



INDIAN INSTITUTE OF TECHNOLOGY DELHI

COL 341
Fundamentals of Machine Learning
Assignment Make-up

Mrunmayi Bhalerao 2019CS50425

Contents

1 Task 1	2
1.a Model Architecture	2
1.b Performance Metric	3
1.c Ablation Study	5
1.c.1 Batch Normalization	5
1.c.2 Dropout	7
1.c.3 Freezing Layers	8
1.c.4 Leaky ReLU	9
1.c.5 Sigmoid	11
1.c.6 Tanh	12
1.c.7 LR = 0.01	13
1.c.8 LR = 0.005	15
2 Task 2	16
2.a Feature Attribution with Integrated Gradients	16
2.a.1 Theory	16
2.a.2 Analysis	16

The code can be found at this link: [https://github.com/MrunmayiB/COL341_Make_U](https://github.com/MrunmayiB/COL341_Make_Up-Assignment)p-Assignment

The links to the model weights can be found at: <https://drive.google.com/drive/folders/1fhvptsXzTXF3ioGD2k7dS9UQsMMu4MKT?usp=sharing>

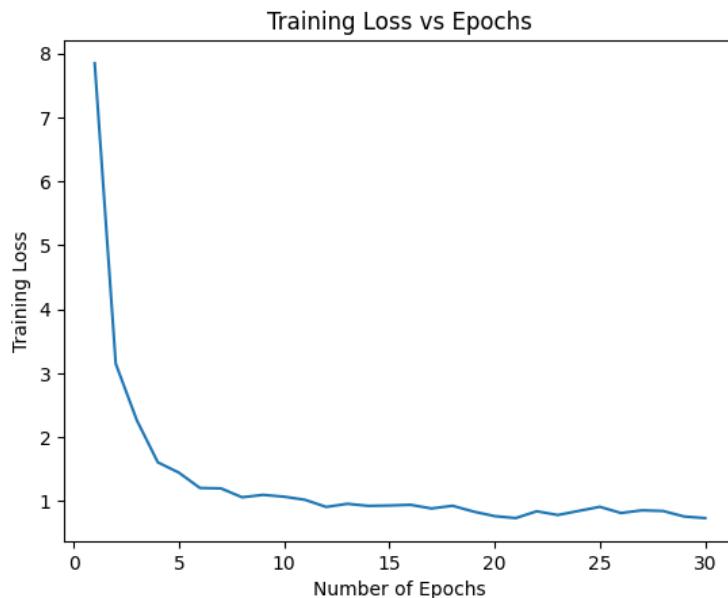
1 Task 1

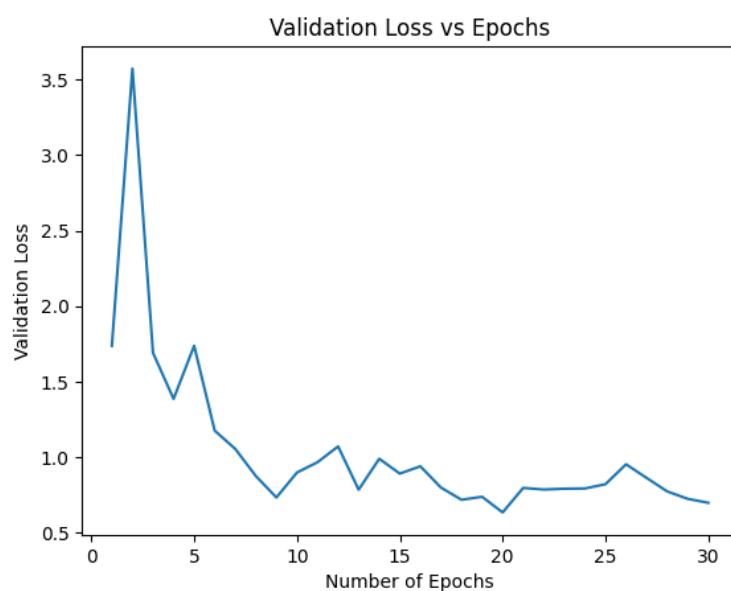
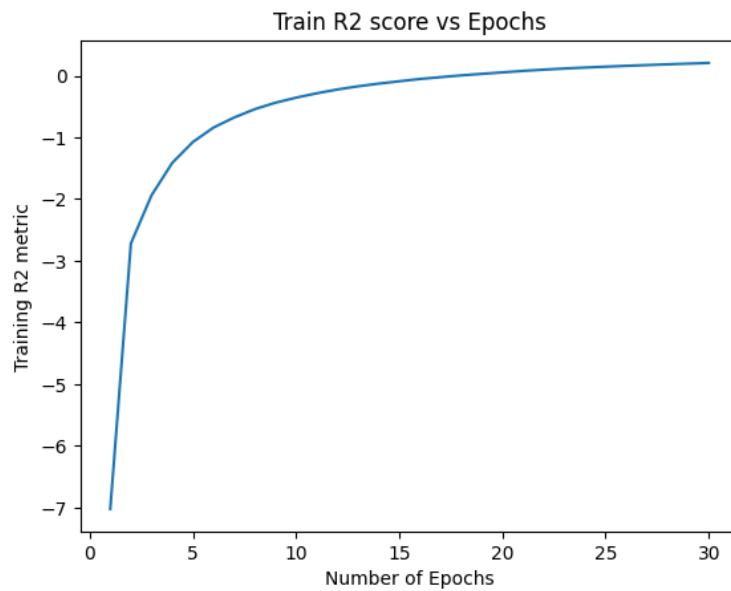
1.a Model Architecture

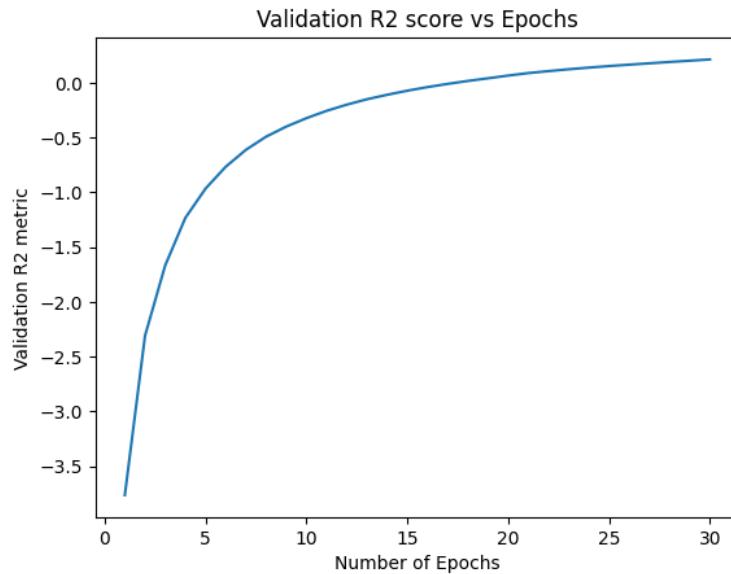
```
1 VGG(
2     features): Sequential(
3         (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
4         (1): ReLU(inplace=True)
5         (2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
6         (3): ReLU(inplace=True)
7         (4): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
8             ceil_mode=False)
9         (5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
10        (6): ReLU(inplace=True)
11        (7): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
12        (8): ReLU(inplace=True)
13        (9): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
14            ceil_mode=False)
15        (10): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
16        (11): ReLU(inplace=True)
17        (12): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
18        (13): ReLU(inplace=True)
19        (14): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
20        (15): ReLU(inplace=True)
21        (16): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
22        (17): ReLU(inplace=True)
23        (18): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
24            ceil_mode=False)
25        (19): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
26        (20): ReLU(inplace=True)
27        (21): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
28        (22): ReLU(inplace=True)
29        (23): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
30        (24): ReLU(inplace=True)
31        (25): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

```
30     (26): ReLU(inplace=True)
31     (27): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
32         ceil_mode=False)
33     (28): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
34         1))
35     (29): ReLU(inplace=True)
36     (30): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
37         1))
38     (31): ReLU(inplace=True)
39     (32): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
40         1))
41     (33): ReLU(inplace=True)
42     (34): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1,
43         1))
44     (35): ReLU(inplace=True)
45     (36): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
46         ceil_mode=False)
47   )
48   (avgpool): AdaptiveAvgPool2d(output_size=(7, 7))
49   (classifier): Sequential(
50     (0): Linear(in_features=25088, out_features=4096, bias=True)
51     (1): ReLU(inplace=True)
52     (2): Dropout(p=0.5, inplace=False)
53     (3): Linear(in_features=4096, out_features=4096, bias=True)
54     (4): ReLU(inplace=True)
55     (5): Dropout(p=0.5, inplace=False)
56     (6): Linear(in_features=4096, out_features=1, bias=True)
57   )
58 )
```

1.b Performance Metric

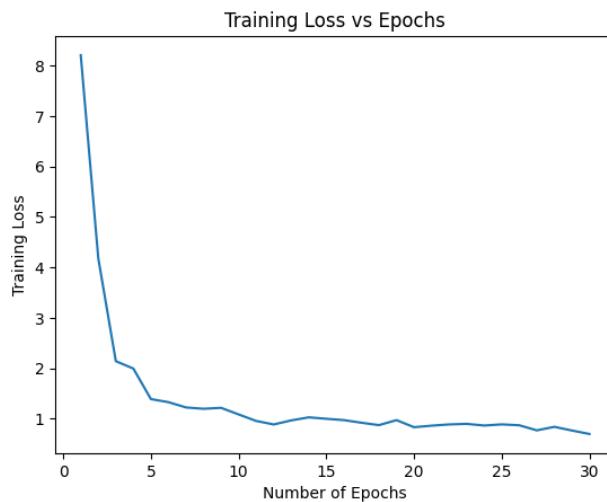




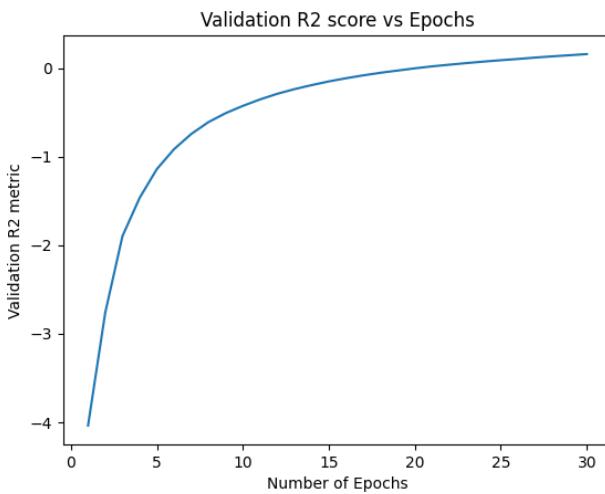
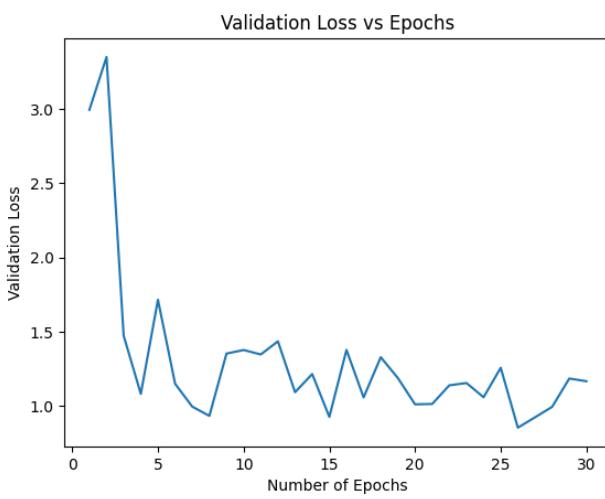
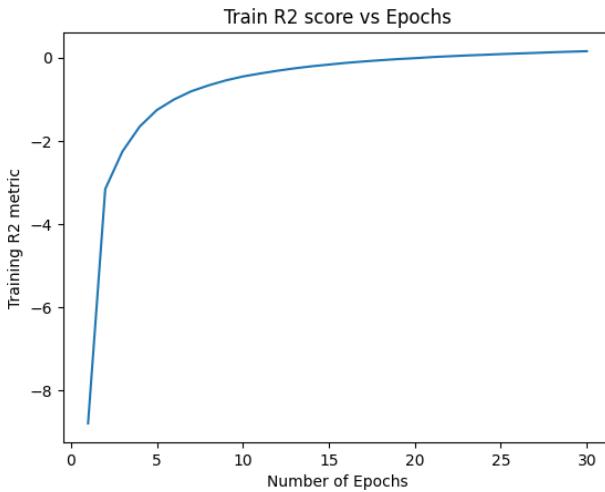


1.c Ablation Study

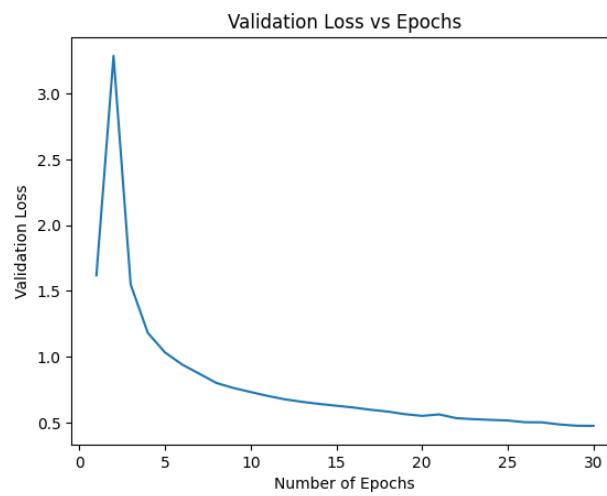
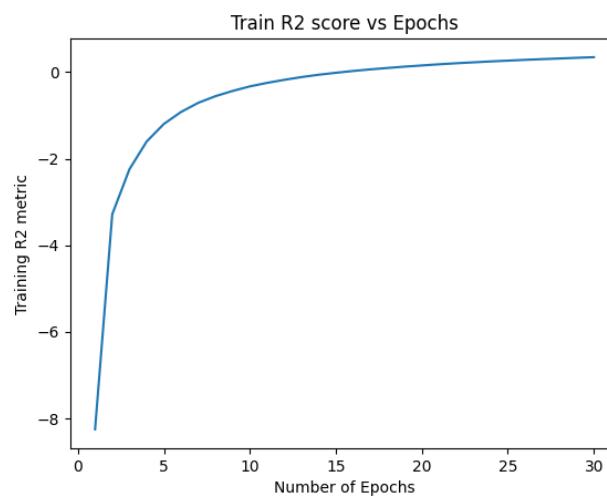
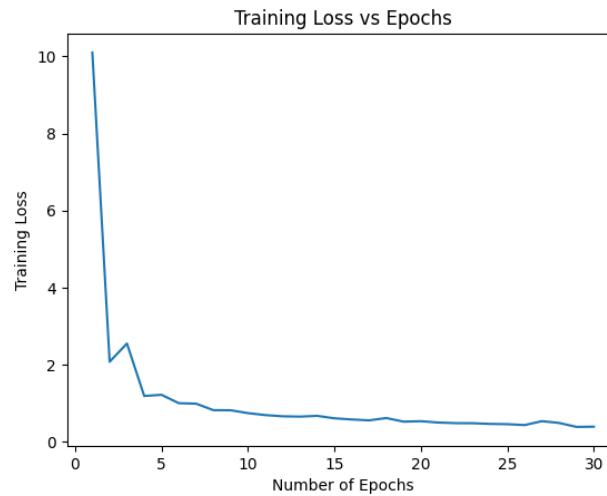
1.c.1 Batch Normalization

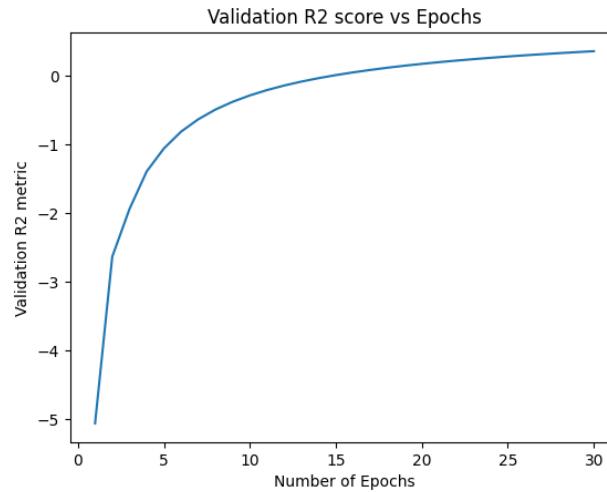


Assignment 2

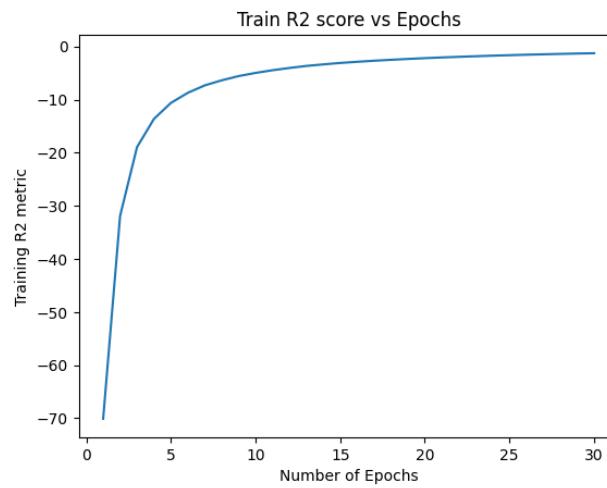
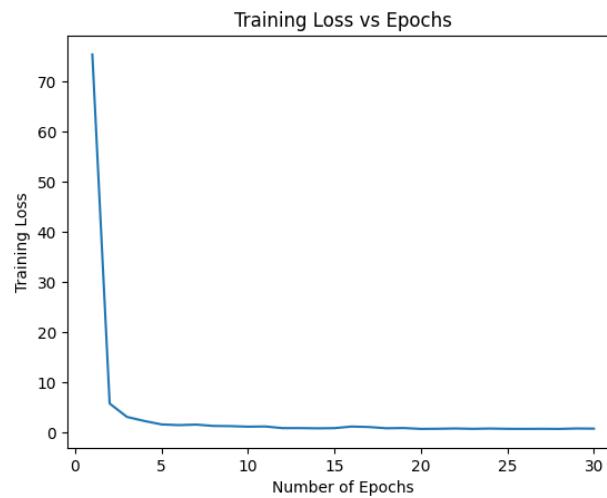


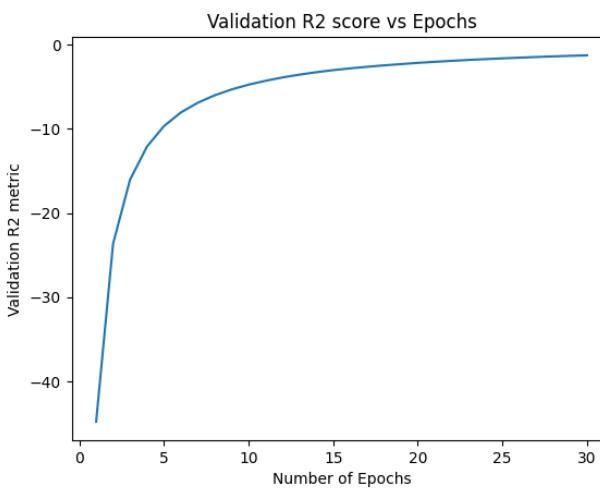
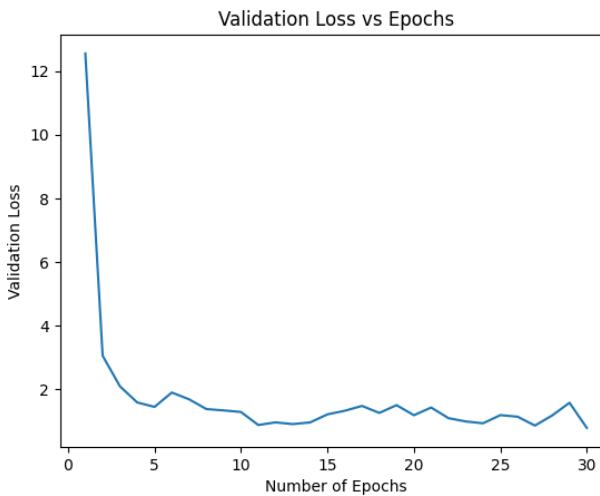
1.c.2 Dropout



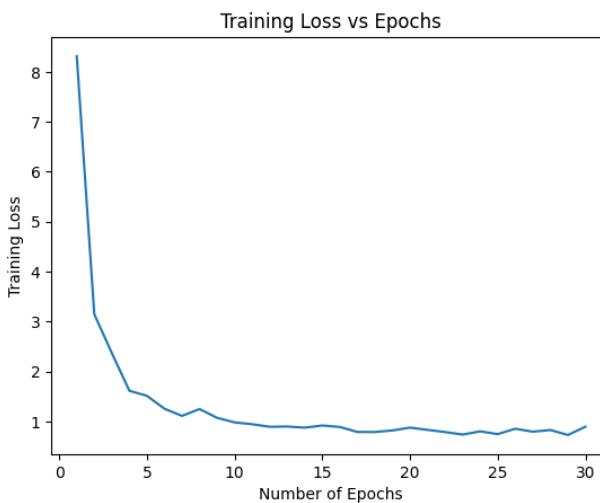


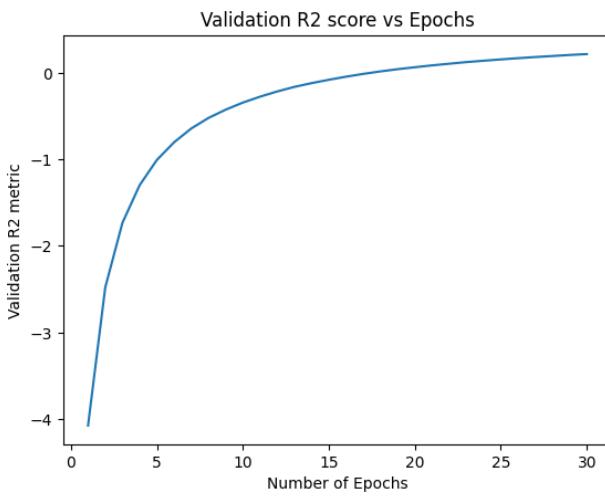
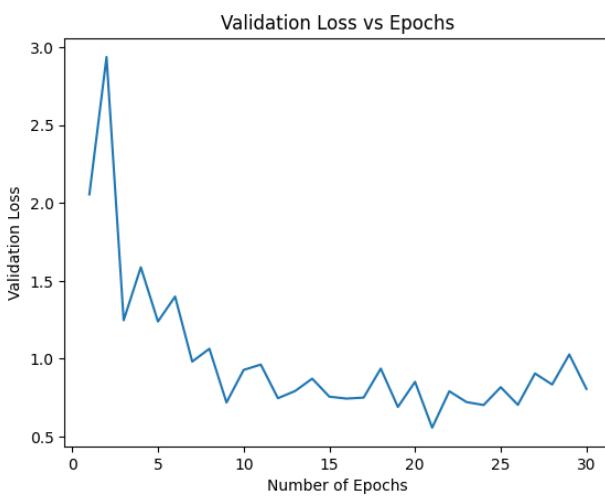
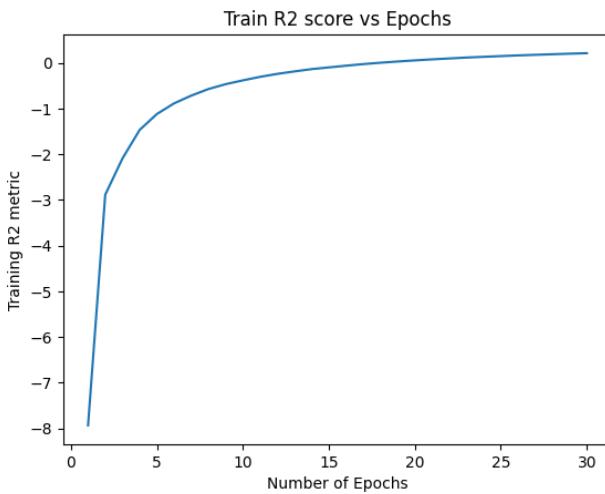
1.c.3 Freezing Layers



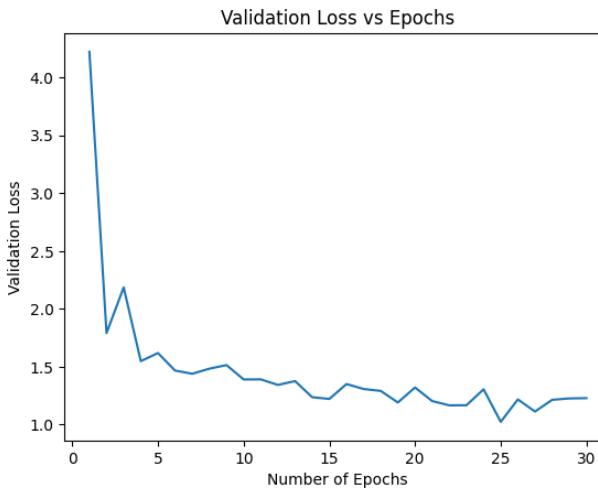
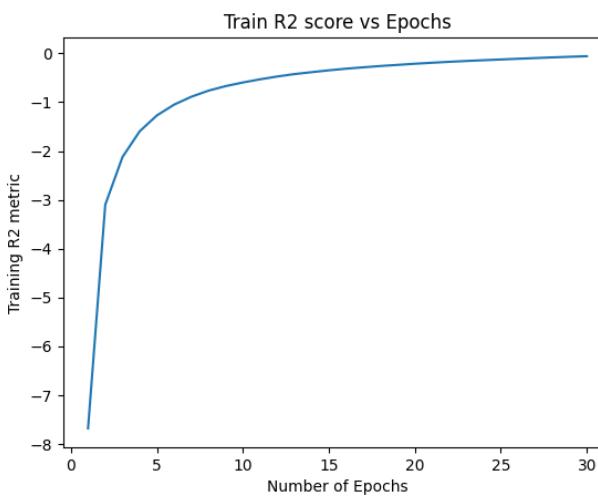
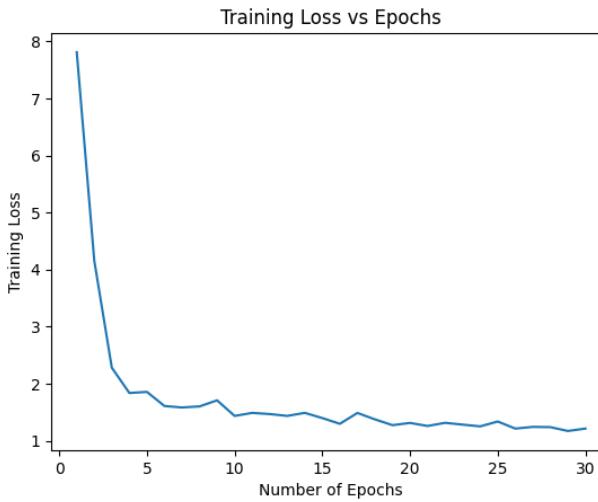


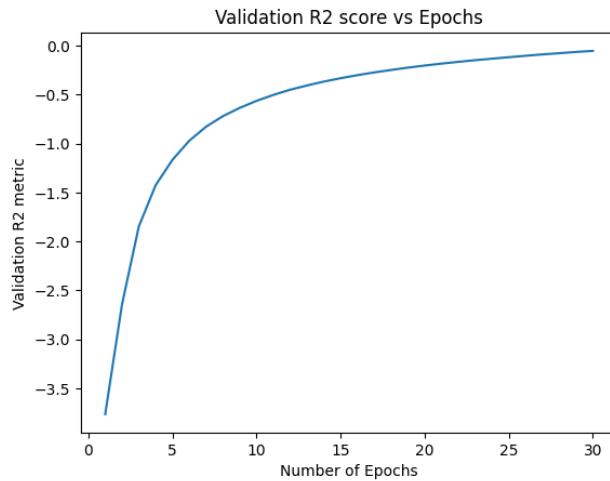
1.c.4 Leaky ReLU



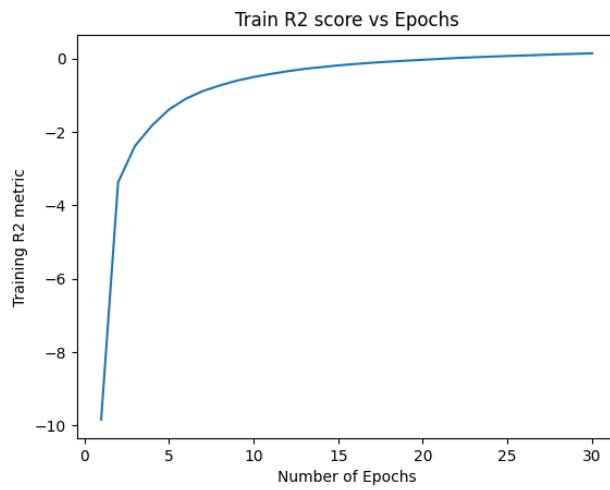
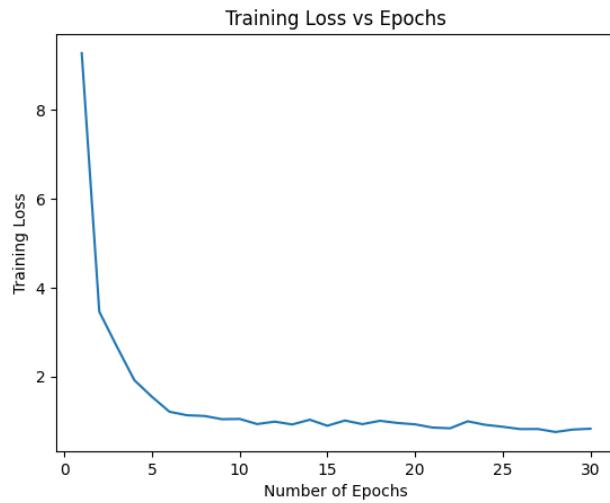


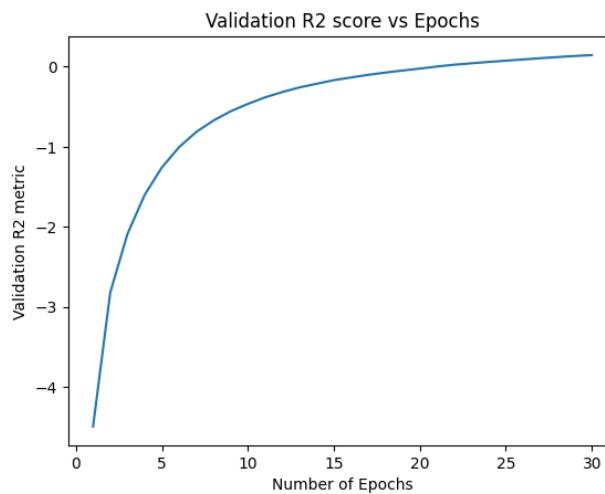
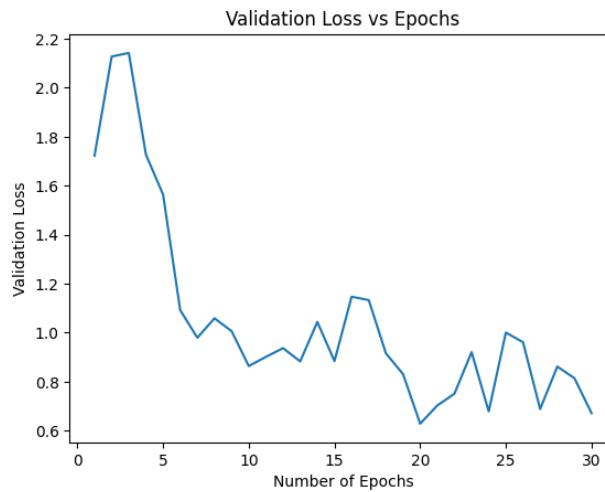
1.c.5 Sigmoid



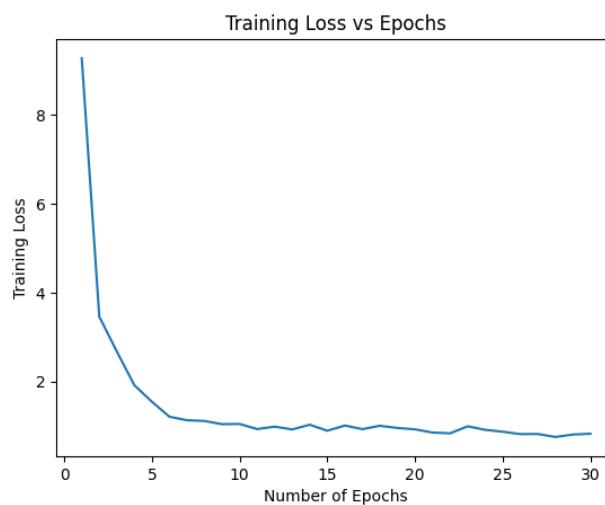


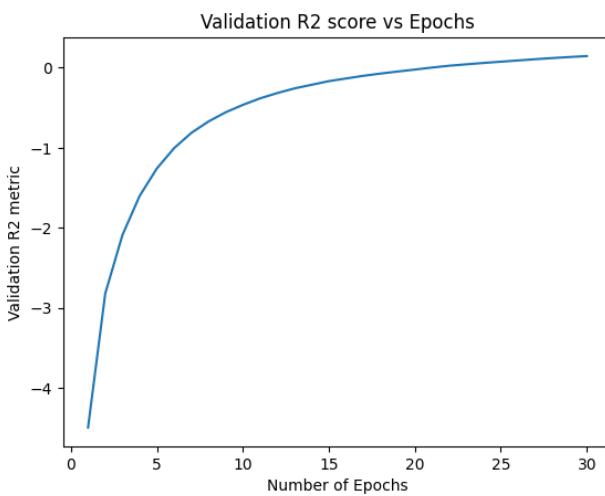
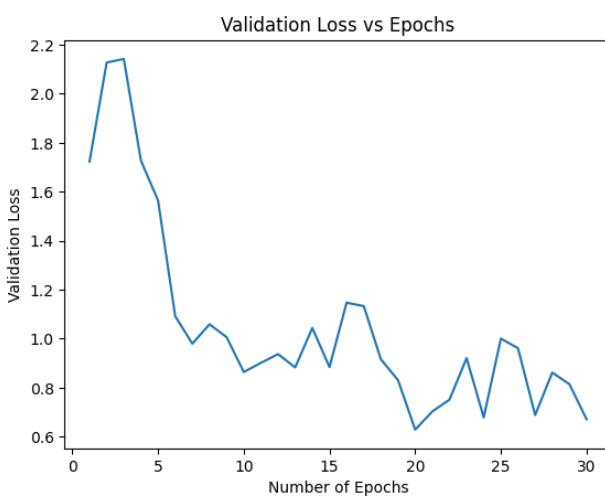
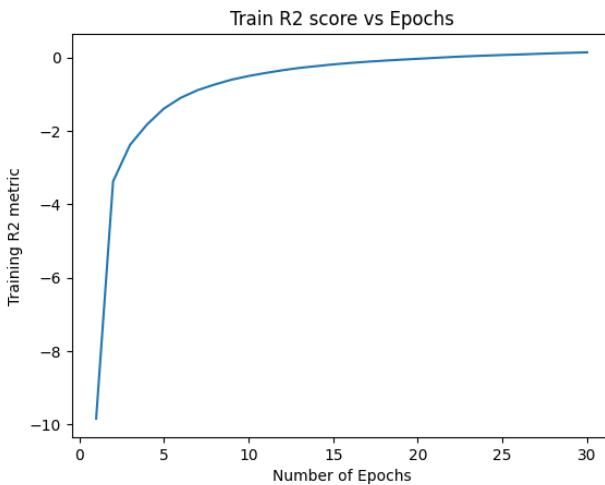
1.c.6 Tanh



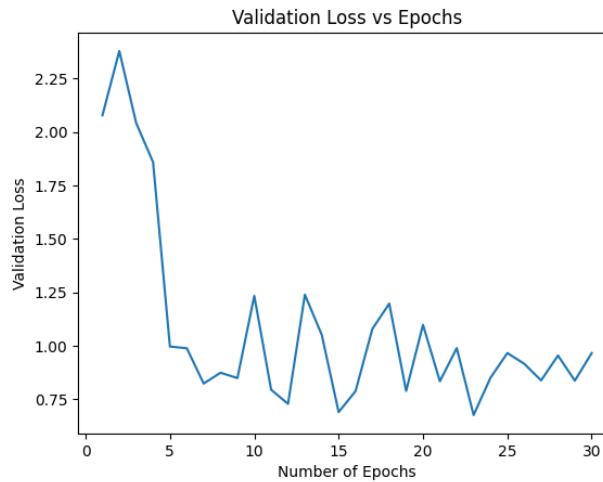
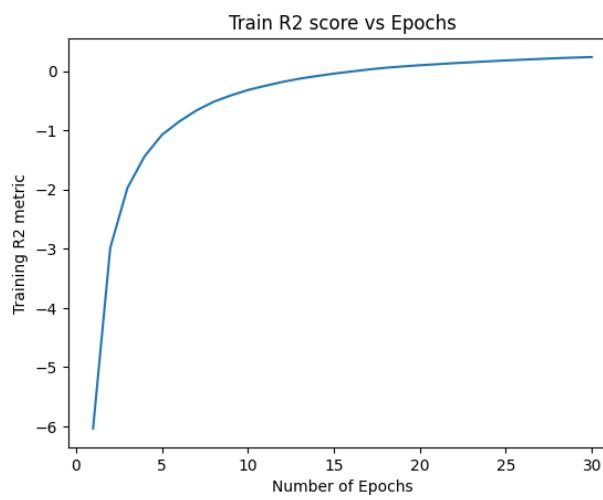
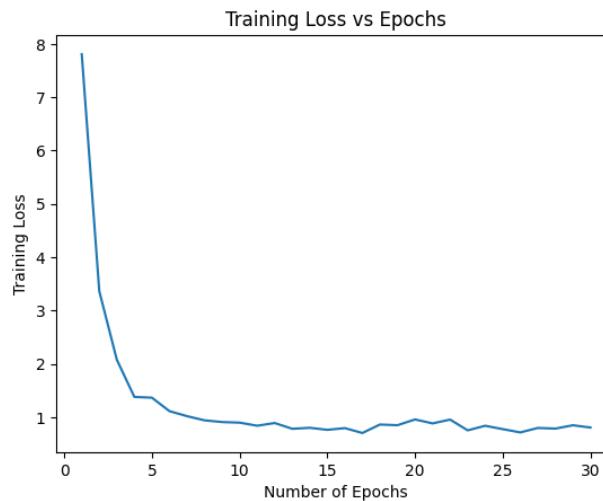


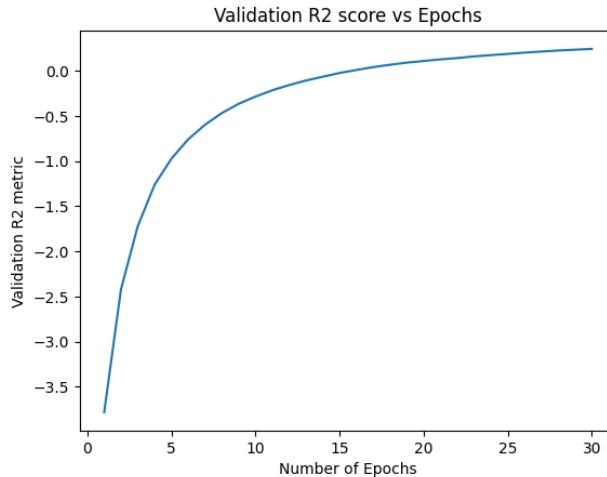
1.c.7 LR = 0.01





1.c.8 LR = 0.005





2 Task 2

2.a Feature Attribution with Integrated Gradients

We finetuned the VGG19 model using the Adam optimiser, while keeping the weights of all layers except the last one frozen. Then we used the method of integrated gradients for computing feature attribution, i.e., each pixel of the input image is assigned a score.

2.a.1 Theory

Integrated gradients computes the integral of the gradients of the output of the model with respect to the input image pixels. It is built on the axioms of sensitivity and implementation invariance.

2.a.2 Analysis

We show 8 images from our dataset below, along with their feature attribution heatmaps. These heatmaps represent which regions of the image the model is most sensitive to. We present each of the images below, followed by some observations.

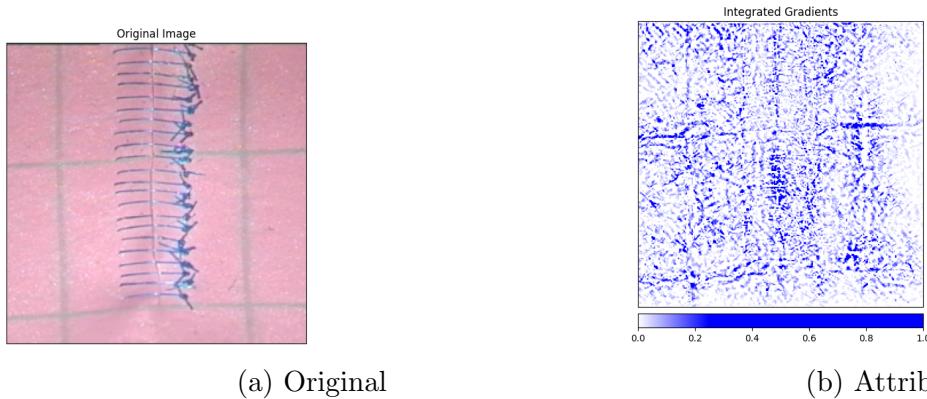
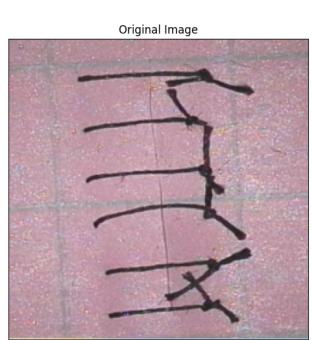
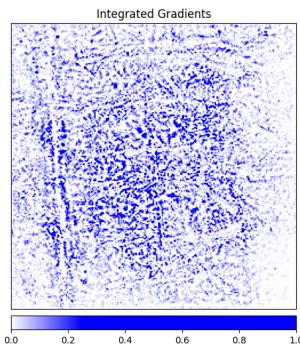


Figure 1: Image 1



(a) Original

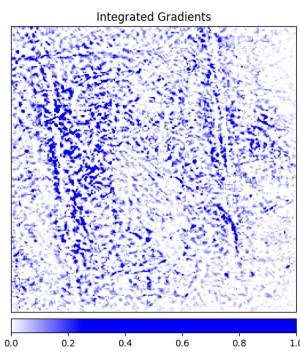


(b) Attribution Map

Figure 2: Image 2

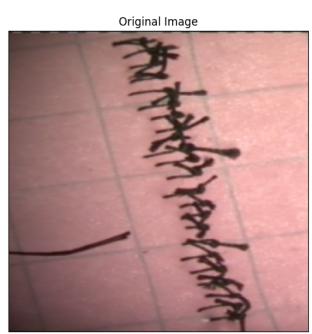


(a) Original

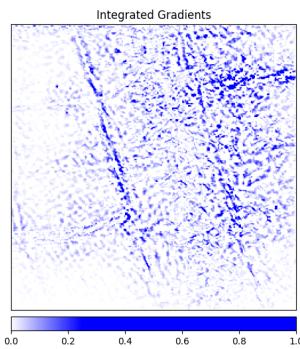


(b) Attribution Map

Figure 3: Image 3

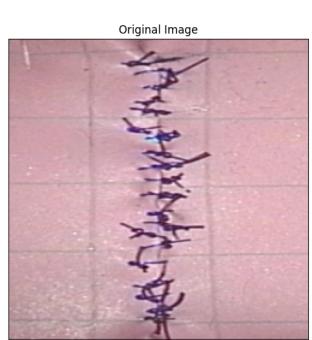


(a) Original

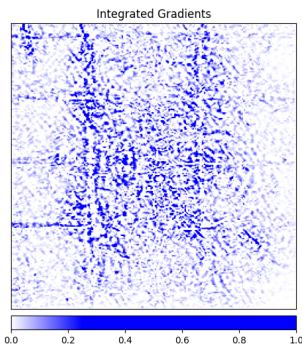


(b) Attribution Map

Figure 4: Image 4

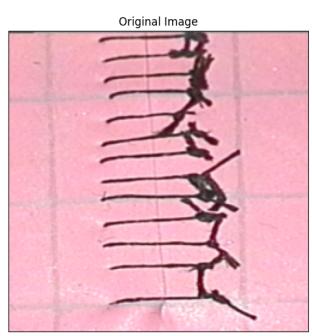


(a) Original

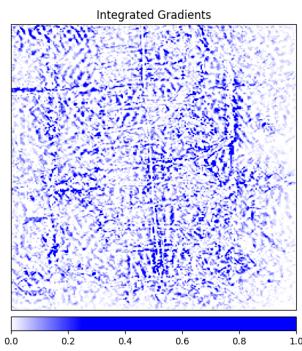


(b) Attribution Map

Figure 5: Image 5



(a) Original

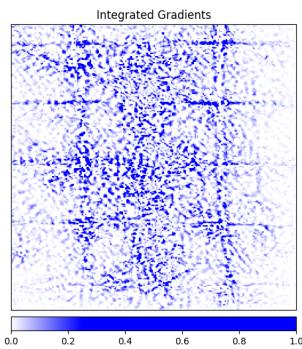


(b) Attribution Map

Figure 6: Image 6



(a) Original



(b) Attribution Map

Figure 7: Image 7

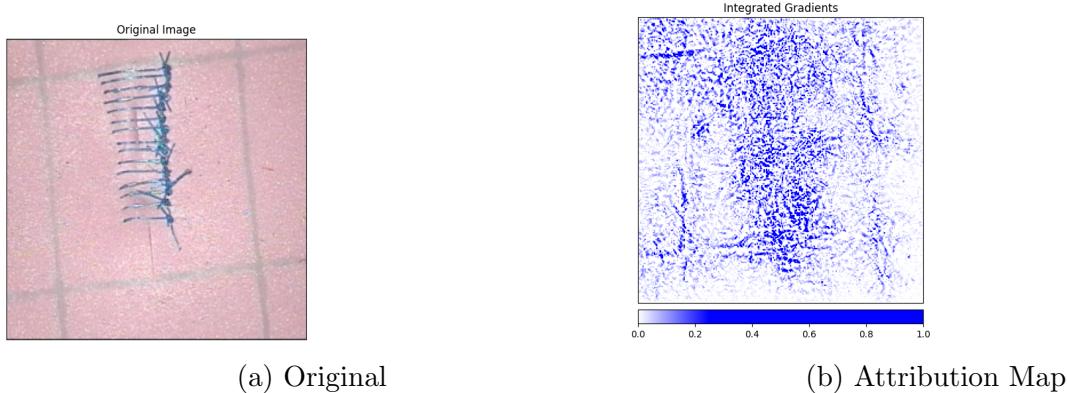


Figure 8: Image 8

- As we can see, the model seems to be very sensitive to the grid-like boundaries present in each image. In all of the images, the pixels along the grid boundaries form a dark-coloured line.
- The model is mostly concentrating on the central part of each image, as the pixels along the right and left boundaries have low attribution scores. This is an encouraging sign, as the sutures are also present in the center of most images.
- In fact, if we look at Images 4, 5, 6, 7 and 8, they suggest that the model concentrates on the grid boundaries which are nearest to/surrounding the sutures. This can be seen by the fact that there are certain grid boundaries which are very dark in colour, but they are in different parts for each image.