

Image Completion Using DeepLearning

RaviChandra Pothamsetty *username:rapotham*

Rajesh Sharma *username:rashar*

Indiana University Bloomington

December 12, 2021

Abstract

Through this course project, we wanted to explore various deeplearning methods for Image completion. Starting from simple **AutoEncoders** to complex architectures like **DCGANS**, **Transformers** we experimented with various methods for image completion task and compare the results among them. The data used for the purpose is celebA dataset. In our case of the human datasets , the DCGANS performed well in re-construction of the missing area of the human faces.

1 Image Completion Introduction

Image completion task can be understood as completing area which is missed from the image. Image completion is very important task in the areas of computer vision and image processing and find its applications in reconstruction of the images, occlusion where we want to crop out the unnecessary objects in image and repaint with the background. This problem has been there from more than 20 years and has been solved in different ways for various purposes and each algorithm has its own limitations. Given the age of deep learning, many of the modern approaches have employing deep neural networks to solve the problem of image completion.

2 Dataset

For the purpose of the problem , we used a *celebA* dataset. *celebA* data set consists of 202599 images of 10177 celebrities. Each image is of resolution 178 x 218. While some models(like Context Auto Encoders) we trained, required us to feed the masked images for the training purposes while some models(GANs, Transformers) required us to train on the original images. For those training tasks which required masked images , we prepared another dataset using *celebA* dataset by randomly masking certain rectangular area of the image with black region. There are two reasons for selection of *celebA* dataset. First one is that there is a structure in the human faces like two eyes, one nose, two lips in that

order from top to bottom which can be easily captured by the models within few training epochs. Second is that with Covid-19, everyone is wearing masks on the face, we looked at the feasibility of our models being able to predict the complete human face when human face is partially covered by Covid-19 mask.

3 Our Approaches

We reviewed the literature work that has been done in the name of Image completion or Image inpainting and narrowed down few areas which seems feasible with our timelines. Before starting going through the discussion of methods, we would like to point out that our training data image size 218 *178 used to cause issues like fail to run due to memory issues, like taking too much time to run etc. To overcome these we resized the images to 64 X 64

1. AutoEncoders
2. DCGANS
3. Visual Transformers

3.1 AutoEncoders

Through out this course, we have learned that autoencoders have been used for the noise removal purpose. AutoEncoders project the data point into lower dimension space and try to reconstruct the same data point from the lower dimension space. AutoEncoders are best used for noise removal and in some way the missing pixel reconstruction can be thought of noise removal. We train the model by taking the masked images as input and original images as output. We experimented with the two types of AutoEncoders.

1. AutoEncoders(with linear layers)
2. CNN Auto Encoders

3.1.1 AutoEncoders

We flattened the image of (64, 64 ,3) size into 12088 dimensions and feed into encoder and decoder to reconstruct the same image. In all these we experimented with various encoder output dimension(latent vector dimension) and observed the reconstruction loss error.

Encoder layers: (12088, 4096), (4096, 2048) ,(2048, 1024) (1024, z) where z is the dimension of the latent dimension

Decoder layers: (z, 1024), (1024, 2048), (2048, 4096), (4096, 12088)

Experiment Details: Data Size: 20000 ,Training Percentage : 80 percent ,Testing Percentage : 20 percent, MSE loss with Adam Optimizer default parameters,all linear layers with leaky relu activation function

Table 1: AutoEncoder Model Results		
Dimension	TrainingMSE	TestingMSE
256	0.0147	0.0187
512	0.0148	0.0185
1024	0.0151	0.0189
2048	0.0141	0.0183
4096	0.0126	0.0164

3.1.2 CNN AutoEncoders

Since we are dealing with images, we can make use of the Convolution Neural Network version of AutoEncoders where each layer is CNN or Transpose CNN. We used 6 layers of convolution layers with Leaky relu activation function and batch normalization on the encoder side and 6 layers on the decoder side with relu activation function. Batch normalization is done expect for the last layer. Tanh activation is used for the last layer. Like in the vanilla autoencoders here we experimented with latent dimension by observing the outcome images, MSE on training and testing data.

Enocder layers: Conv2d(3,64,4), Conv2d(64,64,4), Conv2d(64,128,4), Conv2d(128,256,4), Conv2d(256,512,4), Conv2d(512,512,4), Conv2d(8,64,4,2,1)

Decoder layers: ConvTranspose2d(512,512, 4), ConvTranspose2d(512, 256, 4), ConvTranspose2d(256, 128, 4), ConvTranspose2d(128,64 ,4), ConvTranspose2d(64,64,4), ConvTranspose2d(64,3,4)

Experiment Details: Data Size: 20000 ,Training Percentage : 80 percent ,Testing Percentage : 20 percent, MSE loss with Adam Optimizer default parameters, CNN layers with batch normalization with leaky relu activation function

Obserevations:We picked the best performing latent dimension in both cases

Table 2: CNN AutoEncoder Model Results		
Dimension	TrainingMSE	TestingMSE
256	0.006	0.0196
512	0.0064	0.021
1024	0.0069	0.022
2048	0.0059	0.0213
4096	0.0056	0.02354

for the prediction on the data. Results of predictions of both types of encoders are posted in the page 6 of this file. Normal auto encoders performance prediction is quite poor, they are predicting completely irrelevant images to the input images. These will not serve the purpose. CNN autoencoders are performing better, able to complete the picture but the entire is blurred or not with clarity. CNN auto encoders are performing well in completing the hair or forehead or something that aligns with rest of neighbouring pixels but cannot do well in predicting the

missing regions like nose, mouth, eyes etc. A Example is depicted in the page 6 where mouth region is masked , the CNN autoencoder model failed to give atleast the shape.

3.2 Deep Convolutional Generative Adversial Networks

Generative Adversial Networks(GAN) are extensively used in learning underlying data distribution and to generate points from the data distribution. Deep Convolutional Generative Adversial Networks(DCGANs) are extension to the GANs where the random vector generates image using the layers of series of convolutional, batch norm, relu layers. There are different ways to solve the image inpainting method using DCGAN. The method we employed is first to train GAN model such that the model's generator has the ability to generate good images which looks like from the dataset. Once that is done, now passing a random vector z of 100 dimension with each number value in the interval of $[-1, 1]$ will generate a image Now we can pose the problem statement as follows:

1. Divide the image which needs to be reconstructed into regions a masked regions(pixel values are missed) and unmasked region
2. If we can estimate the z which is \hat{z} such that $G(\hat{z})$ is as close as to unmasked region of the original , then the same area enclosing the masking region in $G(\hat{z})$ a good estimate of the masked region of original image
3. The produced image $G(\hat{z})$ is as close as to real one or atleast the discriminator should able to percieve as real one.
4. The above two notions can be framed in single loss function by constructing this equation as follows

$$L_{totalloss} = L_{contextualloss} + \lambda * L_{perceptualloss}$$

$$L_{contextualloss} = \|M \odot G(z) - M \odot x\|$$

$$L_{perceptualloss} = \log(1 - D(G(z)))$$

5. In the above equations M is a binary tensor whose elements values zero if the pixel is in the masked region or else one
6. Need to find \hat{z} which minimizes the $L_{totalloss}$. This can be done in iterative process via backpropagation. In the above equation λ acts as regularizer which is how much importance the perceptual loss should be given.
7. For our experiments λ value is taken as 0.01
8. insert generator picture, insert change in the picture with epochs, loss as a function of epochs plot

Experiments We split the data into training and testing data. Training data is train the GAN and test data used for the reconstruction purpose. Different architectures of DCGANS are experimented until the generator produce good representation of images. Once this is done, the generator can be used for image completion through iterative process using 5000 iterations. We observed that 5000 is good amount of iterations and beyond 5000 iterations the loss is not decreasing and is almost same.

We have picked some of the images from the test data and masked the regions randomly. Some of the results are pasted in page 6 of this file. In each row the first column is the original image, the second column is the image where the some portion is masked, the third column is the reconstructed image, the fourth column is the image that is generated by the generator.

For the first image, even the major portion of the image is removed, the model able to reconstruct the image which looks similar to the human face even though it is not same as original face. For the second row image, the model could able to generate the cropped upper lip, nose and eyes. Because of the generated image colour is not in tune with the original image, the boundary is quite visible in the reconstructed image(third column).

In the third row , the cheeks and some part of the teeth are generated well and in fourth row missing eye, eyebrow are generated well. In all the examples except the first one the boundary between the masked and unmasked area is quite clearly visible.

3.3 Visual Transformers

Transformers are widely used in NLP and majority of the tasks like classification, tokenization, sentiment analysis, auto completion can be solved by Transformers. Success in NLP making the transformers making in roads into the Computer Vision Area.

We have tried to used the model called Masked Auto Encoders, which has Encoder(Transformer) and Decoder. The model is fed with image patches of the 32 by 32 size which are the tokens and the model randomly masks the few patches and finally learns the latent representation which is fed to the decoder along with masked representation where the decoder tries to reconstruct the image. The Mean Square Error is used as the loss. Using the celebA dataset we trained the model.

During inference, we divided the image into patches and if the missing region falls into the masked patch, it is considered masked patch. Except for the masked patches , the rest of the patches are sent to the Encoder , the output of the encoder is appended with the masked tokens and fed to the decoder which outputs the full reconstructed image.

We tried this approach and tried with 10 epochs , batch size 4 and Adam optimizer with default learning rate. This took more than 20 hours to learn the model. But this did not yielded any good results. The missing region we used to get random pixel values.

4 Conclusion

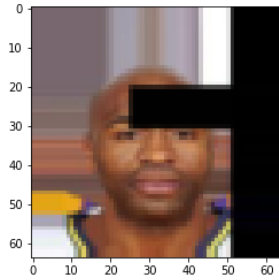
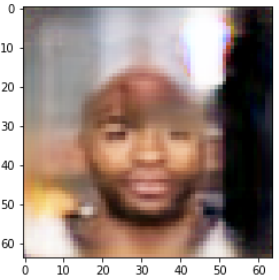
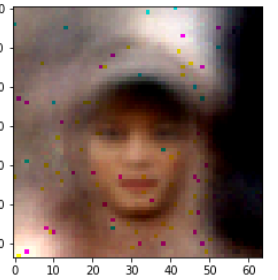
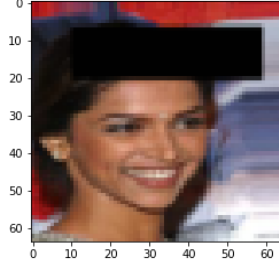
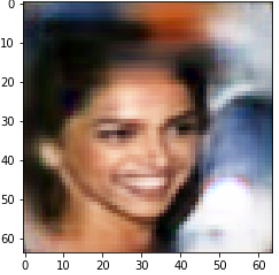
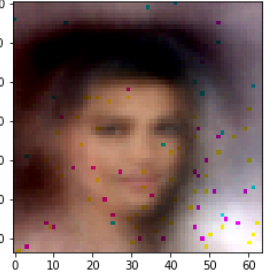
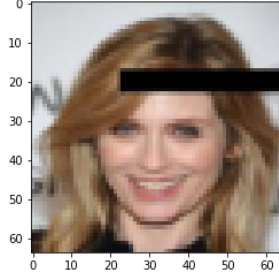
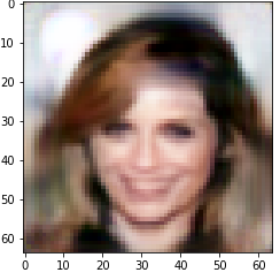
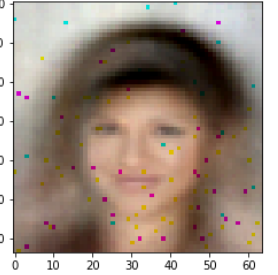
We started with simple models to complex models reviewed lot of research material in the process. Simple autoencoders did not work as these are images and simply flattening the images would miss the spatial structure. The CNN autoencoders worked very well in cases where the missing region is almost similar to the neighbouring region like partial missed forehead cheeks or hair. Coming to the image completion with DCGANS, they seem to be working out better than rest of the ideas for this task as long as we can stabilize GAN in producing good quality images with respect to the data distribution. One more reason is it is intuitive to find the best random vector which produces the nearest image to the given image. For the Visual Transformers overfitting might causing the issue to achieve decent results.

Worth mentioning improvements is to train the models for 218 X 178 images instead of 64 X 64 size images as images taken by camera would be of higher resolution. In the approach of DCGANS, we can look to come up with a processing step after reconstruction so that the boundary between the original region and predicted region is not visible and the boundary is smooth. Regarding the transformers, we feel that it is better to fine tune the pretrained models (which were trained on large datasets like ImageNet)

References

References follow the acknowledgments. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. **Note that the Reference section does not count towards the eight pages of content that are allowed.**

- [1] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.
- [2] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GENeral NEural SIMulation System*. New York: TELOS/Springer-Verlag.
- [3] Hasselmo, M.E., Schnell, E. & Barkai, E. (1995) Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region CA3. *Journal of Neuroscience* **15**(7):5249-5262.

Original Image	CNN AE output	AE output
		
		
		

Best CNN AE architecture is used to predict masked (mouth portion) images		
Original Image	Masked Image	CNN AE predicted image
