DEEP LEARNING FOR GENERATION OF CONCEPT CAR DESIGNS

SARAH MUKUTI MUTISYA	N11/3/1247/017
BONFACE MARTIN OYWA	N11/3/0486/017
JOHN KAMAU NDICHU	N11/3/1171/015
KENNEDY MUTISO MWENDWA	N11/3/0443/017
SHADRACK DOROTHY MUTETI	N11/3/0438/017

Software project in partial fulfillment of the requirements for award of Bachelor of Science in Computer Science degree of Laikipia University

JANUARY 2021

DEEP LEARNING FOR GENERATION OF CONCEPT CAR DESIGNS

SARAH MUKUTI MUTISYA	N11/3/1247/017
BONFACE MARTIN OYWA	N11/3/0486/017
JOHN KAMAU NDICHU	N11/3/1171/015
KENNEDY MUTISO MWENDWA	N11/3/0443/017
SHADRACK DOROTHY MUTETI	N11/3/0438/017

Software project in partial fulfillment of the requirements for award of Bachelor of Science in Computer Science degree of Laikipia University

JANUARY 2021

DECLARATIONS

We hereby declare that this project is our work and has not been submitted to any other university for purposes of examination. All the information given is our own and all the cited sources are quoted and acknowledged accordingly.

Signature: ______________________ Date: ______________________

Signature: ______________________ Date: ______________________

Signature: ______________________ Date: ______________________

Signature: ______________________ Date: ______________________

Signature: ______________________ Date: ______________________

RECOMMENDATION

The project “Deep Learning for Generation of Concept Car Designs” has been presented to the Computing and Informatics Department of Laikipia University. We have reviewed the thesis and recommend it to be accepted in partial fulfillment of the requirement for the Bachelor of Science in Computer Science.

Signature: ____________________ Date: ____________________

Dr. Kirori Mindo

Department of Computing and Informatics, Laikipia University

COPYRIGHT

ACKNOWLEDGMENT

This is to acknowledge the Almighty for His providence and blessings during the whole course of undertaking our project. We acknowledge all the lecturers in the Department of Computing and Informatics, Laikipia University who impacted us with the skills that were essential in executing the project. It would have been an uphill task to work on the project without the prerequisite coding and software engineering tools.

In a special way we acknowledge our supervisor Dr. Kirori Mindo for his guidance and all-time correction and the timely recommendations that enabled us to successfully complete the project. We gladly acknowledge our course mates who gave us insights and advice during the implementation of our project.

We also acknowledge the researchers at Facebook AI Research for their works in the field of Deep Learning. They were a great source of inspiration.

DEDICATION

We dedicate this project to the entire Department of Computing and Informatics, Laikipia University. To our supervisor Dr. Kirori Mindo who patiently listened to our presentations and gave us the guidance we needed to complete the project and make it a success.

To our fellow students who supported us through words of encouragement and technical assistance whenever we reached out to them. We cannot also forget our parents who contributed immensely to our learning process. Their financial and emotional aid ensured that we had peace of mind as we worked on our project and had access to all the necessary materials.

ABSTRACT

Generative Adversarial Networks (GANs) is a technique that has not been in existence for a long time but one which has revolutionized the field of Machine Learning.

A GANs model uses two competing Neural Networks; a Generator model and a Discriminator model. The Generator uses noise as an input to produce sample images as output. The Discriminator takes both the data produced by the Generator and the data from the real distribution or real images and compares the two. The aim is that the two models continuously get better at their functionality whereby the Generator produces more realistic images while the Discriminator gets better at distinguishing the generated samples from the real images fed into it.

GANs can be applied in real world data modelling for different scenarios but as for now they have only successfully been implemented in producing images.

TABLE OF CONTENT

CHAPTER 1 – INTRODUCTION

This chapter gives an introduction of Generative Adversarial Networks (GANs) as a Machine Learning technique and their extensive application in image generation. The chapter further gives a background information for the project and a problem statement, as well as the objectives and significance of the GANs if implemented.

1.1 BACKGROUND

Creating designs is a thought intensive process for the human brain, and it is a normal phenomenon for designers to hit a wall. But what if a computer could be trained to create something that looks somewhat unique based on other objects that have already been created? The Generative adversarial network is a machine learning technique that uses two models; a generator and a discriminator.

The generator generates new images based on the training images fed to it. The discriminator then classifies the image as either real or fake. The two networks (generator and discriminator) work hard to outsmart each other; the generator learns to generate images that seem so perfect like the testing images.

1.2 STATEMENT OF THE PROBLEM

It is with no doubt that people are highly creative and they have the ability to produce so many new designs. However, it can get to the point that the creative capacity of a group or an individual reaches its limits. Someone would then hope for a tool that gives them something somewhat unique with a little bit less effort exerted on them.

A model that is well trained to produce perfect designs that are somewhat indistinguishable, in terms of quality from what already exists would therefore be a welcome relief. It is for this reason that we decided to base our project on studying the foundations of Generative Adversarial Networks and implementing them using Machine Learning's subset, Deep Learning to Generate Concept Car Designs.

1.3 OBJECTIVES

General Objective

To study and implement Machine Learning techniques through Generative Adversarial Networks for randomized generation of Concept Car Designs.

Specific Objectives

i) To create a model that is able to generate new images of cars based on the data set that has been fed to it.

ii) To create a model that is able is able to discern that the car images produced by the generator are not real cars concepts and make the discriminator better at producing better and unique images.

1.4 JUSTIFICATION

Many people and even companies invest a lot of time in finding new designs for their products. For example, mobile phones and cars are items that are designed from very unique inspirations. New concept car designs would also need one to have a unique design. At some point the person tasked with the responsibility of coming up with the design may find it quite overwhelming but if a tool that gave them a unique inspiration based on what has already been created or something completely new.

By simply feeding a series of bits into this model you obtain your design without a breaking a sweat. This project is of great significance when it comes to creating new images based on past designs or even something completely new.

CHAPTER 2 – LITERATURE REVIEW

2.0 INTRODUCTION

This section presents a comprehensive overview of Generative Adversarial Networks (GANs), Deep Convolution Neural Networks (DCNNs), Unsupervised Representation Learning, followed by and in-depth look at Deep Convolution Generative Adversarial Networks (DCGANs), which is used in this project.

2.1 GENERATIVE ADVERSARIAL NETWORKS (GANs)

Generative Adversarial Networks (Goodfellow et al. 2014) is a framework for estimating a generative model via an adversarial process, in which we simultaneously train two models, a generative model G that aims a capturing the data distribution, and a discriminative model D, that aims at estimating the probability that a sample image came from the training data rather than G.

The concept corresponds to the min-max two player game, where during the training process, G tries to maximize the probability of D making a mistake, by classifying its output i.e fake data as real, while at the same time D tried to minimize that probability. This is meant to drive both models to improve their methods until the fake data from G are indistinguishable from the real training data.

Their approach explored the special case when the G model generates sample by passing random noise through a multilayer perceptron. Their D model is also a multilayer perceptron. They also used backpropagation () and dropout () algorithms for samples in the D model, and only forward propagation for the G model, which we heavily borrow for out implementation.

2.2 DEEP CONVOLUTION NEURAL NETWORKS (DCNNs)

Convolution Networks (LeCun, 1989), also known as Convolution Neural Networks (CNNs) are a special kind of neural networks for processing data that has spatial properties thus having a grid-like topology in it, e.g time-series data, can be thought of as a 1-D grid taking samples at regular time intervals, and image data, which can be thought of as a 2-D grid of pixels.

CNNs have had tremendously successful application in practical applications and huge adoption in Computer Vision applications.

The name “Convolution Neural Network” implies that the network employs a mathematical operation called a convolution, which is a special kind of linear operation. Therefor CNN are neural networks that can use these convolutions in place of general matrix multiplication in at least one of their layers. They also pass data through their network in forward propagation style.

The Deep in DCNNs comes by utilizing many hidden CNN layers in a network.

2.3 UNSUPERVISED REPRESENTATIONAL LEARNING

Representation Learning is an approach that uses Machine Learning (ML) to discover not only the mapping from representation to output, but also the representation itself. Think of it like a signal with a feedback look.

Learned representations often result in much better performance than can be obtained with hand-designed representations. They also enable AI systems to rapidly adapt to new tasks, with minimal human intervention.

Unsupervised Representation Learning algorithms experience a dataset containing many features, then learn useful properties of the structure of the dataset. This concept is applied in the G model of GANs, which we'll also use for our Unsupervised Deep Convolution Generative Adversarial Network to be described below.

2.4 DEEP CONVOLUTION GENERATIVE ADVERSARIAL NETWORKS (DCGANs)

A DCGAN is a direct extension of the GAN described above, except that it explicitly uses convolutional and convolutional-transpose layers in the discriminator and generator, respectively. It was first described by Radford et. al.

The discriminator is made up of strided convolution layers, batch norm layers, and LeakyReLU activations. The input is a 3x64x64 input image and the output is a scalar probability that the input is from the real data distribution. The generator is comprised of convolutional-transpose layers, batch norm layers, and ReLU activations.

The input is a latent vector, z, that is drawn from a standard normal distribution and the output is a 3x64x64 RGB image. The strided conv-transpose layers allow the latent vector to be transformed into a volume with the same shape as an image. In the paper, the authors also give some tips about how to setup the optimizers, how to calculate the loss functions, and how to initialize the model weights.

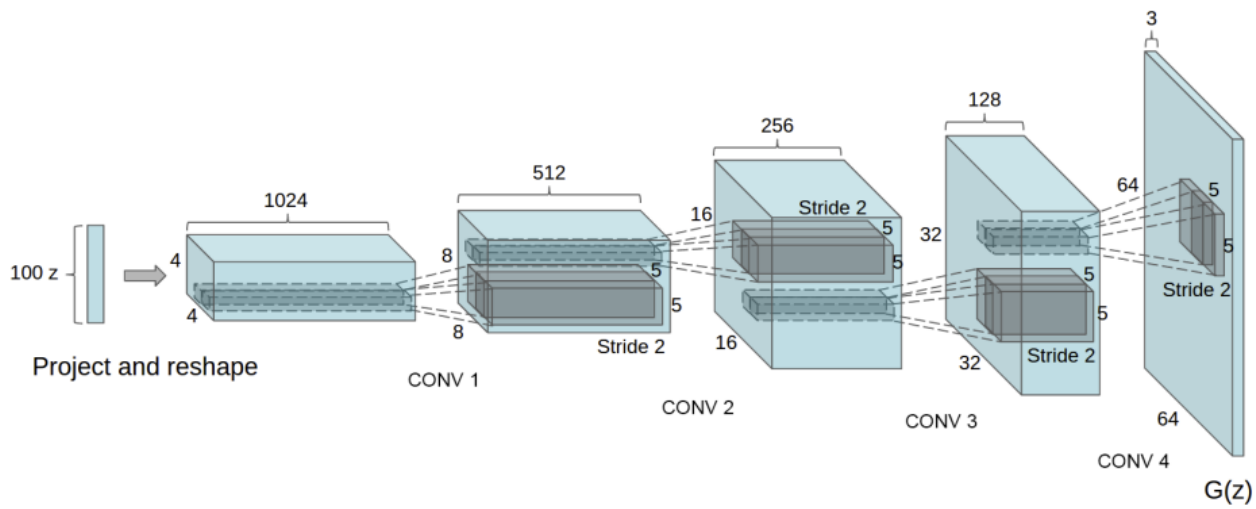
CHAPTER 3 – METHODOLOGY

3.0 INTRODUCTION

The dataset used in the project is a hand-designed portion of the Stanford Cars data set containing images of cars. Since no testing is needed, the entire datasets were used as one unit during training.

The generator, G, is designed to map the latent space vector (z) to data-space. Since our data are images, converting z to data-space means ultimately creating a RGB image with the same size as the training images (i.e. 3x64x64). In practice, this is accomplished through a series of strided two dimensional convolutional transpose layers, each paired with a 2d batch norm layer and a relu activation.

The output of the generator is fed through a tanh function to return it to the input data range of $[-1,1]$. It is worth noting the existence of the batch norm functions after the conv-transpose layers, as this is a critical contribution of the DCGAN paper. These layers help with the flow of gradients during training. An image of the generator from the DCGAN paper is shown below.



The discriminator, D, is a binary classification network that takes an image as input and outputs a scalar probability that the input image is real (as opposed to fake). Here, D takes a 3x64x64 input image, processes it through a series of Conv2d, BatchNorm2d, and LeakyReLU layers, and outputs the final probability through a Sigmoid activation function. This architecture can be extended with more layers if necessary for the problem, but there is significance to the use of the strided convolution, BatchNorm, and LeakyReLUs. The DCGAN paper mentions it is a good practice to use strided convolution rather than pooling to downsample because it lets the network learn its own pooling function. Also batch norm and leaky relu functions promote healthy gradient flow which is critical for the learning process of both G and D.

From the DCGAN paper, the authors specify that all model weights shall be randomly initialized from a Normal distribution with mean=0, stdev=0.02. The aim of weight initialization is to prevent layer activation outputs from exploding or vanishing during the course of a forward pass through a deep neural network. If either occurs, loss gradients will either be too large or too small to flow backwards beneficially, and the network will take longer to converge, if it is even able to do so at all.

Back-end Framework is PyTorch, an open source machine learning framework that accelerates the path from research prototyping to production deployment.

The loss metric was used to monitor while the models were training.

3.1 APPROACH

Only three transforms or data augmentations were performed on the images. Resizing to 64 by 64 for the D model, Transformation to Tensor, since PyTorch models process data in form Tensors, and lastly, Normalization of values / pixels to between -1 and 1, needed for the Hyperbolic Tangent (Tanh) activation function.

The Binary Cross Entropy Loss was used as the loss function since the challenge is a binary classification problem, together with the Adam optimization function.

All training was done on GPU.

The final model was deployed using the Flask Web Microframework and hosted on Heroku.

3.2 SOFTWARE

1. Jupyter Notebooks
2. Python 3.8.4
3. PyTorch
4. Flask

5. Heroku

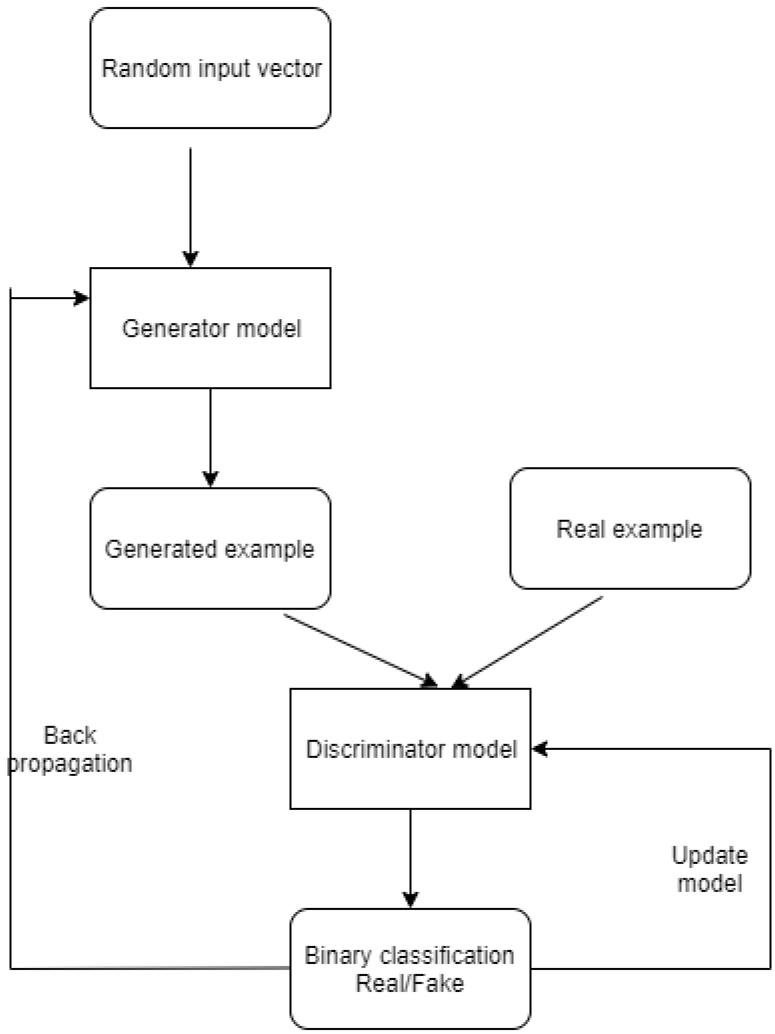
6. Ubuntu 20.04 LTS

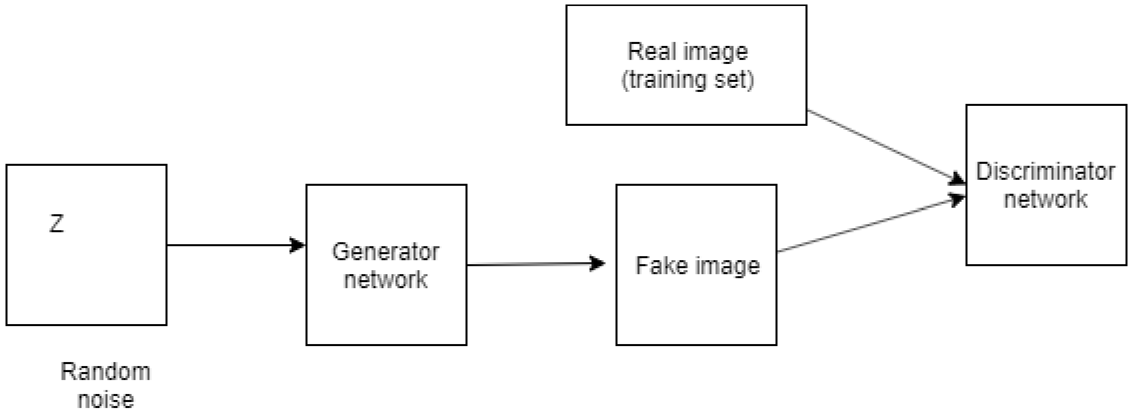
3.3 HARDWARE

1. NVIDIA GTX 1660 GPU

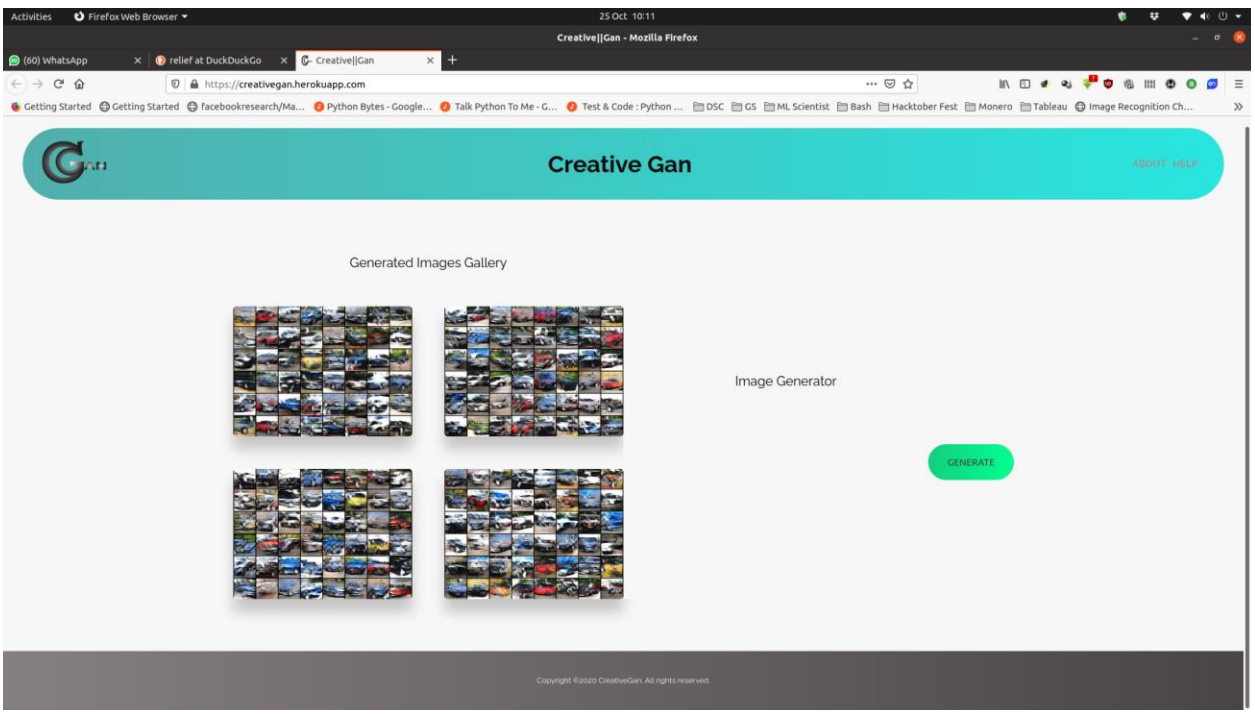
CHAPTER 4 – PROJECT PRESENTATION

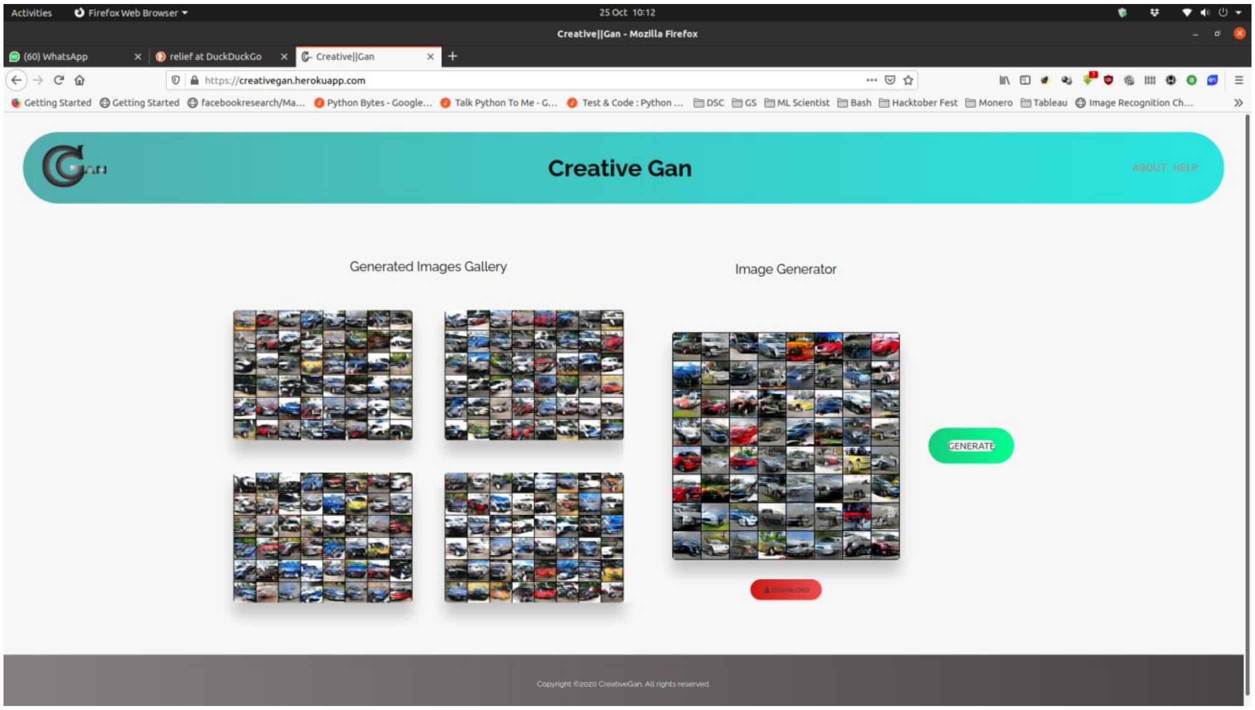
4.1 DATA FLOW DIAGRAMS (DFDs)

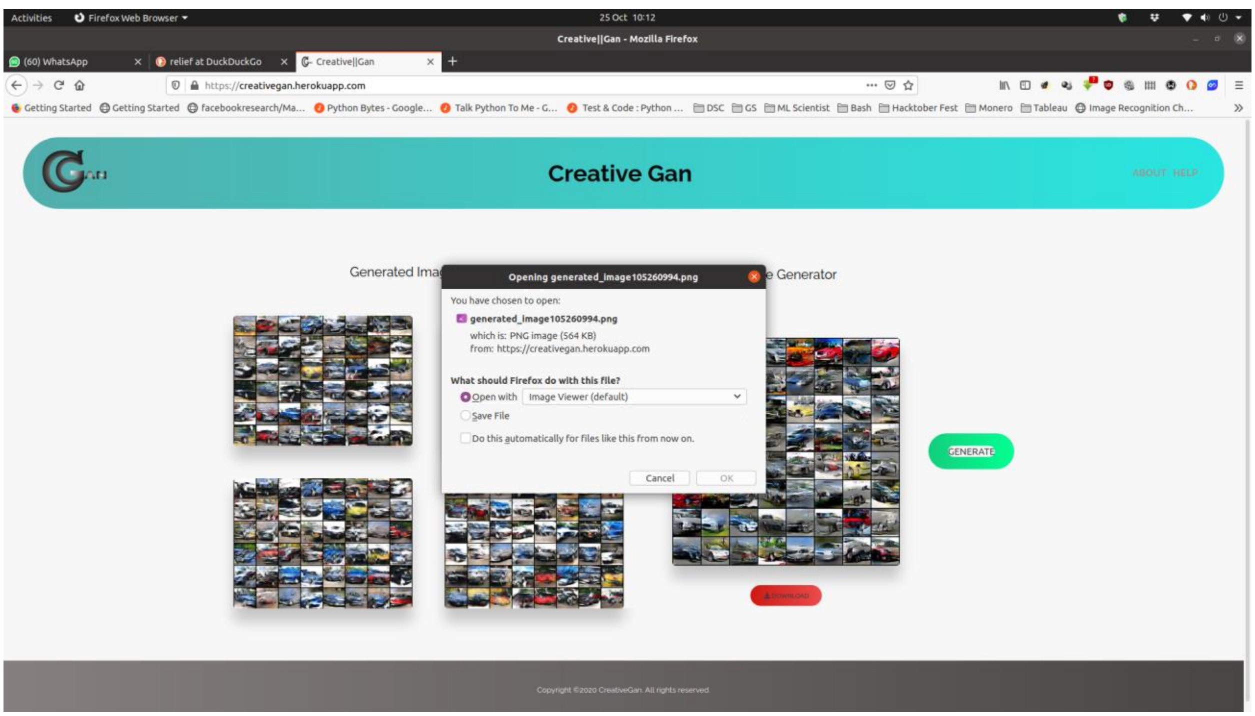


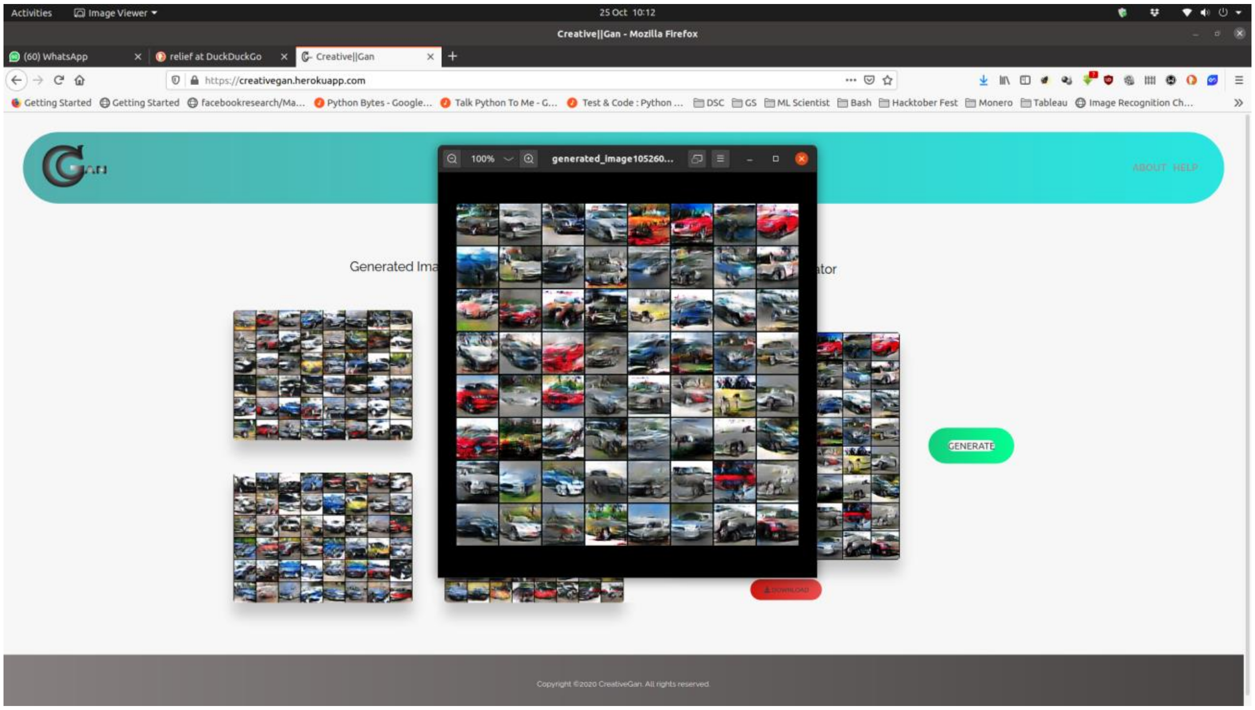


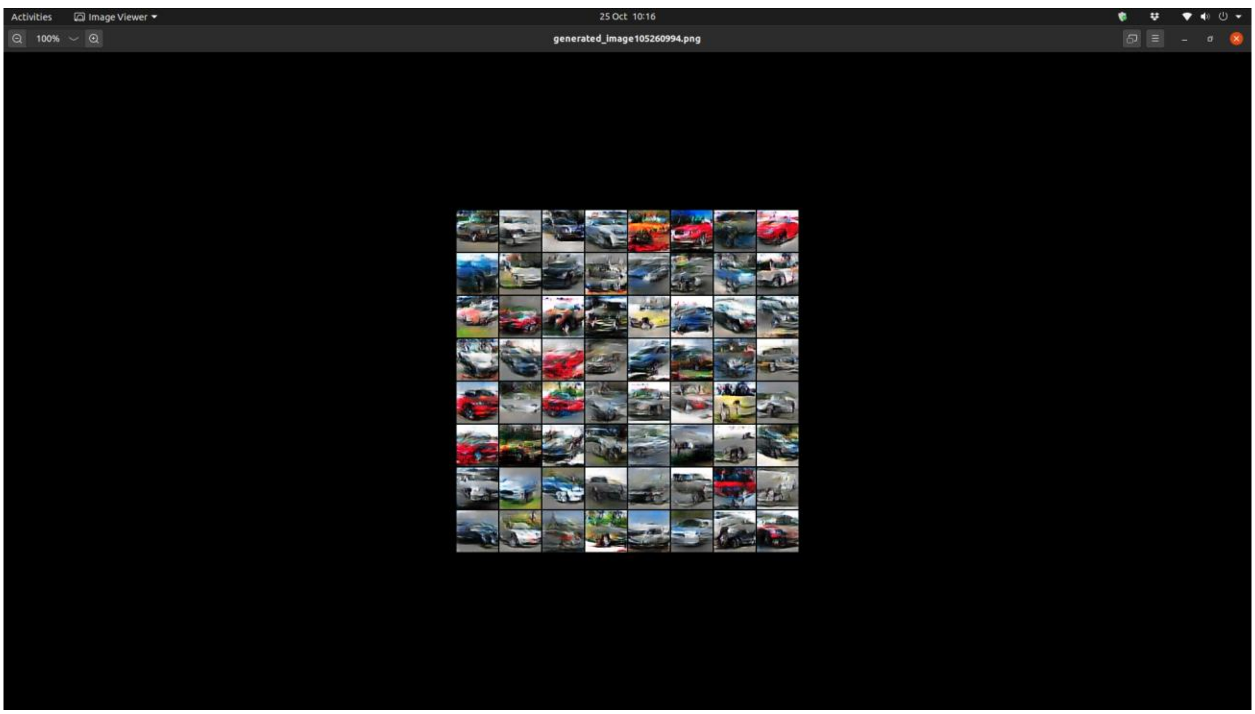
4.2 PROJECT SCREENSHOTS




Creative Gan
Generated Images Gallery
Image Generator
GENERATE
Copyright ©2020 CreativeGan. All rights reserved.


Creative Gan
Opening generated_image105260994.png
You have chosen to open:
generated_image105260994.png
which is: PNG image (564 KB)
from: https://creativegan.herokuapp.com
What should Firefox do with this file?
Open with Image Viewer (default)
Save File
Do this automatically for files like this from now on.
Cancel
OK
GENERATE


Creative||Gan - Mozilla Firefox
https://creativegan.herokuapp.com
generated_image105260...
GENERATE
Copyright ©2020 CreativeGan. All rights reserved.


generated_image105260994.png

CHAPTER 5 – RECOMMENDATIONS AND CONCLUSION

5.0 INTRODUCTION

This chapter presents the recommendations and conclusions for the research on Generative adversarial networks. It also provides information on the application areas of generative adversarial networks in real world data modelling.

5.1 RECOMMENDATIONS ON FURTHER STUDY

Generative adversarial networks are still a new area in machine learning having been discovered less than ten years ago. Therefore, the available research is not quite exhaustive, in depth study is required so as to analyze leverage the application of GANs as a great branch of artificial intelligence. In particular study should be intensified on understanding how recurrent neural networks can be leveraged and integrated into generative adversarial networks to increase their capability and functionality. The integration of recurrent neural networks into generative adversarial networks produces recurrent generative adversarial networks which are in particular applicable in recommender systems. This project worked on the implementation of GANs to come up with unique car concept designs.

5.2 RECOMMENDATIONS ON THE APPLICATION AREAS OF GANS

Having looked at the foundations of generative adversarial networks and how they are designed, the following are some of the recommended application areas of GANs.

In his original paper Ian Goodfellow GANs were used to generate new samples for the MNIST handwritten data set to classify different numerical digits. This has also been applied to generate sample images for other datasets.

Generate photographs of Human faces. Datasets of people's images have been used to generate several images of the same person which remarkably look real.

Generative adversarial networks have been applied to generate real cartoon characters.

Generate new poses for models.

CONCLUSION

Machine learning is a field that will greatly impact how artificial intelligence is applied to provide real world solutions. Generative adversarial networks are one of the major techniques used to implement unique concept designs. After understanding the design of generative adversarial networks, we successfully implemented a generative adversarial network model to generate unique images of car designs using the cars image dataset. We deployed the model on a website platform.

REFERENCES

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541-551.

Djolonga, J. (2017). torch-two-sample Documentation.

Jaiswal, A., AbdAlmageed, W., Wu, Y., & Natarajan, P. (2018). Capsulegan: Generative adversarial capsule network. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 0-0).