Project Report - ECE 176

Chenbo Peng
Department of Electrical and Computer Engineering
A16363723

Yue Yin
Department of Mathematics
Department of Cognitive Science
A16858644

Abstract

As artificial intelligence and machine learning are receiving more public attention and popularity, computer vision as a subcategory is also experiencing massive development. Our goal for this project is to inpainting and restore images by leveraging prediction using related context pixels. We use encoder decoder and convolutional neural network algorithm to accomplish the task. Furthermore, we analyze the specific context to compare the performance.

1 Introduction

The motivation of solving this problem is to connect deep learning knowledge to real world practices. Our project is about a context-based pixel prediction algorithm for image implanting. The key part in this project is a context encoders – a type of convolutional neural network designed to fill in given part of an image using information from the surrounding areas. The designed context encoders need to understand the content information of the given image, and then to predict a plausible solution for the missing part(s). After we complete the building of the encoder, we intend to explore some potential factors may have impact on the quality and accuracy of the predicted area. For this project, the factors we are interested in are classified training dataset and the location where the missing part(s) is. From this result we may refine the project for better prediction. Our project is capable of producing plausible prediction which meets our expectation.

2 Related Work

We reference two related works about inpainting. The Context Encoders: Feature Learning by Inpainting introduces a encoder-decoder model and the concept of joint loss with reconstruction loss and adversarial loss for image inpainting. The Image Inpainting for Irregular Holes Using Partial Convolutions introduces inpainting methods use a standard convolutional network over the corrupted image, using convolutional filter responses conditioned on both valid pixels as well as the substitute values in the masked holes. Our project builds on those two papers by using reconstruction loss and convolutional filter to implement the context encoder for 32 * 32 size images.

3 Method

Our approach is designed to reconstruct specific regions of an image by learning from the context provided by its surroundings. This method stands on the foundation laid by auto-encoders, yet it introduces significant innovations tailored to the task of context-based image reconstruction.

3.1 Detailed Structure

The method employs a convolutional neural network (CNN) that consists of two main parts: the encoder and the decoder. The encoder compresses the input image into a compact representation,

capturing the essential features required for reconstruction. The decoder then uses this representation to generate the missing parts of the image, ensuring consistency with the available context.

Encoder: Begins with a Conv2D layer of 64 filters, followed by batch normalization and max pooling to downsample the image, preserving essential features while reducing dimensionality. This pattern continues, increasing the filter count to 128 and 256 in subsequent layers, each followed by batch normalization and max pooling. The output is then flattened and passed through a dense layer with 4096 units, simulating a fully connected layer that captures global features.

Decoder: The decoder mirrors the encoder structure but in reverse, using UpSampling2D layers to gradually increase the spatial dimensions. Conv2D layers with decreasing filter sizes (256, 128) reconstruct the detailed features of the target image region. The final layer uses a Conv2D to match the output shape to the original image, employing a 'sigmoid' activation function to ensure output values are in the [0, 1] range, suitable for image data.

3.2 Training and Testing Algorithms

Training involves minimizing the mean squared error (MSE) between the reconstructed image and the ground truth, using the Adam optimizer for efficient gradient descent. Testing evaluates the model's ability to reconstruct images it has never seen before, measuring performance using the same MSE metric.

3.3 Novelty and Strengths

Compared to previous work, our method introduces a deeper architecture with an emphasis on batch normalization and a dense layer to better capture and utilize global contextual information. The strengths of choosing this method include:

- Improved Reconstruction Quality: By employing a dense layer within the encoder-decoder architecture, our model can better understand global features and relationships within the image, leading to more accurate reconstructions.
- Flexibility and Adaptability: The model can be easily adapted to different image sizes and types by adjusting the input and output shapes, making it versatile across various tasks.
- Efficient Learning from Context: The method efficiently learns to fill in missing parts by understanding the entire image's context, which is particularly useful in scenarios where the missing information cannot be directly inferred from local features alone.

4 Experiments

4.1 Dataset

CIFAR-10: This dataset is a fundamental benchmark in the field of computer vision, comprising 60,000 32x32 color images spanned across 10 classes, with 6,000 images per class. The dataset is evenly divided into a training set of 50,000 images and a test set of 10,000 images. The data format in the dataset is represented as a 32x32 pixel square, encoded in RGB format, thus having a shape of (32, 32, 3) where 3 denotes the three color channels. For our experiments, labels were not required since the focus was on reconstructing images from their context. Therefore, we normalized the pixel values to a range of [0, 1] for computational efficiency and to facilitate training convergence.

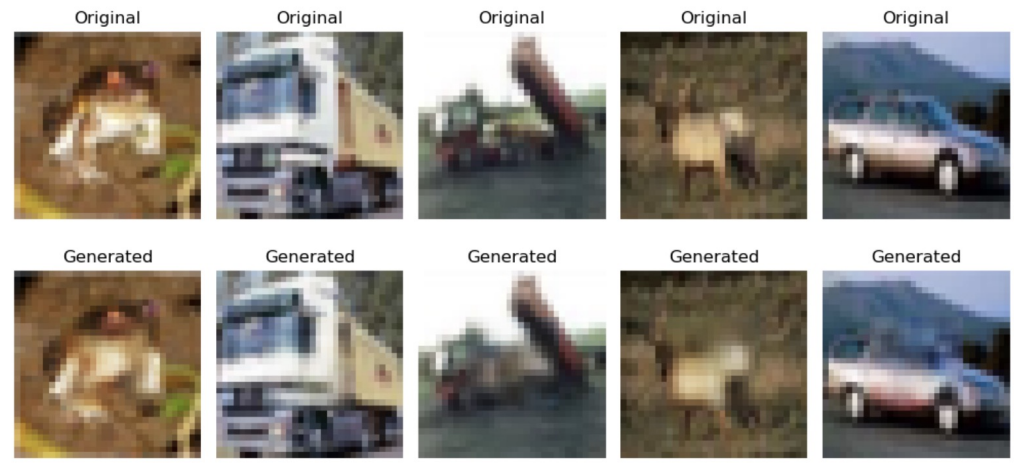
4.2 Result

Comparing Training Images with Generated Ones

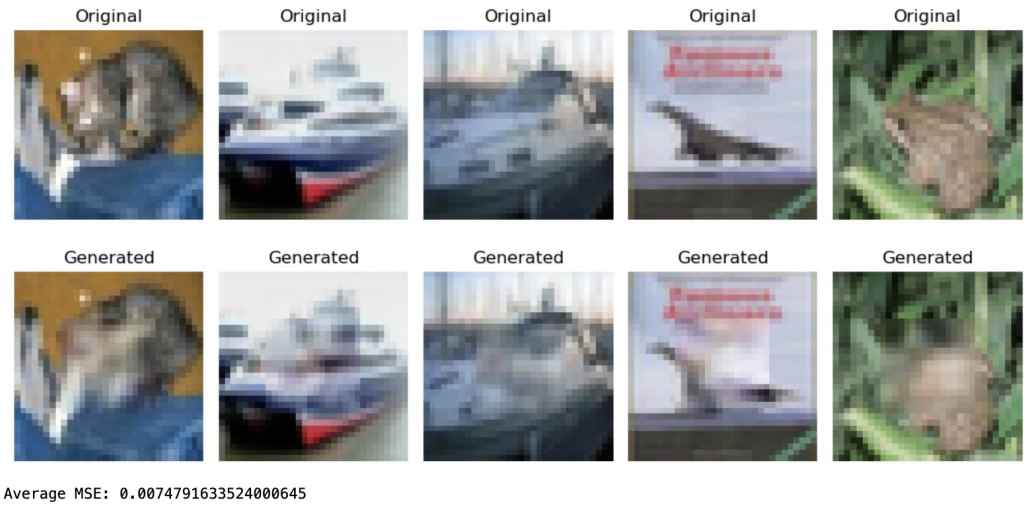
Below are the original images from the training set and their counterparts generated by our model with MSE.

Comparing Test Images with Generated Ones

Below are the original images from the testing set and their counterparts generated by our model with MSE.



(a) Training image and its generated counterpart



(a) Test image and its generated counterpart

4.3 Further Approach

Specified Dataset

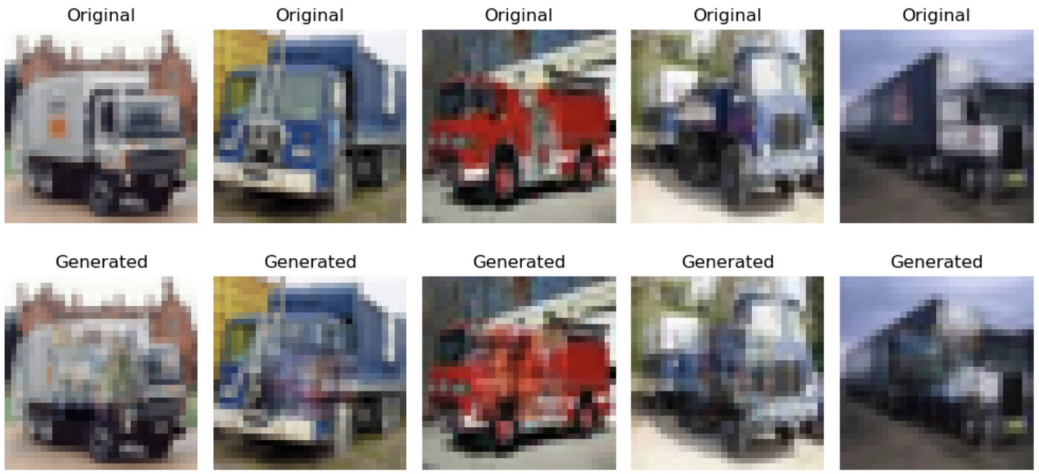
Instead of training with the complete CIFAR-10 dataset, we also experimented with training using only a single class from CIFAR-10. We focused on truck images for this test, and here is the result with Testing image.

As shown in the above figure, truck images has a higher MSE than the general model, proving that the general model has a better performance. This may due to the fact that there are much less truck image for the model to train or the center mask cover may make the model more difficult to predict. Therefore, we performed another experiment with masking different positions on the image.

Mask Location

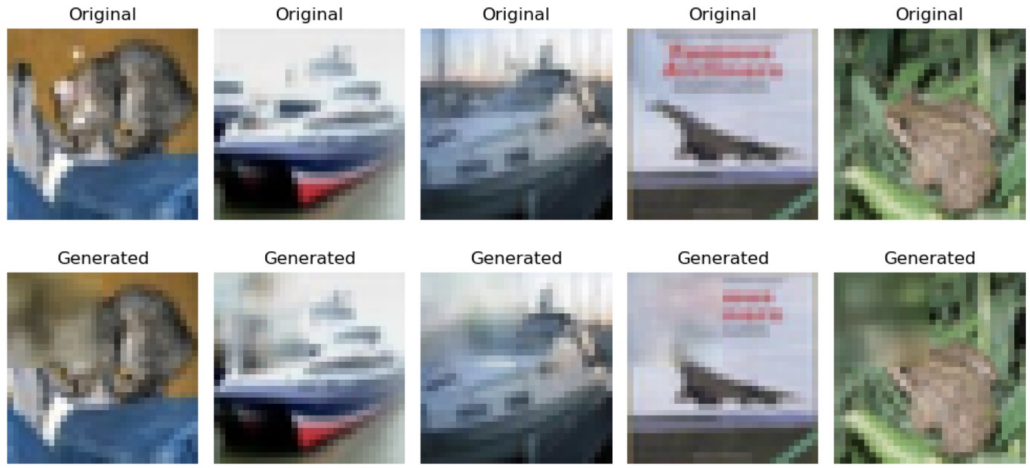
We further tested the performance with different mask location on our dataset. In this experiment, we moved the mask to the upper left corner of the images to perform the inpainting task.

The MSE is slightly lower than the test image result of general model. Visualize this by human eyes, we found that the upper left corner mask location gave a more reasonable result. This could be due



Average MSE: 0.009005878120660782

(a) Test image and its generated counterpart



Average MSE: 0.007227609399706125

(a) Upper left corner mask test image and its generated counterpart

to the higher probability of masking the most important information when adding a mask to the center of the dataset, making it hard to generate accurate results without useful references from the surrounding areas.

5 Supplementary Material

Final Presentation: https://youtu.be/UY3PGVM10yQ

References

[1] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, Antonio Torralba, *Places: A 10 million Image Database for Scene Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017. `http://places2.csail.mit.edu/`

[2] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros, *Context Encoders: Feature Learning by Inpainting*, arXiv preprint arXiv:1604.07379, November 21, 2016. `https://arxiv.org/abs/1604.07379`

[3] Alex Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*, Tech. Rep., University of Toronto, 2009. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf