VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANA SANGAMA, BELAGAVI 590018

Project Entitled

IMAGE REGENERATION WITH GENERATIVE MODELS

Submitted in partial fulfillment of the requirements for the award of degree of
BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING
For the Academic year 2017-2018

Submitted by

ABHIJITH C.	**1MV14CS004**
RAGHAVA G. DHANYA	**1MV14CS077**
SHASHANK S.	**1MV14CS131**

Project carried out at
Sir M. Visvesvaraya Institute of Technology
Bangalore-562157

Under the Guidance of
MRS. SUSHILA SHIDNAL
Assistant Professor, Department of CSE
Sir M Visvesvaraya Institute of Technology, Bangalore.

Department Of Computer Science & Engineering
Sir M. Visvesvaraya Institute Of Technology
Hunasamaranahalli, Bangalore 562157

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANA SANGAMA, BELAGAVI 590018

Project Entitled

IMAGE REGENERATION WITH GENERATIVE MODELS

Submitted in partial fulfillment of the requirements for the award of degree of
BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING
For the Academic year 2017-2018

Submitted by

ABHIJITH C.	**1MV14CS004**
RAGHAVA G. DHANYA	**1MV14CS077**
SHASHANK S.	**1MV14CS131**

Project carried out at
Sir M. Visvesvaraya Institute of Technology
Bangalore-562157

Under the Guidance of
MRS. SUSHILA SHIDNAL
Assistant Professor, Department of CSE
Sir M Visvesvaraya Institute of Technology, Bangalore.

Department Of Computer Science & Engineering
Sir M. Visvesvaraya Institute Of Technology
Hunasamaranahalli, Bangalore 562157

SIR M VISVESVARAYA INSTITUTE OF TECHNOLOGY

BENGALURU 562157

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

It is certified that the project work entitled "Image Regeneration With Generative Models" is carried out by Abhijith C. (1MV14CS004), Raghava G. Dhanya (1MV14CS077), Shashank S. (1MV14CS131) bona-fide student of Sir M Visvesvaraya Institute of Technology in partial fulfillment for the award of the Degree of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2016-2017. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the course of Bachelor of Engineering.

MRS. SUSHILA SHIDNAL	**PROF. DILIP K SEN**	**DR. V. R. MANJUNATH**
Asst. Prof & Internal Guide	**Head of Department**	**Principal**
Dept. of CSE, Sir MVIT	**Dept. of CSE, Sir MVIT**	**Sir MVIT**

Name of the examiners **Signature with date**

1)

2)

DECLARATION

We hereby declare that the entire project work embodied in this dissertation has been carried out by us and no part has been submitted for any degree or diploma of any institution previously.

Place: Bengaluru
Date:

Signature of Students

Abhijith C.
1MV14CS004

Raghava G. Dhanya
1MV14CS077

Shashank S.
1MV14CS131

ABSTRACT

Current advances in Generative Adversarial Networks allow us to obtain near realistic images of faces but it is still quite distinguishable from actual photographic images. The technology is also not very amiable to changes in the orientation of faces in Convolutional Neural Networks(CNN). Additionally, the amount of data required to train the network must be exhaustible, for example, in case different perspectives of a face are required the various perspectives must be explicitly present in the training data to achieve the result. Thus the network requires humongous amounts of data.

In this project we propose a novel approach to accomplish the same results using CapsNet. CapsNet employs a dynamic routing algorithm which replaces the scalar-output feature detectors of the CNN with vector-output capsules. A capsule is essentially a group of neurons describing a specific part of object or image. Active capsules at one level make predictions, via transformation matrices, for the instantiation parameters of higher-level capsules. In essence, the CapsNet is the reverse of the common Computer Graphics pipeline where we convert objects to their renders. The CapsNet works from the pixel level and works up towards the object.

We propose that the amount of data required to train a comparable model is very small while it gives comparable, if not better, results.

CONTENTS

CHAPTER 1

INTRODUCTION

"What I cannot create, I do not understand."

Richard Feynman

One of the main aspirations of Artificial Intelligence is to develop algorithms and techniques that enrich computers with ability to understand our world. Generative models are one of the most promising approaches towards achieving this goal.

1.1 Generative Models

A generative model is a mathematical or statistical model to generate all values of a phenomena. To train such a model, we first collect a large amount of data in some domain (e.g., think millions of images, sentences, or sounds, etc.) and then train a model to generate data like it.

A generative algorithm models how data was generated to classify a data instance. It poses the question: according to my generation hypotheses, which category is most likely to generate this data instance? A discriminative algorithm does not care about how the data was generated, it just classifies a given data instance; that is, given the features of a data instance, they predict a label or category to which that data belong. Discriminative models learn the boundary between classes while Generative models model the distribution of individual classes; that is, a generative model learns the joint probability distribution $p(x, y)$ while a discriminative model learns the conditional probability distribution $p(y|x)$ "probability of y given x".

The trick is that the neural networks that we use as generating models have a significantly smaller number of parameters than the amount of data on which we train them, so the models are forced to effectively discover and internalize the essence of the data to generate it.

There are multiple approaches to build a generative models

1.1.1 Generative adversarial networks

Generative adversarial networks (GANs) are a class of generative algorithms used in unsupervised machine learning, implemented by a system of two neural networks competing in a zero-sum game framework. They were presented by Ian Goodfellow *et al*. [**?**]. This technique can generate photographs that seem at least superficially

authentic to human observers, having many realistic features (though in tests people can tell real from generated in some cases).

1.1.2 Variational Autoencoders

An autoencoder network is actually a pair of two connected networks, an encoder and a decoder. An encoder network receives an input and converts it into a smaller, denser representation that the decoder network can use to convert it back to the original input. Variational Autoencoders (VAEs) have one fundamentally unique property that separates them from vanilla autoencoders, and it is this property that makes them so useful for generative modeling: their latent spaces are, by design, continuous, allowing easy random sampling and interpolation. Variational Autoencoders (VAEs) allow us to formalize generative modeling problem in the framework of probabilistic graphical models where we are maximizing a lower bound on the log likelihood of the data

1.1.3 Autoregressive models

Autoregressive models such as PixelRNN, on the other hand train a network that models the conditional distribution of every individual pixel given previous pixels (to the left and to the top). These models efficiently generate independent,exact samples via ancestral sampling. This is similar to plugging the pixels of the image into a char-rnn, but the RNNs runs both horizontally and vertically over the image instead of just a 1D sequence of characters.

1.2 Generative Adversarial Networks

Generative Adversarial Networks, which we already discussed above, pose the training process as a game between two distinct networks: A neural network, called the generator, generates new instances of data, while the other, the discriminator, evaluates their authenticity; discriminator network tries to classify samples as either coming from the true distribution $p(x)$ or the model distribution $\hat{p}(x)$. Every time the discriminator notices a difference between the two distributions the generator adjusts its parameters slightly to make it go away, until at the end (in theory) the generator exactly reproduces the true data distribution and the discriminator is guessing at random, unable to find a difference.

The generator takes noise as input and attempts to produce an image that belongs to the real distribution; that is, it tries to fool the discriminator to accept it as real image. Discriminator takes a generated image or a real image as input and attempts to correctly classify the image as real or fake (generated).

To learn the distribution of the generator p_g over data $\boldsymbol{x}$, we define a prior on input noise variables $p_z(\boldsymbol{z})$, then represent a mapping to data space as $G(\boldsymbol{z};\theta_g)$, where G is a differentiable function represented by a neural network with parameters θ_g. We define a second neural network $D(\boldsymbol{x};\theta_d)$ that outputs a single scalar. $D(\boldsymbol{x})$ represents the probability that $\boldsymbol{x}$ came from the data rather than p_g. We train D to maximize the probability of assigning the correct label to the training examples and samples of G. We simultaneously train G to minimize $\log(1 - D(G(\boldsymbol{z})))$.

This can be represented minimax game

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{1.1}$$

1.3 Convolutional Neural Networks

Before we can jump to understanding Capsule Networks, we need to know about Convolutional Neural Networks(CNNs). Convolutional neural networks are very similar to ordinary neural networks, they consist of neurons that have learn-able weights and biases. Each neuron receives inputs, performs a scalar product and possibly follows it with a nonlinearity. The entire network expresses a single differentiable score function: raw image pixels at one end to class scores at the other end. And they still have a loss function on the last layer.

The major difference is that CNN explicitly assumes that the inputs are images, which allows us to encode certain properties in the architecture. These then make the forward functions more efficient to implement and significantly reduces the amount of parameters in the network.

Ordinary neural networks don't scale well to full images, for example, A colour image of dimensions of 150x150 has a shape (150,150,3), a fully connected neuron on first layer which receives this image would require 67500 weights. Unlike an ordinary neural network, the layers of a CNN have neurons arranged in 3 dimensions: width, height, depth, the neurons in a layer will only be connected to a small region of the layer before it, instead of all of the neurons in a fully-connected manner. CNN will reduce the full image into a single vector of class scores, arranged along the depth dimension.

CHAPTER 2

LITERATURE SURVEY

"Adversarial training is the coolest thing since sliced bread"

Yann LeCun,
Director of AI Research at Facebook and Professor at NYU

GANs were first introduced by Ian Goodfellow *et al*. [**?**] in Neural Information Processing Systems 2014. The paper proposes a completely new framework for estimating generative models via an adversarial process. In this process two models are simultaneously trained. According to [**?**] the network has a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. This original work by Ian Goodfellow uses fully connected neural networks in the generator and the discriminator.

Since then, there has been tremendous advancements in Deep Learning. A convolutional neural network (CNN, or ConvNet) [**?**] is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery. The convolution layer parameters consist of a set of learn-able filters, also called as kernels, which have a small receptive field, but they extend through the full depth of the input volume. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input.

A breakthrough development that occurred in Adversarial Networks was the introduction of "Deep Convolutional Generative Adversarial Networks" by Alec Radford *et al* [**?**]. He applied a list of empirically validated tricks as the substitution of pooling and fully connected layers with convolutional layers.

The power of the features encoded in the latent variables was further explored by Chen at al. [**?**]. They propose an algorithm which is completely unsupervised, unlike previous approaches which involved supervision, and learns interpretable and disentangled representations on challenging datasets. Their approach only adds a negligible computation cost on top of GAN and is easy to train.

Today, most GANs are loosely based on the former shown DCGAN [**?**] architecture. Many papers have focused on improving the setup to enhance stability and performance. Many key insights was given by Salimans *et al*. [**?**], like Usage of convolution with stride instead of pooling, Usage of Virtual Batch Normalization, Usage of Minibatch Discrimination in DD, Replacement of Stochastic Gradient Descent with Adam Optimizer [6], Usage of one-sided label smoothing.

Another huge development came with the introduction of Wasserstein GANs by Martin Arjovsky [**?**] . He introduced a new algorithm named WGAN, an alternative to traditional GAN training. In this new model, he showed that the stability of learning can be improved, remove problems like mode collapse, and provide good learning curves useful for debugging and hyperparameter searches.

This recently proposed Wasserstein GAN (WGAN) [**?**] makes progress toward stable training of GANs, but sometimes can still generate only low-quality images or fail to converge. Ishaan Gulrajani with Martin Arjovsky proposed an alternative in [**?**] to fix the issues the previous GAN faced. This proposed method performs better than standard WGAN and enables stable training of a wide variety of GAN architectures with almost no hyperparameter tuning, including 101-layer ResNets [**?**] and language models over discrete data.

Work by Mehdi Mirza *et al*. [**?**] introduced the conditional version of GAN which can be constructed by simply feeding the data, y, we wish to condition on to both the generator and discriminator. The CGAN results were comparable with some other networks, but were outperformed by several other approaches – including non-conditional adversarial nets.

Sebastian Nowozin *et al*. [**?**] discussed the benefits of various choices of divergence functions on training complexity and the quality of the obtained generative models. They show that any f-divergence can be used for training generative neural samplers.

Ming-Yu *et al*. [**?**] proposed coupled generative adversarial network (CoGAN) for learning a joint distribution of multi-domain images. The existing approaches requires tuples of corresponding images in different domains in the training data set. CoGAN can learn a joint distribution without any tuple of corresponding images.

A big breakthrough in the field of Deep Learning came with the introduction of CapsNets or Capsule Networks [**?**] by the Godfather of Deep Learning, Geoffrey Hinton. CNNs perform exceptionally great when they are classifying images which are very close to the data set. If the images have rotation, tilt or any other different orientation then CNNs have poor performance. This problem was solved by adding different variations of the same image during training.

CHAPTER 3

OBJECTIVE

"Any A.I. smart enough to pass a Turing test is smart enough to know to fail it."

Ian McDonald,
River of Gods

The broad objective is to use the existing Generative Adversarial Networks technologies to aid in the generation of human faces such that the GAN generated images is indistinguishable from the images of the real people used to train the network, i.e fake images should look very much real. This would be then extended to completion of faces, ie. reconstruction of facial features given a partial face.

The internal specific objective would be to achieve the above said objectives using a ground breaking technology released in fall 2017, the Capsule Nets. The existing latest state-of-the-art GAN architectures use Convolution Neural Networks in their Generators and Discriminators. The CNNs are said to have the drawbacks as mentioned before, where they cannot understand orientation and spatial relationships unless they are extensively trained with all possible images. This major drawback is handled by Capsule Networks.

Using the CapsNet architecture into the Generator/Discriminator could improve these Adversarial Networks quite drastically. This mating of the revolutionary Generative Adversarial Networks along with the ground-breaking Capsule Networks, resulting in "Capsule Net GANs" is the overarching objective.

CHAPTER 4

SCOPE

"By far the greatest danger of Artificial Intelligence is that people conclude too early that they understand it."

Eliezer Yudkowsky,
Machine Intelligence Research Institute

Generative Adversarial Networks are one of the hottest topics in Deep Learning right now. The applications of GANs are far ranging and immense. Creating Infographics from text, creating animations for rapid development of marketing content, generating website designs are to name a few. Our focus in this project is to implement a way to complete images of faces by generating the missing pieces using a GAN.

This particular implementation of the technology would be immensely useful in a variety of circumstances. A few straightforward applications include face sketching of suspects in a crime using eye witness accounts, super resolution of CCTV camera footage to enhance faces, filling in of old degraded color photos, etc.

CHAPTER 5

METHODOLOGY

"Artificial intelligence, in fact, is obviously an intelligence transmitted by conscious subjects, an intelligence placed in equipment."

Pope Benedict XVI

The first step would be to implement the state-of-the-art in image regeneration to guage the improvements. We use DCGAN to start of with. The results of the training and testing will be recorded to compare it with the results of our CapsNet-based approach later. We will be using CapsNet as the underlying technology to implement our GAN (CapsGAN). The goal is to replace the CNN inside DCGAN with CapsNet and compare the results. The GAN internally consists of two components - a generator and a discriminator - which we build out of CapsNet. The discriminator is initially trained separately to distinguish real and fake data, and later they work together to improve upon their performance by acting as adversaries.

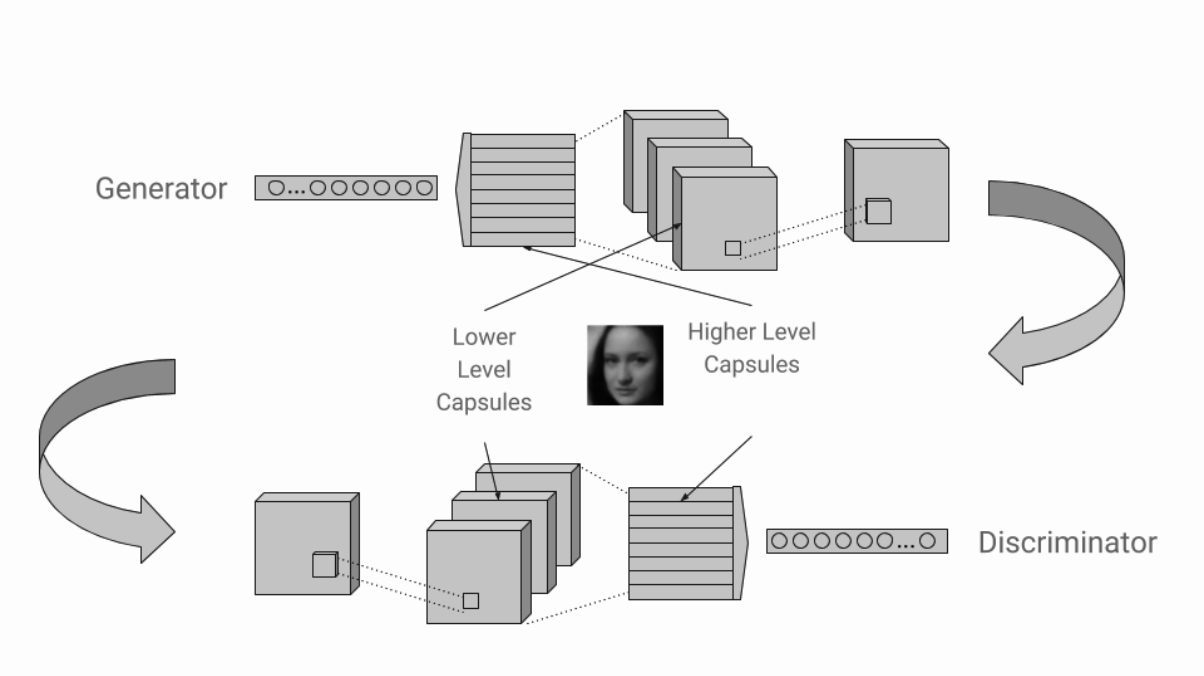

Figure 5.1: Proposed architecture

The generator will use noise as input to generate faces. We will use random data as this noise. This ensures the data is unique and across the spectrum while retaining a normal distribution.

The CapsNet making up the discriminator consists of a small convolutional network to convert low level data in the form of pixels into an artifact called "pose". These poses can be anything, like nose, ear, eye, etc. These poses are then passed on as input to the later lower layers consisting of components called Capsules. A capsule is analogous to the human brain containing different modules to handle different tasks. The brain has a mechanism to route the information among the modules, to reach the best modules that can handle the information.

A capsule is a nested set of neural layers. Each capsule is able to handle one particular pose and communicate its calculation to other capsules which can use that calculation. This calculation is in the form of a probability prediction of the current pose that takes place in its logistic unit. This working is fundamentally different from convolutional networks, which utilizes Max Pooling. Max pooling selects the most active input node from the next layer to pass on the information. CapsNet on the other hand selects the next capsule based on which capsule would be capable of handling that information. This is called Dynamic routing. This results in invariance of information to the position and orientation of features in an object while ignoring the invariance in very low level features as, at the pixel level, this does not matter.

The generator is built using an architecture that is a mirror image of the discriminator. Inside, the data flow is in the opposite direction. The job of the discriminator is to take the images given out by the generator and discriminate it against a ground truth. The discriminator selects the class based on how close the images are in agreement. The underlying principle is that when multiple entities agree with each other in higher dimensions, the chances of it happening due to complete coincidence is exponentially minimal. This ensures the understanding by the CapsNet of the world is remarkably similar to humans.

CHAPTER 6

TECHNOLOGY

"I think, therefore I am"

René Descartes,
French philosopher and scientist

- **Adversarial Training:** Two models undergoing training simultaneously by competing against each other. The output of each model acts as an adversary for the other to improve upon.
- **Generative Model:** Any model capable of generating completely new realistic data of a class.
- **Discriminative Model:** Here, as a Discriminator, a model which is capable of distinguishing between actual ground truth from the true distribution and generated information from a model.
- **Gradient Descent:** A method of optimizing an objective function using a first-order iterative approach to find a local minimum.
- **Gaussian Model:** A model that fits the data in the shape of a Gaussian function, identified by a characteristic "bell curve". CapsNet uses this to find agreeing probability outputs of the "capsules".
- **TensorFlow:** An open source software library for numerical computation using data flow graphs. It was originally developed by the Google Brain Team within Google's Machine Intelligence research organization for machine learning and deep neural networks research.

Other essentials:

- **CPU:** 3GHz quad core, x86-64 architecture (or)
- **GPU:** NVIDIA or any other TensorFlow supported GPU, CUDA or cuDNN
- **Python:** 2.7 or 3.4 and above

CHAPTER 7

CONCLUSION

"A year spent in artificial intelligence is enough to make one believe in God"

Alan Perlis,
First Turning award recipient

During the course of this project, we wish to replicate the results of the existing state-of-the-art in Generative Models. Using this as a stepping stone, we would like to incorporate a hitherto unexplored option in CapsNet for Generative Models. Our motivating assumption is that CapsNet would provide a performance improvement.We base this on the idea that it is more capable of understanding the variances in objects. This in turn should lead to lower data requirements during training of the model and consequently lower power consumption.

We wish to provide a comparison between our novel CapsNet-based approach and other implementations of GAN for the same task. We would like to implement a proof of concept by developing a application to complete incomplete images of human faces. This could later on be used in enhancement of hazy CCTV footage to identify individuals, which would be immensely helpful to law enforcement personnel.