# GROUP No: 12

## Project Title: Image Inpainting and Completion

**Members:**

**Ayush Shah, Gurkirat Sarabjeet Singh Nagpal, Atharva Marathe and Sarvesh Yenarkar**

**Contents:**

# 1) Project Description

Image completion, also known as image in painting, is an active computer vision research problem that aims to automatically fill in a missing portion of an image in a content-aware way. Various approaches have been engineered by academia dedicated to this problem, modern ones even applying deep convolutional neural networks to have a smoother and more realistic output images.

By content-aware, it means that an algorithm should consider the neighbor pixel information of the missing portion of the image it is completing when it produces the final completed output. State of the art such algorithms are often realized with convolutional neural networks, and in this paper, we aim to reproduce one of the novel research results of such neural network structures.

Image inpainting is the process of reconstructing missing parts of an image so that observers are unable to tell that these regions have undergone restoration. This technique is often used to remove unwanted objects from an image or to restore damaged portions of old photos.

Content-aware fill is a powerful tool designers and photographers use to fill in unwanted or missing parts of images. Image completion and in painting are closely related technologies used to fill in missing or corrupted parts of images. There are many ways to do content-aware fill, image completion, and in painting.

Existing methods which extract information from only a single image generally produce unsatisfactory results due to the lack of high level context. In this paper, we propose a novel method for semantic image inpainting, which generates the missing content by conditioning on the available data. Given a trained generative model, we search for the closest encoding of the corrupted image in the latent image manifold using our context and prior losses. This encoding is then passed through the generative model to infer the missing content. In our method, inference is possible irrespective of how the missing content is structured, while the learning based method requires specific information about the holes in the training phase. Experiments on three datasets show that our method successfully predicts information in large missing regions and achieves pixel-level photorealism, significantly outperforming the state-of-the-art methods.

## HOW ???

We humans rely on the knowledge base(understanding of the world) that we have acquired over time. Current deep learning approaches are far from harnessing a knowledge base in any sense. But we sure can capture spatial context in an image using deep learning. A convolutional neural networks or CNN is a specialized neural network for processing data that has known grid like topology – for example an image can be thought of as 2D grid of pixels. It will be a learning based approach where we will train a deep CNN based architecture to predict missing pixels.

## How would you fill in the missing information?

There are two types of information:

1. **Contextual information:** You can infer what missing pixels are based on information provided by surrounding pixels.
2. **Perceptual information:** You interpret the filled in portions as being "normal," like from what you've seen in real life or from other pictures.

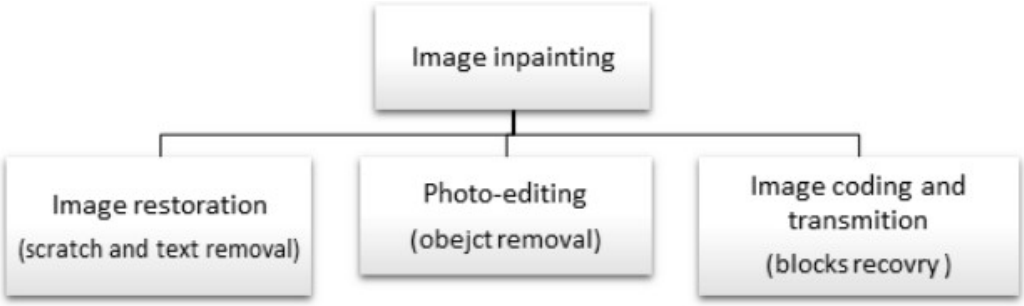
Both of these are important. Without contextual information, how do you know what to fill in? Without perceptual information, there are many valid completions for a context. Something that looks "normal" to a machine learning system might not look normal to humans.

It would be nice to have an exact, intuitive algorithm that captures both of these properties that says step-by-step how to complete an image. Creating such an algorithm may be possible for specific cases, but in general, nobody knows how. Today's best approaches use statistics and machine learning to learn an *approximate* technique.

## Abstract:

Most image completion methods produce only one result for each masked input, although there may be many reasonable possibilities. In this paper, we present an approach for pluralistic image completion the task of generating multiple diverse and plausible solutions for image completion. A major challenge faced by learning-based approaches is that here the conditional label itself is a partial image, and there is usually only one ground truth training instance per label. As such, sampling from conditional VAEs still leads to minimal diversity. To overcome this, we propose a novel and probabilistically principled framework with two parallel paths. One is a reconstructive path that extends the VAE through a latent space that covers all partial images with different mask sizes, and imposes priors that adapt to the number of pixels. The other is a generative path for which the conditional prior is coupled to distributions obtained in the reconstructive path. Both are supported by GANs. We also introduce a new short+long term attention layer that exploits distant relations among decoder and encoder features, improving appearance consistency. When tested on datasets with buildings (Paris), faces (CelebAHQ), and natural images (ImageNet), our method not only generated higher-quality completion results, but also with multiple and diverse plausible outputs.

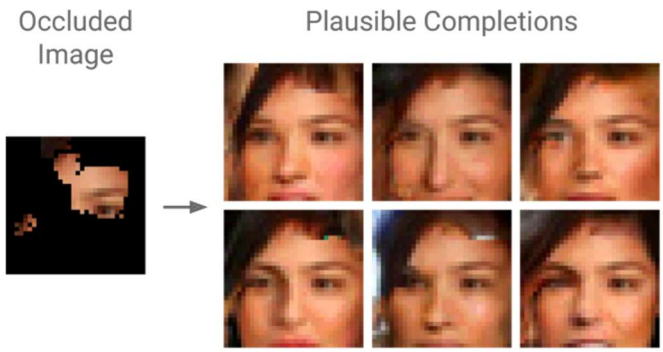
## 2) Applications




### 1. Crime Investigation

The purpose of using video surveillance images to carry out the investigation is that investigators use the video monitoring data collected by themselves to get clues about the cases, then found and confirmed the identity of the suspect. The core task of image detection is to analyse video image to access to the case clues and evidence which is help to solve cases to public security organs, so as to achieve the purpose of image detection. At present, in the process of video analysis, investigators often encounter the problem that the key video footage is fuzzy, but the picture or a frame related to the features of the criminal suspect or the vehicle license plate number, so image processing technology plays an important role in solving the fuzzy video information.The common fuzzy image in monitoring system can be divided into the following several types: (1) the low contrast image caused by underexposure or overexposure; (2) the degraded image which is shoot in bad weather such as greasy weather; (3) noise blurred image; (4) motion blurred image caused by a fast-moving target; (5) the defocus blurred image caused by the lens out of focus; (6) low resolution image.

Image inpainting gives a reconstructed image through textual descriptions or AI technique, and provides a rough image of the suspect. When the image is run through the criminal database complete details may be found.



## 2. Cosmetology – Dermatology

Cosmetology requires image processing for all its operations to detect the after effects of procedure performed on the patient.

One of the main problems of image processing in dermatology is the presence of hair which must be removed before image analysis. All existing hair removal algorithms involve two main stages:
Detection of pixels covered by hair and restoration of pixels in hair region with minimum distortion.
Hence, it uses inpainting by nonlinear-PDE based diffusion algorithms and example based methods.

## 3. Damaged Pictures Restored

Old pictures that are scraped, torn, or in any form of damaged condition can be restored using image inpainting.

Several historical artefacts, portraits and paintings are recovered through this mechanism.

# 3) Project Requirements

The following requirements are derived from an objectives tree by taking in consideration the mission goals and expectations. The requirements will be categorized as mandatory and desirable requirements. These categorires are further divided and the requirements are classified as performance requirements which are functional requirements with an associated performancve measure and non-functional requirements.

## 3.1 Mandatory Performance Requirements

M.P.1 An attractive and easy to use U.I so that all the features are accessible and easily visible to all the users.

M.P.2 A varied Data Sets of people with different height, race, and colour so that a huge base is covered during training of the network.

M.P.3 Add different types of images and have a huge database cause neural network work best when given huge data to process.

M.P.4 implementing a low Loss as well as Cost implementation of CVAE-GAN for Image Generation.

M.P.5 Achieve a score greater than 0.3 when implementing over the Huge and Varied database (Score is so low cause varied data such as landscapes are present which do not match if test element is human)

M.P.6 keep an option for the original image to appear on the screen so that it can be compared with as it is in the frame.

M.P.7 Remove specific things from an image like people in the background

M.P.8 Different parts of the face so that it will help in the cosmetology department

M.P.9 Take into consideration the case where people will want to save changes and update the database

## 3,2 Mandatory Non-Performance Requirements

The Network shall:

M.N.1 Operate using a framework that meets specifications of testcase such that it should be easy to develop and deploy the code on and should be compatible with our large training and testing set

M.N.2Operate within a windows operating system environment

M.N.3 Maintain a check on the complie time of the train set and test set

M.N.4 Be compatible with for different back prop methods to see which works best dynamically

In addition to the mandatory performance and non functional requirements, we have also identified certain desirable requirements. These additions are nice to have and extend the project scope in exchange for having a more robust, reliable and valuablle system. The desirable requirements are formulated to extract the greatest amount of information even when operating under different conditions.

## 3.3 Desirable Performance requirements

The Network shall:

D.P.1 Virtual background and diff effects to improve the GUI add karenge

D.P.2 People can add their own image to check change kar ke dekh sakte apan kaisa lagenge

D.P.3 People ke sath locations and sab bhi add karenge

D.P.4 Different database for criminals so they can be faster to search

D.P.5 Inform the last police station where he was caught

D.P.6 Try functions like Nose – if nose likha then nose delete ho jaana chaiye aisa

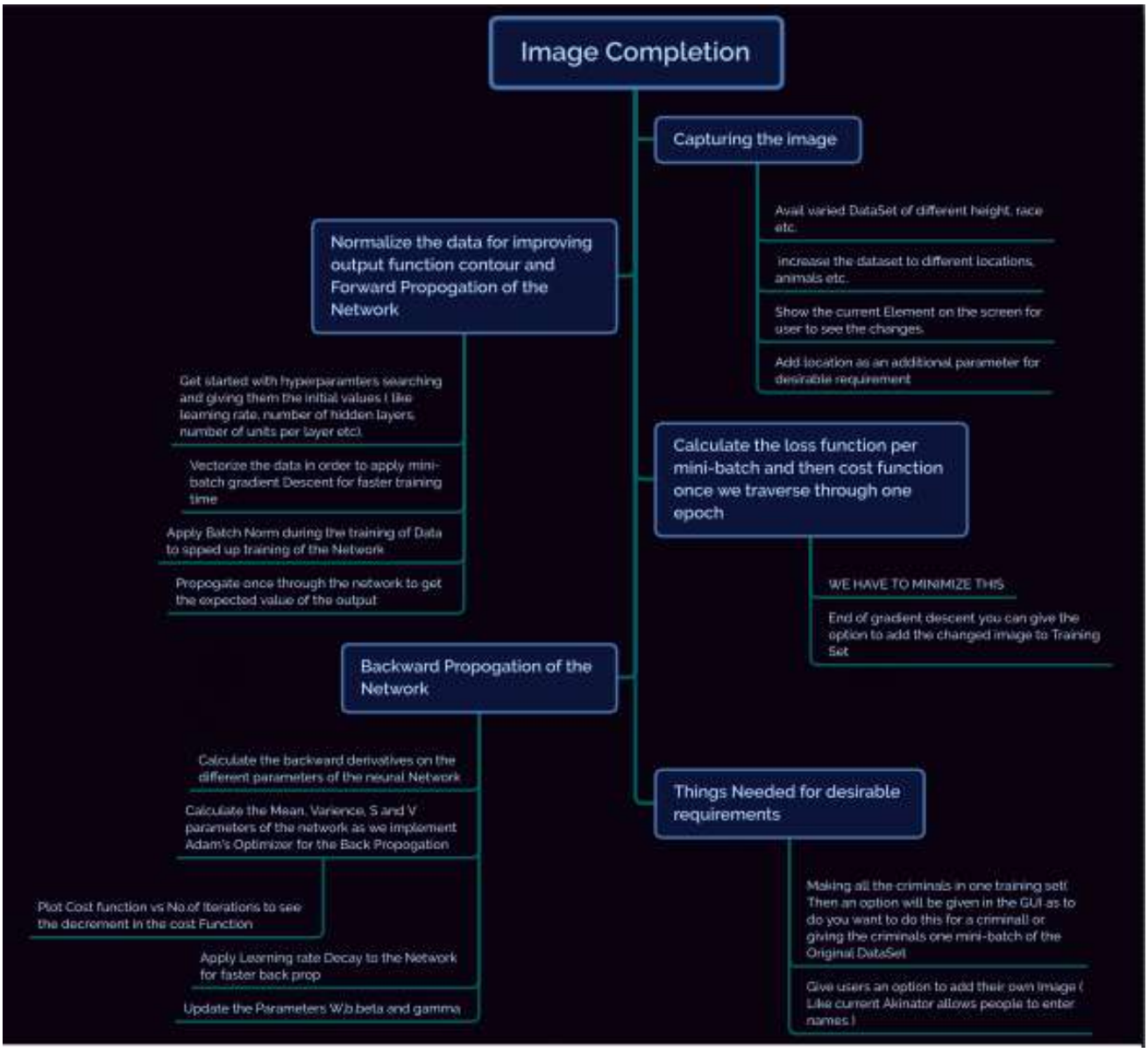
## 3.4 Desirable Non-Performance Requirements

D.N.1 Take the train and Test set Error into consideration when updating the Back Prop method

# 4) Functional Architecture

The architecture outlined below shows the functions that the system must execute to fulfill the previously mentioned requirements. The functions are derived assuming we have already have the image we want to complete or implanting.

The main functions of Pluralistic Image Completion:



# Different functions under Image Completion :

1) Capturing the image
- For image completion of Humans avail varied DataSet of different height, race etc.
- For the desired requiremnts we will increase the dataset to different locations,animals
- etc.
- Show the current Test Element on the screen so that user can see the changes for
- Themselves

2) Normalize the Input data to improve the contour for Output function

3) Forward Propogation of the Network
- Get started with hyperparamters searching and giving them the initial values ( Some of the parameters will be learning rate, number of hidden layers, number of units per layer etc.
- Vectorize the data in order to apply mini-batch gradient Descent for faster training time
- Apply Batch Norm during the training of Data to spped up training of the Network
- Propogate once through the network to get the expected value of the output

4) Calculate the Loss Function after each mini-batch is passed

5) Calculate the cost function once we traverse through one epoch
- WE HAVE TO MINIMIZE THIS
- Once you have gone through all the iterations of the gradient Descent you can give the option to add the changed image to Training Set

6) Backward Propogation of the Network
- Calculate the backward derivatives on the different parameters of the neural Network
- Calculate the Mean, Varience, S and V parameters of the network as we try to implement Adam's Optimizer for the Back Propogation
- To check if this the correct method for backward propogation we will plot Cost function vs Number of iterations and check that it keeps on decreasing
- Apply Learning rate Decay to the Network for faster back prop
- Update the Parameters W,b,beta and gamma

7) Things Needed for desirable requirements
- Making all the criminals in one training set(Then an option will be given in the GUI as to do you want to do this for a criminal) or giving the criminals one mini-batch of the Original DataSet
- Give users an option to add their own Image (Like current Akinator allows people to enter names )

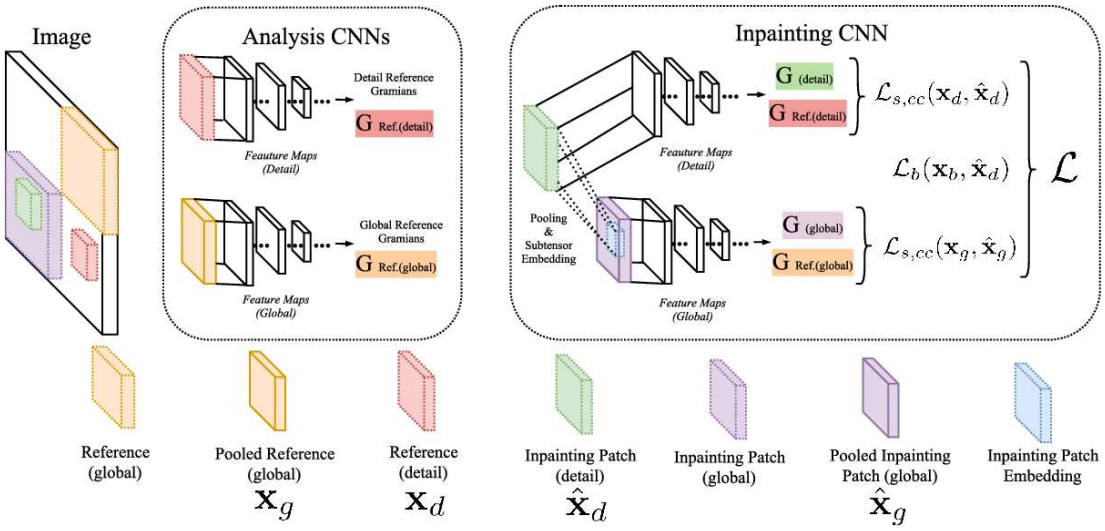
# 5) Subsystem descriptions:

## 1) Approach:

Suppose we have an image, originally Ig, but degraded by a number of missing pixels to become Im (the masked partial image) comprising the observed / visible pixels. We also define Ic as its complement partial image comprising the original missing pixels. Classical image completion methods attempt to reconstruct the original unmasked image Ig in a deterministic fashion from Im (see fig. 2 “Deterministic”). This results in only a single solution. In contrast, our goal is to sample from p(Ic|Im).

## 2) CNN Approach:

CNN-based approaches Recently, the strong potential of deep convolutional networks (CNNs) is being exhibited in all computer vision tasks, especially in image inpainting. CNNs are used specifically in order to improve the expected results in this field using large-scale training data. The sequential-based methods succeed in some part of image inpainting like filling texture details with promising results, but still the problem of capturing the global structure remains [25]. Several methods have been proposed for image inpainting using convolutional neural networks (CNNs) or encoder-decoder network based on CNN. Shift-Net based on U-Net architecture is one of these methods that recover the missing block with good accuracy in terms of structure and fine-detailed texture [25]




## 3) GAN Process:

The generative ResNet that we compose consists of down-sampling, residual blocks and up-sampling parts using the architectural guidelines introduced in [14]. Downsampling layers are implemented by using strided convolutions without pooling layers. Residual blocks do not change the width or height of the activation maps. Since our network performs completion operation in an end-to-end manner, the output must have the same dimension with the input. Thus, in the configuration of all our experiments, the number of down-sampling and up-sampling layers are selected as equal

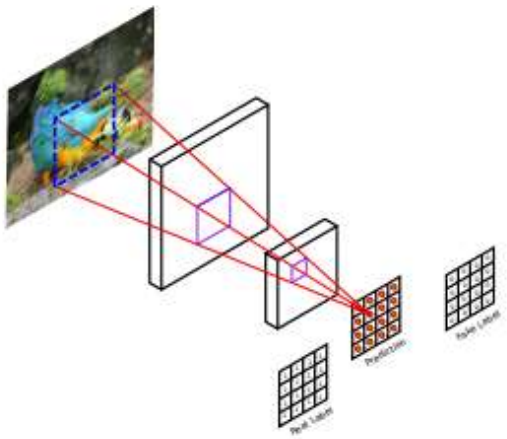


Figure is a GAN Discriminator

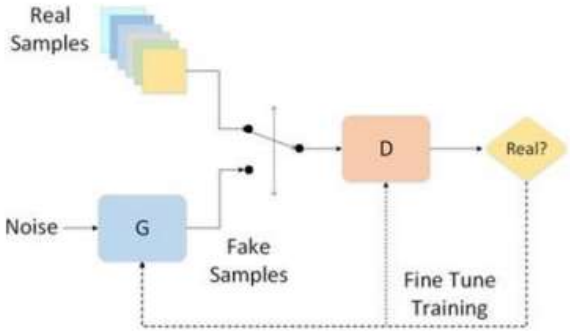



Figure 4: Framework of GANs.

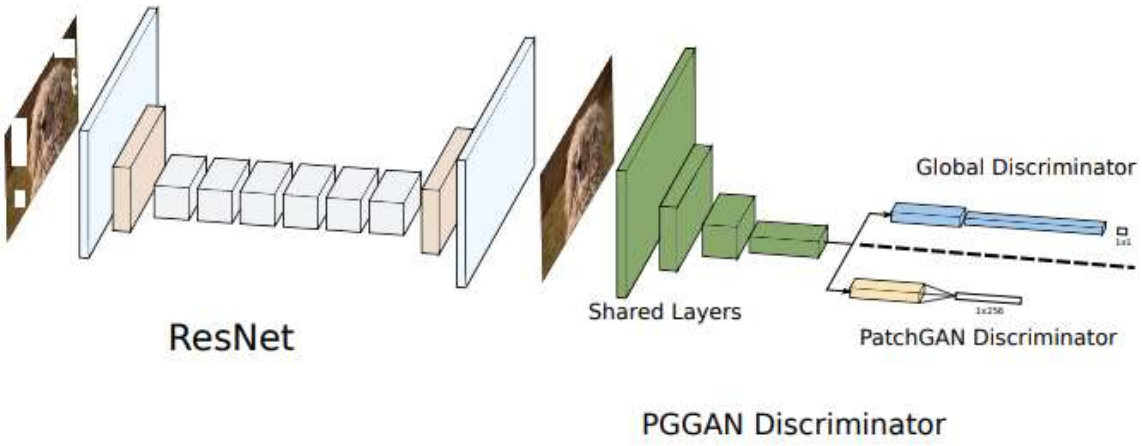



Figure is Resnet Architecture and PGGAN discriminator.

## 4) Capturing the image :

Image inpainting methods use many public and large datasets for evaluating their algorithms and comparing the performance. The categories of images determine the effectiveness of each proposed method. From these categories we can find natural images, artificial images, face images, and many other categories. In this work, we attempt to collect the most used datasets for image inpainting including Paris StreetView, Places, depth image dataset , Foreground-aware, Berkeley segmentation, ImageNet and others. We also try to cite the types of used data such as RGB images, RGB-D images and SST images. Figure below represents some frame examples from the cited datasets. Where Table 3 describe various datasets used for image inpainting approaches.

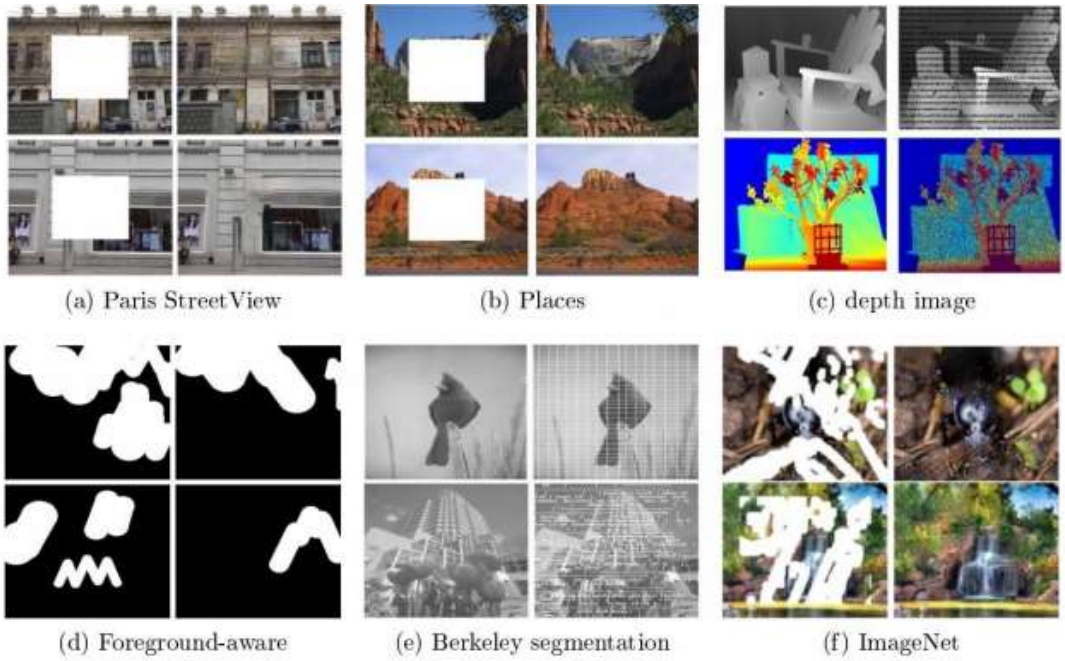
**Paris StreetView** is Collected from Google StreetView that represent a large-scale dataset that contains street images for several cities around the world. The Paris StreetView composed of 15000 images. The image's resolution is for 936 537 pixels.

**Places 1 [59]** datasets built for human visual cognition and visual understanding purposes. The dataset contains many scene categories such as bedrooms, streets, synagogue, canyon and others. The dataset is composed of 10 million images including 400> image for each scene category. It allows the deep learning methods to train their architecture with a large-scale data

**Depth image** dataset is introduced by [8] for evaluating depth image inpainting methods. The dataset is composed of two types RGB-D images and grayscale depth images. Also, 14 scene categories are included such as Adirondack, Jade plant, Motorcycle, Piano, Playable and others. The masks for damaged images are created including textual makes (text in the images) and random missing masks.

**Foreground-aware** dataset is different from the other's dataset. It contains the masks that can be added to any images for damaging it. It named irregular hole mask dataset for image inputting. Foreground-aware datasets contains 100,000 masks with irregular holes for training, and 10,000 masks for testing. Each mask is a 256 256 gray image with 255 indicating the hole pixels and 0 indicating the valid pixels. The masks can be added to any image for which can be used for creating a large dataset of damaged images.

**Berkeley segmentation database** is composed of 12 000 images segmented manually. The images collected from other dataset contains 30 human subjects. The dataset is a combining of RGB images and Grayscale images.



(a) Paris StreetView (b) Places (c) depth image

(d) Foreground-aware (e) Berkeley segmentation (f) ImageNet

**ImageNet4** is a large-scale dataset with thousands of images of each subnet. Each subnet is presented of 1000 images. The current version of the dataset contains more than 14,197,122 images where the 1,034,908 annotated with bounding box human body is annotated.

**USC-SIPI image database** contains several volumes representing the many types of images. The resolution in each volume can vary between 256x 256, 512x 512 and 1024 x1024 pixels. Generally, the datasets contain 300 images representing four volumes including texture, aerials, Miscellaneous and sequences.

**CelebFaces Attributes Dataset** is a recognized and public datasets for face recognition.it contains more that 200K celebrity images representing 10 000 identities with a large pose variations.

**Indian Pines7** consist of images representing images of three scenes including agriculture, forest and natural perennial vegetation with resolution of 145x 145 pixels.

**Microsoft COCO val2014 dataset** is a new image recognition, segmentation,

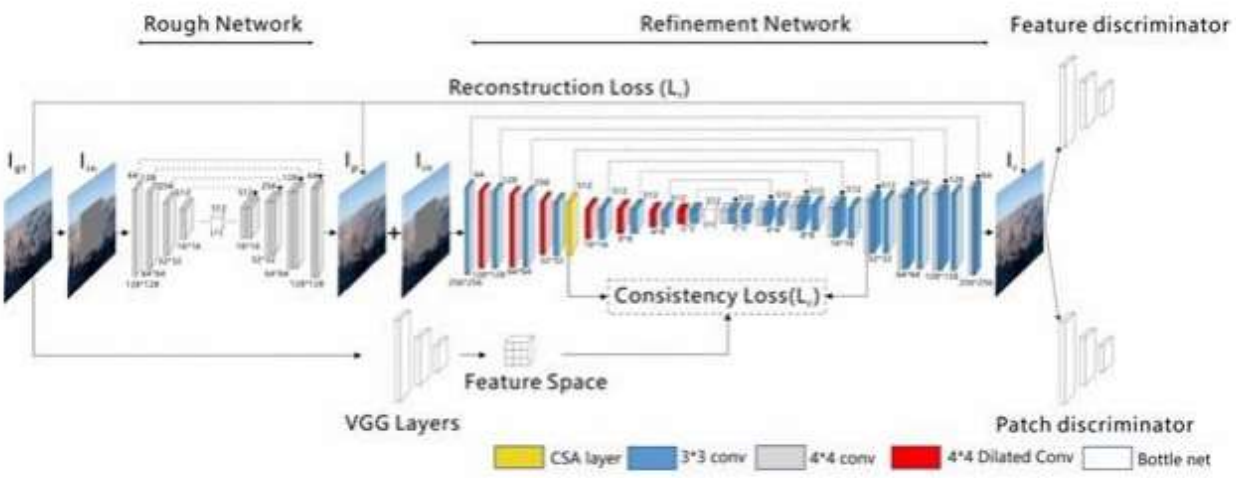
and captioning dataset. Microsoft COCO has several features with a total of 2.5 million labeled instances in 328k images.

Benchmark dataset **ICDAR 2013** is a handwritten datasets with two languages including Arabic and English. The total number of writer is 475 where the images have been scanned. The datasets contain 27 GB of data.

**SceneNet dataset** is a dataset for scene understanding tasks including semantic segmentation, object detection and 3D reconstruction. It contains RGB image and the corresponding RGB-D images which are in total 5 million images.

**Stanford Cars dataset** is a set of cars represent 196 categories of cars with different size. The datasets contain 16 200 images in total.

**Cityscapes dataset** is large scale dataset of stereo videos of street scene of 50 cities. The images contain about 30 classes of objects. Also, about 20 000 annotated frames with coarse annotations. **Middlebury Stereo** datasets contain many versions we present the two new ones [70] and [71]. **Middlebury 2006** is a depth grayscale dataset that contains images captured from 7 view with different illuminations and exposures. The images resolution is defined by three categories full-size with 1240x 1110 pixels, half size with 690x 555 pixels and the third resolution with 413x 370. **Middlebury 2014** is an RGB-D datasets unlike the other version.




Encoder and Decoder Model

## 5) Normalize the Input data to improve the contour for Output function:

By normalizing all of our inputs to a standard scale, we're allowing the network to more quickly learn the optimal parameters for each input node.Additionally, it's useful to ensure that our inputs are roughly in the range of -1 to 1 to avoid weird mathematical artifacts associated with floating point number precision. In short, computers lose accuracy when performing math operations on really large or really small numbers. Moreover, if your inputs and target outputs are on a completely different scale than the typical -1 to 1 range, the default parameters for your neural network (ie. learning rates) will likely be ill-suited for your data.

## 6) Forward Propogation of the Network:

Get started with hyperparamters searching and giving them the initial values (Some of the parameters will be learning rate, number of hidden layers, number of units per layer).Gradient descent is an optimization algorithm often used for finding the weights or coefficients of machine learning algorithms, such as artificial neural networks and logistic regression. It works by having the model make predictions on training data and using the error on the predictions to update the model in such a way as to reduce the error.

Batch normalization is a technique for training very deep neural networks that standardizes the inputs to a layer for each mini-batch. This has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks.Propogate once through the network to get the expected value of the output.

## 7) Calculate the Loss Function after each mini-batch is passed:

Neural networks are trained using stochastic gradient descent and require that you choose a loss function when designing and configuring your model.

Typically, with neural networks, we seek to minimize the error. As such, the objective function is often referred to as a cost function or a loss function and the value calculated by the loss function is referred to as simply "***loss***."

## 8) Calculate the cost function once we traverse through one epoch:

We're going to be implementing gradient descent to create a learning process with feedback. Each time — each step really — we receive some new information, we're going to make some updates to our estimated parameter which move towards an optimal combination of parameters. We get these estimates using our cost function from before.

## 9)Backward Propagation of the Network:

Back propagation is a short form for "backward propagation of errors." It is a standard method of training artificial neural networks. This method helps to calculate the gradient of a loss function with respects to all the weights in the network.

Picking the right optimizer with the right parameters, can help you squeeze the last bit of accuracy out of your neural network model. In this article, optimizers are explained from the classical to the newer approaches.

Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data.


## 10) Things Needed for desirable requirements:

Making all the criminals in one training set( Then an option will be given in the GUI as to do you want to do this for a criminal) or giving the criminals one mini-batch of the Original DataSet Give users an option for example checking if he/she is a history of cheating.

## 6) Work Management within the project

The majority of the work in this project is Software based. Our approach will be to focus and decrease the cost function of the neural network, work on the GUI of the application and work on acquiring the data sets in our Odd semester and then work towards the specification of Criminal works along with landscapes and deletion of objects from the image during the even semester. This will allow us to verify the functionality of each individual subsystems and isolate any errors, making the eventual integration smoother.

Lastly, our work would plan reflects the need to practice project management throughout the duration of the project

| SOFTWARE | | |
|---|---|---|
| **Data acquiring process** | 1.1 | For image completion of Humans avail varied DataSet of different height, race etc. |
| | 1.2 | For the desired requirements we will increase the dataset to different locations,animals etc. |
| | 1.3 | Show the current Test Element on the screen so that user can see the changes for themselves |
| | 1.4 | Give an option to the user to add |
| | 1.5 | Normalize the Input data to improve the contour for Output function |
| **Forward Propogation of the Network** | 2.1 | Get started with hyperparamters searching and giving them the initial values ( Some of the parameters will be learning rate, number of hidden layers, number of units per layer etc. |
| | 2.2 | Vectorize the data in order to apply mini-batch gradient Descent for faster training time |

| | | |
|---|---|---|
| | 2.3 | Apply Batch Norm during the training of Data to spped up training of the Network |
| | 2.4 | Propogate once through the network to get the expected value of the output |
| **Calculate the Loss Function after each mini-batch and Calculate the Cost Function after each epoch** | 3.1 | We have to minimize this. Once you have gone through all the iterations of the gradient Descent you can give the option to add the changed image to Training Set |
| **Backward Propgation** | 4.1 | Calculate the backward derivatives on the different parameters of the neural Network |
| | 4.2 | Calculate the Mean, Varience, S and V parameters of the network as we try to implement Adam's Optimizer for the Back Propogation |
| | 4.3 | Plot Cost function vs No.of Iterations and a downward slope indicates you are correct |
| | 4.4 | Apply Learning rate Decay to the Network for faster back prop |
| | 4.5 | Update the Parameters W,b,beta and gamma |
| **Things needed to be done for desired requirements** | 5.1 | Making all the criminals in one training set(Then an option will be given in the GUI as to do you want to do this for a criminal) or giving the criminals one mini-batch of the Original DataSet |
| | 5.2 | Give users an option to add their own Image (Like current Akinator allows people to enter names ) |

## 7) References

[1] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In Proceedings of the IEEE International Conference on Computer Vision, pages 3730–3738, 2015.

[2] https://arxiv.org/pdf/1903.04227.pdf

[3] H. Li, W. Luo, J. Huang, Localization of diffusion-based inpainting in digital images, IEEE Transactionson Information Forensics and Security 12 (12) (2017) 3050-3064

[4] Y.-L. Chang, Z. Yu Liu, W. Hsu, Vornet: Spatio-temporally consistent video inpainting for object removal, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019.

[5]http://openaccess.thecvf.com/content_CVPR_2019/papers/Zheng_Pluralistic_Image_Completion_CVPR_2019_paper.pdf

[6] http://www.chuanxiaz.com/publication/pluralistic/

[7]Neural Networks and Deep learning by deeplearning.ai at Coursera

[8] https://arxiv.org/ftp/arxiv/papers/1909/1909.06399.pdf

[9] http://bamos.github.io/2016/08/09/deep-completion/