
CS60010 - Deep Learning

Final Project Report

Group - 12

Group Members: Sachin Vashistha (22CS91R02), Sidharth Vishwakarma (20CS10082), Tarun Mohanty (22CS60R28), Saras Umakant Pantulwar (20CS30046)

Project Title: Kaggle Photo Reconstruction Challenge

1. Best score achieved

The best score achieved by Group 12 is **0.15866**.

2. Summary of the work carried out: Details of the Models used

2.1 Model 1: UNET model. The details about hyper-parameters is as follows:

- Batch Size: 32
- Number of Epochs: 20
- Input Image size = Output Image size: (256, 256, 3)
- Optimizer: Adam

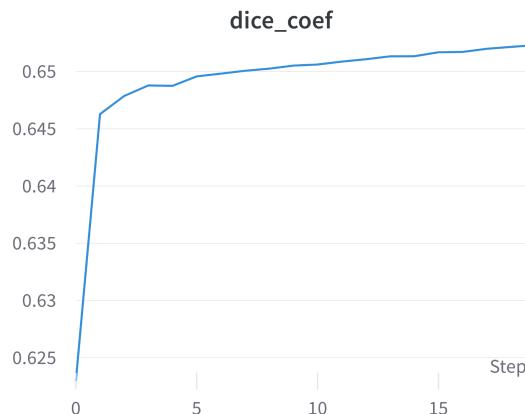
2.1.1 Experiments Carried Out:

- **Choice of metrics:** In tasks like Image reconstruction, two commonly used metrics are IoU (Intersection over Union) and Dice Coefficient. In this model, Dice Coefficient is used whose goal is to maximize the area of overlap between the predicted pixel and the ground truth pixel divided by their union.
- **Loss Function:** Loss function “Mean Absolute Error”(MAE) is used in the experiment.

2.1.2 Results and Analysis

- Plot of Dice Coefficient

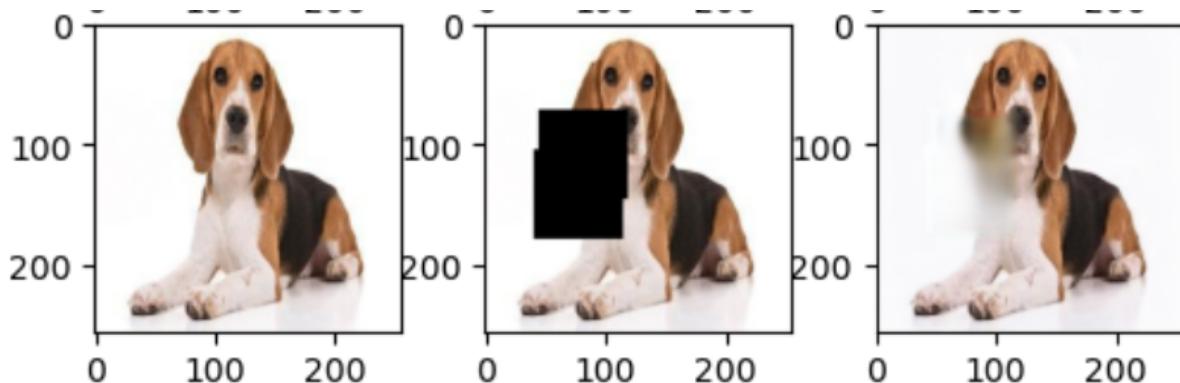
Plot of Loss Function



Dice Coefficient Value at the end of 20 epochs: 0.6523
Loss Value at the end of 20 epochs: 0.0234

2.1.3. Model Performance / Score

The score achieved using Model 1 is **0.21662**. This score is achieved by making a wrong assumption about channels that 0 belongs to RED and 2 belongs to BLUE. Then, the team corrected the order. The best score achieved is then **0.15866**. This is the best score Group-12 has achieved. The performance shown by the **Model 1** is as shown:



The first image is the original image. Second image is the masked image having two square holes. The third image is the image predicted by **Model 1**.

2.2 Model 2: Partial Convolution 2D model.

The details about hyper-parameters is as follows:

- Batch Size: 32
- Number of Epochs: 20
- Input Image size = Output Image size: (256, 256, 3)
- Optimizer: Adam

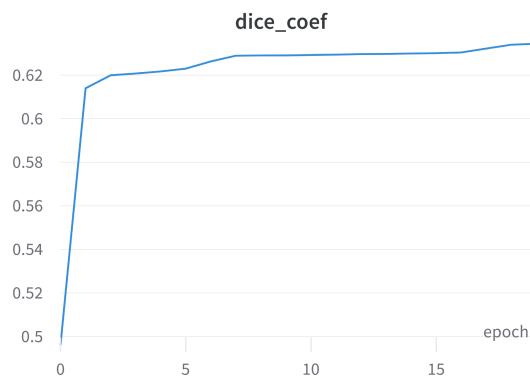
2.2.1 Experiments Carried Out:

- **Choice of metrics:** In this model, Dice Coefficient is used, whose goal is to maximize the area of overlap between the predicted pixel and the ground truth pixel divided by their union.
- **Loss Function:** Loss function “Mean Absolute Error”(MAE) is used in the experiment.

2.2.2 Results and Analysis

- **Plot of Dice Coefficient**

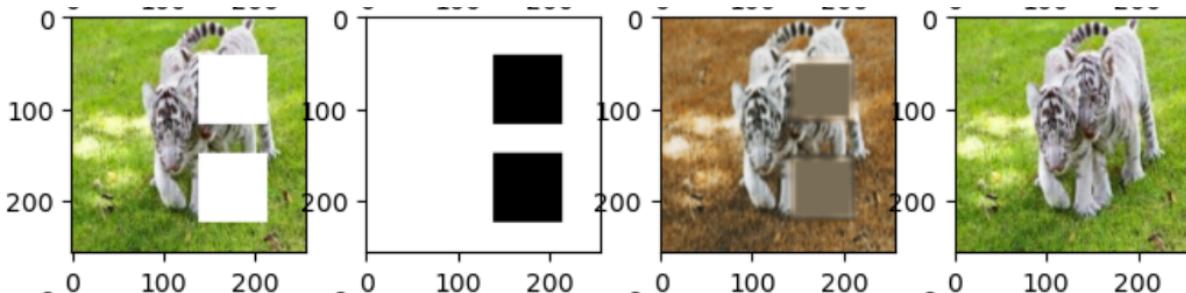
Plot of Loss Function



Dice Coefficient Value at the end of 20 epochs: 0.6344
 Loss Value at the end of 20 epochs: 0.0559

2.2.3. Model Performance / Score

The performance shown by **Model 2** is as shown:



The first image is the masked image (we changed the color of squared boxes from black to white). The second image is a mask. The third image is the predicted image given by **Model 2**, and the fourth image is the desired or original image.

As it is clearly visible that **Model 2** just created grey boxes in the predicted image. Hence the team decided not to go ahead with Model 2 and dropped the idea of generating a prediction file for test images.

2.3. Model 3: pix2pix Model

2.3.1 Model Architecture and Hyperparameters:

Network's architecture includes the following:

- A generator based on the [U-Net](#) protocol.
- A discriminator represented by a convolutional PatchGAN classifier (As in the [pix2pix](#) paper).

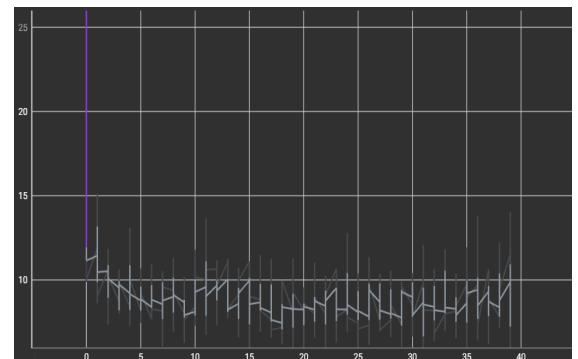
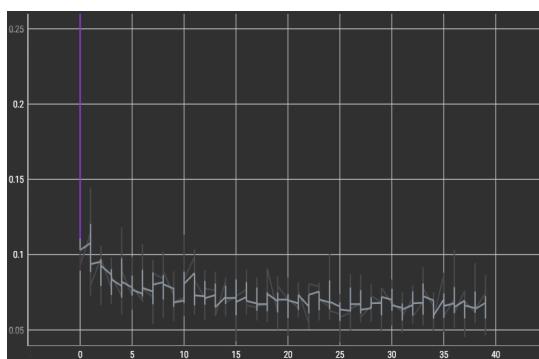
The details about hyper-parameters are as follows:

- **Batch Size:** 1
- **Number of Epochs:** 100
- **Input Image size = Output Image size:** (256, 256, 3)
- **Optimizer:** Adam
- **Learning Rate:** 2e-4

2.3.2 Results:

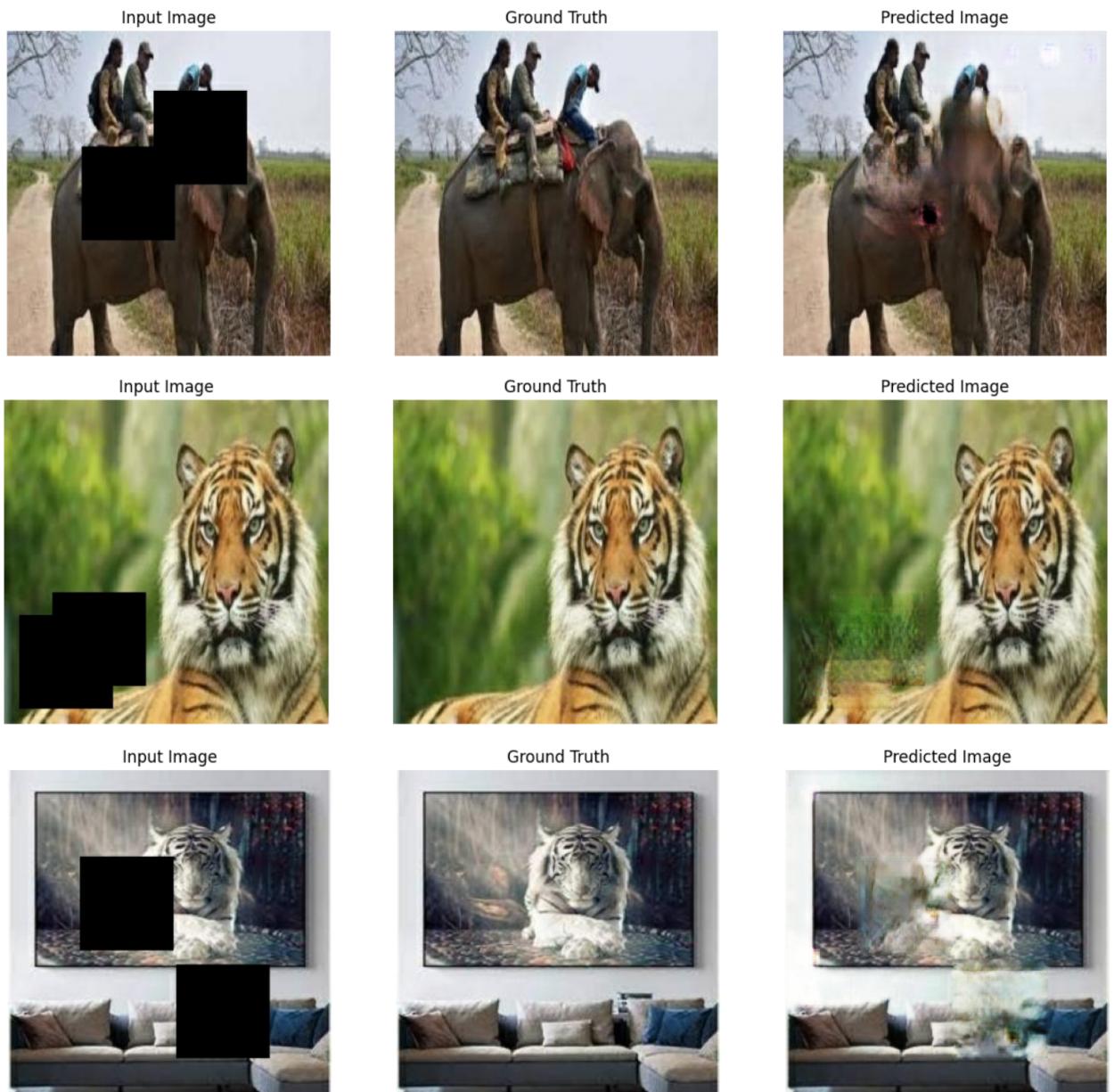
- generator L1 loss plot

generator total loss plots



2.3.3. Model Performance / Score

The performance shown by **Model 3** is as shown:



The score achieved using Model 3 is **0.24997**.

3. Details about the Final submission

The team has submitted the following files:

- This report named "**Final_Project_Report_Group12.pdf**"
- The jupyter notebook named "**photo-reconstruction-images.ipynb**". Please use this notebook for carrying out inference on 60% of the remaining test data. The cells containing the inference code are present at the bottom of the notebook with the Heading: "**Inference code: Loading model weights and testing it on test file**".