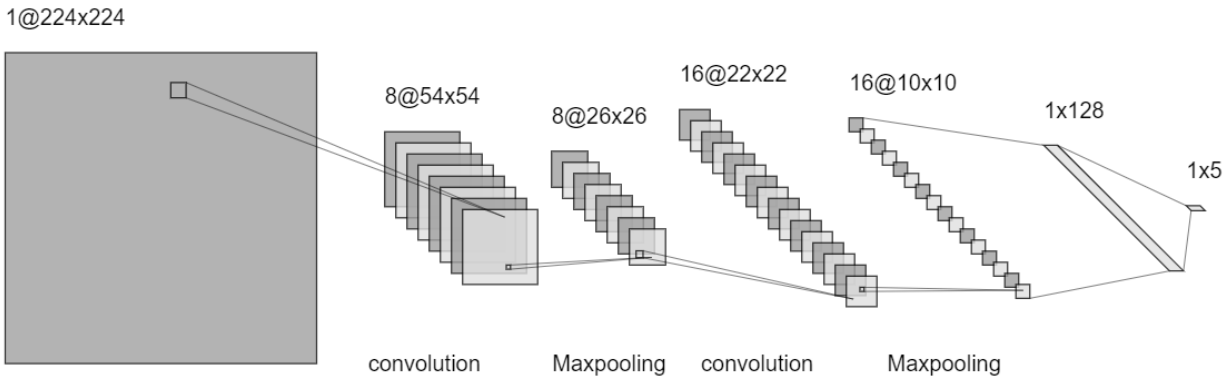
    $D_2 = K_1 = 16$
- Max Pooling operation
  - Kernel: 16 | Filter Size: 3*3 | Stride: 2| Padding: 0

    $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 10$

    $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 10$

    $D_2 = K_1 = 16$

- Then flatten it and add two fully connected layers.
  - Layer 1: 128 hidden nodes with **ReLU** activation function.
  - Layer 2 (Output Layer): 5 neurons with **Softmax** activation function.



Architecture 2:

- Image Size
  - Height: 224, width: 224, channel: 3
- Convolution Layer 1
  - Kernel: 8 | Filter Size: 11*11 | Stride: 4 | Padding: 0

  $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 54$

  $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 54$

  $D_2 = K_1 = 8$
- Max Pooling operation
  - Kernel: 8 | Filter Size: 3*3 | Stride: 2| Padding: 0

  $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 26$

  $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 26$

  $D_2 = K_1 = 8$
- Convolution Layer 2
  - Kernel: 16 | Filter Size: 5*5 | Stride: 1 | Padding: 0

  $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 22$

  $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 22$

  $D_2 = K_1 = 16$
- Max Pooling operation
  - Kernel: 16 | Filter Size: 3*3 | Stride: 2| Padding: 0

  $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 10$

  $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 10$

  $D_2 = K_1 = 16$

- Convolution Layer 3
    - Kernel: 32 | Filter Size: 3*3 | Stride: 1 | Padding: 0

      $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 8$

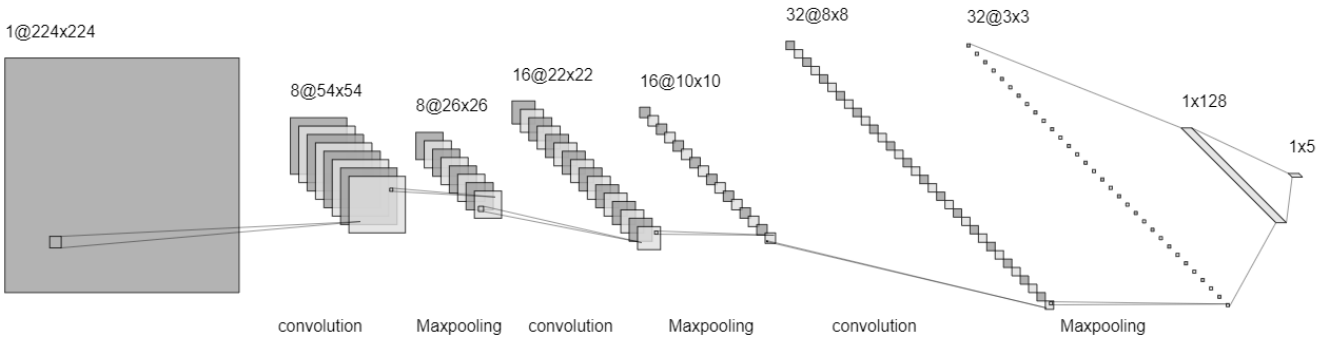
      $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 8$

      $D_2 = K_1 = 32$
- Max Pooling operation
    - Kernel: 32 | Filter Size: 3*3 | Stride: 2| Padding: 0

      $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 3$

      $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 3$

      $D_2 = K_1 = 32$
- Then flatten it and add two fully connected layers.
    - Layer 1: 128 hidden nodes with **ReLU** activation function.
    - Layer 2 (Output Layer): 5 neurons with **Softmax** activation function.



Architecture 3

- Image Size
    - Height: 224, width: 224, channel: 3
- Convolution Layer 1
    - Kernel: 8 | Filter Size: 11*11 | Stride: 4 | Padding: 0

      $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 54$

      $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 54$

      $D_2 = K_1 = 8$

- Max Pooling operation
  - Kernel: 8 | Filter Size: 3*3 | Stride: 2| Padding: 0

    $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 26$

    $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 26$

    $D_2 = K_1 = 8$
- Convolution Layer 2
  - Kernel: 16 | Filter Size: 5*5 | Stride: 1 | Padding: 0

    $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 22$

    $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 22$

    $D_2 = K_1 = 16$
- Max Pooling operation
  - Kernel: 16 | Filter Size: 3*3 | Stride: 2| Padding: 0

    $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 10$

    $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 10$

    $D_2 = K_1 = 16$
- Convolution Layer 3
  - Kernel: 32 | Filter Size: 3*3 | Stride: 1 | Padding: 0

    $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 8$

    $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 8$

    $D_2 = K_1 = 32$
- No max pooling
- Convolution Layer 4
  - Kernel: 64 | Filter Size: 3*3 | Stride: 1 | Padding: 0

    $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 6$

    $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 6$

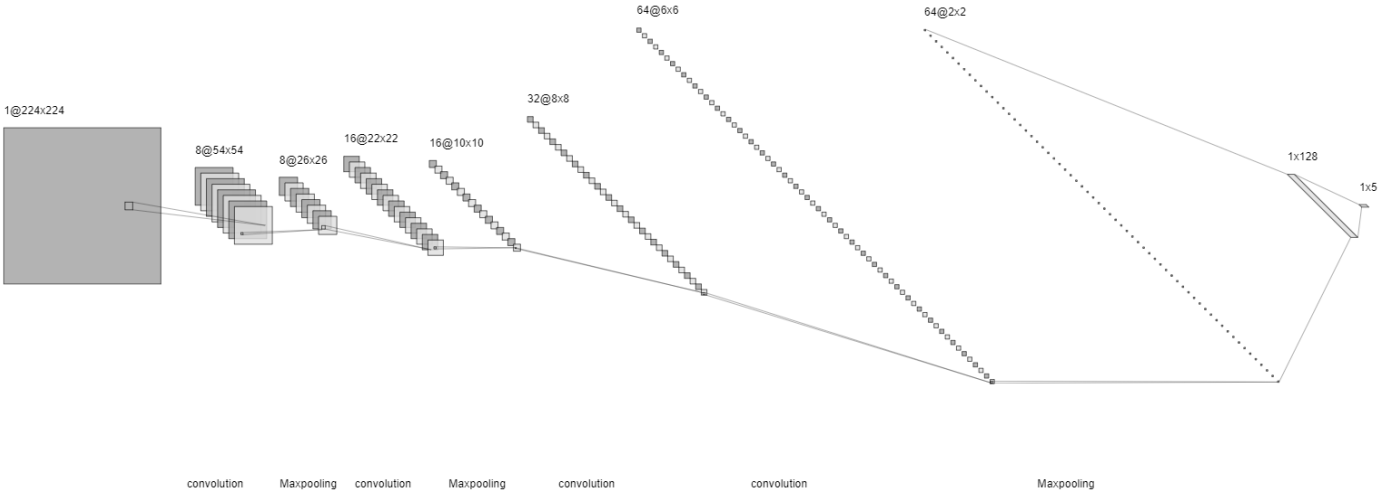
    $D_2 = K_1 = 64$
- Max Pooling operation
  - Kernel: 16 | Filter Size: 3*3 | Stride: 2| Padding: 0

    $w_2 = \frac{(W1 - F + 2P)}{S} + 1 = 2$

    $H_2 = \frac{(H1 - F + 2P)}{S} + 1 = 2$

    $D_2 = K_1 = 16$

- Then flatten it and add two fully connected layers.
    - Layer 1: 128 hidden nodes with **ReLU** activation function.
    - Layer 2 (Output Layer): 5 neurons with **Softmax** activation function.



Overall Result:

Architecture	Architecture Description	No. of Epochs	Training Accuracy	Validation Accuracy	Testing Accuracy
Architecture 1	Input Layer Convolution Layer Max-pool Layer Convolution Layer Max-pool Layer Dense Hidden Layer Output Layer	**11**	**1.0000**	**0.8199**	**0.7400**
Architecture 2	Convolution Layer Max-pool Layer Convolution Layer Max-pool Layer } Architecture 1 Convolution Layer Max-Pool Layer Dense Hidden Layer Output Layer	10	1.0000	0.7599	0.6900
Architecture 3	Convolution Layer Max-pool Layer Convolution Layer Max-pool Layer } Architecture 1 Convolution Layer Convolution Layer Max-pool Layer Dense Hidden Layer Output Layer	7	0.9919	0.6999	0.7000

* Mark in bold is Best architecture according to the validation accuracy

Confusion Matrix of CNN Architecture:

1. **Architecture 1**

Predicted label	Actual Label				
	Bonsai	Chandiler	Kangroo	Laptop	Watch
Bonsai	14	4	1	1	0
Chandlier	3	13	1	2	1
Kangroo	3	1	15	1	0
Laptop	1	0	2	16	1
Watch	1	1	2	0	16

Testing Accuracy: 0.74

2. **Architecture 2**

Predicted label	Actual Label				
	Bonsai	Chandiler	Kangroo	Laptop	Watch
Bonsai	15	1	1	1	2
Chandlier	2	12	3	2	1
Kangroo	2	2	16	0	0
Laptop	0	1	5	13	1
Watch	1	1	5	0	13

Testing Accuracy: 0.69

3. **Architecture 3**

Predicted label	Actual Label				
	Bonsai	Chandiler	Kangroo	Laptop	Watch
Bonsai	15	1	4	0	0
Chandlier	2	15	3	0	0
Kangroo	2	0	18	0	0
Laptop	2	2	5	11	0
Watch	2	2	5	0	11

Testing Accuracy: 0.7

Best Architecture , Architecture 1 Results:
Image used for getting the feature map

Fig. Image from “bonsai” class

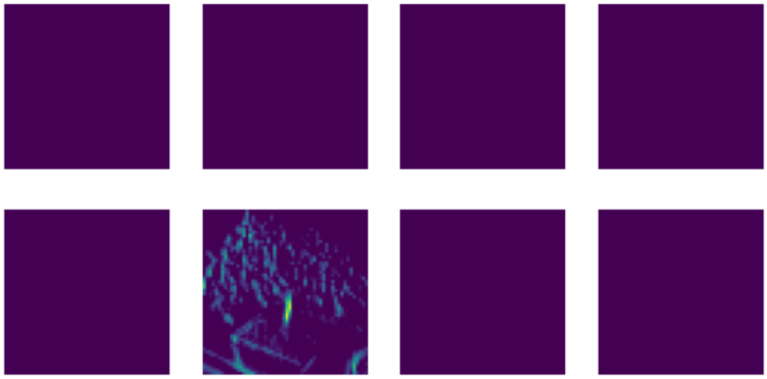


Fig. 8 Feature map from the First Convolution Layer

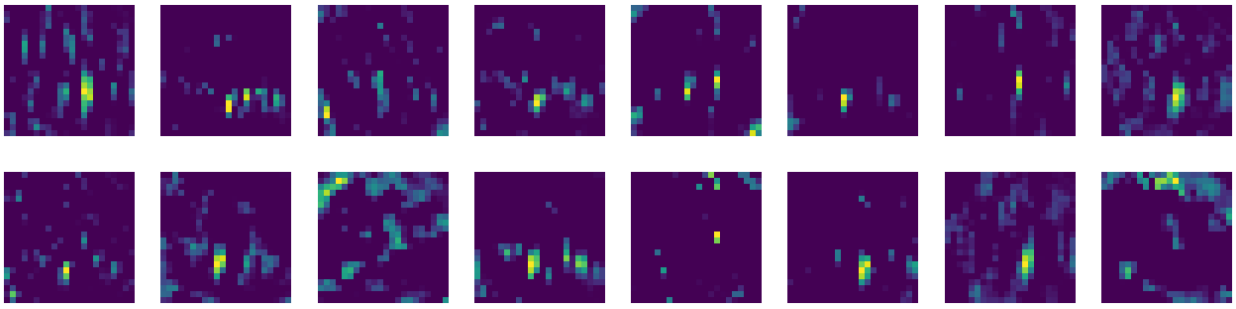


Fig. 16 Feature map from the Second Convolution Layer

Task 2:

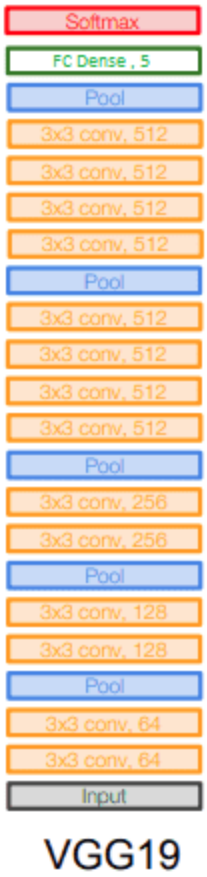


Fig. Showing the Architecture of the VVG19

Architecture	**Architecture Description**	**No. of Epochs**	**Training Accuracy**	**Validation Accuracy**	**Testing Accuracy**
VVG 19 Architecture	Input Layer Convolution Layer Max-pool Layer Convolution Layer Max-pool Layer Dense Hidden Layer Output Layer	10	1.0000	0.9800	0.9900

Predicted label	Actual Label					
		Bonsai	Chandiler	Kangroo	Laptop	Watch
	Bonsai	20	0	0	0	0
	Chandlier	0	20	0	0	0
	Kangroo	1	0	19	0	0
	Laptop	0	0	0	20	0
	Watch	0	0	0	0	20
						Testing Accuracy: 0.99

Guided-backpropagation

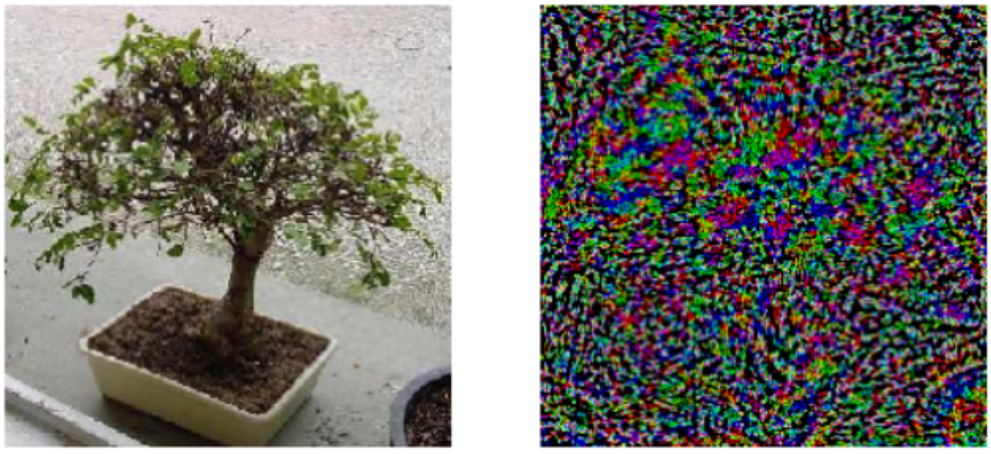


Fig. Visualizing the influence of the neuron in "Bonsai class".

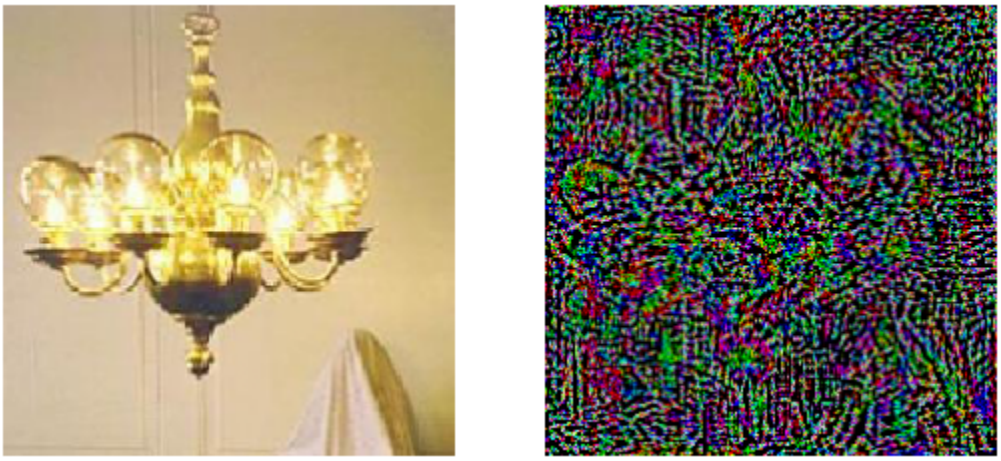


Fig. Visualizing the influence of the neuron in "Chandlier class".

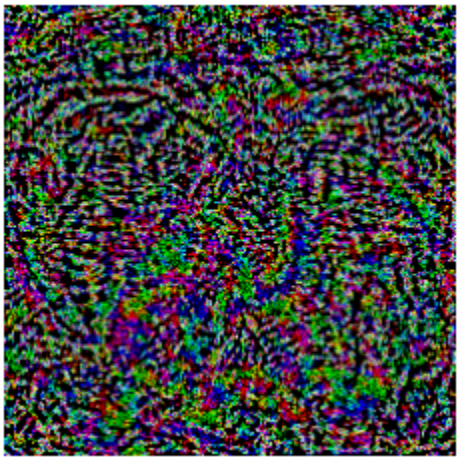



Fig. Visualizing the influence of the neuron in "Kangaroo class".


Fig. Visualizing the influence of the neuron in "Laptop class".

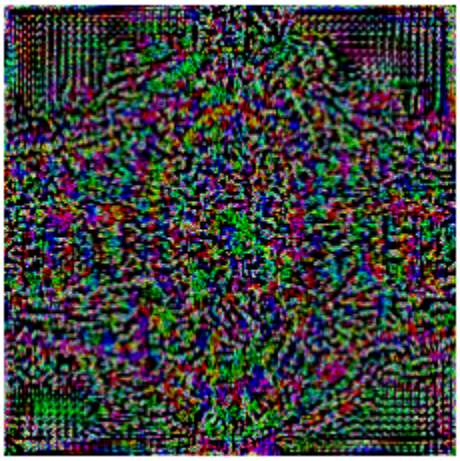
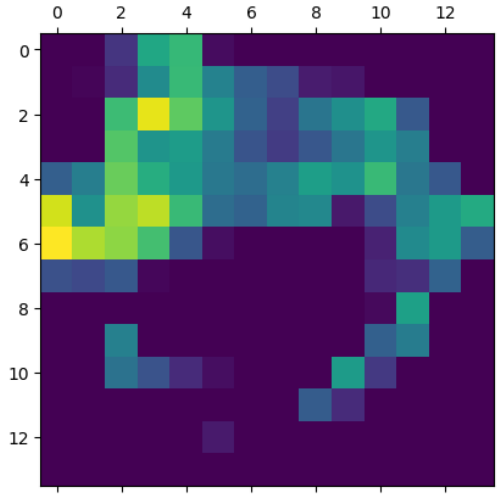



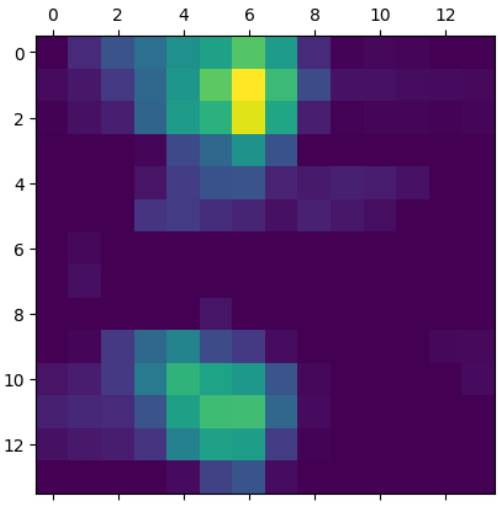
Fig. Visualizing the influence of the neuron in "Watch class".

Gradcam output:



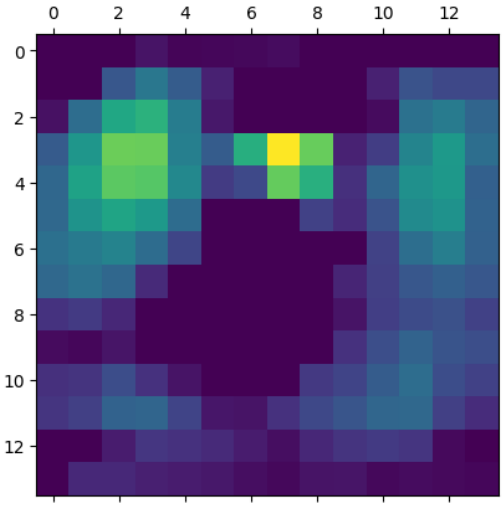
a. **b.**

Fig. a.Activated Neuron of the "Bonsai Class" as a heat map. b. Overlapping the Heat map on to the image to visualize the part that activate the neuron



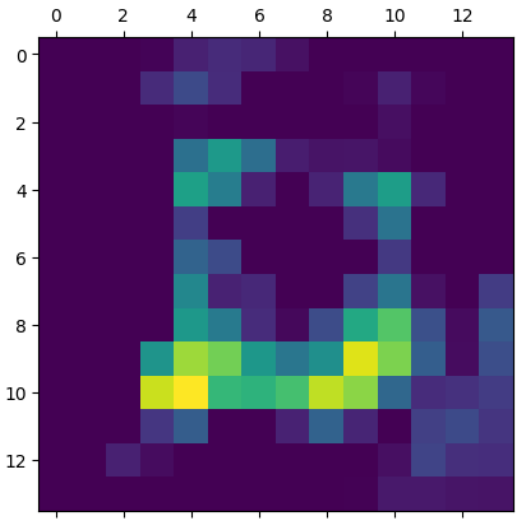
b. **b.**

Fig. a.Activated Neuron of the "Chandelier Class" as a heat map. b. Overlapping the Heat map on to the image to visualize the part that activate the neuron



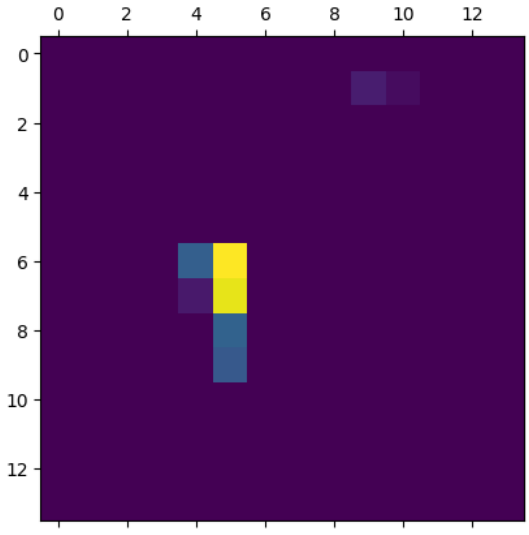
c. **b.**

Fig. a.Activated Neuron of the "Kangaroo Class" as a heat map. b. Overlapping the Heat map on to the image to visualize the part that activate the neuron



d. **b.**

Fig. a.Activated Neuron of the "Laptop Class" as a heat map. b. Overlapping the Heat map on to the image to visualize the part that activate the neuron



e. **b.**

Fig. a.Activated Neuron of the "Watch Class" as a heat map. b. Overlapping the Heat map on to the image to visualize the part that activate the neuron