# Image Completion using Feature Engineering in Convolution Neural Networks

**Rashmi Shettigar[1], Nikhil Chandran[2], Visal Rajeev[3], Mayuresh Pitale[4] and Prof. Aju Paleri[5]**

[1,2,3,4,5] *Department of Computer Engineering, Pillai College of Engineering, Navi Mumbai, India – 410206*

---

**Abstract -** *Image completion is one of the most challenging problems, as the reconstruction algorithms should render the missing pixels. Recent studies have achieved promising advances in photorealistic human face synthesis and generation. However, these approaches are limited to deal with general or structure specified images which involve large datasets. When there is a lack of training data, it becomes difficult to handle the algorithm, for which Low-Shot learning is introduced. In this project, we aim to combine the perceptual way with the Low shot learning using Feature Engineering to increase the size of the training dataset in ways such that, it improves the accuracy and robustness of Image Completion that overcomes the missing values. Based on extensive experimental study, we will conduct the analysis on how each dataset affects the identification accuracy. Based on the recent studies on this topic it suggests that the algorithm for completing images is effective in improving the identification accuracy and precision.*

***Key Words***: Photorealistic Image Synthesis, Low-shot Learning, Feature Engineering, Image Completion, Image Inpainting

## 1. INTRODUCTION

An Image Completion System inputs an incomplete jpeg image or a masked image and uses machine learning algorithms to complete it. It uses previously trained images to determine how newly input images will be completed. It replaces black pixels with image patterns in accordance with the background image. Also, it allows removing unwanted objects and generates occluded regions.

## 2. LITERATURE SURVEY

***A. Face Generation for Low-shot Learning using Generative Adversarial Network:*** Authors - Junsuk Choe, Song Park, Kyungmin Kim Joo Hyun Park, Dongseob Kim and Hyunjung Shimks tried techniques to improve the increase in the size of training dataset in various ways to improve the accuracy and robustness of face recognition. This was proposed to handle the lack of training data in machine learning. The paper presents studies and experiments to conclude that the proposed algorithm for generating faces is effective in improving the identification accuracy and coverage using both the base and novel set. [1]

***B. Deep Portrait Image Completion and Extrapolation***: It was developed by Xian Wu, Rui-Long Li, Fang-Lue Zhang, Jian-Cheng Liu, Jue Wang, Ariel Shamir and Shi-Min Hu where deep portrait image completion technique was used. A general image completion and extrapolation method fails as the task requires more accurate structure details and appearance synthesis. Whereas the deep portrait technique performs a two stage deep learning framework to tackle the limitations faced by general image completion methods. Hence it provides a better and more efficient structural map recovery and a faster training and generation cycle. [2]

***C. Image Completion with Deep Learning in Tensorflow:*** In this paper authors Richard Davies and Brandon Amos use the GANs algorithm for the process of image completion. In this experimental study the images are interpreted as being samples from a probability distribution. This interpretation of images helps learn how to generate fake images. This understanding leads to finding the best fake images that are most similar to the nearby surrounding regions of the missing pixels. For this the image generator and discriminator has to be trained first, so as to generate an approximate image similar to the real one and which also fools the discriminator to be able to generate a real like and consistent image. [3]

***D. High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis:*** This technique is used by Yang, Chao and Lu, Xin and Lin, Zhe and Shechtman, Eli and Wang, Oliver and Li, Hao. Given an image, we use the content and texture network to jointly infer the missing region. A multi-scale neural patch synthesis approach is proposed based on joint optimization of image content and texture constraints which produce for high-frequency details by matching and adapting patches with the most similar mid-layer feature correlations of a deep classification network. This approach produces for a sharper, coherent and more accurate image inpainting. [4]

***E. Globally and Locally Consistent Image Completion using celebA datasets:*** The technique of fully convolutional neural network was used by Satoshi Iizuka, Edgar Simo ,Hiroshi Ishikawa and Waseda to complete images of arbitrary resolutions by filling-in missing regions of any shape. Local and global discriminators were used to assess the realness and the completeness of the image while the image completion network is trained to fool both the discriminators. Thus this approach can distinguish the images and generate fragments that allow to naturally complete the images of objects with familiar and highly specific structures. [5]

### 2.1 Literature Survey Summary

The overview of different works is given in Table 1.

Table 1. Overview of literature survey

| Literature | Advantage | Disadvantage |
|---|---|---|
| Junsuk Choe, Song Park, Kyungmin Kim-(2017) [1] | Realistic visual and impressive content result generated | Huge computation and tough to train |
| Xian Wu, Rui-Long Li, Fang-Lue Zhang, Jian-Cheng Liu, Jue Wang, Ariel Shamir(2018) [2] | Once trained, predictions are pretty fast, can be trained with any number of layers | Slow training time if low quality GPUs are used, high computational cost |
| Richard Davies, Brandon Amos (2017) [3] | GANs can learn messy and complicated distributions of data. | "Realistic" but not real. Fake patterns can be created for small scale structures or non-nature objects like text. |
| Satoshi Iizuka, Edgar Simo, Hiroshi Ishikawa(2017) [5] | TensorFlow helps to retrieve discrete data onto an edge and therefore offers a great debugging method. | TensorFlow does not offer symbolic loops feature, but there is a workaround using finite unfolding. |

## 3. PROPOSED WORK

Inpainting is the process of restoring the lost or disintegrated portions of images and videos. In the museum world, in case of a treasured painting, this task would be carried out by a skilled art conservator or art restorer. In the digital world, inpainting refers to the appliance of sophisticated algorithms to replace lost or corrupted parts of the image data. A discriminator network, such as the one in a conventional GAN, can prove useful at such points. Its main use in such a scenario is to ensure that the final image obtained after filling in the gaps doesn't look obviously fake. When compared with the original image, it needs to look reasonably similar, containing minimal differences.

### 3.1 System Architecture

The system architecture is given in Figure 1. Each block is described in this Section.

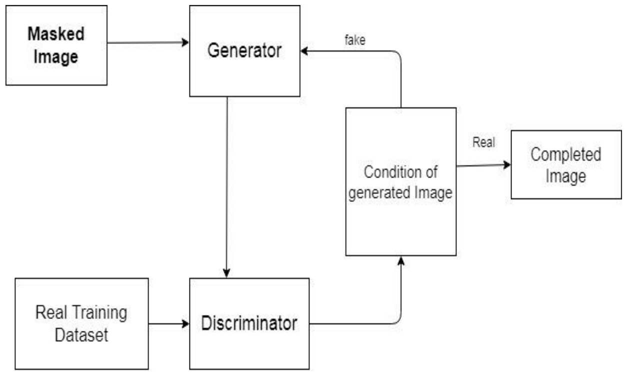


Fig. 1 Proposed system architecture

***A. Masked Image:*** Masking of the image is the first part of the architecture. Image masking is the process of removing a specific feature or background from the image. Masking involves setting some of the pixel values in an image to zero, or other "background" values. It can be done manually or using a vector mask using Photoshop and other tools.

***B. Generator:*** From the name itself, we can understand that it's a generative algorithm. Generator is a type of inverse Convolutional Neural Net, which means it works exactly opposite to what CNN does. In CNN, an real image is provided as input and a corresponding label is obtained as an output. But in Generator, a random noise (vector having some values) is supplied as input to the Inverse CNN and a generated image is expected as an output. In simple terms, it generates new data from an existing piece of data on its own.

***C. Discriminator:*** Discriminator is a Convolutional Neural Network consisting of many hidden layers and a single output layer. The big difference here is the output layer of GANs can have only two outputs, unlike CNNs, which can have outputs equal to the abels it is trained on. The output of the discriminator is binary because of a specifically chosen activation function for this task, if the output is 1 then the given data is real and if the output is 0 then it suggests that the data is fake. Discriminators are trained on the real data so that they learn to recognize what real data looks like and based on what features should the data be classified as real.

***D. Condition of Generated Images:*** The Generator starts to generate data from a random input and then that generated data is supplied to Discriminator as input. Now, Discriminator analyzes the data and checks its proximity to realness. If the data generated lacks enough features to be classified as real by the Discriminator, then the data and the weights associated with it are sent back to the Generator using back-propagation, so that the weights associated with the data can be readjusted and new data, significantly superior to the previous data, can be generated. This newly generated data is again passed to the Discriminator and it continues. This process keeps imitating for as long as the Discriminator keeps classifying the generated data as fake, for every time the data is classified as fake and with every back-propagation the quality of data keeps improving and there comes a time when the Generator becomes so accurate that it becomes hard to distinguish between the real data and the data generated by the Generator.

***E. Image Dataset:*** The dataset that has been used in the architecture is the CelebA and Places2 dataset. CelebA is great for training and testing models for face detection, particularly good for identifying facial attributes such as locating people having brown hair, grinning, or wearing spectacles. The Places2 dataset contains millions of unique images of various sceneries and backdrops, and is particularly good for identifying the attributes like sky color, elevation and various landscapes.

***F. Completed Image:*** The Discriminator keeps classifying the generated data as fakes, for every time data is classified as fake and with every back-propagation the quality of data keeps improving and there comes a time when the Generator becomes so accurate that it becomes hard to distinguish between the real data and the data generated by the Generator. A completed image is obtained when the discriminator is not able to identify whether the input is real or fake.

## 4. REQUIREMENT ANALYSIS

### 4.1 Software

The experiment setup is carried out on a computer system which has the software specifications

| | |
|---|---|
| Operating System | Windows 7/ Ubuntu |
| Programming Language | Python 3.6 |
| Softwares | Keras, Tensorflow, OpenCV |

### 4.2 Hardware

The experiment setup is executed on a computer system which has the hardware specifications

| S. No. | Hardware | Description |
|---|---|---|
| 1. | Processor | Intel Core i5 or above |
| 2. | Storage | 180 GB(SSD Recommended) |
| 3. | RAM | Minimum 8GB RAM |
| 4. | Graphics | NVIDIA GeForce 940mx and above |

### 4.3 Libraries Used

**Tensorflow** is a math library, which is usually used for machine learning applications like neural networks. We will be using this for the development of the neural network.

**Keras** is a neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

**OpenCV (Open Source Computer Vision Library)** is a computer vision and ML software library. OpenCV was built to provide a common infrastructure for computer vision applications and to complement the usage of machine perception in industrial products.

### 4.4 Dataset and Parameters

**CelebFaces Attributes Dataset (CelebA)** is a massive face features dataset with more than 2 lakh celebrity images, each having around 40 attribute annotations. Images in this dataset cover large poses as well as background clutter. CelebA consists of large diversities, quantities, and rich annotations, including 10k+ number of identities, 2lakh+ number of face images, and 5 landmark locations, 40 binary attributes annotations for each image. The dataset can be employed as the training and testing dataset for computer vision tasks like face feature recognition, face detection, landmark localization, and face manipulation & synthesis.

**Places Dataset**, developed by Bolei Zhou, Agata Lapedriza, Aditya Khosla, Antonio Torralba, Aude Oliva, MIT contains 10million+ images having 400+ unique scene categories. The dataset features 5000 to 30,000 training images per class which is consistent with real-world frequencies of occurrence. Using convolutional neural networks, Places dataset allows learning of deep scene features for various scene recognition tasks, with the goal to establish new state-of-the-art performances on scene-centric benchmarks.

## 5. IMPLEMENTATION

### 5.1 Algorithm used

### Fast Marching Algorithm

The fast marching method is a numerical approach created by James Sethian for solving boundary value problems of the Eikonal equation: Typically, such an issue describes the evolution of a closed surface as a function of your time with speed within the normal direction at some extent on the propagating surface.

The fast marching algorithm slowly moves inward and takes the weighted sum of the neighboring pixels to form the colour pixel that will replace the mark region. When we mark the region on the image a separate layer consisting of the mark vision will be created. This layer mask will be imported along with the base image without the pixels marked on the layer mask. Now, when the image inpainting begins the fast marching algorithm will first record the pixel mark on the mask layer.

For the outermost pixel the algorithm takes the average of the weighted sum of the neighborhood pixels to create a colour approximate for the pixel at the outermost part. This, along with the previously trained data is now used to detect the colour of the pixel which will replace the marked pixel. Thus, the outermost pixel present on the marked region is now replaced with coloured pixel which is a close approximation to the boundary of the image.

### 5.2 Sample Implementation

Places dataset model was first trained using the described algorithm. For testing, sample images of nature from the internet were used and the following results were obtained.

Input Image Overlying mask Inpainted Image

Input Image Overlying mask Inpainted Image

Input Image Overlying mask Inpainted Image

## Accuracy:

Accuracy in proposed model - 70%

The accuracy of the model is solely dependent. on the size of the dataset used to train it. Here, as the data set used is small the accuracy remains constricted to 70%. However, if we use a larger dataset to train the model, we can achieve a higher percentage of accuracy.

## Conclusion:

Image generation techniques are attractive in various computer vision applications, especially because of the difficulties of labeled data collection for training. Among those, we attempted to generate face images and place images with several attributes and poses using GAN, enlarging the novel set to achieve increased performance. With the increased dataset, we verified increased performance on the missing portions of the image. Moreover, we demonstrate that duplicating original data effectively regularized excessive influences from augmented data.

## REFERENCES

1. Junsuk Choe, Song Park, Kyungmin Kim, Face Generation for Low-shot Learning using Generative Adversarial Networks, Yonsei University (2017)

2. Xian Wu, Rui-Long Li, Fang-Lue Zhang, Jian-Cheng Liu, Jue Wang, Ariel Shamir, Members, Deep Portrait Image Completion and Extrapolation, IEEE Computer Society (2018

3. Richard Davies, Brandon Amos, Image Completion with Deep Learning in TensorFlow (2017)
4. Chao Yang, Xin Lu, ZheLin, Eli Shechtman, Oliver Wang , Hao Li, High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis (2017)

5. Satoshi Iizuka, Edgar Simo, Hiroshi Ishikawa, Globally and Locally Consistent Image Completion using celebA datasets, Waseda University (2017)