Project Report
on
Kaggle Photo Reconstruction Challenge

Submitted by: Group 29
Debtanu Datta (22MA91R04), Ganjikunta Vijay Kumar (20CS30018),
Rupshali Dasgupta (22CS72P07), Shah Dhruv Rajendrabhai (20CS10088)

April 16, 2023

1 Introduction

Photo reconstruction is a challenging task in computer vision that aims to reconstruct high-quality images from incomplete or degraded inputs. It has various applications in fields such as satellite imagery, medical imaging, and artistic rendering. In recent years, deep learning models have been widely used for photo reconstruction tasks due to their ability to learn representations from large datasets and capture high-level features.

In this report, we have presented a detailed analysis of the given photo reconstruction task. We have explored different deep neural networks such as BiLSTM, Pix2Pix, Context Encoder, and Stable Diffusion for reconstructing high-quality images from degraded inputs. The degraded inputs have missing pixels i.e. the inputs contain masks. We have performed some experiments including fine-tuning to evaluate the performance of our models, and have analyzed the results.

2 Dataset Description

The training dataset used in this project consists of images of four classes of animals, namely Cat, Dog, Elephant, and Tiger. For each class, there are two types of images - Masked and Unmasked. Every masked image in the dataset is a 256 x 256 image with precisely two masks which are squares of size 75 x 75. And for each category, there is a csv file named *masked_info.csv* that contains information about the location of the masks (holes) in each image. Each unmasked image in the dataset is a complete image of size 256 x 256. The Detailed size of the train dataset is given in Table 1.

Category	#Train-masked (80%)	#Val-masked (20%)	#Train-unmasked (80%)	#Val-umasked (20%)
Cat	1400	350	1400	350
Dog	1400	350	1400	350
Elephant	1400	350	1400	350
Tiger	1400	350	1400	350
Total	5600	1400	5600	1400

Table 1: Number of data in Train and Validation (created from given Training set) set corresponding to each animal

In the test dataset, we have only **200** masked images of the same size as 256 x 256 with exactly two masks of the same size as 75 x 75. And there is exactly one csv file that contains the information about the location of the masks (holes) in all the 200 test-masked images.

3 Data Preprocessing

We have done the following data pre-processing steps for this Image Reconstruction task:

1. **Resizing:** The input images can be resized to a smaller size to reduce computational overhead and memory requirements. In this project, we have fed the images as input to the model by resizing them as 64 x 64, 128 x 128, and 256 x 256.

2. **Normalization:** The pixel values in the given images have been normalized to make them more amenable to the training process. We have done Normalization by subtracting the mean pixel value and dividing by the standard deviation.

3. **Creating masks:** In this task, the input-masked images have holes that need to be filled in. Therefore, we have created masks that indicate the location of the holes. These masks have been created using the information provided in the masked_info csv files of the train and test set.

4. **Converting the data to tensors:** As we have used PyTorch for the implementation of several models. It requires input data to be in tensor format. Therefore, the input-masked images, unmasked images, and masks have been converted to tensor format before training and validation of the models.

4 Methodology

We have so far tried implementing recurrent neural network as *BiLSTM*, conditional GAN as *Pix2Pix*, & *Context Encoder*. Also, we have done inferencing of the pre-trained text-to-image diffusion model *Stable Diffusion* for this image inpainting task.

1. **BiLSTM:** This RNN model is basically bidirectional Long Short-Term Memory (BiLSTM) neural network. It is a famous model to learn contextual representations. Here, it takes as input a tensor with shape (sequence_length, batch_size, size_in), where sequence_length is the length of the input sequence, batch_size is the number of sequences processed in parallel, and size_in is the size of the input features. The BiLSTM layer has layers, each with size_h hidden units per direction. The layer processes the input sequence in both forward and backward directions, and the output of each direction is concatenated to form a tensor of shape (sequence_length, batch_size, size_h*2). The output of the BiLSTM layer is passed through a linear fully connected layer that maps the output to the same size as the input. The final output tensor has shape (sequence_length, batch_size, size_in). A linear layer is added instead of any non-linearity so that capturing of image tasks very well because in images pixels change slowly relative to the surroundings (context) but not rapidly.

2. **Pix2Pix:** It is a conditional generative adversarial network (cGAN) type popular for image-to-image translation tasks. We have tried this with hyperparameter tuning for the given inpainting task. Here, the generator network takes the masked image as input and generates a completed image. The discriminator network takes pairs of a completed image and a corresponding unmasked image as input and outputs a probability score indicating whether the pair is real or fake. The loss function combines a content loss and an adversarial loss. The content loss measures the difference between the completed image and the corresponding unmasked image in terms of pixel-wise mean squared error. The adversarial loss encourages the generator network to produce indistinguishable images from real images according to the discriminator network. During training, the model iterates over batches of paired images and updates the generator and discriminator networks alternately.

3. **Stable Diffusion:** It is a pre-trained text-to-image diffusion model capable of generating photo-realistic images given any text input. We have done inferencing of Stable Diffusion. The masked image and the mask generated from the masked_info file have been passed as input along with suitable prompts like ”This is an image of cat” or ”This is an image of dog”, for inference. To make Stable diffusion more efficient in our task we have to fine-tune this with the given dataset. This our one of the main objectives of the final submission.

4. **Context Encoder:** For this task, It is trained to generate the contents of an arbitrary image region conditioned on its surroundings.

5. **Average of outputs of Pix2Pix & Stable Diffusion:** We trained our model using pix2pix and got output as the normalized pixel scores for pix2pix, as well as got output from stable diffusion as normalized pixel scores. Both of these normalized pixel scores are averaged and a new CSV file was generated to test for RMSE.

4.1 Hyperparameter Tuning

- **Hyperparameter Tuning for Pix2Pix:** The hyperparameters such as Image Size, number of epochs, batch size ad learning rate, are tuned for the Pix2Pix model. The RMSE scores for different hyperparameters have been described in Table 2.

Trial No.	Input Image Size	No of Epochs	Batch Size	Learning Rate	RMSE
Trial 1	64 x 64	10	16	0.0002	0.30387
Trial 2	128 x 128	15	32	0.002	0.30482
Trial 3	256 x 256	20	16	0.0002	0.29391
Trial 4	256 x 256	25	16	0.002	0.28931
Trial 5	256 x 256	35	16	0.002	0.34751
Trial 6	256 x 256	10	4	0.001	0.26639

Table 2: Different hyperparamters and the corresponding RMSE for Pix2Pix

- **Hyperparameter Tuning for Stable Diffusion:**
  - Using the filenames in the test dataset, suitable prompts like ”This is an image of cat” or ”This is an image of dog”, are fed to the pre-trained text-to-image Stable Diffusion for the inference.
  - Using both Pix2Pix and Stable Diffusion, we fed the output of Stable Diffusion to Pix2Pix and took the normalized values of the pixels of both models.

  - A CSV file was generated by averaging the normalized score of all the pixel values for pix2pix and Stable Diffusion. The stable diffusion part of this method intakes a prompt that says something about the image to be inpainted and the pix2pix is trained using hyperparameters that were giving the best results, as aforementioned.

- **Hyperparameter Tuning for BiLSTM:** In BiLSTM, we used two fully connected layers and learning rate was 0.001. Table 3 shows the different hyperparameters applied to test out BiLSTM model.

Trial No.	No of Epochs	Batch Size	Early Stopping	RMSE
Trial 1	50	32	Yes	0.21048
Trial 2	10	1 (Default)	No	0.19913

Table 3: Different hyperparameters and the corresponding RMSE for BiLSTM

5 Performance and Analysis

- Initially, we tried the **Context Encoder** architecture, but we have not gotten any expected results so far. The performance and the learning were not sufficient enough to be included in this report.

- From our experiments, we have analyzed that so far **Pix2Pix** is giving promising results in terms of learned models. The Pix2Pix model with the input image size of 256 x 256 to the Generator and output image size of 256 x 256 from the Generator and the Batch Size of 4, Learning Rate of 0.001, and Number of Epochs of 10 is giving the best results till now in terms of RMSE score of **0.26639**.

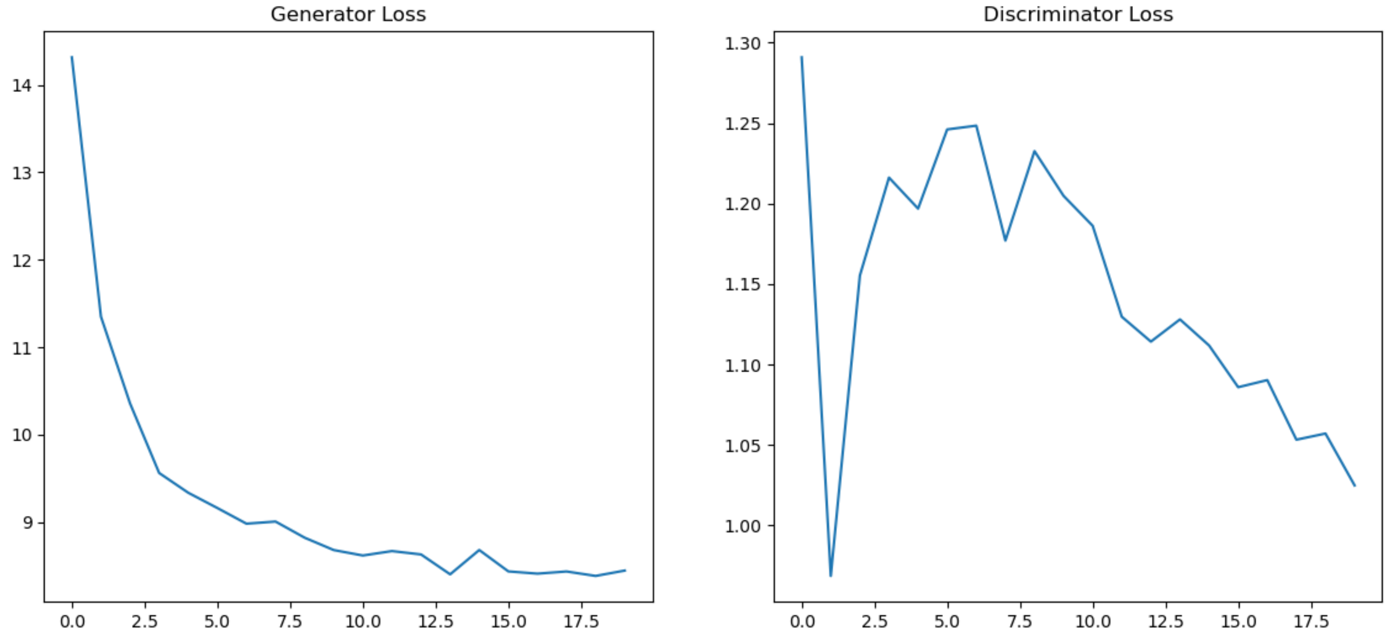

Figure 1: Graphs of generator loss and discriminator loss for Pix2Pix

- Inferencing of **Stable Diffusion** gives RMSE score as **0.26438**.

- By averaging the normalized pixel scores of Pix2Pix and the normalized pixel scored of Stable Diffusion, the RMSE score stands out to be **0.27662**.

- **BiLSTM** is also giving promising results. Using the BiLSTM of two fully connected layers with a default batch size of 1 and no early stopping with 10 epochs, we got the best RMSE of **0.19913**.

Model	Best RMSE achieved
Pix2Pix	0.26639
Stable Diffusion	0.26438
Pix2Pix and Stable Diffusion (averaging the normalized pixel values)	0.27662
BiLSTM	0.19913

Table 4: Different models and the corresponding RMSE

As is evident from Table 4, BiLSTM gives the best results out of all the techniques applied, so the inferencing was done using the BiLSTM model.

6 Conclusion

So far, **BiLSTM** is giving the best RMSE value of **0.19913**. Previously, the minimum RMSE score for Pix2Pix is **0.28931**, and by tuning the hyperparameters of our model, we have been able to achieve the lowest RMSE score for Pix2Pix equal to **0.26639**. From the inferencing of Stable Diffusion, we get the minimum RMSE score of **0.26438**, after submitting our result on the Kaggle Competition platform. BiLSTM is a context-based recurrent neural network. Hence, it can generate appropriate and better pixel values in the masked position, based on the context of the surrounding pixels. Therefore, for this particular task of photo reconstruction of masked regions in images, we can see that BiLSTM works better than the inferencing of pre-trained Stable Diffusion model, due to its property of capturing the context in both directions.

The best checkpoint corresponding to the BiLSTM model which gives the best RMSE score, can be accessed through this google drive link.