Paper

by Daksha Singhal1

Submission date: 02-Dec-2020 03:40PM (UTC+0530)
Submission ID: 1462338797
File name: Daksha_singhal.docx (1.05M)
Word count: 6655
Character count: 36518

Dissertation on

Intelligent Fashion Generation and Retrieval System

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE17CS490A – Capstone Project Phase - 1

Submitted by:

Kavya Khatter	**PES1201701136**
M Adithya Vardhan	**PES1201700788**
Daksha Singhal	**PES1201700847**

Under the guidance of
Prof. KS Srinivas
Professor
PES University

August - December 2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

Intelligent Fashion Generation and Retrieval System

is a bonafide work carried out by

Kavya Khatter	**PES1201701136**
M Adithya Vardhan	**PES1201700788**
Daksha Singhal	**PES1201700847**

in partial fulfilment for the completion of seventh-semester Capstone Project Phase - 1 (UE17CS490A) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period Aug. 2020 – Dec. 2020. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th-semester academic requirements in respect of project work.

Signature	Signature	Signature
Prof. KS Srinivas	Dr Shylaja S S	Dr B K Keshavan
Designation	Chairperson	Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

DECLARATION

We hereby declare that the Capstone Project Phase - 1 entitled **Intelligent Fashion Generation and Retrieval System** has been carried out by us under the guidance of Prof. KS Srinivas and submitted in partial fulfilment of the course requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester August – December 2020. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES1201701136	**Kavya Khatter**
PES1201700788	**M Adithya Vardhan**
PES1201700847	**Daksha Singhal**

ACKNOWLEDGEMENT

I would like to express my gratitude to Prof. KS Srinivas, Department of Computer Science and Engineering, PES University, for his continuous guidance, assistance, and encouragement throughout the development of this UE17CS490A - Capstone Project Phase – 1.

I take this opportunity to thank Dr. Shylaja S S, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan, Dean of Faculty, PES University for his help.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro-Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

The fashion industry has always been one of the most lucrative markets in terms of products and design. Every designer is always looking for new ways to come up with ideas and inspirations. Every new design and concept could be a new trend in fashion. In this paper, we discuss the applications of Generative Models to train and develop a vast array of fashion apparel, given an adequate dataset. Characteristics such as texture, shape, design and material are some of the parameters that we take into consideration while creating variations in the output. Features, specific to a designer, can be achieved by loading the previous works of the designer into the model. We will cover the various methodologies used to achieve the output and the best solution that we found. Such applications would find an interesting use case in designer boutiques such as H&M and ZARA as well as existing online boutiques such as SHEIN.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

Every item of clothing we wear, from the turquoise sweater we wear in the cold mornings to the checkered-pattern shirt we would wear to an office outing, is made by designers spending indescribable periods of time picking the best combinations of colors, textures and patterns to bring out the expression that is engendered within the apparel. These fashion designers compete in a highly competitive industry to make a mark and bring in a customer flow towards their own designs. One of the main challenges faced by fashion houses is the constant innovation and creativity that is demanded from the fashion industry that changes at every whim of the customer-What was in trend this summer, might be yesterday's clothing to the new trend that rose up in the interim. A strong intuition of the market trend along with a constant flow of creative thinking is imperative to stay afloat in the industry.

It is therefore easy to deduce that machine learning models are actively researching on methodologies for image generation of clothing and designs. Research in this field usually overlooks generation of images based on the brands and fewer developed in operational capacity to the actual apparel production. Instead, more emphasis is developed to increase the comfort of users or improve efficiency in the user's perspective.

CHAPTER-2

PROBLEM STATEMENT

In the fashion industry, every designer has a unique taste, a unique sense of style and a designated aesthetic to their products. To deliver such a large variety of designs, fashion designers spend months together in putting up a montage suited to a particular theme in the time and place and are compensated accordingly. But what if we could cut short that process by automating the process? Study the previous designs of the designer and produce a large variety of apparel that can be selected to help push the process of elimination.

Using variations of generator models, we aim to develop a model that takes in several inputs varying in colour, shape, design and texture, then used to create randomized outputs of different textures and colours which bear a similar resemblance to the original works of the designer but also new combinations the designer could contemplate in order to gain a better insight.

To implement this, we propose to use an improvised version of GANs to provide an unsupervised separation of high-level attributes of the generated images. The generated images have been known to suffer from low-resolution, a problem to which we find a solution in the application of Super-Resolution GANs.

The application of such a product lies in many fields within fashion. Besides the proposed use case in apparel, implementations lie in designing of large draperies, accessories and jewellery.

CHAPTER-3

LITERATURE SURVEY

In this chapter, we present the current knowledge of the area and review substantial findings that help shape, inform and reform our study.

3.1 Generating varied fashion designs using GANs

3.1.1 DeepWear: a Case Study of Collaborative Design between Human and Artificial Intelligence[1]

Natsumi Kato, Naoya Muramatsu , Hiroyuki Osone, Yoichi Ochiai

Through this research paper, the author presents a novel technique to generate images using DCGANs. This approach may be used within our application to aid designers in generating designs tailored to their taste and needs.
The model takes brand-specific clothing designs in various categories, extracts the features of the apparel, generates images that resemble the clothes, and thus provide inspiration to clients using them.
The benefits of the DeepWear model is:
i) Useful to learn unsupervised image representations
ii) Patterns may be extracted from one image and superimposed onto another image
Hypothesis: Segregating the products based on brands

3.1.2 A Style-Based Generator Architecture for Generative Adversarial Networks[2]

Tero Karras ,Samuli Laine, Timo Aila NVIDIA

This paper uses concepts similar to the work presented in style transfer literature to pose a distinct architecture for a GAN network which helps in developing an unsupervised separation of high-level attributes as well as a stochastic variation in generated images , while enabling scale-specific control of the creation.

In the process of measuring interpolation quality and disentanglement in a quantifying manner, the paper posits new methods applicable to any architecture specific to generators.

Hypothesis: In order to control distinctive visual highlights, we use the Mapping Network in the hope of encoding an input vector into an intermediate vector whose elements could perform the required action.
However, converting the vector is no easy step as there are limitations to the process of controlling the visual highlights of the input vector, relating to the fact that it should necessarily follow the input data's probability density. For instance, if there is a prevalence of people with round noses in the training dataset, then a larger number of input values would be mapped to that feature. Thus, the model does not possess the ability to map components of the input to the features(vector elements).

For the sake of maintaining the quantitative nature of their argument regarding feature separation, the paper discusses the following novel ways to quantify feature disentanglement:
Perceptual path length - estimating the VGG16 embeddings of successive images while introducing irregular information sources. Huge changes infer that various highlights have changed together and they may be trapped.
Linear Separability -a measure to coordinate better characterization into the model to consider higher distinguishableness of the highlights. For example, classifying inputs into classes such as male or female.
Furthermore, they also introduced a high-quality, new dataset of human faces commended for it's high variation.

3.1.3 Progressively Growing GANs for improved Quality, Stability and Variation [3]

Alec Radford, Luke Metz, Soumith Chintala

Through this research paper, the author presents a novel technique for generative adversarial networks (GANs) wherein the standard philosophy is to train the discriminator and generator in a progressive style.
Moving Forward from low resolution images i.e. 4x4, PG GAN architecture adds new layers that characterize progressively fine subtleties as preparing advances.This process improves the speed of the model while vastly improving the stability of the model, that allows the model to cultivate high quality images.
Furthermore, it allows us to achieve a significantly higher inception score by increasing the variety of the generated designs.Meanwhile, the paper also talks about multiple details regarding how the model doesn't encourage anything besides healthy rivalry between the discriminator and the generator. In its conclusion, the paper prescribes new metrics based on variation and quality of the image, to evaluate the results of the GAN.

Hypothesis: Most GAN networks fail to optimize the finer texture details of images since they aim to grasp the large scale details of the picture as well as the finer details simultaneously. These models would then require large amounts of computation power and time before it could optimize the finer details of all images at an acceptable threshold.
This paper aims to overcome this requirement of significant computation time by spending the first few iterations learning the large scale features of the image,such as the clothing structure and the body outline, and then adding finer detail layers to the model. This approach significantly cuts down the number of iterations/epochs needed for generating acceptable images.

A few approaches the paper employs to further their evaluation are:
MiniBatch Standard Deviation - Salimans et al.(2016) suggest an innovative solution called "Minibatch Discrimination" in his paper, which provides a solution to the GAN model's tendency to capture only a part of the training dataset's variation.

Pixel-wise Feature Vector Normalization— To reign in the magnitudes in the generator and the discriminator within healthy competitive parameters, we normalize the feature vector within a pixel

to a unit length after each convolutional layer within the generator.

3.2 Improving the resolution of generated images

3.2.1 Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network [4]

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi

The behaviour of optimization-based super-resolution methods is mainly driven by the choice of the objective function.
Minimization of the mean-squared error(MSE) and the maximization of the peak signal-to-noise ratio(PSNR) which are defined on the basis of the pixel-wise differences in the image. While these images obtained from these methods possess higher PSNR ratios, they are not perceptually satisfying.

Hypothesis: To side-step the problem of dropping finer texture details in Super-resolution at larger upscale factors, research has recently been focusing on reducing the MSRE. Consequently, the resulting estimates often lose out on the more frequent details and tend to be perceptually unconvincing, in spite of having higher PSNRs.

3.2.2 Generate high-resolution images with Generative Auto-Encoder [5]

Abhinav Sagar

An innovative solution of using a hybrid Variational Auto-Encoder model along with the Generative Adversarial Network to create high-resolution images. This paper uses the results of a generator as well as an encoder to form the data inputted into the discriminator to identify and train for authenticity. This bypasses the pitfalls that GANs and VAEs encounter while running by themselves.

3.2.3 On Single Image Scale-Up Using Sparse-Representations[6]

Roman Zeyde, Michael Elad, and Matan Protter

Given an obscure and down-scaled image, the objective of the methodology outlined in the paper is to extrapolate the high-quality version of the same. Since this problem is poorly represented, a

previous example is needed in order to regularize it.
The changes implement a significant simplification of the process in terms of the architecture exercised by the algorithm, by utilizing a varied approach to training the dictionary -pair and bringing in operational ability without a training-set, as well as the computational complexity.

Hypothesis: Given a noisy image, the literature posits different approaches to generate the detailed-Scaled up version of the image ranging from simple linear space-invariant interpolation schemes, to spatially-adaptive and non-linear filters of various sorts.

CHAPTER-4

DATA

Chapter serves to describe the data under consideration. Understanding the ways in which all the data work together is vital in the process of creating a good solution to the problem at hand.

To train and test our model, we were required to use varying datasets at different steps.

Training Images for Bounding Box

For the pre-processing steps, Deep Fashion dataset is used which is a large-scale clothes database and is described with rich information of clothing items. Every design image in this collection is marked with bounding box and clothing landmarks, as well as labelled with 50 categories and 100 descriptive attributes.

Training Images for GAN Models

In the process of training the model for the purpose of generating images, we use the Kaggle Product Images dataset. Since the training of the image has to be done on a particular apparel category, the number of images per category are less. Generated images are expected to be of high quality and to do so 10K images of each category were downloaded and annotated from google image search using certain keywords. Very less percentage of the resulting images do not contain persons, i.e., only clothing items.

CHAPTER-5

SYSTEM SPECIFICATION AND PROJECT REQUIREMENTS

5.1 Project Scope

The fashion industry has always been one of the most lucrative markets in terms of products and design. Every designer is always looking for new ways to come up with ideas and inspirations. Every new design and concept could be a new trend in fashion. In this project, we will discuss the applications of Generative Models to train and develop a vast array of fashion apparel, given an adequate dataset. Characteristics such as texture, shape, design and material are some of the parameters that we take into consideration while creating variations in the output. Features, specific to a designer, can be achieved by loading the previous works of the designer into the model.

Such applications would find an interesting use case in designer boutiques such as H&M and ZARA as well as existing online boutiques such as SHEIN.

5.2 Product Perspective

Every organisation in the fashion industry invests large amounts of time and money into creating the best designs in a fashion that, in turn, generates the best traction in that season. Designers spend months on creating a sketch of all the different designs to apply, the different textures that would bring out the best details in the design.

Our application objective is to aid in the process, in that we provide a vast array of different kinds of designs that could be possible from a set of parameters relating to the features of the dress.

5.2.1 Product Features

The major features of the product are:

1. Generator model with a capability to produce large amounts of images based on the features specified by the user.
2. HD images created by a deep-learning model allowing the user to gain a better insight into the designs created by the generator model.
3. Feature-based image segregation allows the user to access different apparel created by the generator model based on the features of the apparel, including but not limited to, sleeve length, apparel type and formal/informal wear.

5.2.2 User Classes and Characteristics

Based on the features of the product and the objective aimed at with the product, the major classes of users fall under :

1. Large companies: Companies that invest heavily into the designs coming out into their product line depend on their army of fashion designers to keep up with the demand and the dynamic market. Our model would help them focus on the features they like to repeat in their products while bringing out new features into the apparel.
2. Boutiques: Medium to small enterprises that pride themselves in possessing designs unique to their brand would be encouraged to use a tool that upholds that value while offering them a vast array of choices in different products.

5.2.3 Operating Environment

Our product would operate in a windows system environment (Window 10). Due to the significant processing power required for the different models specified in the plan, we would require systems with GPU-enabled CUDA systems.

5.2.4 General Constraints, Assumptions and Dependencies

The general constraints posed by this product are:

- the size and features of the dataset entered
- The cloud platform used to run the model should have the requisite computing power

5.2.5 Risks

The problem usually faced with GAN based generator models is creating low-resolution

images. To match our users' expectations of the quality of output, we must create images with high resolution and attention to detail. Another significant obstacle to overcome is the lengthy computation required to bring out acceptable images from the model. Priority should be given to faster computation and cleaner code.

5.3 Functional Requirements

The functional requirements of our product are:

- The data entered into the model will be pictures selected from the client for the optimal designs required by the client.
- The generated images will be stored in a secure database to allow for sharing with people with authorised access, when necessary.

5.4 External Interface Requirements

5.4.1 User Interfaces

The model shall provide a very intuitive and simple interface to the fashion houses so that they can upload their fashion apparels like shirts and pants, and the model can generate new fashion images that stand on the same line of the particular fashion company. The user is allowed to upload more than a certain number of images and particular resolution such that the images produced by the model have an expected degree of resolution. The time taken by the generator model to generate images would take not more than a week but may differ given the resolution and number of images. The UI will pop an error message if the number of images provided by the user is less than a certain threshold and the resolution is lower than set by the admin.

5.4.2 Hardware Requirements

The server is a high GPU based system to run the GAN model in order to generate images faster with higher resolution. The client-side requires a browser such as Chrome which supports HTML & Javascript where they need to ready the design and upload it on to the server. Web frameworks such as Django will be implemented to link the client and server where the client can upload his choice of apparels and the server can then generate new fashion clothes and send it back to the client.

5.4.3 Software Requirements

Anaconda

- A tool that helps ease the package management and deployment processes, Anaconda is a non-proprietary software distribution of Python programming language for data science tools.

- Version / Release Number:2020.02
- Operating Systems: Windows 10, Ubuntu, IOS
- Tensorflow, PyTorch

Windows

- OS developed by Microsoft Corp. to allow for GUI access and operations to files and resources within the system.
- Version / Release Number:10

Database

- For the purpose of this project, we will be using SQL, a Relational DataBase Management System manipulation language to perform basic operations such as Creation, Insertion, Update and Delete.
- Version / Release Number:15
- Operating Systems: Windows 10, Ubuntu, IOS

5.4.3 Communication Interfaces

The communication from one interface within the framework to another is important to the product we are developing since they depend on each other. However, the approach adopted isn't significant for the framework and can be taken care of by the hidden frameworks working with the online interface.

5.5 Non-Functional Requirements

5.5.1 Performance Requirement

Usability: An easy-to-use interface is preferred in this case since the majority of our customers lie in the non-technical domain. Training and running the model on the images provided should be effortless and coordinated. The GUI should also be one helpful to the user to understand everything he needs to do.

Reliability: This software will be developed with machine learning, feature engineering and deep learning techniques. Likewise, the information given by fashion houses will be utilized to contrast the outcome and measure dependability. With current AI techniques, the user gained information should be enough for dependability if enough information is acquired. When admins want to update, it takes as long as upload and updates time of executable on server. The users can be reached and use the program at any time, so maintenance should not be a big issue.

Performance: We estimate that the process of training and generating would not take more than a week. These estimates vary significantly based on the response time and the size of the datasets provided, given that the performance of the servers are up to the task as well. The certain session limit is also acceptable at early stages of development.

5.5.2 Safety Requirements

In the event of extensive damage to a large portion of the database, requisite information should be stored in a separate location, preferably in a secure environment as well. This data should be easily recoverable and operable.

5.5.2 Security Requirements

Our application provides areas of security which impose requirements such that the new images generated by our model are not leaked as well as the images provided by the fashion houses are also protected. Therefore, any stored user credentials are encrypted to secure their data and no plaintext account information is stored on our application.

CHAPTER-6

System Design

This segment portrays plan objectives and contemplations, gives an elevated level outline of the framework engineering, and depicts the information configuration related with the framework, just as the human-machine interface and operational situations

6.1 High-Level System Design

6.1.1 Logical user group

Based on the features of the product and the objective aimed at with the product, the major classes of users fall under :

1. Large companies: Companies that invest heavily into the designs coming out into their product line depend on their army of fashion designers to keep up with the demand and the dynamic market. Our model would help them focus on the features they like to repeat in their products while bringing out new features into the apparel.
2. Boutiques: Medium to small enterprises that pride themselves in possessing designs unique to their brand would be encouraged to use a tool that upholds that value while offering them a vast array of choices in different products.

6.1.2 Project Components

We have identified the following project components of our product:

1. Client Tier: The User Interface of our product, enabled through a website. This component allows the client to interact with our model and develop brand-specific designs without gaining access to other privileged information.
2. Data/Storage Tier: This component is tasked with storing information coming from the client as well as the Processing Unit. Tasks also include validating login information and storing time-specific task data.
3. Processing Tier: This component is the crux of our product. Containing the model ready to be trained for brand-specific apparel.

6.1.3 Application Components

The application components are arranged based on the project components they are a part of, as the following

6.1.3.1 Client Tier

The client tier contains applications that allow the client to gain seamless access to our product

1. Login
2. Submit designs
3. View designs
4. Text-Based design retrieval

6.1.3.2 Data Tier

The data tier contains applications that allow us to design at a redundant capacity.

1. Store input designs
2. Store generated designs
3. Validate logins
4. Store task data

6.1.3.3 Processing Tier

The processing tier consists of applications that interact with the model

1. Train model on brand-specific designs
2. Generate designs

6.1.4 System Functionalities

6.1.4.1 Login

Functionality to allow users to gain access to their designs and personalized data

Figure 1: Interaction Diagram for Login

6.1.4.2 Train Model

Functionality for users to input brand-specific designs to the model and train it to generate the most accurate designs

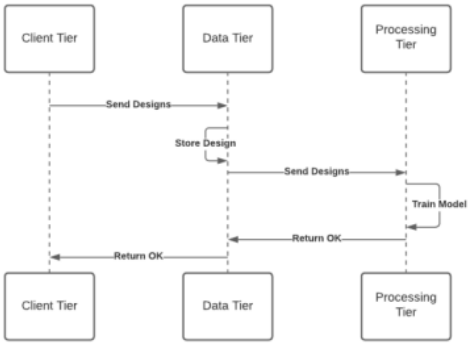


Figure 2: Interaction Diagram for training the model

6.1.4.3 Generate Designs

Functionality to use the trained model to generate new designs specific to the brand

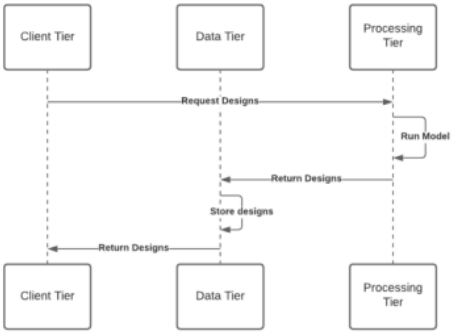


Figure 3: Interface Diagram of Generating Designs

6.1.4.4 Text-Based design retrieval

Functionality to retrieve designs based on high-level features

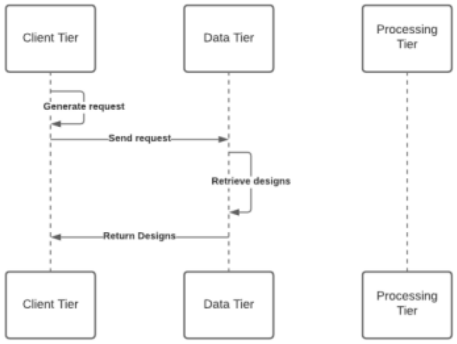


Fig 4: Interface Diagram for Text-Based Retrieval

6.1.5 Security

Security is an essential part of the system. Designers working with our product would be working within extremely private workspaces accessible only to them. Designs would be allowed to be shared/exported only by direct collaborators of the project.

6.2 Design Description

The main design features include 4 different parts: the Client UI, the database, the admin and the Processing component. Each component would be modularized efficiently for better productivity and reuse.

The Client UI component resides within the systems of our end users. It would include the interfaces for creating projects, tasks and displaying data, as well as providing an interface to view, share and export the designs from our model.

The Database component would be stored on a cloud platform. This would allow for better transferability and redundancy. The database component would allow for efficient storage and retrieval of images/data. The database component would store designs inputted from the model, the designs generated from the model and the user information.

The Processing Component would also be executed upon a cloud service. This would allow for better access to the required computation at significantly lower costs. The Processing component would allow for easy access to the model along with easy retrieval of designs generated.

Finally, we have the admin component which oversees the other components and generates reports for the same.

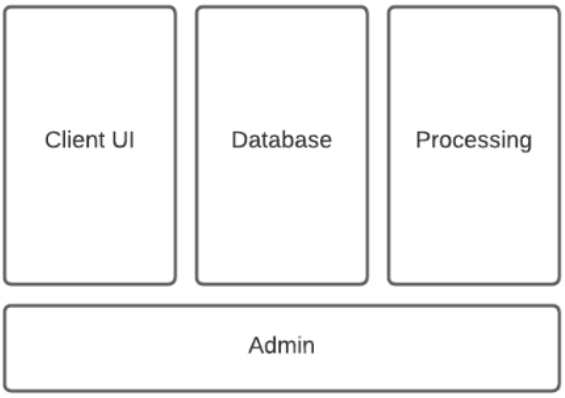
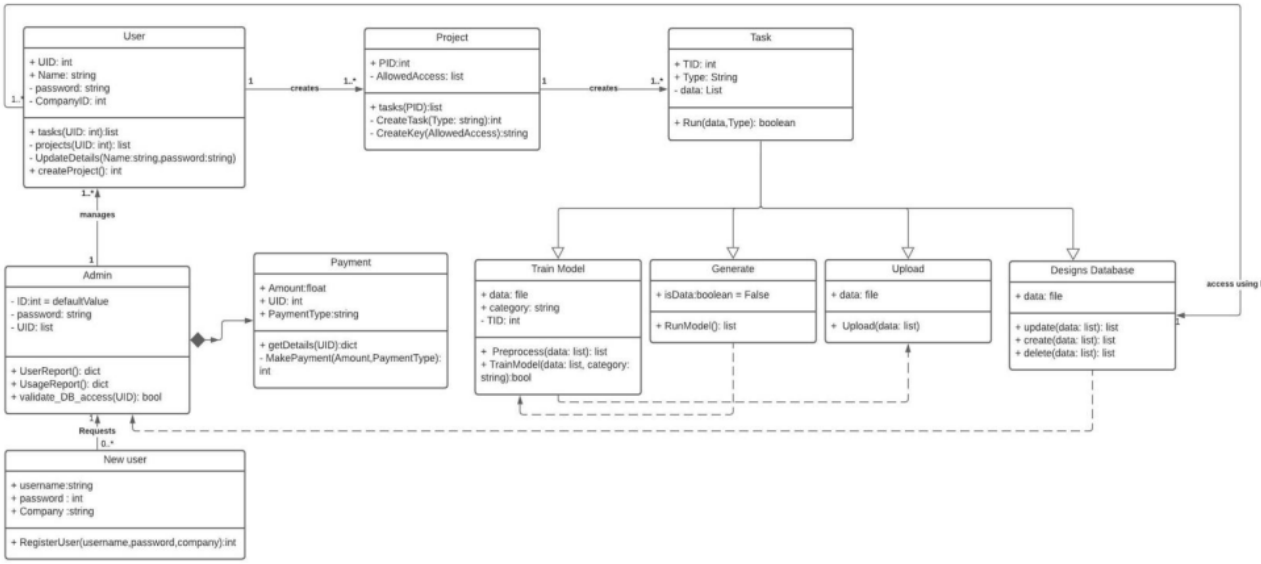


Figure 5: Design Interface Diagram



6.2.1 Master Class Diagram

Figure 6: Master Class Diagram

6.2.2 Reusability Considerations

For considerations regarding reusability of code and methodologies, we have:

- Singleton design pattern for database access and better security
- Decorator design pattern for different types of tasks assigned

Activity Diagram

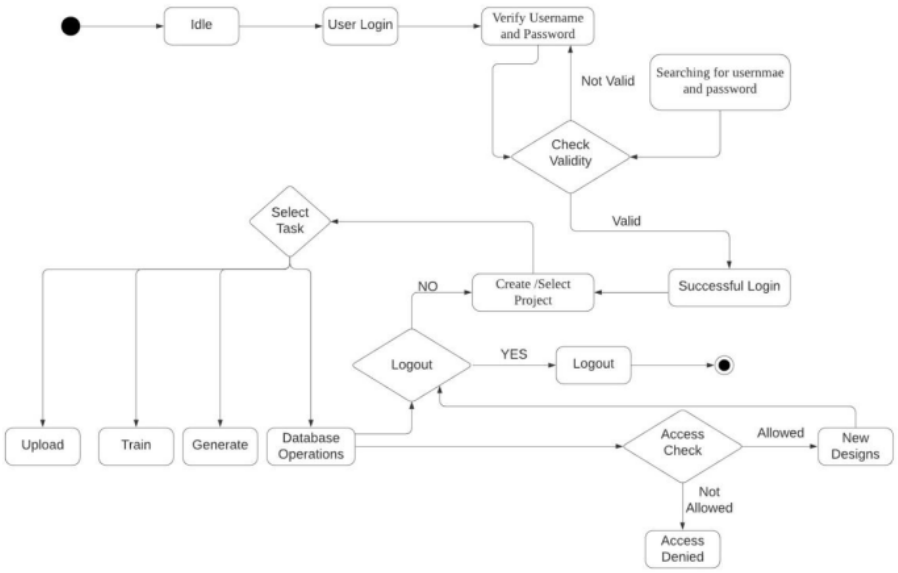


Fig 7: Activity diagram

User Interface Descriptions

Login.html - the first page of the UI, where the user logs into his profile

FrontPage.html - Display the different projects and tasks currently run by the user

Project.html - Display the features of a particular project and all the designs generated

Tasks.html - Display the list of currently running tasks as well as previous tasks run

Train.html - Page to upload images to train the model

Generate.html - Page to load images generated by the model

Admin.html - admin page to view user traffic and change settings for users

Report Layouts

Synchronous Reports would be generated in a periodic fashion, to describe user interactions as well as the tasks performed by the Processing unit.

User reports would include, but not be limited to, update on the number of users, average product utilisation per user, the median frequency of use and other usage patterns.

Task reports would include, but not be limited to, the number of tasks executed in unit time(week/month/year), size of the task executed, time is taken for the task, specifications of the task and any inconveniences during task execution.

CHAPTER-7

Implementation And Pseudocode

This chapter describes the implementation of details of various approaches to build the fashion generation system.

7.1 Data preprocessing

Classify the images into predefined categories

Given input images of different fashion styles, we have built the model with 3 layers (1 layer to flatten the image to a 28 x 28 = 784 vector, 1 layer with 128 neurons and Relu activation function & 1 layer with 10 neurons and the Softmax function). We have 10 Neurons because we have 10 labels for the image data set.

T-Shirt (0)	Trouser (1)	Pullover (2)	Dress (3)	Coat (4)
Sandal (5)	Shirt (6)	Sneaker (7)	Bag (8)	Saree (9)

Table 1: Label for Clothing Classification

Input type: Image

Output Type : Category Number(0-9)

Bounding Box approach

An object detection model was built using YOLO architecture that takes real-world images and detects and classifies apparel. Thus, using this model we detected the apparels in the images and cropped the specific clothing item and stored in the database.

7.2 Modeling

This section describes the model incorporated to generate high-resolution apparel images from the preprocessed dataset.

The underlying architecture for the generation of fashion images is generative adversarial network (GAN) architecture. There are primarily 2 parts of generative adversarial network model :

- The **generator** is a function that learns to generate synthetic data from random data in terms of a probabilistic model.
- The **discriminator** takes the generated image from the generator as negative samples and then learns to differentiate the generator's fake image from the actual image. The discriminator penalizes the generator for producing implausible results.

7.2.1 DCGANs + SRGANS

DCGANS

A DCGAN is an annexe of the vanilla GAN besides its use of convolutional layers in the discriminator and convolutional-transpose layers in the generator. Three major changes are made in the DCGAN architecture that is different from CNN architecture.
First, Deterministic spatial pooling functions with stridden convolutions are replaced by All Convolutional Net letting the generator to acquire its own spatial downsampling.
Second, the fully connected layers such as the global average pooling are eliminated that is on top of convolutional features as it reduces the convergence speed greatly. Concurrently, the foremost GAN layer receives as input which is a uniform noise distribution and outputs a four-dimensional tensor to be used as the start of the convolution stack. In the case of the discriminator, the final convolution layer is flattened and fed into a single sigmoid yield. Finally, both the discriminator and the generator undergo batch normalization which steadies learning by normalizing the contribution to every unit to have zero mean and unit difference and blocks the generator from deteriorating all examples to a solitary point.
Within the generator, most of the activation functions fall under ReLU activation excluding the output layer which uses a Tanh function. In the case of the discriminator, it uses Leaky ReLU activation function for all instances.

SRGANS

Generative Adversarial Networks are extremely strong structures to be used in developing images of high quality while also being perceptually satisfying. The model outlined in the paper uses the concept to form a perceptual loss function for Super-Resolution photo realistic. This model contains three important components to be discussed, the perceptual loss employed and the changes made to each of the Generator and Discriminator.
Two losses, the adversarial loss and the content loss, together form the perceptual loss function proposed by the paper, in the attempt to train the generator G into fooling the discriminator D which is trained to distinguish the real images from the generated ones.
The perceptual loss is calculated by computing the two parts of the loss relevant to maintain the perceptual clarity of the image, content loss and adversarial loss. Rather than relying on the popular loss at a pixel-level, the authors of this research define a loss based on the ReLu activation layers called VGG loss. The adversarial loss motivates the generator in approaches that help fool the discriminator, by resorting to images that border at the edge of the natural images.

At the core of the Generator, networks are a certain number (B) residual blocks of the identical layout. The representation of the architecture is presented in figure 1. Our approach outlines the use of two convolutional layers with 64 feature maps and miniature 3x3 kernels followed by a layer performing batch-normalization and finally, an activation layer with ParametricReLu as the activation function.

To distinguish between the Super-Resolution images generated by the generator G and the real high-Resolution images, we use the Discriminator model of the GAN architecture as shown in the figure. Similar to the generator model, we employ LeakyReLu activation but we circumvent max-pooling throughout the network. This model consists of 8 convolutional layers with varying number of third-degree filter kernels, each greater than the previous by a factor of 2, starting at 64 and working itself all the way upto 512 kernels as in the network specified previously as the VGG network. The 512 feature maps that arrive as output to this model are then followed by two dense layers and a sigmoid activation function. This promotes perceptually improved solutions found in the context of the real high-resolution images.

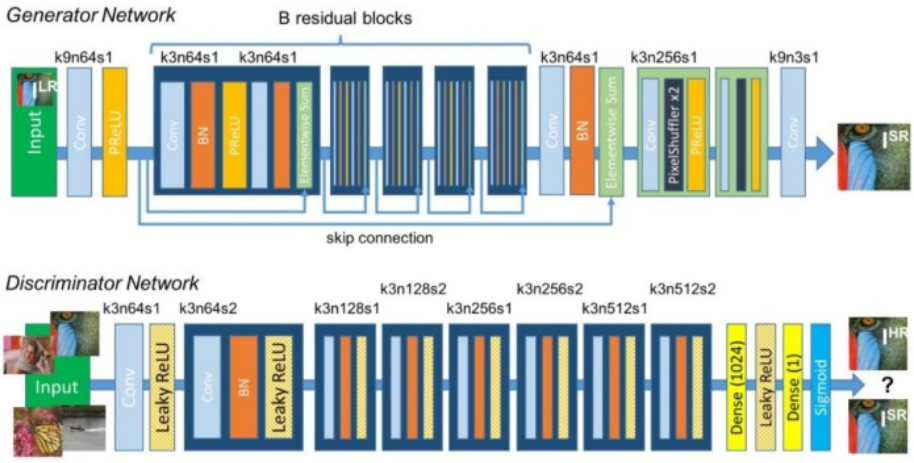


Figure 8: Architecture of DC-GANS

7.2.2 Progressive Growing GANS

The PgGANs is built on the premise of initializing at very small resolutions such as 4x4 and 8x8, and further developing at each stage to comprehend higher resolutions until the requisite resolution is achieved. This process is achieved in two steps, namely the fading in of new blocks to support the higher resolution after each stage and the finetuning for the same.

The Progressive Growing GANs is a well-known method in developing 1024x1024 resolution images which we may also use in the generation of fashion apparel for our use case. PGGANs incrementally enhances the size of the image within the generator, starting at a small resolution such as 4x4(as required by the user) and working itself up to a desired resolution

such as 512x512 or 1024x1024. The model also includes a fine-tuning process at regular intervals between generations followed by repetitions of merging in the new blocks into the generator since all the blocks within the model are open to training during the entire process. The process of appending convolutional layers to the model allows both the generator and the discriminator to understand the coarse-level details and later finer-details, permitting the model to discover the bigger structure details of the distribution of the picture and then adjust the attention towards the increasingly finer details of the higher-resolution images, in contrast to the usual practice of learning both scales at the same time.

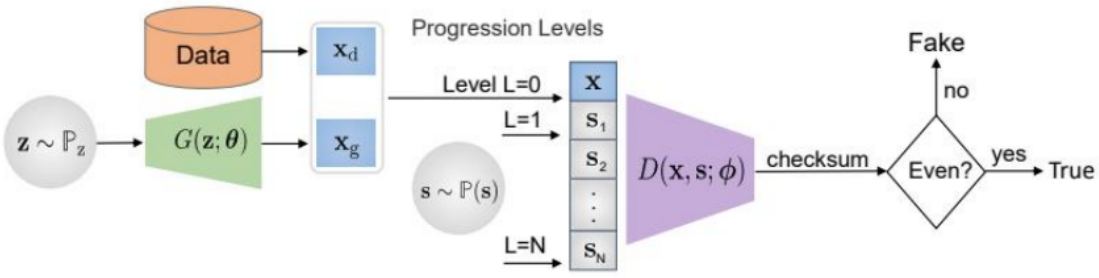


Figure 9: Architecture of PG-GANS

CHAPTER-8

Conclusion of Capstone Phase - 1

We have successfully built an application where the client can request for new designs by inputting his original design images. For generating new designs we have implemented 2 GAN models i.e. DC-GANs and P-GANS, both provided us with the desired results given the good quality adequate number of images.

CHAPTER-9

PLAN OF WORK FOR CAPSTONE PROJECT PHASE - 2

Integrating Generation of Fashion styles with Dialogue based Retrieval System

We aim to build a user-friendly application where we integrate the fashion generation model with the image retrieval system that would learn to seek natural language feedback from the user and iteratively retrieve the generated images from the database. This retrieval system learns to construct the most optimal sequence of candidate images to obtain the most informative user feedback and automatically learn to optimize the retrieval objective.

Product Testing by real-world customers

It is believed that in order to release products as a service effectively, it is important for us to learn the behaviour of the people as this tool navigates increasingly challenging times. It will strive us to understand users' (in our case fashion houses) behaviour and design our application that help us get closer to their needs.

Releasing our Product as a service

Once the application design and the behaviour of the user align properly, the next step would be to launch the product into the market and make it available for the targeted user for purchase. This would help build the application and expectations for the provided service, collect useful feedback from early users, and create push and momentum and recognition from the target industry for the products.

REFERENCES/BIBLIOGRAPHY

[1] Natsumi Kato, Naoya Muramatsu , Hiroyuki Osone, Yoichi Ochiai , Daitetsu Sato University of Tsukuba,*DeepWear: a Case Study of Collaborative Design between Human and Artificial Intelligence*

[2] Tero Karras ,Samuli Laine, Timo Aila NVIDIA,*A Style-Based Generator Architecture for Generative Adversarial Networks*

[3] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi Twitter,*Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*

[4] Roman Zeyde, Michael Elad, and Matan Protter,*On Single Image Scale-Up Using Sparse-Representations*

[5] Alec Radford, Luke Metz, and Soumith Chintala,Unsupervised representation learning with deep convolutional generative adversarial networks In International Conference on Learning Representations (ICLR), 2016.

[6]Seyedarian Hosseini, Thomas Boquet, Wojciech Stokowiec, Ying Zhang, Christian Jauvin, Chris Pal,Fashion-Gen: The Generative Fashion Dataset and Challenge,Negar Rostamzadeh

[7] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens,and L. Carin, "Variational autoencoder for deep learning
of images, labels and captions," in Advances in neural information processing systems, 2016, pp. 2352–2360.

Appendix A: Definitions, Acronyms and Abbreviations

GPU: A graphics processing unit is a specialized, electronic circuit intended to quickly control and change memory to quicken the making of pictures in a casing support expected to yield a showcase gadget.

CUDA: CUDA is a parallel computing platform and application programming interface model.

NT: Windows NT is a processor-independent, multiprocessing and multi-user operating system.

GAN: A generative adversarial network is a generative model used to generate data using the adversarial method

HTML: Hypertext Markup Language is a markup language for web pages and used in the frontend designed to be displayed in a web browser.

PHP: PHP is a scripting language specially used as a backend language in web development.

Paper

ORIGINALITY REPORT

13%	9%	7%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

4%

★ export.arxiv.org

Internet Source

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Paper

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33