Batik Pattern Synthesis For Virtual Try On Application

BY

Chen Sin Yee

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

June 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Batik Pattern Synthesis For Virtual Try On Application

Academic Session:_____June 2024______

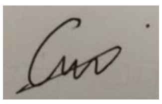
I _________________________CHEN SIN YEE________________________

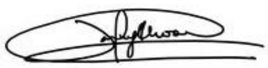
(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,





(Author's signature) (Supervisor's signature)

Address:
___171, Tamen Permai Jaya 34100
Selama,Perak___________________

Tan Hung Khoon
Supervisor's name

Date:_________9/9/2024________ **Date**: 11/9/2024

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **21 JUNE 2011**	Page No.: **1 of 1**

FACULTY/INSTITUTE* OF __Faculty Information and Communication Technology__

UNIVERSITI TUNKU ABDUL RAHMAN

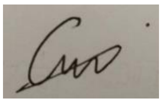
Date:______9/9/2024_________

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***Chen Sin Yee*** (ID No: ***2005680***) has completed this final year project/ dissertation/ thesis* entitled "**Batik Pattern Synthesis For Virtual Try On Application**" under the supervision of___________Dr. Tan Hung Khoon_________(Supervisor) from the Department of_______Computer Science_______________, Faculty/Institute* of _____ Faculty of Information and Communication Technology, and_______Tan Joi San_______________ (Co-Supervisor)* from the Department of_________Computer Science__________________, Faculty/Institute* of__________Faculty of Information and Communication Technology _.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,



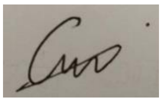
____

(CHEN SIN YEE)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Batik Pattern Synthesis For Virtual Try On Application**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.



Signature : ____________________________

CHEN SIN YEE

Name : ____________________________

9/9/2024

Date : ____________________________

ACKNOWLEDGEMENTS

I was full of gratitude when I presented the project. The journey, which was both challenging and rewarding, culminated in the study of GAN patterns in fabric pattern design, especially in complex batik pattern design. This effort is made possible by the collective support and wisdom of many. I sincerely thank them.

I would like to start by expressing my heartfelt gratitude to University Tunku Abdul Rahman University for the valuable opportunity to participate in such an interesting and innovative field. The research and creative environment provided by the university was inspiring.With great respect and gratitude, I express my deepest gratitude to my mentor, Dr. Tan Hung Khoon. His knowledge and insightful guidance was the compass that guides this project. Dr Tan's patience, encouragement and deep knowledge of reproductive adversary networks were instrumental in the conception and implementation of this project.

My acknowledgment would be incomplete without mentioning the generous financial support and resources. These contributions are essential to the development of the project, enabling in-depth exploration of the unique and complex patterns of batik design.Also, I am eternally grateful to my family and friends for their unwavering faith in me. Their constant encouragement and support has been a source of strength and motivation, reminding me of the joy and passion that lies at the heart of this academic endeavor.

Finally, I am happy to share the achievements of this project, which were achieved through strong dedication and expertise. Thank you all for being a part of this unforgettable journey.

ABSTRACT

Fabric pattern design is a dynamic blend of culture, tradition, and human ingenuity. It serves as a canvas reflecting cultural identity and aesthetics across various societies, conveying stories, beliefs, and a sense of belonging. However, the labour-intensive and time-consuming nature of traditional pattern design hampers creativity and customization.

Generative Adversarial Networks (GANs), a subset of deep learning techniques, offer a promising solution to this challenge. By automating the design process, reducing human intervention, and enhancing the accuracy and realism of generated patterns, GANs bridge the gap between tradition and modernity. However, current GAN-based approaches suffer from inconsistencies, diversity limitations, and control issues.

This study aims to leverage GANs to automate fabric pattern generation while preserving cultural heritage. It seeks to enhance GAN architectures, diversify training datasets, and gradually increase pattern complexity. This research empowers designers to create unique fabric patterns, revitalizes traditional designs, and enriches the fashion industry with culturally rich fabric models. Ultimately, it paves the way for a new era of efficient, creative, and culturally significant fabric pattern design.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

List Of Abbreviations

GAN	Generative Adversarial Network
DCGAN	Deep Convolutional Generative Adversarial Network
WGAN	Wasserstein Generative Adversarial Network
VGG-19	Visual Geometry Group-19
FID	Fréchet Inception Distance

Chapter 1

1.1 Introduction

Designing fabric pattern is an art that can combine the creativity , tradition and human innovation. For instance , the batik and traditional pattern design on fabric in ancient of Chinese. For the designer that involved in textile and fabric pattern design , the activities to produce and generate the complex pattern is like weaving stories on the fabric , where by using color , shape and texture blended together to reflect a particular culture and style. For instance , Batik , a venerable art from that originated in different cultures and is the evidence of the deep connection between fabric and identity. In some country such as Indonesia , Malaysia, batik is not just a cloth , but for them they are a cultural artifact that conveys stories , belief and sense of belonging. Not only that , batik pattern are just a aesthetic choice , but they often reflect the local tradition, social status. Similarly , in ancient China, the fabric pattern design are also important to reflect position in the society. The delicate fabric pattern are often inspired by natural , and spiritual belief , similar to the batik , they are not just decorative , but convey message and emotions. For example the intricate dragon pattern symbolize imperial power , phoenix symbolizes harmony and virtue.

Designer are always forced to constantly come out with a new and attractive ideas on the fabric pattern design , but most of the time being hindered by the creativity and the process of design fabric pattern is too complex as paper[1] mentioned that the process of produce and generate an new idea of fabric pattern design will require the designer to manually extract the interest and attractive style features from the existing patterns , which lead to very time consuming to do those collection. In addition , it is also being mentioned in the paper [6] and [5] that there were lacking of younger and employee to involve in fabric pattern design that blend with traditional pattern. For example in generate and create diverse batik pattern. Batik is an ancients textile art that originated from Indonesia , India[10]. Batik is a complex pattern formed by wax-resistant dyeing. Despite batik has beautiful and traditional pattern that reflect a particular culture and it has been selected by UNESCO as an Intangible Cultural Heritage on October 2,2009.[11], but research indicated fewer people engage in batik pattern creation due to the process itself is labor-intensive process ,time consuming and required skilled people [6].

From that , it is important that a creativity , skills and time become an important factor in producing the pattern design. But , in this century , where technology is becoming more and more sophisticated , fabric pattern design can be generated and created very easily and automatically with the advance of technology. By recent introduced technology – Generative Adversarial Network (GAN), a subset of deep learning , which could be applied in fashion field [7] such as the aim of this research , GAN can have significance advantage in fabric pattern design.

To explain how the GAN can actually being applied in fabric design and how it works, these network consist of two components that are known as generator and discriminator. Where the generator is responsible to generate or craft the pattern from the input images we provided to the generator. This generator also will learn the underlying patterns , style and textures that presented in the input images we during the training process. The discriminator component playing a role to evaluate and differentiate the real fabric pattern from training and fake fabric pattern generated from generator. The result we said is successful when discriminator unable to distinguish that the generated pattern is a fake pattern that generated by generator.

1.2 Problem Statement

The conventional way of fabric pattern creation are characterized by complexity and labor . For instance, designing the fabric pattern is sometime time consuming and difficult for some designer because as a textile or fabric pattern designer they have to collect a lot of samples such as texture , textile pattern , color , shape to get some new ideas to be included in a new pattern design. Which is not effective and sometime difficult to provide customization for customer and difficult to have adequate creative in mindset all the time as being mentioned in the paper [1] In contrast , the emergence of Generative Adversarial Network (GAN) technology offer promise to automate the process with minimal human intervention .It is worth noting that some studies have shown that , GAN is not that perfect in some cases which they were inconsistencies in producing high resolution ,lack diversity pattern, high quality of fabric pattern and sometime resulting in blurring [12] .

In addition , limited pattern diversity poses another obstacle that lead to repetitive and uninspired design as mentioned in [2]. Furthermore , some research also shown that some type of GAN only able to generate the simple pattern , but when there were complex pattern involved , the performance of the model is decreasing such as the cheetah patten , ethnic and geometric pattern as in the paper [3].Furthermore , although there already exist of AI in fabric pattern design which using two patch of different kind of fabric pattern style and to generate a new pattern , but in the research [4] but the resulting output pattern is still uncontrollable, and some of the original batik properties are gone.[4]. Addressing these challenge is essential to fully tap the potential of GAN in fabric pattern design revolution

1.3 Motivation

Therefore , with the problem statement above , it is ideal that apply a method which can help to automatic and enhance the Batik fabric design process and generate a lot of diverse unique pattern . Thus , we delve into the capabilities of GAN that can act as tools to solve the problem , and by focusing on the development of an application that can help to facilitate the automatic generation of fabric pattern , we aim to solve the need for efficiency , diversity and customization. The expected outcome of this research is to create and develop a user friendly application which offer user wide array of unique , culturally inspired Batik pattern and allow the user to adjust and choose the desire fabric pattern design , and enable users to visualize these patterns in real-world application such as the virtual try on feature.

1.4 Project Scope

At the end of this project, a dynamic software platform designed to generate and explore Batik fabric patterns will be delivered. The platform will possess a robust database of Batik motifs sourced from the significant research conducted by [14], who synthesized Batik motifs by utilizing a Diffusion -- Generative Adversarial Network. The curated data will be utilized to train the StyleGAN2 model, thereby enabling the creation of numerous, superior Batik patterns that commemorate the extensive cultural heritage of this traditional art form while introducing novel concepts.

Users will have the ability to interact with the generative model through an intuitive interface, enabling them to customize and combine Batik patterns in accordance with their personal or commercial design requirements. A virtual try-on feature will be integrated into the platform, enabling a realistic simulation of how these designs would appear on various attires and materials. This feature aims to provide instant visual feedback and encourage more interactive and user-centered design exploration. The project aims to create a novel tool that empowers designers, artists, and fabric lovers to invent within the realm of Batik pattern design. It will make it accessible to a broader audience and boost global appreciation for Batik art.

1.5 Project Objectives

The goal of the project is to improve creativity and decision-making for fabric fans, designers, and artists by introducing cutting-edge technology like virtual try-on tools and Generative Adversarial Networks (GANs) into the Batik pattern design process. The three main goals of this project are to combine modern digital innovation with traditional Batik artistry to provide users with a distinctive platform that allows them to create, explore, and visualize fabric designs in a dynamic and interactive manner.

The primary goal of the research is to use StyleGAN2, a potent machine learning model recognized for its capacity to produce high-quality, varied pictures, to generate a broad variety of unique Batik ideas. The project's goal is to create a variety of patterns that faithfully represent the rich cultural legacy and creative diversity inherent in Batik by training StyleGAN2 on a carefully chosen dataset that embodies classic Batik designs. These created patterns will provide users with a wide range of options, as each design is original and distinctive, opening them many creative possibilities. This method allows users to find and play with patterns that go beyond the bounds of conventional art, honoring the traditional art form while also promoting creative discovery. The goal is to make Batik more accessible and appealing in the digital age, fostering a spirit of experimentation and creativity in fabric design.

The second objective is to provide users with a virtual visualization feature for customized Batik patterns. Leveraging VITON-HD, an advanced virtual try-on technology, the project will allow users to visualize how their selected or customized patterns would look on different garments. This will be achieved by overlaying the generated Batik patterns onto images of models, offering a realistic and interactive view of how the designs will appear on various types of clothing and fabrics. This functionality is designed to empower users to make more informed choices about pattern selection and application by enabling them to see the final result before committing to actual production. The virtual try-on feature will enrich the creative process by allowing for real-time adjustments and experimentation, making it more intuitive and engaging for users. It serves as a bridge between imagination and reality, allowing designers and fabric enthusiasts to see their ideas brought to life in a virtual environment, thus enhancing their confidence in their creative decisions.

The third objective focuses on the seamless integration of advanced machine learning models to support creative exploration and user interaction. By utilizing StyleGAN2 for generating diverse Batik patterns and VITON-HD for virtual try-on capabilities, the project aims to create a platform that facilitates artistic exploration and user engagement with machine-generated designs, without requiring any deep technical expertise. This objective is geared towards democratizing access to advanced design tools, making them more widely available to a broader audience. The application is designed to provide artists, designers, and fabric enthusiasts with the opportunity to experiment with different patterns, visualize them in real time, and customize them to their preferences within a user-friendly interface. This combination of technology and creativity opens up new possibilities for fabric design, expanding the potential for innovation in both traditional and digital mediums.

Overall, the project aims to bridge the gap between traditional Batik craftsmanship and digital innovation by providing a dynamic platform that enhances the creative process for users. By generating a variety of unique Batik patterns, offering virtual visualization for customized designs, and integrating advanced machine learning models, the project seeks to open new avenues for artistic expression and innovation in the field of fabric design. The focus is on the practical application of these cutting-edge technologies to provide users with a unique, interactive experience that fosters creativity and supports decision-making, contributing to the evolution of textile art in the modern era.

1.6 Contributions

Our research is very beneficial to the field of design and technology by developing an application that automate the generation of diver and unique fabric pattern design such as the Batik pattern which streamlining the design process and enhance the creativity of the designer while help to retain the traditional pattern by incorporate with the machine learning algorithm. Furthermore with the model we developed and improved , it will be able to develop and application that generate the patten and produce a more realism pattern quality and enhance the diversity generated pattern. In addition which it also can preserve the cultural heritage such as some traditional pattern and generate more unique diverse patterns at the same time. Lastly , with the application we developed it will integrated with a feature that allow users to virtual try on garment with their chosen patterns, improving their experience and provide them better decision-making.

Chapter 2

Literature Review

2.1 Previous Works on GAN

2.1.1 Model Architecture Generative Adversarial network for Fabric Pattern Generation

Generative Adversarial Networks (GANs) is the cutting-edge technology that being introduced in by Ian Goodfellow and his colleagues [14] in their published paper "Generative Adversarial Nets" in 2014. GAN become a popular application used in the fashion industry to create the fabric pattern design or apparel design as it provide automation for the conventional fabric pattern design process.

2.1.2 Image Synthesis /Generation from random noise to generate new fabric patterns

In this paper[3] , the primary goal of the [3] is to utilize the GAN to generate new textile patterns that typically used in fashion and design industries. As to achieve their goal ,the dataset employed in [3] was compiled from various source on Google images. Each images within their datasets has its own unique resolution. The data collected is organized into different categories. In [3] – [12] – [9] all undergo the pre-processing step as to ensure that all the dimension of each images was standardized. For [3] data organized into batches of 64 images with (64,64,3) dimensions, [12]- [9] resizing all the images to a standardized dimension of 64x64.

The model architecture that introduced [3]closely align with the DCGAN framework that being introduced in [8]. However , hyperparameters were adjusted based on the GAN-hacks which is to get enhanced performance. Figure 4 in [3] depict the pipeline of GAN. As described in the study , discriminator responsible to differentiate between real fabric pattern and generated 1 (indicate Real). Conversely , generator acting as counterpoint , where generator using the noise to synthesize images.

Furthermore , training strategy in[3], the optimizer that was utilized is "Adam" optimizer , activation functions like LeakyReLU and Tanh/Sigmoid was used. Where the "Adam" optimizer used to update parameter of neural network during training , leakyRelu used for prevent neuron from becoming inactive during training and Tanh/Sigmoid were used to squash input value between -1 & 1 and 0 -1 respectively.

Hyperparameter chosen for discriminator and generator are both different in term of their learning 0.00015 and 0.0002 respectively in [3]. For the training process in [3] 350 epchs , which refer to the number of time the entire training dataset passed through the model during training was set to 350, while each parameter that being produced every 5 epochs was saved and 3 images were considered in each epochs.

After done the training on 350 epochs , the model in[3] was continue trained with other pattern separately , and 80 epochs was chosen for trained number.

The model then evaluated after the training process on various fabric patterns. It could be observed that vary degree of quality of the pattern were being generated , however cheetah patterns stands out due to its simplicity as compare to the geometric pattern or ethnic design due to the complexity of patterns.

Figure 2.1 pipeline of GAN working in [3]

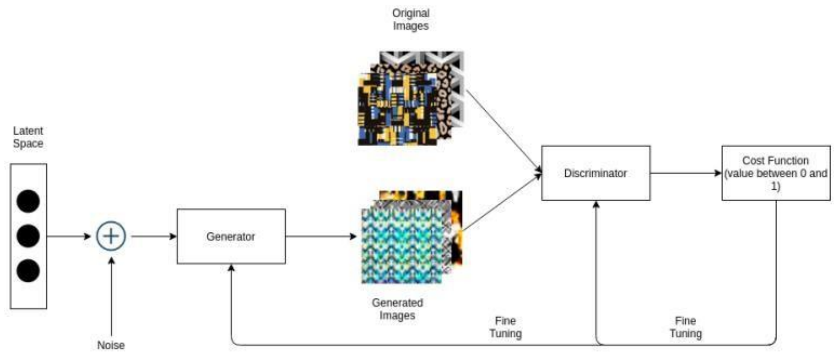


Fig. 4: Overall architecture of the proposed model

In[12] model architecture that being emoployed is based on DCGAN principles but compare to the model architecture in [3]it performed some modification on the DCGAN to be more suitable for specific task of generating real pattern. The modification that being performed include the number of filter , learning rate and modification of code. The architecture adapted in[12] differs from conventional DCGAN as it replace polling layers with stride convolution and involve Batch Normalization for improve the training stability. As similar to the [3], activation function that being selected were LeakyReLU , "Adam" as optimizer . For the learning rate , [12]applied and replaced the original Adam optimizer value with 0.002 but momentum unchanged. From [3],

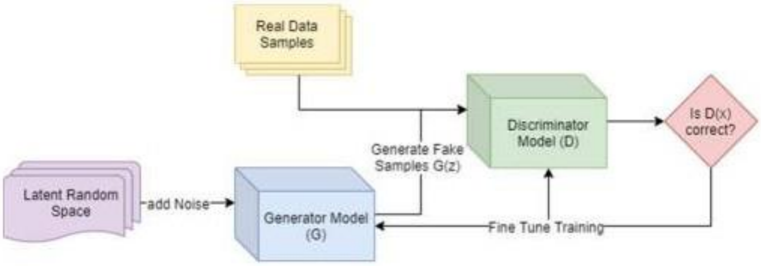
[12] indeed more focus on addressing the overfitting and aimed to generate unique pattern using GAN model. In contract [3]did not handle for this.[12]is crucial as to avoid model to be too specifically fit to training datasets and unable to generalize well to unseen data. Therefore , in [12], it mentioned the methods such as heuristic approximation , latent space interpolation and smoothness of latent space are considered to make sure for overfitting.

The outcome from the[12] involved using the trained generator to produce realistic pattern images. For the 1st epochs , the results showed were not perfect as it was far from the result. From the results showed , there was a high difference output between discriminator and generator when they are presented with initial random noise . However , after 35th epochs , it has a noticeable advancement , where some features and colors emerge , which reflect that it presented some pattern.

The culmination of generated patterns provided in [12]indicated that the model falls short of producing image with high quality.

[9] delved into GAN implementation using two competing model known as generator and discriminator, which aim to compete with each other to get better performance , and most of the time described as compete in zero-sum. The figure to show the generator and discriminator were showed in Figure3 in [9] described that the particular framework align well with fundamental concept of GAN. The models compete with each other to distinguish the real from fake samples , which is the role of discriminator , this subsequently leading to increase the realistic pattern that being generated by the generator. In other words, these model that engage in a competitive process were aiming to increase own accuracy to distinguish actual data and increase accuracy to mimic the original sample. This iterative process that described in[9] will continue enhance their performance until the achieve the objective of rendering fake data indistinguishable from real data input.

Figure 2.2 Structure of GAN in paper [9]



Similar to the paper [12]both were highlight the significance of proper weight initialization. In [9] , it presented two difference schemes with difference standard deviation and different learning rate for generator and discriminator.

In this study [9], compare with the two paper[3] [12]it emphasized on the choice of loss function , which is the Binary Cross Entropy , although in [12] did mentioned but it was not the primary focus of the optimization process , instead that paper emphasized on addressing overfitting. In [9], it justified the choice by explained that BCE is more suitable for binary classification problems , which is the output will either 0 or 1.

The experimental evaluation of the method includes thorough, including analysis of G and D losses to measure the convergence of the model. In addition, this article goes beyond numerical measurements to provide a fabric pattern generated for visual evaluation. These generated patterns are carefully examined compared to real fabric patterns, emphasizing the model’s ability to capture complex details and similarities.

2.1.3 Image style transfer for fabric pattern design

[4] introduced an approach by utilized the power of GANs for texture synthesis with a specific focus on Batik. There were two-stage training strategies being employed by [4], the methodology involved patch generation and subsequent combination, promise to seamlessly mix different batik patch while effectively addressing challenges such as block artifact.

In the [4], a patch generator, employing WGAN with Gradient Penalty is established to generate batik patches. After that , batik image generator is trained and learned a to place user-selected patches harmoniously , with the aim to generate a authentic Batik Image. The two distinct stages, strategically addressing the challenge of mixing rich and regular patches in single images.

This paper [4] introduce a comprehensive framework for Batik texture synthesis by utilize GANs. First, since the authors were not satisfied with the color and style attribute of generated image from GAN , so they decide use style transfer that extract features from VGG-19 model. With this integration, it help to make the fusion of texture more realism. The series of layers used in convolutional filters are to extract features from input images. Each filter scan and capture specific pattern. Gram matrix and L2 distance (Euclidean distance) are the mathematical computation that adopted for to measure the distance between the two points (real and generated) and qualify how different two set of value are.

Next , in[4] the author also focus on tackle the challenge of harmonizing the blocking artifact as they realized that BatikGAN_S showed overfitted result to the patch basedL2 distance. Therefore , they introduces local discriminator to evaluate the smaller local region within generated images. They trained the local discriminator by utilized the generated image and the corresponding real image. Local discriminator the is trained to distinguish whether the pair are from same image or not. In[4], they used the quantitative metric of Fréchet Inception Distance (FID) to asses the performance between the real and generated distribution. Batik Generator with Style Features emerges as key highlight as it integrates insight from pre-trained model to infuse different style of generated images, which make pattern more visually rich and realistic.

In addition , BatikGAN with style and local features also important for focusing on smaller part of images to ensure seamlessly fit together , with this consideration , it gradually reduce the block artifacts problem. The result showed BatikGAN_SL has smallest FID , means the distance between the real distribution and generated distribution are smaller, which indicated it closer to real Batik patterns.

Figure 2.3 Performance Comparison Result in [4]

Table 1: Performance comparison between the BatikGAN, the BatikGAN_S, and the BatikGAN_SL models.

	BatikGAN	BatikGAN_S	BatikGAN_SL
FID	168.42	163.92	**137.87**

2.1.4 Analysing and Improving Image Quality of StyleGAN2

The foundation of StyleGAN2 was based on the original styleGAN framework ‘s thorough review , which sought to eliminate the artifacts that compromised image quality. Through the creaful examination of deficiencies , the author of the paper [StyleGAN2] identified that the cause of the artifact and prompted several changes in the architecture in order to resolve the issue. The elimination of progressive education stands out among them. Although progressive upscaling was innovative for stabilizing GAN training, it inadvertently introduced artifacts that degraded image quality, which was particularly noticeable in applications that required accurate surface textures, such as fabric pattern. StyleGAN2's redesigned architecture avoids this approach and opts for a more advanced model scaling approach that improves image quality without compromising stability and scalability.

In the paper [StyleGAN2] , it discussed that for enhance the training stability and efficiency , the methodological innovations of StyleGAN2 also extend to its training program, greatly increasing the stability and efficiency of the model. The introduction of adaptive discriminant augmentation is an important improvement that cleverly solves the challenge of overfitting, a constant concern when training deep learning models, especially GANs. This technique dynamically adjusts boost levels based on discriminator performance, ensuring robust training even on limited datasets. This method is particularly advantageous in particular fields, like fabric patterning, where special effects such as unique textures or specific design motifs may be missing from training data. This adaptive mechanism ensures that StyleGAN2 can be effectively trained on a narrow dataset without compromising the diversity or quality of the results generated.

In addition , to solve these problems, the StyleGAN2 architecture differs from incremental growth and adopts a fixed architecture that trains all resolutions simultaneously. This change fundamentally changes the way the model scales and processes images, ensuring more consistent detail creation at different scales. By training the network holistically from scratch, StyleGAN2 avoids the discontinuities that occur when adding layers gradually, resulting in smoother transitions and more accurate texture rendering in final images. This method significantly reduces the artifacts associated with incremental resolution increases, improving the quality and realism of the generated images.

Another important architectural improvement in StyleGAN2 is the improvement of normalization techniques, especially the replacement of an adaptive expression. Normalization (AdaIN) with weight demodulation. In StyleGAN, AdaIN used to inject style information at various points in the generator, which allowed manipulation of the style elements of the generated images. However, this approach also introduced normalization artifacts that could degrade image quality. StyleGAN2's weight demodulation technology provides a more nuanced method of style control, effectively removing these artifacts while maintaining the ability to create highly versatile and stylized images. This refinement highlights a deeper understanding of how style information can be integrated into the creation process without compromising the integrity of image content.

In addition, StyleGAN2 introduces path length adjustment, a new concept that aims to improve image consistency and predictability. This regularization technique encourages the model to produce changes in the generated images that are as linear as possible in response to changes in the hidden state of the input. The result is a more intuitive and manageable hidden space, where small adjustments lead to predictable and smooth transitions in the output. This enhancement not only improves the user's ability to manipulate the generated images, but also improves the overall accuracy and quality of the images by ensuring that the variations between the different generated instances are consistent and realistic. Despite these advances, it is important to acknowledge potential limitations and future research areas. Although StyleGAN2 greatly improves on the original model, the complexity of its architecture and the computational resources required for training and inference can be problematic, especially for applications that require real-time production or implementation with limited hardware. Additionally, while StyleGAN2 reduces artifacts and improves image quality, creating certain textures or highly detailed patterns can still be difficult, indicating that GANs still have room for improvement in handling complex image details..

2.1.5 D-batik Application

D-Batik is a digital application designed to create Batik patterns for beginners and small businesses. The app is accessible on both PC and Android platforms and offers a range of tools to simplify the Batik production process. One of the standout features of D-Batik is its "Smooth Draw" feature, which allows users to draw Batik patterns with their fingers or mice, providing a direct tactile experience. In addition, the "symmetric drawing" feature makes it easier to create symmetrical designs, which are common in traditional Batik patterns. The app aims to make digital Batik creations easier for those who may not have extensive design skills, thereby democratizing the process by lowering the barriers to entry for new designers. It also supports direct integration with fabric production, meaning designs can easily transition from digital format to physical textiles.

However, D-Batik lacks advanced features such as artificial intelligence or machine learning algorithms that can automatically generate new and unique Batik patterns. The application relies heavily on the user's manual input and creativity, which may limit the diversity and innovation of the resulting design. While this may preserve the traditional elements of Batik's production, it may also mean that users invest a lot of time and effort in learning the nuances of digital design tools. In addition, the app does not provide any guidance on color theory or pattern matching, which may be helpful for novice users. Despite these limitations, D-Batik is a valuable tool for those interested in exploring Batik design, especially in educational or amateur settings.

2.1.6 jBatik Application

jBatik is another digital application focused on Batik design that features the use of fractal geometry to create complex and complex patterns. The software allows users to generate Batik designs using mathematical principles, making them unique compared to traditional drawing tools. jBatik's method is particularly interesting because it utilizes geometric patterns to replicate repetitive but different patterns that are common in Batik. Users can select basic shapes such as triangles or squares as the base elements and then convert them into complex fractal designs that can be used as Batik patterns. This approach introduces a more scientific approach to design that allows for patterns that are both aesthetically pleasing and mathematically coherent.

However, while jBatik offers a unique way to design Batik patterns, it also has some limitations. Using fractal geometry can make the design process less intuitive, especially for users with no background in mathematics or geometry. For beginners, the software's user interface can seem complex and overwhelming, which can hinder its accessibility. Also, like D-Batik, jBatik does not integrate AI technology to automatically generate patterns or suggest design changes based on user preferences. This lack of automation can be seen as a limitation on users' search for more dynamic design tools. Despite these weaknesses, jBatik remains a valuable tool for designers who want to explore new ways to create Batik patterns, especially those interested in combining art with mathematics.

2.1.7 Pattern Monster

Pattern Monster is an online tool that enables users to create repeatable SVG (Extensible Vector Graphics) Batik patterns. Unlike traditional Batik design software, Pattern Monster focuses on generating simple geometric patterns suitable for a variety of purposes, such as website background, clothing, branding, and packaging design. Basic tools such as stroke width adjustment, connection mode customization, and color selection are provided for beginners to use. However, the app does not include any AI capabilities that automatically generate unique Batik patterns, instead relying on user input and manual adjustments. The platform allows users to download their work in multiple formats, such as SVG or PNG, making it available for different applications from digital to print media.

The main limitation of the pattern monster is that it provides only a simple geometric design that may not capture the intricate and cultural nature of traditional Batik patterns. Users seeking to create more complex or traditional Batik patterns may find the tool limited because it lacks the ability to draw free forms or organic shapes that are common in Batik art. Pattern Monster, however, offers an easy-to-use solution for users who want to quickly create simple, repeatable business models.

2.2 Comparison between Previous Work

Table 2.2.1a Comparison between Previous Work

Model	Original GAN	DCGAN	StyleGAN2
Architectural Innovation	Used basic Gan structure	Some adjustment made for fabric pattern , use DCGAN framework	Elimination of progressive growing and weight demodulation
Image Quality	Moderate	Moderate	Very high
Training Stability	Variable	Improve with specific adjustment	Enhance with adaptive discrimination augmentation
Texture and Detail Handling	Basic	Can be enhanced through pre-processing and model adjustment	Superior , with high fidelity in texture and details
Artifact Reduction	Does not specified	Need to adjust	Significant reduction in artifact
Control Over Generation Process	Limited	Moderate	High with improved latent space control

Evolution from original GAN/DCGAN to style GAN2 has made significant progress in generative modelling, especially in structural pattern creation. The original GAN laid the foundation for the basic structure that allowed preliminary research into the generating application. However, these models often have moderate image quality, variable training stability, and limited control over the generation process, making them less suitable for applications that require high texture and detail accuracy, such as fabric pattern generation. From the research, as cited in [3], [9] and [12], builds on these and introduces architectural adaptations that adapt to fabric patterns, often using DCGAN frameworks to improve image quality and realism. These efforts have made progress in improving texture and detail handling, targeted preprocessing, and model adaptability, resulting in modest improvements in scalability and performance. However, control of the creative process remained modest, and efforts to reduce artifacts were only partially successful. StyleGAN2 represents the pinnacle of this development, introducing architectural innovations that significantly improve image quality and realism. Removing progressive gains and introducing weight demodulation directly removes previous limitations, allowing excellent texture and detail handling. By adding adaptive separators, the stability of the training can be greatly improved and robust training can be performed even under limited data sets. Furthermore, despite the complexity of the StyleGAN2 architecture, it achieves high scalability and efficiency, and provides unprecedented control over the generation process thanks to improved hidden state control. In particular, the StyleGAN2 greatly reduces the appearance of artifacts and sets a new standard for image generation in applications that require the highest accuracy, such as fabric pattern design. Basically, previous GAN models, including DCGAN and its fabric-specific adaptations, mark a significant step forward in the adoption of GANs in the fashion industry.StyleGAN2's extensive improvements in architectural innovation, image quality, training stability and efficiency define it as a transformative force that greatly increases the likelihood of creativity and practical application in fabric pattern generation.

Table 2.2.1b Comparison between Different Batik Design App

Feature	**D-Batik**	**jBatik**	**Pattern Monster**
Platform	PC, Android	PC	Online
Pattern Generation	Manual drawing tools like Symmetry and Smooth Draw	Manual creation using fractal patterns	Simple geometric shapes
AI Integration	No	No	No
Design Flexibility	Moderate - allows creativity without AI	High - can make complex patterns, but is harder	Low - limited to basic shapes

Unique Features	Easy tools for symmetrical pattern creation	Uses fractal math for unique, complex patterns	Quick and simple pattern creation
Main Weaknesses	No AI, relies on user input for creativity	Complex and hard to use for beginners	Very basic, not suitable for traditional Batik

To sum up, D-Batik is perfect for novices and small enterprises since it is easy to use and accessible. Nevertheless, it is devoid of sophisticated functions like artificial intelligence (AI) that generate original designs automatically. With its fractal geometry, jBatik offers a more advanced method, although it could be too difficult for novices. Though it lacks the depth required for traditional or more intricate designs, Pattern Monster provides an easy-to-use online platform for making geometric Batik patterns. Every app has advantages and disadvantages based on the demands and skill level of the user.

2.3 Strength and weakness of the previous work

The advancement of Generative Adversarial Networks (GANs) in the domain of fabric pattern generation has been marked by a series of innovative advancements, each bringing forth its distinct contributions and encountering distinct obstacles. The introduction of GANs, including their DCGAN variant, established the foundation upon which generative modelling was to be built. These initial models, which were celebrated for their pioneering structure, offered simplicity and accessibility, democratizing the field of generative modelling. However, their nascent architecture was marred by several limitations, including a propensity for training instability and a lack of fidelity in the generated images. This inadequacy was particularly evident in applications that necessitate intricate textures, such as fabric pattern generation, wherein these models frequently failed to capture the intricate details essential for authenticity.

Following the original GAN framework, subsequent efforts were undertaken to tailor the GAN architecture specifically to the nuances of fabric pattern generation. These works represented a significant advancement in aligning the generative process with the specialized requirements of textile pattern design, achieving enhancements in image quality, and introducing a degree of control over the generated outputs. During this phase of evolution, a concerted effort was made to mitigate the artifacts that plagued previous models, albeit with varying degrees of success. The persistence of such artifacts, coupled with the challenges pertaining to training efficiency and model scalability, emphasized the limitations that still impede the full potential of GANs in this specific domain.

StyleGAN2 represents a paradigm shift that addresses the multifaceted challenges inherited from its predecessors. This iteration serves as a testament to the meticulous examination and inventive prowess of its creators, showcasing architectural innovations that significantly enhance image quality and texture authenticity. By eliminating progressive growth and adopting weight demodulation, StyleGAN2 eliminates the artifacts and inconsistencies that previously hindered the authenticity of the image. Furthermore, the introduction of adaptive discriminator augmentation brings new levels of training stability and efficiency, enabling the model to overcome the limitations of its training dataset and achieve unprecedented levels of fidelity in fabric pattern generation.

Nevertheless, the sophistication of StyleGAN2 introduces its own set of complexities, especially in the realm of computational resource requirements. The architectural advancements that underpin its superior performance also entail a heightened requirement for processing power, potentially creating obstacles to accessibility for individuals or institutions with limited computational resources. Moreover, the model's capability for intricate detail and superior output raises concerns regarding the possibility of overfitting, particularly when operating within the limitations of limited training data.

The evolution from the original GANs to StyleGAN2 demonstrates the relentless pursuit of perfection in the domain of fabric pattern generation. Each stage of this journey has yielded layers of comprehension, expertise, and insight, culminating in a model that significantly extends the scope of possibilities in generative modelling. StyleGAN2, with its unsurpassed image quality, control, and efficiency, sets a new benchmark for realism in digital fabric pattern design. Nonetheless, its accomplishments are accompanied by novel obstacles in terms of complexity and resource requirements, thus marking the subsequent set of obstacles to be overcome in the ongoing advancement of generative adversarial networks.

This section will discuss the strength and weakness of the reviewed existing application.

D-Batik is designed to be user-friendly, making it an excellent choice for beginners and small businesses interested in creating Batik designs. Its main advantage is its simplicity and ease of use, providing simple tools such as "smooth drawing" and "symmetric drawing" to help users easily create symmetrical Batik patterns. These features enable users to start creating patterns with minimal design experience without the need for a steep learning curve. However, its weaknesses include a lack of advanced features, such as integrating artificial intelligence that automatically generates new and unique patterns. Dependence on manual input means that users must be creative and patient enough to create different designs.

By combining fractal geometry, jBatik provides Batik design with a more advanced approach that allows users to create complex patterns that are difficult to implement with traditional drawing tools. For designers who prefer more mathematical or algorithmic design methods, the app offers a unique way to generate visually complex Batik patterns. Despite these advantages, jBatik has obvious weaknesses, including a more complex user interface and a higher learning curve, which may not be suitable for beginners. Also, like D-Batik, it doesn't use AI to automate or suggest new designs, which can be a limitation for users seeking more dynamic or automated design tools.

Pattern Monster provides a very simple and easy-to-use platform for creating simple geometric Batik patterns. It's perfect for beginners and users who need a quick, repeatable design for business purposes, such as brand or website background. The strength of the application lies in its simplicity and the ability to download designs in multiple formats (SVG, PNG). However, its main drawback is its limited design capabilities and lack of support for more complex or culturally rich Batik models. Unlike D-Batik and jBatik, it doesn't offer tools to create traditional or complex Batik designs, making it less suitable for users who want to explore Batik's full range of art. All three applications share a common limitation of not integrating AI, which can enhance their functionality and user experience by providing more innovative and diverse design options.

Chapter 3

3.1 Methodologies and General Work Procedures

The methodology that used for develop an mobile application in this project is Agile Methodology. This is because agile methodology is more suitable for a project that is consider medium scale with about 7month of development timeline. In addition , since for the project it is not complex and do not required extensive upfront analysis before the work therefore , this can reduce the time of development and make the development process more faster by reducing the documentation.

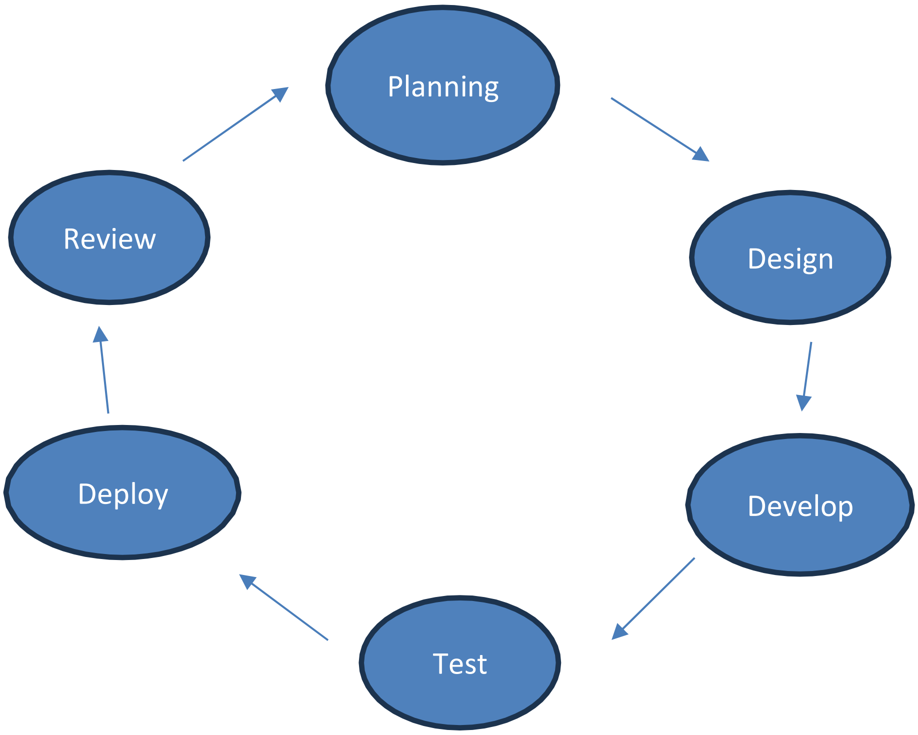


Figure 3.1 Agile Methodology used in mobile application

<u>Stage 1: Planning Phase</u>

During the design phase of the project, an in-depth review of the StyleGAN technology to fully understand its complexity and capabilities was conducted. This phase aimed to deepen understanding of reproductive adversarial networks and the nuances of their training in culturally significant batik patterns. Agile design practice allowed to create an adaptive design timeline that took into account the complexity of training StyleGAN on a curated batik dataset. This phase included strategizing the collection and preparation of this dataset, developing a training program, and creating an architecture for subsequent application development.

Design Phase:

During the design phase, we carefully designed the system architecture to match the StyleGAN2 integration. The user interface patterns have been created to ensure a smooth interaction with the created batik patterns. Detailed architectural diagrams including use cases and block diagrams were developed to define the technical components and their interactions. This phase also determined the essential hardware and software requirements needed to effectively train and run StyleGAN2 using the custom batik dataset.

Development phase:

The development phase began by installing and training the StyleGAN2 model using prepared batiks dataset This training was crucial in creating high quality batik patterns. At the same time, we developed the mobile application in Android Studio, preparing it for the integration of the teachable model. Each sprint was about implementing and refining the functionality of the application, ensuring readiness for AI model integration.

Testing phase:

Testing was done in two critical segments: validating the performance of the StyleGAN2 model and ensuring the functionality of the mobile application. We tested StyleGAN2 by generate result to evaluate the quality and variety of batik patterns it creates. At the same time, the application underwent rigorous testing to identify bugs and ensure that all functions work as intended, especially StyleGAN2 model integration points.

Implementation phase:

When testing confirmed the stability and performance of both StyleGAN2 model and a mobile application, we continued with the implementation. This step involved launching the application in a live environment where users can start using it.

Continuous monitoring was carried out so that potential operational issues could be quickly resolved and the application worked smoothly.

Review and feedback phase:

After implementation, we actively collected user feedback to evaluate the application's performance and satisfaction with the created batik patterns . This feedback was crucial for iterative improvements and future updates. The insights gained led to adapting the StyleGAN2 model and improving the application, focusing on user experience and improving pattern quality.

3.2 System Architecture Diagram

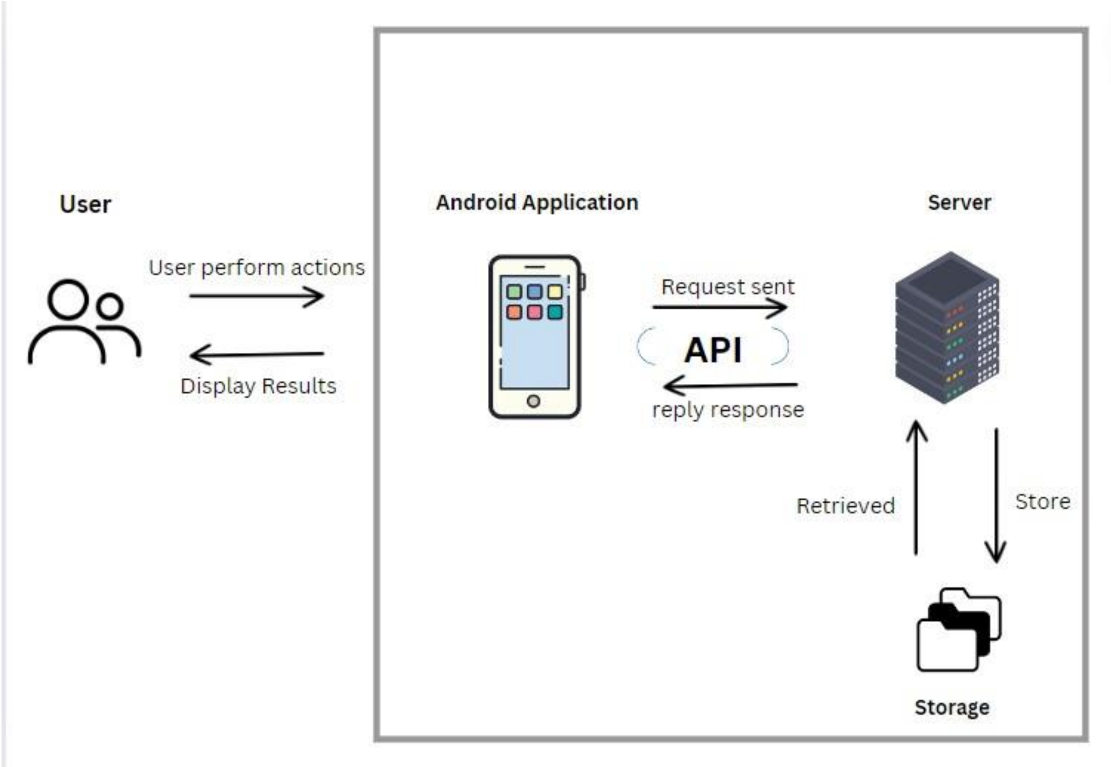


Figure 3.2 Figure of Architecture Diagram

In this Figure 3.2 architecture, the server is responsible for performing complex AI tasks to generate and customize Batik patterns. It leverages powerful AI models such as StyleGAN and VTON-HD to produce these results. The StyleGAN (Generative Adversarial Network) generates new and unique Batik patterns by learning from datasets of existing patterns, allowing the creation of complex designs that reflect

Batik's traditional art while introducing new variants. At the same time, Virtual Fit Network High Definition (VTON-HD) is used for virtual fitting functions, allowing users to see the appearance of the generated Batik pattern on different clothing items or models. This AI model processes user input, such as the selection of clothing and patterns, and creates realistic visual representations of these combinations.

The server handles these AI tasks, performing computationally cumbersome operations such as training and inference using Style. GAN generates Batik mode and applies VTON-HD to simulate fitting. Intermediate results, including AI-generated images and frames, are temporarily stored in the server's storage. This ensures efficient data management and allows for quick retrieval when needed. The server also performs auxiliary tasks such as cropping and resizing images to optimize them for display on the client side.

Android applications act as clients, providing an interface for users to perform operations such as generating patterns, customizing designs, and fitting clothes. It communicates with the server to request the necessary AI-generated output, such as a unique Batik pattern or a virtual fit image. The application retrieves the results stored on the server and displays them to the user, maintaining a smooth and responsive experience by offloading all the onerous processing tasks to the server. This setup ensures that users can easily customize and visualize their Batik mode without putting too much burden on their devices.

3.3 Use Case Diagram

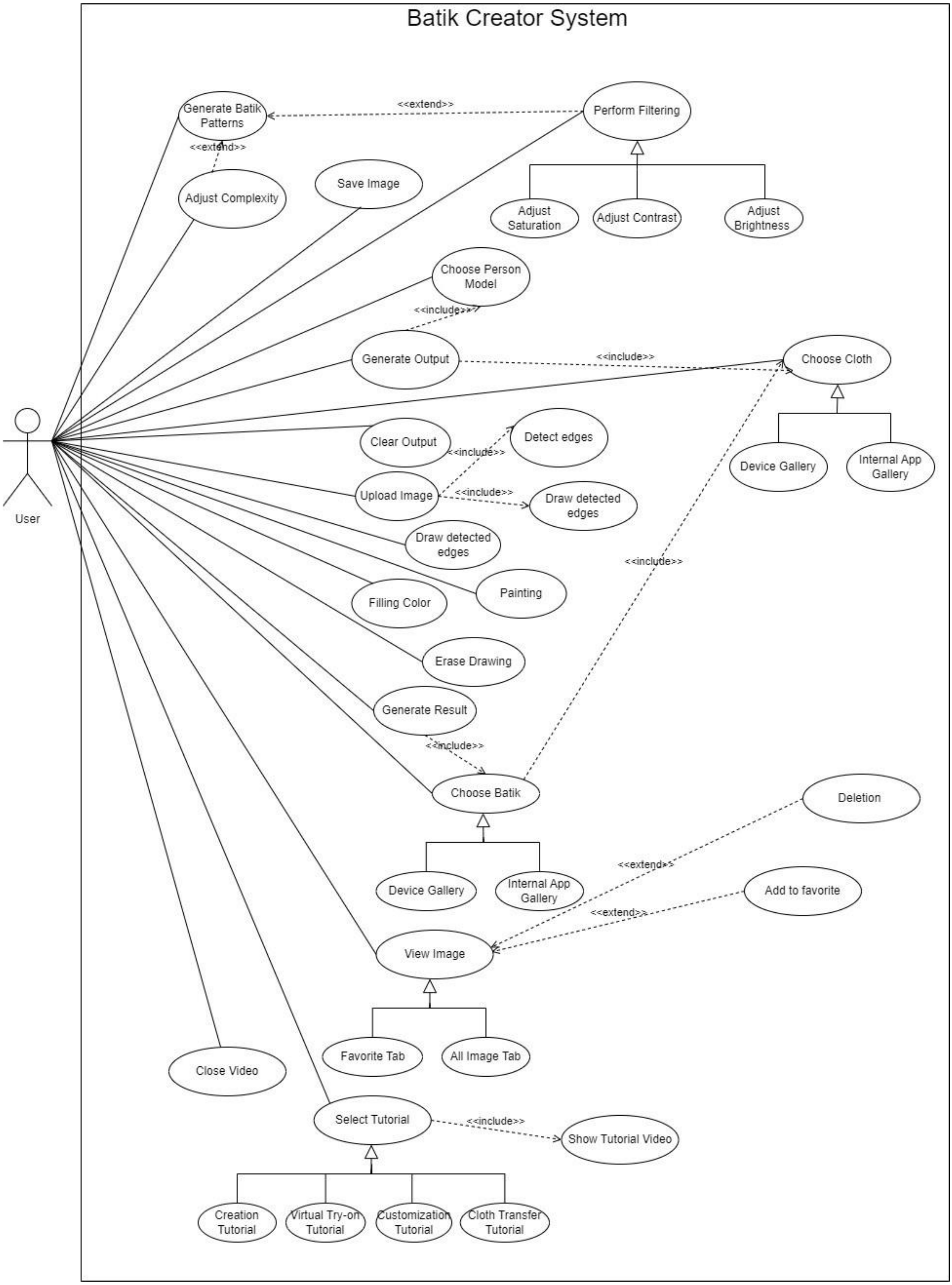


Figure 3.3 Use Case Diagram

3.4 Activity Diagram

3.4.1 Use Case : Generate Batik Pattern

When the user clicks Generate Batik Pattern, the Android application requests the Flask server to create patterns using AI models such as StyleGAN. The server processes and returns the image received by the application, which can be cropped and cached if needed. The user can then view, adjust, or filter the image and save it to the application library.

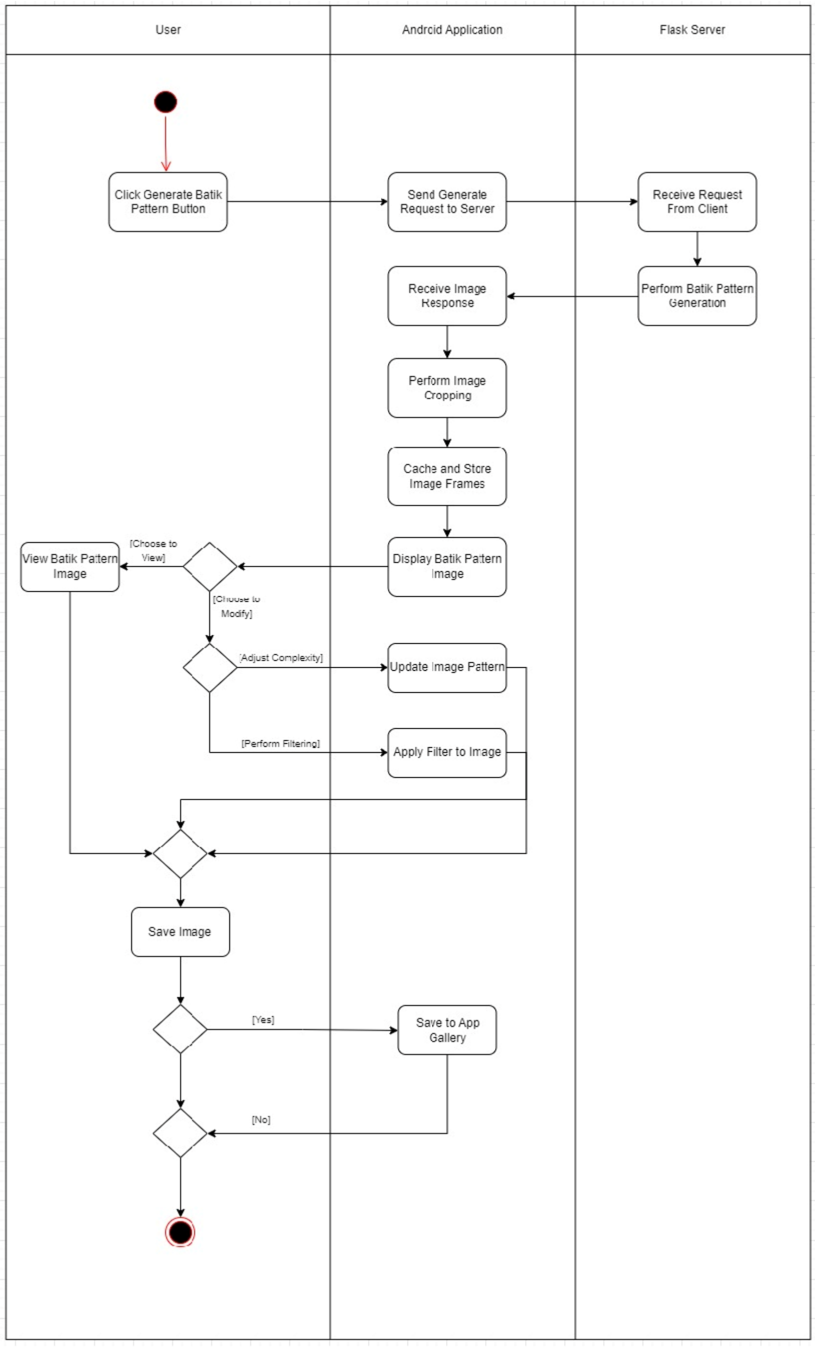


Figure 3.4.1 Activity Diagram (Generate Batik Pattern)

3.4.2 Use Case : Virtual Try On

The user clicks the Select Image button to open the file selector and select an image from the device. The application loads the image, converts it to gray scale, and applies Gaussian blur to reduce noise. Canny's edge detection can identify and refine edges, further smooth edges, and perform color inversion with Gaussian filters for clarity. The processed image is then converted to bitmap format and displayed on canvas.

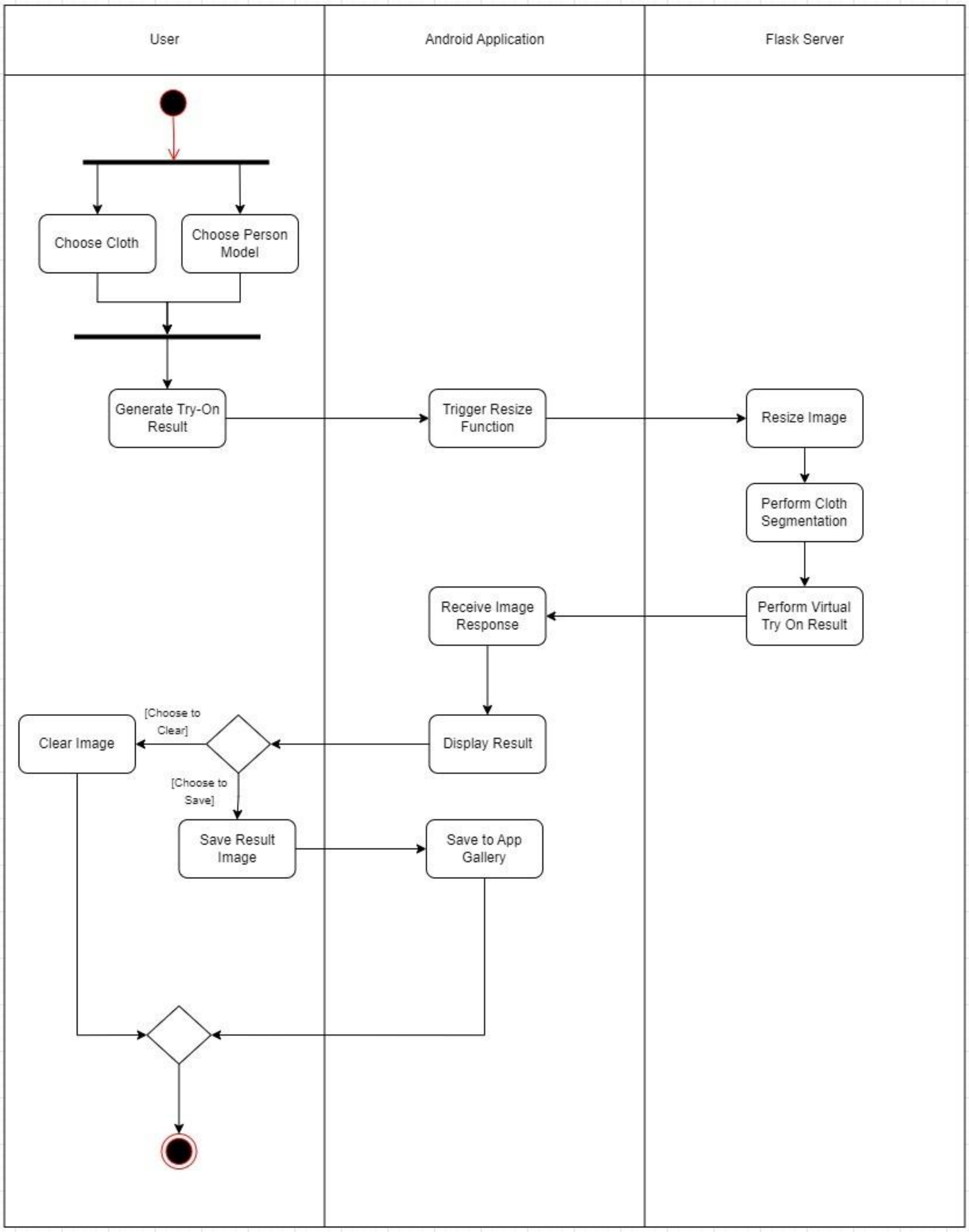


Figure 3.4.2 Activity Diagram (Virtual Try On)

<u>3.4.3</u> <u>Use Case : Customization – Upload Image and detection edges</u>

The user clicks the Select Image button to open the file selector and select an image from the device. The application loads the image, converts it to gray scale, and applies Gaussian blur to reduce noise. Canny's edge detection identifies and refines edges and further smoothes the edges and performs color inversion through a Gaussian filter for clarity. The processed image is then converted to bitmap format and displayed on canvas.

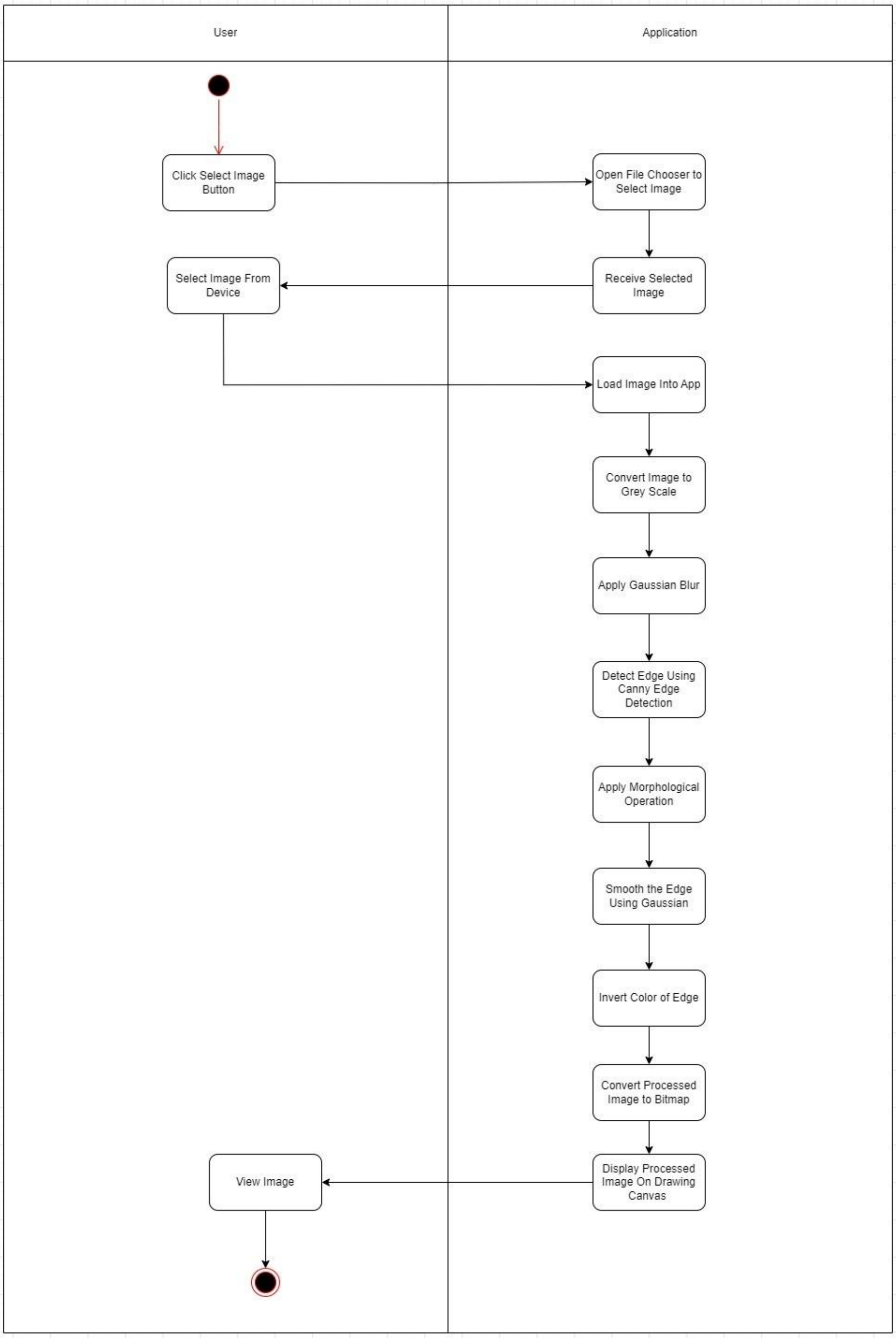


Figure 3.4.3 Activity Diagram(Upload Image and detection Edges)

3.4.4 Use Case : Customization – Painting

The user first selects the brush mode in the Android application to start drawing. After selecting the brush mode, the application sets the current mode to Draw to prepare the application to capture any drawing operations performed by the user.

When the user touches the screen to draw, the app captures these touch events and interprets them as drawing inputs. The application then draws the path based on the user's input, creating a visual representation of the graphics on the screen. After the path is drawn, the display is updated in real time to reflect the new drawing, providing immediate feedback to the user as they continue to draw the screen. This process ensures an interactive and responsive drawing experience within the application.

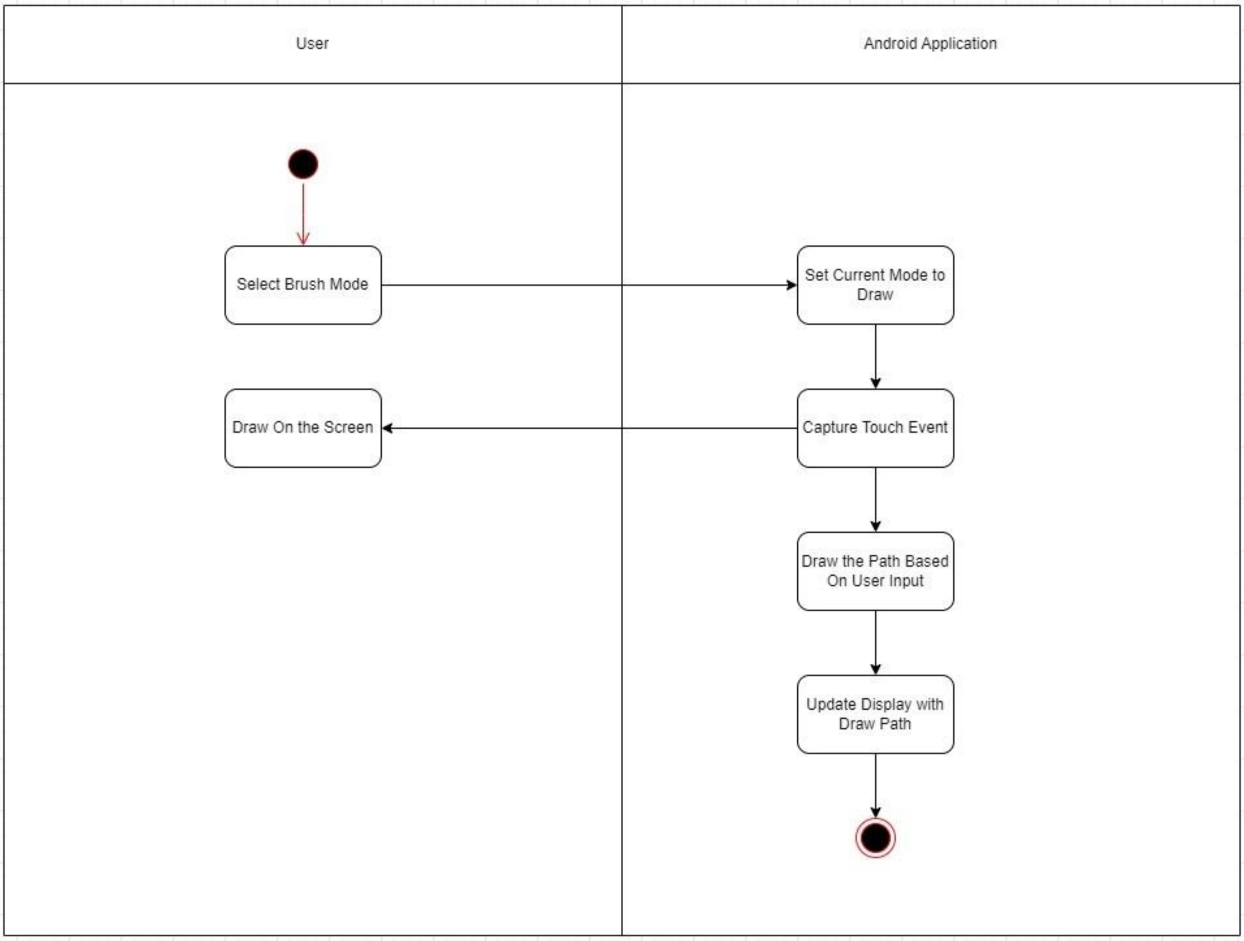


Figure 3.4.4 Activity Diagram(Customization - Painting)

3.4.5 Use Case : Customization -Erase

Users first select "Erase Mode" in the Android application. The application then sets the current mode to "erase", ready to capture any erasure operation performed by the user. When the user moves his finger on the screen, the app captures these touch events.

The application interprets touch events to draw circles or shapes that represent areas to be erased. As users continue to move their fingers, the app dynamically updates the display to show deleted areas, providing immediate visual feedback to users, ensuring an intuitive and interactive erasure experience.

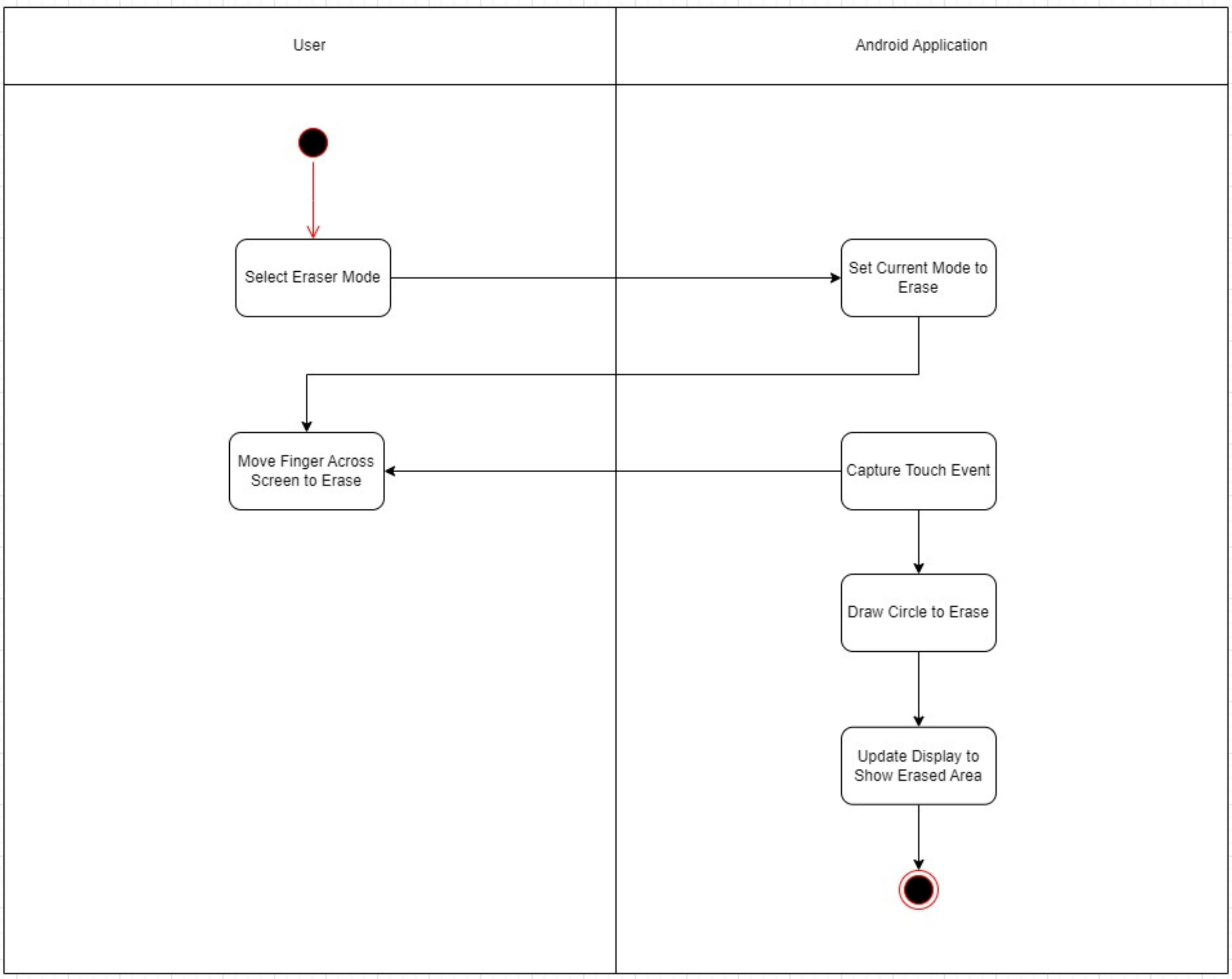


Figure 3.4.5 Activity Diagram (Customization -Erase)

3.4.6 Use Case : Customization – Filling Color

The user first clicks the Fill Color button in the Android application and opens the color selector to select the desired color. The application sets the mode to Fill and applies the selected color. When the user clicks on the canvas, the application detects the click, captures the coordinates, and checks the target color. If the target color is different from the selected color, the app performs overflow fill, replaces it with the selected color, and updates the display to show changes.

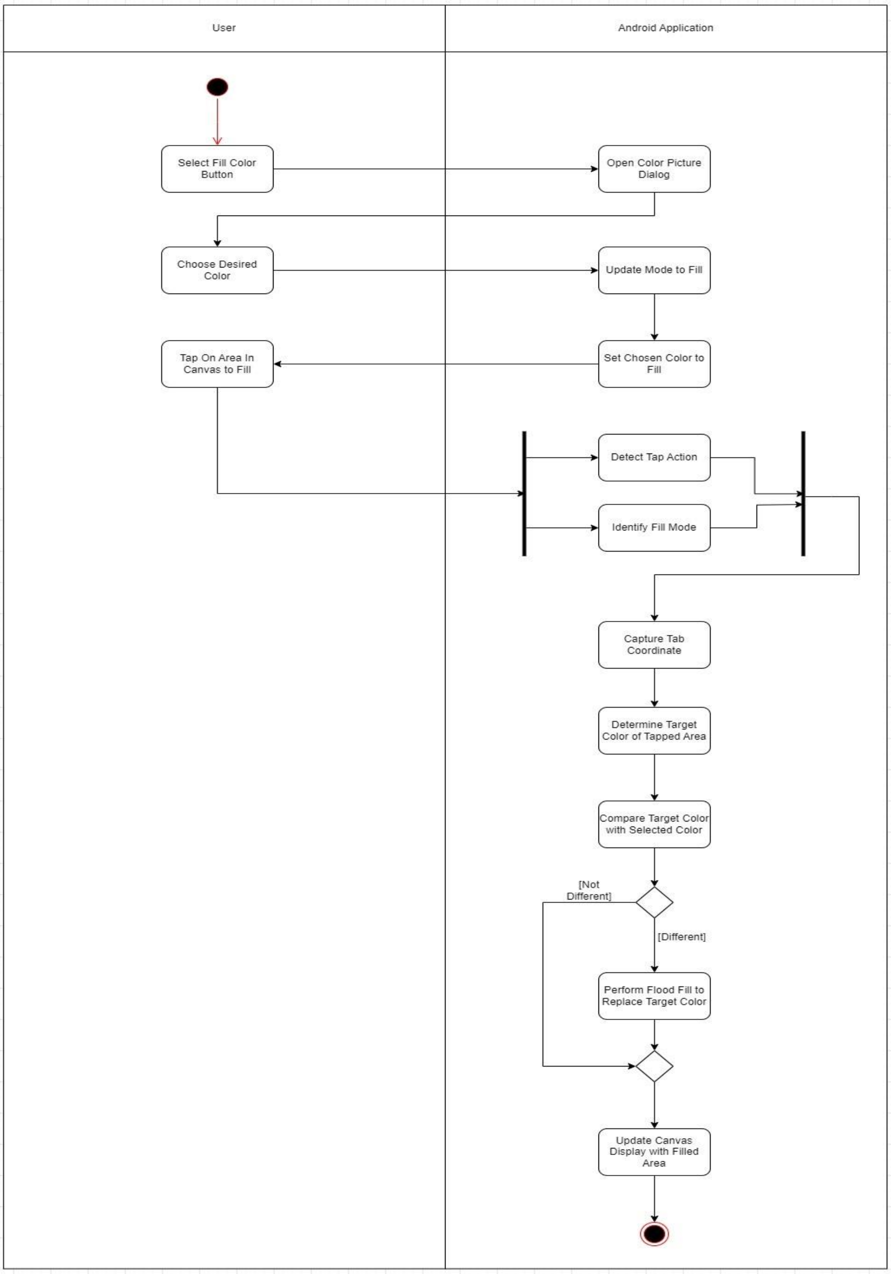


Figure 3.4.6 Activity Diagram (Customization – Filling color)

3.4.7 Use Case : Customization – Save Image

The user first clicks the "Save" button in the Android application. The application checks if it has the storage permissions required to save the image. If no permissions are granted, the app requests permissions from the user. The user can grant or deny this permission.

If permission is granted, the application converts the current drawing on the canvas to bitmap format and continues to save the image to storage. After saving the bitmap successfully, the application displays a message indicating success, confirming that the image has been saved correctly. This process ensures that user data is handled securely and with appropriate administrative privileges.

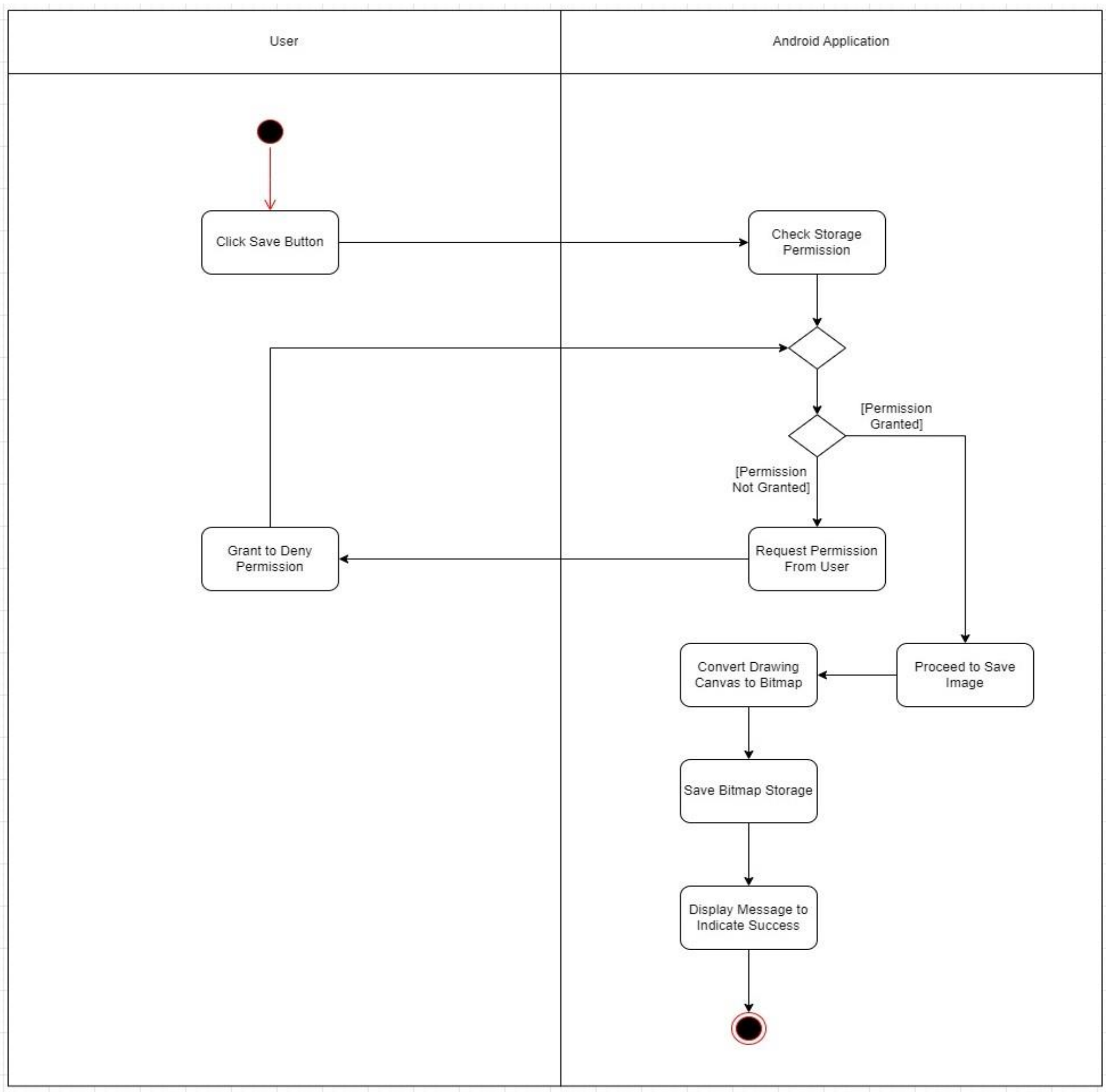


Figure 3.4.7 Activity Diagram (Customization -Save Image)

3.4.8 Use Case: Batik Cloth Transfer

The user first clicks the "Add Batik" button in the Android app and selects the image source. After selecting an image from the gallery, the application loads and displays it. The user then selects a cloth image for texture transfer. The app retrieves Batik and fabric images, applies texture transfer, and finally generates and displays a composite image of the Batik pattern applied to the selected fabric.

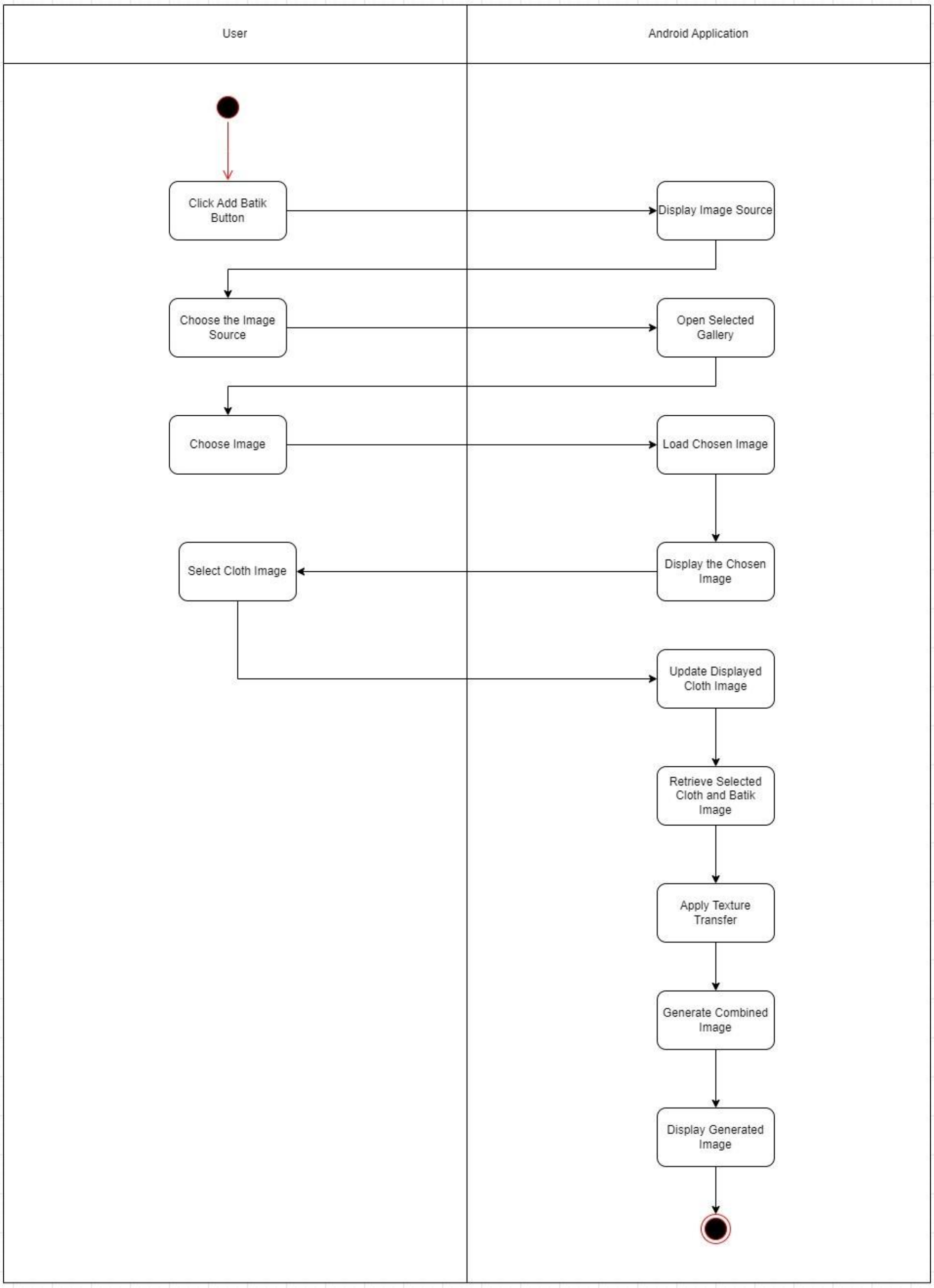


Figure 3.4.8 Activity Diagram (Batik Cloth Transfer)

3.4.9 Use Case : View and Select Tutorial Video

Users first click on the "Tutorial Tips" button in the main menu. The application will then display a list of available tutorial videos for the user to choose from. Once the user selects a tutorial video, the app starts playing the selected video.

When playing video, the user can choose to control playback. They can choose to turn off the video, continue watching, or use controls to play or pause the video at any time. If the user decides to turn off the video, the app will stop playing and close the video dialog, effectively ending the tutorial session.

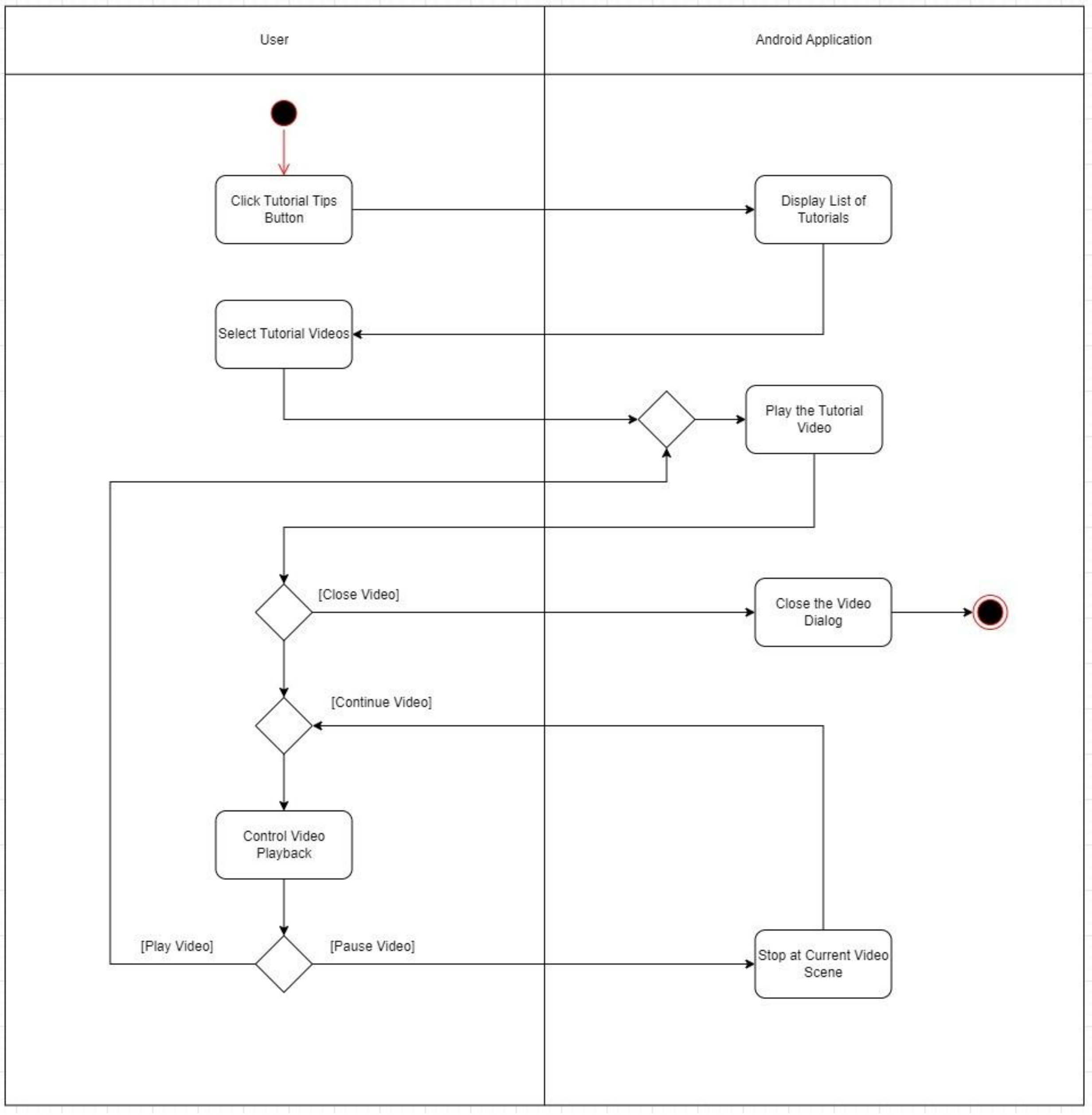


Figure 3.4.9 Activity Diagram (View and Select Tutorial Video)

3.4.10 Use Case : View Gallery

Users can choose from two tabs: "All Images" or "Favorite Images." Depending on the selection, the app displays all images or only favorites. Users can add, delete, or share images to favorites. Adding to favorites places the image in the favorites list, while deleting it deletes it and refreshes the page. For sharing, the user selects the image, selects a platform and shares it. If no further action is taken, the user returns to the main menu. This process enables efficient image management.

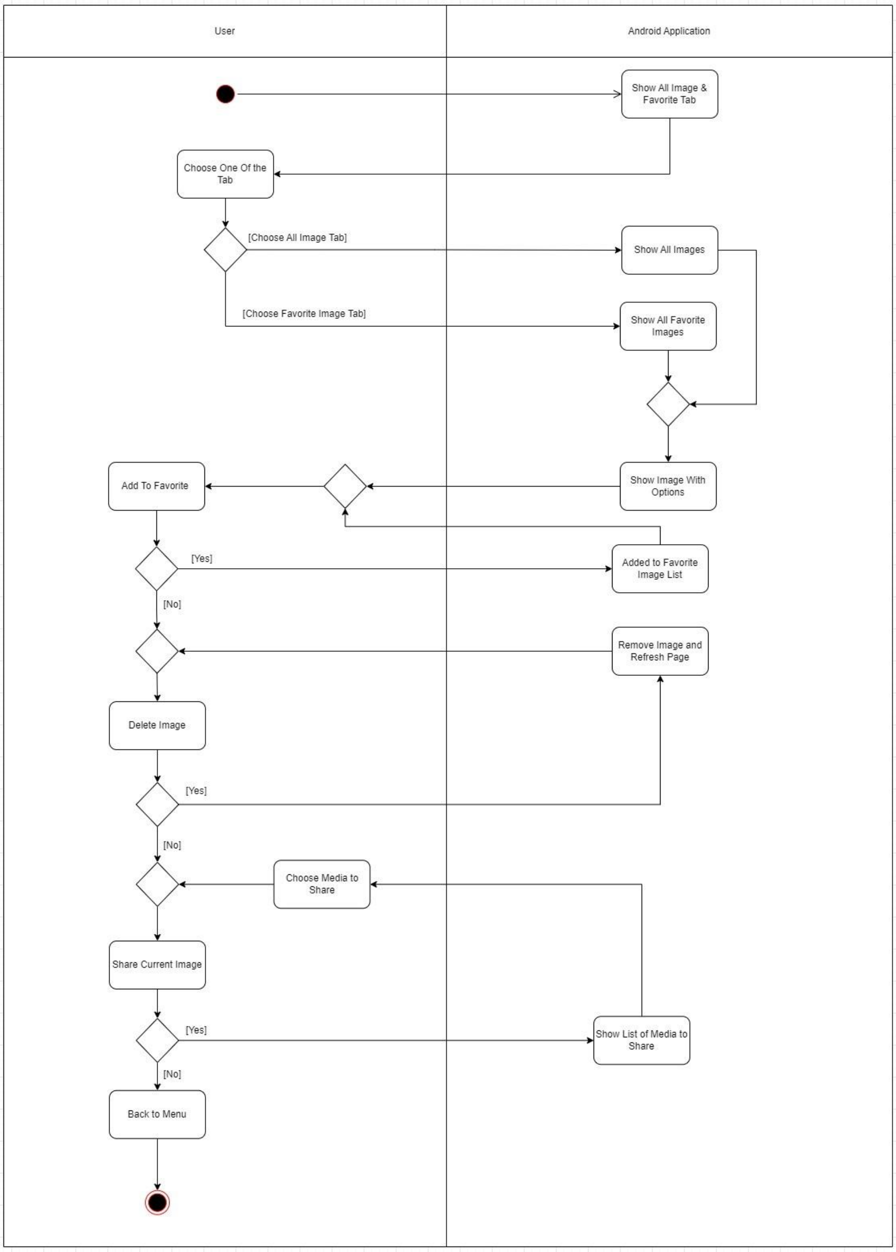


Figure 3.4.10 Activity Diagram (View Gallery)

Chapter 4

System Design

4.1 System Block Diagram

In this project , training style Gan is first step. Which is training the StyleGAN2 model with the prepared dataset and develop a workable UI interface that able to generate the batik image from the host server and use that generated pattern for virtual try-on. Below show the styleGAN2 model that used for generate batik pattern image and the overall design of the entire application that will be developed at the end of the project 2.

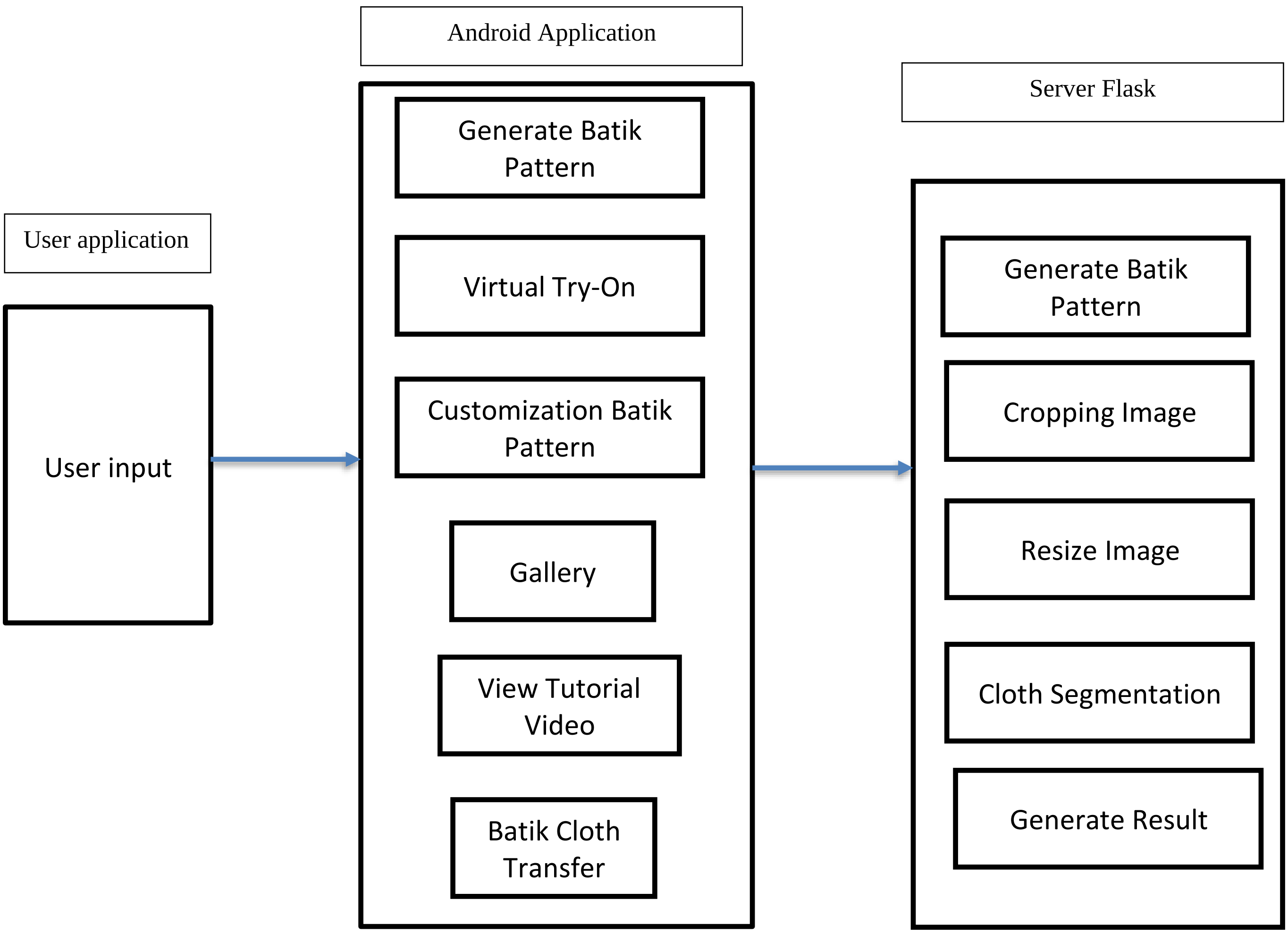


Figure 4.1 Overall System Block Diagram

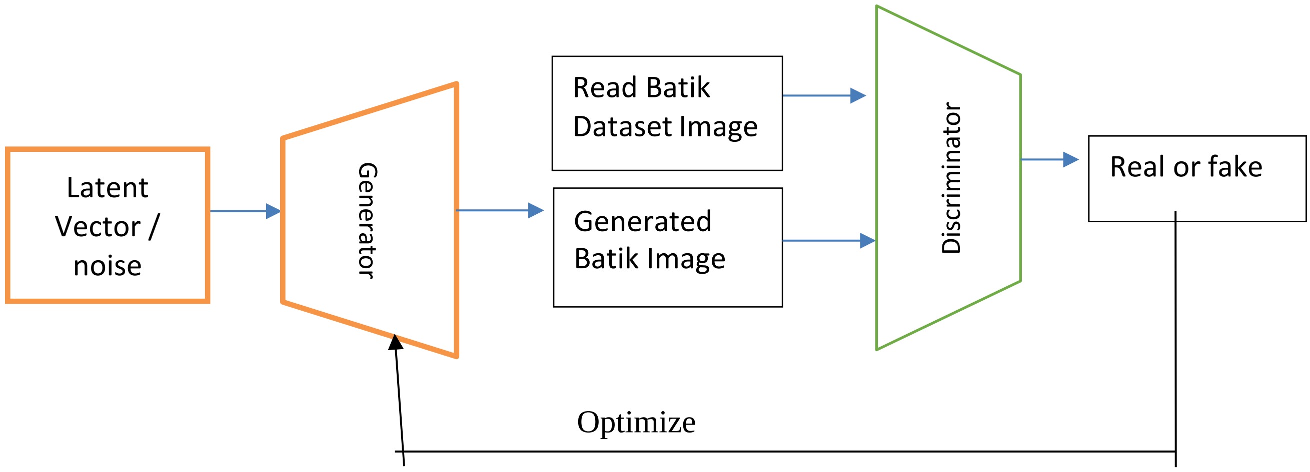


Figure 4.2 Block Diagram of the proposed system design (Train the StyleGAN2 model with batik dataset)

The project's goal is to create an application that lets users create distinctive Batik patterns with Style, a potent generative model that produces pictures with excellent quality. The procedure starts with input from the user via an interactive interface. The user may change the style's concealed size by dragging the slider. Possible area for the GAN2 model to occupy. Color, form, and pattern complexity are just a few of the design components that are directly impacted by these changes.

After the user makes changes, a structured latent vector is created from the input. The Style GAN2 model, which creates visual Batik patterns depending on user-defined parameters, is used by the server to handle this vector. To guarantee that the output is ready for display and to enhance its visual quality, post-processing is used.

After that, the user's smartphone receives the created Batik pattern, which they may use to examine and edit their creations. Additional features of the program include virtual fitting and Batik Cloth Transfer, which are made possible via a server-side Flask framework that manages picture scaling and cropping. using this integrated approach, users may create and customize Batik patterns using an entertaining and adaptable tool.

4.2 System Component Specifications for StyleGAN2 Training Model

Before start to develop and integrate the styleGAN2 into the application that we developed , it is a must to understand how the **training in styleGAN2** actually done and how it can generate the batik pattern.

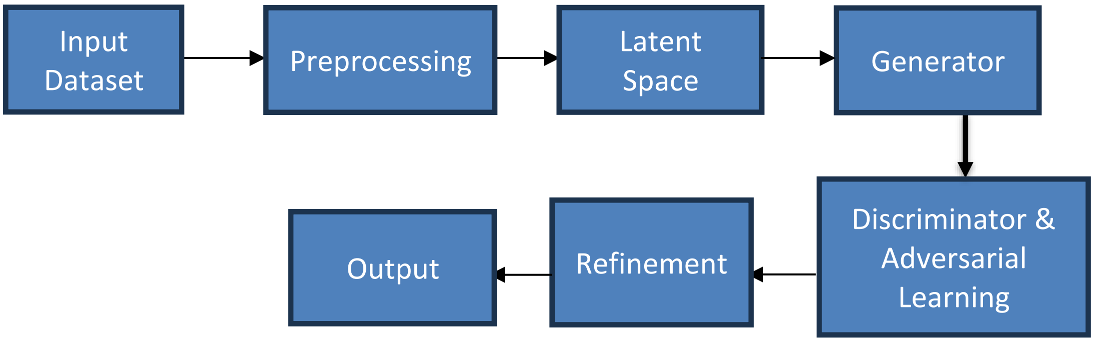


Figure 4.2.1 System block diagram of StyleGAN2

The styleGAN2 start with the data **pre-processing** , the dataset that we collected will all undergo the preprocessing steps it is a crucial and necessary steps to the foundation of a building to ensure standardization of image. The pre-processing step involve resizing the image to a uniform scale where there is a class named Dataset that will perform the preprocessing phase, the function **"resize_to_minimum_size"** was being used to resize and standardizing input images and applying random augmentation to introduce variation by using **"RandomApply"** transformation .From the pre-processing code that provided by the author , there is one step involve in pre-processing step which is the conditional augmentation. Where this step is to introduce a variety such as random horizontal flipping or color shift. This can help to make the dataset to be more diverse and help to enhance the model's ability to generalize and create variety if output.

After that , the central of the generative process is the **latent space**. Where in this step random latent vectors such as numerical representation of potential image are synthesized. These latent vectors are initially just a abstract points in high dimensional space , passing through the generator network. Each of the layes in generator will progressively perfect or refine the vectors ,shaping them with learned attributes and details. The network architecture enables the systematic transformation of random vectors into complex, structured output that gradually mirrors the target image's content and style. The function **noise_list** initiate the generation process by generating latent vector that act as abstract seed from which the images emerges. These vectors are fed through a generator network and undergo successive refinements are each layer to gradually gain detail and contents.”G(w_styles,noise)” G here stand for the generator and it is taking style and noise to generate the new images.

After the image is created, the **discriminator**, a neural network that critically observes the details, checks the synthesized images against the real samples. Its job is to distinguish created images from real ones. The discriminator estimates provide feedback to the generator, which contributes to the iterative refinement process. As the generator improves over time, it becomes more and more difficult for the discriminator to distinguish between real and generated images, indicating the progress of the generating model in achieving photorealistic results.**” real_output, real_q_loss = D_aug(image_batch, **aug_kwargs)”** , D_aug here stand for discriminator , which is evaluating the image batch of real image against those generated by model. As the generator keep learning from discriminator’s feedback , it get better and better, the discriminator has to work harder to spot the difference between the real and fake batik patterns.

The training process is shortened by gradient penalties and path length equalization. These mechanisms act as the model's internal compass, maintaining the integrity of the training journey by punishing shortcuts and deviant behavior. The goal of the system is to ensure that the creativity of the generator is based on learned characteristics that contribute to the realism and variety of the output. The evaluation phase is the final step in which the resulting images are recorded and scrutinized. Extended image sets that use clipping techniques and moving average processing provide a smooth representation of model properties. This shows the model's learning efficiency in synthesizing the essence images of the input dataset.

4.3 System Components Specification for mobile application with integration of SyleGAN2

After finish trained the styleGAN2 model with batik dataset, this section will discuss and understand how to integrate the trained model into the developed application. Below is the explanation of each block functionality.

4.3.1 Generate Batik Pattern

StyleGAN2 Model hosted in Ubuntu environment

Mobile Application

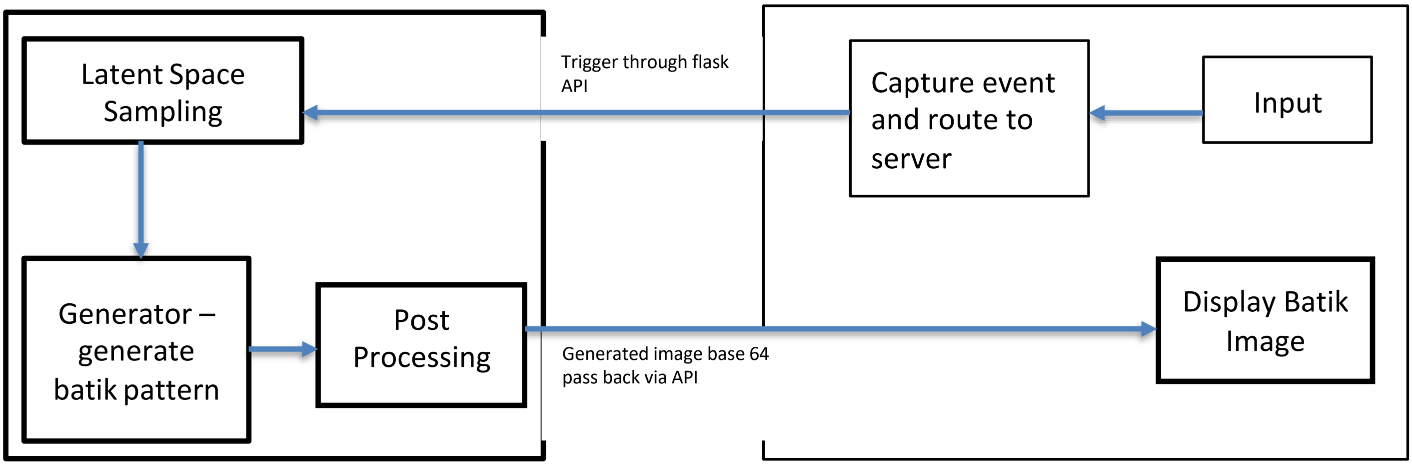


Figure 4.3.1 Component Specification of Generate Batik Pattern

User Interaction

Input and pre-processing step in application development

The generation process starts with an input to the system, usually a random noise vector or seed. The user can randomly generate this vector or manipulate a slider or input field to influence the result. Preprocessing is a key step to standardize this input to meet the requirements of the neural network. After the image is rendered, it is scaled to the desired fixed resolution of the model, typically using double-cube interpolation. The pixel values are normalized to the range used in the model, typically [-1, 1] or [0, 1], to ensure that the statistical distribution of the inputs matches the distribution of the data used in the model.

Server Flask API

The API is the channel through which the mobile application communicates with the StyleGAN2 model hosted on the Ubuntu server. After receiving input from the mobile application, the API begins by paste the generate command in the terminal for generate the batik results. This process starts with the mobile app sending inputs to the user's API, which the user can change, for example, by adjusting sliders or input fields. The API then processes this preprocessed input and passes it to the latent space of the StyleGAN2 model for sampling.

Latent Space Sampling

In the StyleGAN2 architecture , the preprocessed input is mapped to a latent space, a multidimensional fabric where each coordinate corresponds to a latent batik pattern. The task of mapping requires the use of a process called style blending, which involves learning how different parts of the latent vector affect the image's appearance, such as fine lines or complex shapes. This potential vector is the seed of image growth. It encapsulates the underlying features and styles that can be represented visually, depending on the learning models included in the model and the complexity of training on the batik dataset. From the code , noise vector of size 512 is generated using **torch.randn(1, 512).cuda()** to act as the initial seed for the generation process. That vector is crucial because it brings randomness to the system, ensuring the same pattern for many different outputs. Running the operation on the GPU, as demonstrated by **.cuda(),** significantly speeds up the computation process, highlighting the synergy between advanced neural network architectures and powerful hardware accelerators.

The noise vector is then converted to a style vector by the mapping network by calling **loader.noise_to_styles(noise, trunc_psi = 0.7)**. This step is crucial because it refines the random noise into a structured form that determines the stylistic elements of the resulting image. The trunc_psi parameter balances variation with consistency and ensures that images are not only unique, but also visually pleasing. After that with

loader.styles_to_images(styles), which converts style vectors to specific images. This feature demonstrates StyleGAN2's core ability to transform abstract mathematical representations into concrete visual art, a process that reflects the complexity and creativity of human artistic expression. Finally, the **save_image(images, './sample.jpg')** function of the **torchvision.utils** module allows you to save these created images.

Synthesis of batik Pattern in Generator

The generator, which consists of several convolutional layers, each building on the last layer, builds the image step by step. It starts by creating broad shapes and colors and builds on them with subsequent layers to add details such as intricate patterns that symbolize Batik art and repeating geometric patterns. The generator uses Adaptive Instance Normalization (Ada) at each layer, enabling latent performance ,vectors modulate feature maps stylistically.

Flask API pass back data in base 64 to mobile application

Once the generator in the Ubuntu environment has created series of frames batik pattern, a 64 base image returned via API back to the mobile application. Post-processing steps are done here to reconstruct and refine batik patterns for display. The API ensures that the image data is properly formatted and transferred between the server and the mobile application, enabling a smooth and responsive user experience.

Post-processing

Once a Batik pattern image is generated, post-processing is required to prepare it for display to the user. In the post-processing step, using the Flask framework, a function named `DivideImage` is invoked via the route `/dimage/`. This function processes the image by dividing it into smaller segments, enhancing its suitability for rendering in different image views and improving visibility by slightly enlarging the segments.

The function begins by defining two paths: `folder_path`, where the original image is stored, and `folder_path2`, where the segmented images will be saved. It reads the first image from `folder_path` using OpenCV's `cv2.imread` function and determines the image's dimensions (height and width). The image is then divided into 64 equal grids (8 horizontally and 8 vertically) by calculating the height (`h`), width (`w`), and coordinates (`x` and `y`) of each segment. A cropping operation is performed for each segment, creating a clipped portion of the original image. Each clipped segment is saved as a JPEG file in `folder_path2`, with a unique file name consisting of a timestamp and grid indices.

This post-processing step ensures that the image segments are optimized for display, allowing them to be rendered properly in different image views and slightly enlarged for better visibility.

4.3.2 Virtual Try On

Below diagram provides a general idea of how a mobile application interacts with a Flask server to generate a virtual trial image of Batik mode on a selected human model. The concept is developed collaboratively, and detailed processing steps will be handled by the collaborators.

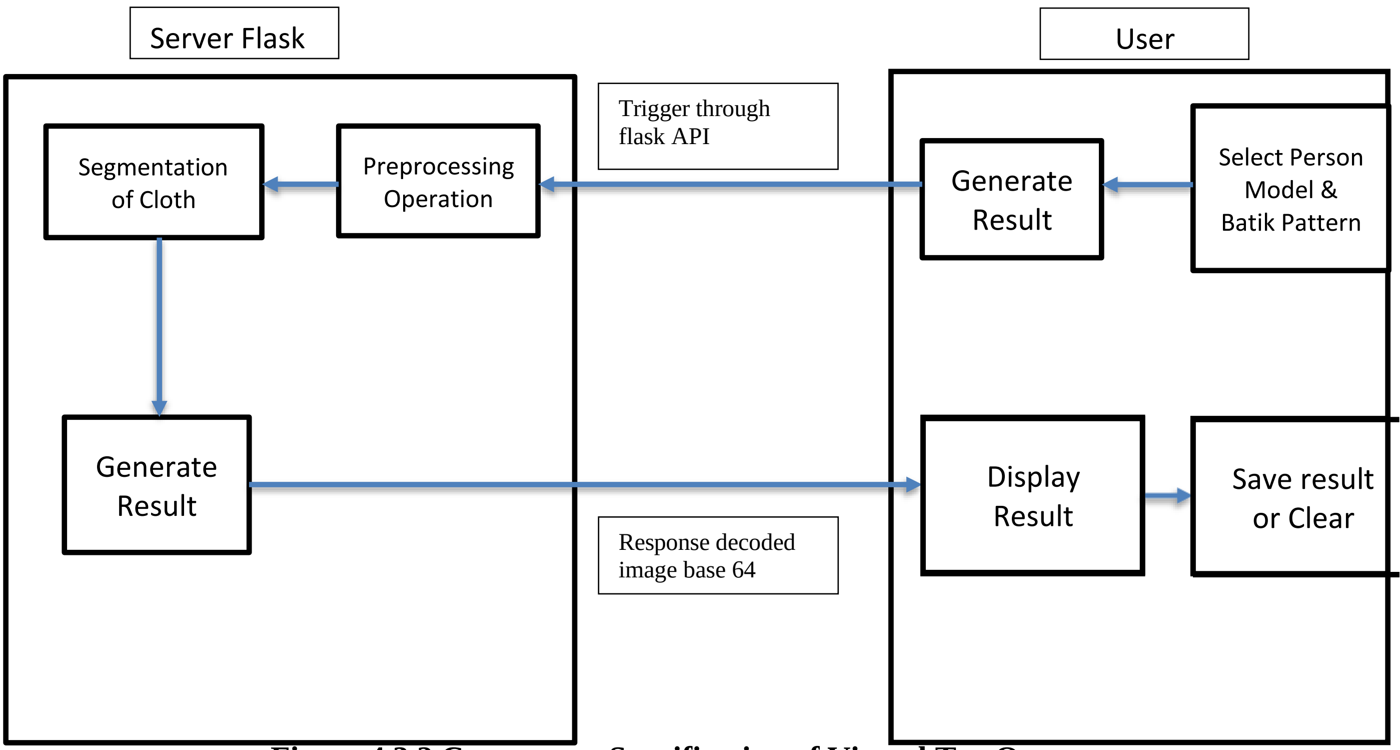


Figure 4.3.2 Component Specification of Virtual Try On

User Interaction

Selecting Model and Batik Pattern

The process begins when users select the persona and Batik modes in the Android app, providing a customized virtual trial experience by selecting the model (persona or mannequin) and Batik modes to be visualized. The choice involves browsing through a variety of models and Batik designs to provide users with a tailored experience.

Generating Results

Once a choice is made, the user clicks the **"Generate Results"** button in the application. This action sends a request to the server via the Flask API, triggering backend processing to generate a virtual trial image.

Server Preprocessing Operation

Flask servers handle the intensive processing tasks required to create realistic visualizations. The first step on the server side involves preprocessing operations that standardize and prepare input data to meet the requirements of neural networks. This step may include resizing the image, normalizing pixel values, and other adjustments to ensure data consistency and optimizing inputs to produce accurate images.

Segmentation of Cloth in Server

Server then performs cloth segmentation. In this step, the server segments the clothing worn by the selected person model, effectively isolating the clothing from the rest of the image. This subdivision is essential to accurately apply Batik patterns to the right areas of a model's clothing. Using advanced image processing techniques, the server identifies clothing areas and separates them from the background and other non-relevant parts of the image. This process reduces computational complexity and improves accuracy by ensuring that Batik patterns are applied only to predetermined parts of the image.

Generating Virtual Try-On Result

After preprocessing and structural segmentation, the server uses the VTON-HD*(virtual fitting network for high-resolution image synthesis) to generate virtual fitting results. VTON-HD applies the selected Batik pattern to the segmented clothing area of the selected person model. The model effectively combines fabric design with clothing to create realistic virtual fit images, taking into account details such as texture, folds, and lighting.

Send response image to user

After generating the image, the server encodes it in Base64 format and sends it back to the mobile application via the Flask API. Base64 encoding ensures that binary image data is transmitted securely over the network, maintaining the integrity and security of the data. The mobile app decodes the Base64 encoded image and displays it to the user, allowing them to visually evaluate the results of the virtual fitting. Users can see how Batik patterns appear on clothes worn by models, providing valuable tools for design evaluation and decision-making.

4.3.3 Customization of Batik Pattern

In Android applications customized for Batik mode, multiple components work together to provide users with an interactive and dynamic experience for creating, modifying, and saving Batik designs. The system allows users to upload images, use OpenCV to detect edges, draw detected edges as the basis for painting, apply painting and filling algorithms, erase parts of the image, and finally save custom designs. The key components of the process are outlined below, with a detailed explanation of its function and purpose.

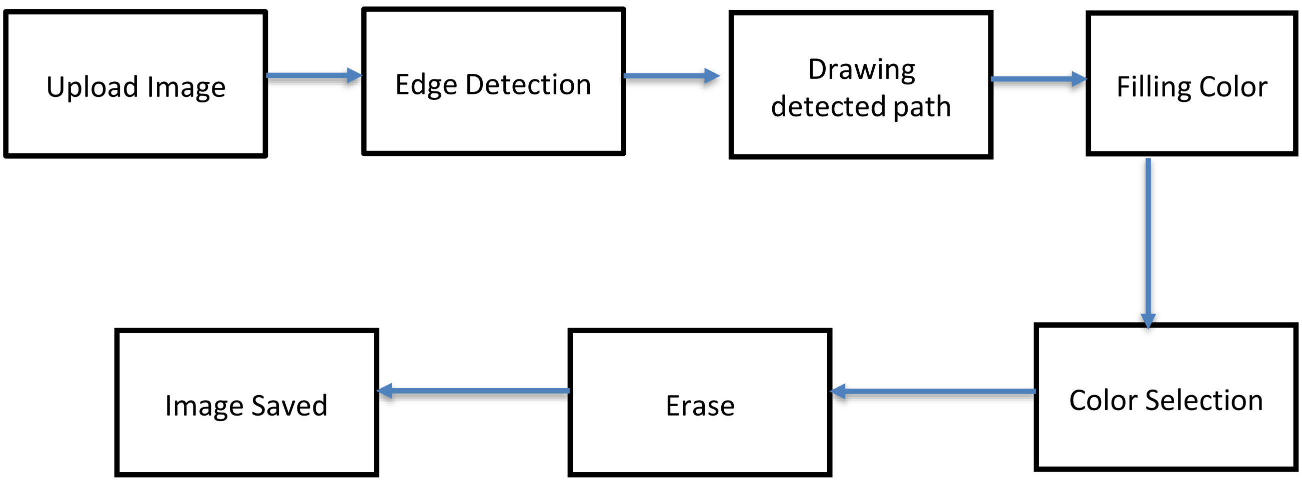


Figure 4.3.3 Component Specification of Customization of Batik Patterns

Upload Image

The Upload Image component is the first step in the application workflow. This component allows users to select images from the device's storage, which will be used as a basis for further customization. When selected, Android Intent with operational intent will be displayed. Create ACTION_GET_CONTENT, prompting the user to select an image file from the library or file manager. The selected image is then processed and displayed on the application's drawing canvas. The selected image is read as a bitmap, providing a manageable format for further operations, including edge detection and color filling.

This component is critical because it lays the foundation for all subsequent processes. Bitmap formats allow efficient pixel operations, which are essential for edge detection and custom operations. In addition, it ensures that users have full control over their image choices, thereby increasing user engagement and satisfaction.

Edge Detection

Batik's custom application includes edge detection components that are essential to recognizing and displaying contours in uploaded pictures. This procedure prepares the groundwork for next stages, which include painting, sketching, and filling in particular spaces with various colors. The program makes use of the OpenCV package, a potent resource that is frequently employed for image manipulation and computer vision processing. To begin with, this library assists in executing a number of image processing methods, such as edge detection for usage as personalized guides.

Imgproc, an OpenCV function, is first used in the edge detection procedure. The uploaded image is converted to gray scale using cvtColor(). By reducing picture data from three color channels (red, green, and blue) to one intensity value per pixel, grayscale conversion simplifies image data. This decrease is important because it reduces the amount of data that has to be processed by the edge detection algorithm in order to capture the essential variations in light intensity that characterize the edge.

After the image has been converted to grayscale, add Gaussian blur using Imgproc. A smoothing method that aids in lowering picture noise and detail is called GaussianBlur. Noise in image processing is the term used to describe arbitrary pixel value changes that can interfere with edge identification and do not yield useful information. By reducing the influence of these irrelevant information, the algorithm is better able to locate critical edges in the image by blurring it. The smoothness is controlled by the kernel size, which also determines the degree of blurring. Through meticulous calibration, the algorithm is guaranteed to preserve fundamental image characteristics while removing superfluous noise.

Imgproc used the Canny edge detection technique after smoothing. Canny() This algorithm works very well at identifying a broad variety of edges in different kinds of pictures. It uses a step-by-step process, beginning with the computation of picture gradients to pinpoint regions with notable intensity variations. By using the Sobel operator to determine each pixel's gradient direction and size, the Canny approach makes it possible to identify possible edges by monitoring the rate at which pixel values vary.

The edges are then fine-tuned to a single pixel width by the algorithm using non-maximum suppression, which removes any pixels that are not local maximums. Through the retention of only the strongest and most relevant edges, this stage assures precision. Next, two specified thresholds are used to apply double thresholds, which differentiate between strong, weak, and non-edge data. Strong edges are identified by pixels with gradient sizes above the upper threshold while noise is eliminated by pixels with gradient values below the lower threshold. Weak edges are those pixels that are between the two thresholds; they are only reserved if they are related to strong edges. This method guarantees that only the most significant edges are kept while further reducing noise.

Tracking the edge using hysteresis is the last phase in the Canny edge detection procedure. Currently, the algorithm only keeps the weak edge that is related to the strong edge after verifying the relationship between the two edges. This technique aids in creating continuous edge representations, guaranteeing the accuracy, clarity, and noise-free generation of edge mappings.

The program employs morphological procedures, such dilation and erosion, to further refine the edges once they have been recognized. These processes are carried out using OpenCV's Imgproc.morphologyEx() function. Morphological closure, which entails dilatation and erosion, aids in bridging narrow spaces and joining disparate edge segments. Dilation efficiently smoothes the contours and removes small flaws by enlarging the bounds of the detected edges and reducing them back.

Following edge refining, the picture will be processed and then converted to bitmap format so that it can be used with the Android canvas drawing framework. Because the application's drawing capability relies on a bitmap, this conversion is essential. DrawingCanvas is then used by the application. By rendering precise edge mappings onto a canvas, the drawBitmap() function gives the objects in the picture a distinct visual contour. Users may examine and interact with edges directly in this visualization, which serves as a reference for further custom actions like sketching and filling.

Because edge detection gives users distinct limits, it is an essential stage in the customizing process of batik. In order to carry out intricate tasks, like painting and filling, without impacting unanticipated regions, users must respect these limits. The final customized Batik picture quality is strongly impacted by the precision and clarity of the edges that are identified. More intricate and aesthetically pleasing designs are made possible by accurate edge detection, which guarantees that changes are exactly contained to the intended region.

Furthermore, there is a smooth transition between the edge detection components and the remainder of the program. The Drawing Detected Path component uses the detected edges as a point of reference and interacts with the user to display and enhance creative possibilities. The definition of the borders of color fill is another important function of edges for fill color components. The flood-fill algorithm prevents fresh colors from spilling into other sections of the image by limiting them to the anticipated area utilizing edges as a barrier.

Drawing Detected Path

After identifying the boundaries of the uploaded picture, the Drawing Detected Path component takes over and is essential to displaying these edges on the canvas. DrawingCanvas is used by this component. The bitmap representing the detected edges is rendered directly into the drawing canvas via the drawBitmap() function. By doing this, it gives the user an exact and transparent outline of the picture profile, which serves as the foundation for additional customisation.

As it offers a clear reference for later activities, such painting or filling a specific region, this visualization phase is essential to improving the user experience. Users are better able to notice the boundaries they may modify by highlighting the identified edges. Because of the accuracy with which these margins are generated, users may securely make adjustments to certain sections of the image without worrying about accidental overflow or overlap. The drawing modes supported by the component, which include DRAW, ERASE, and FILL, enable users to work with pictures in a multitude of imaginative ways. For instance, users may draw directly on identified routes in DRAW mode, erase portions of the drawing in ERASE mode, and add colors within predefined borders in FILL mode. This adaptability increases the app's creative flexibility and gives consumers a more customized and interesting experience.

Filling Color

Another key feature of the program is the Fill Color component, which lets users apply their preferred color to particular regions of the image. The fillArea() method's implementation of the flood-fill technique makes this operation easier. The algorithm identifies the target color for each spot a user picks on a picture and keeps replacing it with a new color the user selects.

The flood-fill approach, also referred to as the "boundary fill" method, fills the whole linked region with the desired color by repeatedly or recursively scanning neighboring pixels until it comes across pixels of different hues. The method is reliable and efficient, guaranteeing that filling processes stay contained inside a defined region and do not seep into the surrounding areas of the picture.

This ability is very helpful for creating batik designs, as intricate and distinctive patterns need filling in different regions of the design with different colors. Users may play around with different color combinations to expand the application's creative potential. The visual integrity of intricate batik designs depends on the capacity to precisely manage filled regions, which guarantees that each component is distinct and aesthetically pleasing.

Color Selection

Users may choose from a variety of interfaces when choosing colors for drawing or filling thanks to the color selection component. AmbilWarnaDialog, a color selector library that gives users access to an extensive color wheel from which they may pick any color, powers the component. Next, drawView. The active color is set to the desired color using the setDrawColor() function.

This function makes sure that the chosen color is applied consistently throughout the customizing process throughout any further drawing or filling processes. Users may experiment with various color combinations and patterns because to this component's versatility, which enhances the ability to create a very customized Batik experience. The app's large color selection inspires users to be imaginative and try new things, which eventually results in more distinctive and varied creations.

Erase

The purpose of erase components is to enable users to take out undesirable portions of their creations. Using PorterDuffXfermode (PorterDuff). mode.CLEAR) technology, the program will delete pixels in the designated region by generating a circle full of transparent colors when the erase mode (mode. ERASE), where the user may swipe over the area of the image, is active.

This function, which enables users to undo or fix mistakes without having to start from scratch, is essential to enhancing the design. This focused erase tool allows users to make exact alterations that improve creative control, unlike the basic "clear all" capability. The program makes sure that users may fine-tune their designs down to the last detail by allowing users to erase particular portions of images. In the case of intricate Bartik designs, where even little mistakes might compromise the entire appearance, this accuracy is especially crucial.

Image Saved

The final stage of the customized process is the Image Saved component, which enables users to store edited photographs to the device's storage. The draw canvas's current state is bitmapped and saved to an external storage directory or a specially made application folder by the saveImageToStorage() function.

Preserving the user's creative effort requires taking this vital step. The saving procedure makes sure that the design's integrity is preserved by flattening all of the layers, edits, and colors made during the session into a single picture file. After the image has been saved, it may be utilized for a number of things, such posting it on social media, printing it on fabric, or doing more editing using other programs. With the help of this feature, the user's creative process comes to a close and they may share and save their unique batik design for posterity.

Applications enable strong workflows that lead users from the first picture upload to the last save step by smoothly combining these components. Every component builds upon what the one before it produced, resulting in a unified and simple user interface that encourages customization and innovation. This connection retains the final design and guarantees consistency across the customizing process, from edge detection to color applications.

4.3.4 Batik Cloth Transfer

An outline of the essential ideas of the method is given in this section. Our partners' specialized sections will include specific information, such as technological approaches, implementation stages, and comprehensive methodology. To set the stage for more in-depth investigation, we offer below a general overview of the procedure.

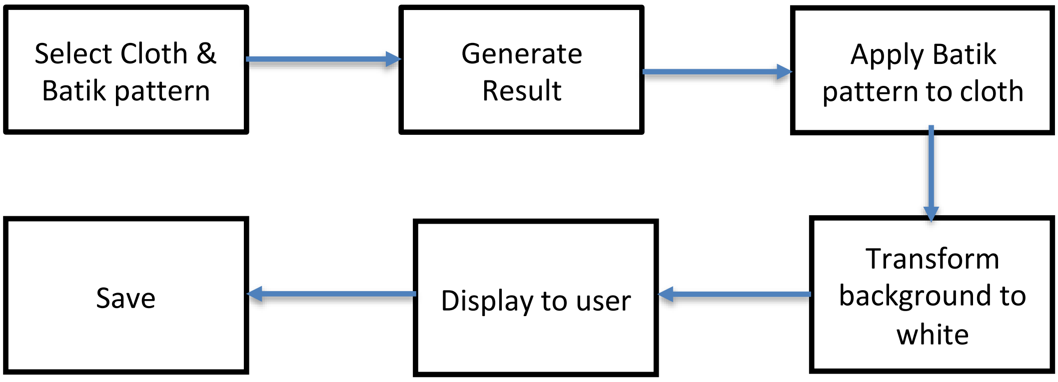


Figure 4.3.4 Component Specification of Batik Cloth Transfer

Select Cloth & Batik Pattern

Choosing the fabric and Batik design at the outset of the procedure is a fundamental step that establishes the framework for the remainder of the customized workflow. A user-friendly interface that allows customers to select from a range of fabric alternatives and Batik designs helps reach this objective. A preloaded collection of Batik designs and different kinds of fabric textures, colors, and styles are provided by the selecting interface. Additionally, users have the flexibility to upload personalized textiles or Batik designs straight from the device or app store. This part is essential because it gives consumers a creative sandbox to experiment with various appearances by letting them see their initial selections before making any changes. Once chosen, the fabric and pattern choices will be saved and ready for the following action, guaranteeing that the program efficiently manages the user's precise preferences.

Generate Result

The "Generate Results" step of the system is initiated once the user has made a selection. In order to produce an accurate virtual representation of the chosen fabric and Batik design, this step entails challenging computational difficulties. This procedure starts with scaling and placing the Batik design onto the cloth template. In order to modify Batik's design to fit the precise dimensions and profile of the chosen cloth, an algorithm is needed. After that, texture mapping techniques are used to merge the pattern with the fabric texture, simulating the cloth's physical characteristics like folds, curtains, and shadows.

The quality and authenticity of virtual representations may be raised with the application of sophisticated AI models, such as Generative Adversarial Networks (GANs). Models such as Virtual Try-On High Definition (VTON-HD), for instance, can enhance the combination of textiles and Batik patterns, guaranteeing a smooth integration that emulates real-world applications. The final result of this computationally demanding stage is high-fidelity user design visualizations, which greatly depend on the device's processing capacity to accomplish these complex rendering tasks.

Apply Batik Pattern to Cloth

The "Apply Batik Patterns to Cloth" component is activated when the results are obtained. This component entails adding Batik designs to particular fabrics in an elegant and natural manner. The method overlays Batik motifs onto fabric templates by using sophisticated image processing algorithms. In order to guarantee that the Batik design fits precisely without distortion or overlap, one of the primary strategies utilized here is edge detection, which locates the borders of the fabric template. Another technique for modifying the pattern's direction and proportion to fit the fabric's natural form and contour is called pattern warping. The objective is to produce a realistic portrayal that preserves the integrity of the Batik design while making it appear as though it has been organically woven into the cloth.

Transform Background to White

When the desired outcomes are achieved, the Apply Batik Pattern on Fabric component is triggered. This component calls for the tasteful and organic application of Batik designs to certain textiles. This technique uses sophisticated image processing techniques to superimpose Batik designs on cloth templates. Edge detection, which finds the limits of fabric templates, is one of the primary tactics employed here to guarantee that Batik designs match precisely without distortion or overlap. Pattern warping is a different method for altering a pattern's direction and proportion to match the fabric's natural shape and contour. The goal is to depict Batik's pattern realistically while maintaining its integrity and giving the impression that it was woven naturally into the fabric.

Save Result

Ultimately, users may save their unique designs for later use by using the Save component. Users have the option to save the finished image to cloud storage services or device storage after checking their work and making any required corrections. The Save function takes a picture in its present state, with all of its layers, edits, and effects applied, and saves it as a JPG or PNG file. This process guarantees that customers have an everlasting documentation of their inventiveness and enables them to distribute, print, or alter their works further. The process of saving components is the apex of a customized journey that offers users a fulfilling way to wrap up their creative experience.

4.3.5 View Tutorial Video

With the help of the tutorial video module, users may access tutorial videos straight from the program, giving them an interactive learning experience. Using video presentations, the module walks users through the creation of Batik, virtual fitting, customization, and fabric transfer, among other features of the tool. Users can see a tutorial, choose a particular tutorial, watch a video, adjust playback, and dismiss the video dialog by following the steps outlined in the flowchart. A thorough explanation of every element included in this module is provided below:

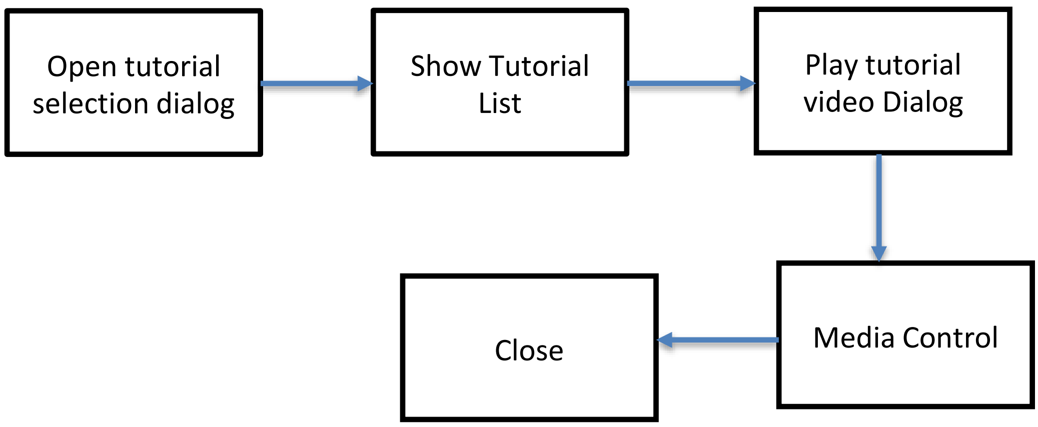


Figure 4.3.6 Component Specification of View Tutorial Dialog

Open Tutorial Selection Dialog

When the user selects the help icon (ivHelp), the Open Tutorial Selection Dialog component opens, initiating the procedure. The available tutorials are listed in a custom dialog that is opened by this action (showTutorialListDialog()). Several instructional choices, including Creation Batik, Virtual Try On, Customization, and Cloth Transfer, are displayed in the dialog using ListView. This part is in charge of designing an intuitive user experience that makes it simple to go through the tutorials that are accessible. To guarantee that any additions or modifications to the instructional list are instantly reflected in the user interface, ListView is populated with ArrayAdapter, which dynamically sets the tutorial name.

Show Tutorial Lists

The Show Tutorial List component is activated after the tutorial selection dialog box has been opened. This part controls how a list of tutorials appears and how users may interact with it. Every item on the list links to a particular instructional video. The name of the selected tutorial and the accompanying video resource ID are passed to the TutorialDialog() function, which displays when the user clicks on a list item. The listener records the event. After that, the conversation will end so that the video may resume. A smooth transition from selection to playing is essential to preserving a user-friendly interface.

Play tutorial Video Dialog

The Play video Video dialog box opens when a user chooses a video from the list. This part is in charge of making a second custom dialog to show the chosen video. A VideoView in the dialog box dynamically determines the appropriate video URI in accordance with the user's selection. Additionally, it has text components like tutorialTitle, which gives the playing video context. With features like playback, pause, and recovery, users may customize playing to suit their tastes thanks to VideoView's media controls. To make these controls easier, MediaController is configured to use VideoView.

Media Control

The Media Control component makes the viewing of instructive videos more interactive. The user has the option to click on the video to pause or restart playback. By briefly showing an indicator that indicates the current status (play or pause), the showPlayPauseIcon() function improves user feedback. With MediaController, the user has total control over how videos playback—they may fast-forward, rewind, or move the timeline. The smooth and responsive playing is made possible by the VideoView component's setOnClickListener(), which makes sure that state changes (pause or playback) are handled appropriately.

Close Video Dialog

Users may always close the tutorial video dialog by using the Close component. The video dialog has an ImageButton (btnClose) to close it. tutorialDialog is called by onClickListener upon the click of the Close button. dismiss() eliminates the dialog box from the display. Make a call to tutorialVideo as well. stopPlayback() stops unexpected background activity, frees up resources, and stops video playing. This element contributes to a simple and easy-to-use interface by making sure users can easily quit the instructional and go back to the main screen.

4.3.6 View Gallery

An application's view library is a crucial component of its image management system. Users may view, pick, like, and remove photographs from the gallery using this component. By offering a number of features for effectively managing photos, it is intended to improve the user experience. Each subcomponent's functioning is described in depth in the sections that follow.

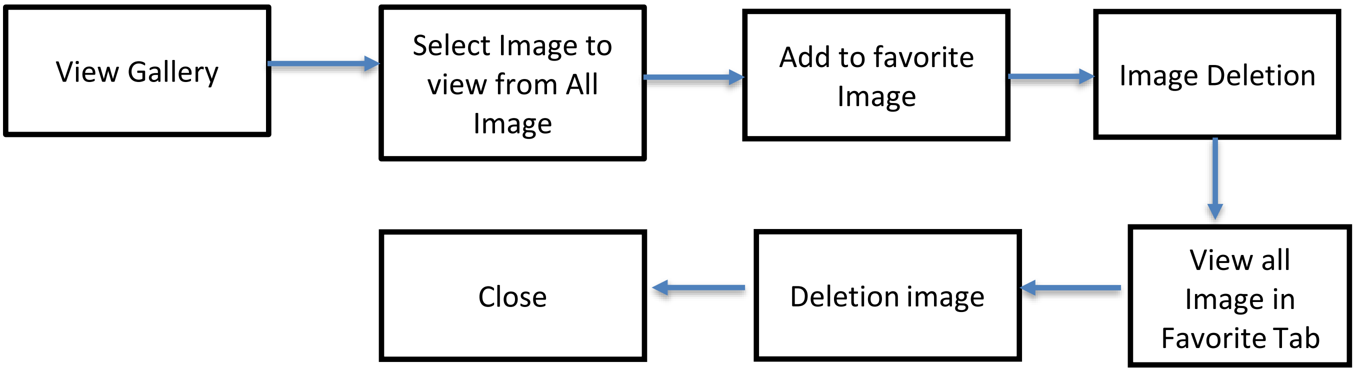


Figure 4.3.6 Component Specification of View Gallery

View Gallery

The user may browse all of the available photographs in the library by starting at the view library. Users may peruse the collection of photographs saved in the program through an organized display. This component usually allows users to examine numerous photos at once by displaying them in a grid manner utilizing layouts similar to RecyclerView. The dynamic updating of gallery displays occurs in response to user inputs, such as the addition or deletion of photographs. It provides the foundation for utilizing additional functions, such as choosing a picture, adding to favorites, and removing.

Select Image to View from All Image

The user may choose a picture from the gallery to examine details by using the Select picture subcomponent. An image opens in a bigger size and typically opens in a different ImageDetailActivity format when a user clicks on it. This subcomponent detects user clicks and displays the selected picture by using a listener method. The feature increases user engagement by offering focused views, which allow users to examine the image more clearly, take actions (like favorite or delete), and carry out additional tasks (like sharing or editing).

Add to Favorite Image

Users may like certain photographs by using the Add to Favorites image function. To make it easier for people to obtain photographs they enjoy or believe are significant, this functionality is vital. The application transfers a picture to a different Favorites tab when a user chooses to mark it as a favorite. This might be anything as simple as moving an image file to a designated location or labeling it with database tags. The software notifies you visually when an image is added to your favorites list, using icons in the shape of hearts. In order to make sure that the Favorites tab displays the most current modifications, this procedure also broadcasts the intention to refresh the gallery.

Image Detection

Users can remove photographs they no longer need or desire by using the Image Delete component. With the help of this function, users may better manage storage space and maintain the organization of their collections in their galleries. To avoid unintentional deletion, users are presented with a confirmation dialog box when they want to remove an image. If verified, the program modifies the data structure containing the image reference, deletes the picture from the file system and library, and broadcasts the intent to update all pertinent views. To maintain uniformity across the program, deletion may be done from the Favorites and main gallery tabs.

View All Image in Favorite Tab

Users may view all of the photographs they have marked as favorites in one convenient location by using the View Favorites tab subcomponent. Usually, TabLayout or Fragment is used to construct this tab; these tools load dynamically and show all photos that have been copied or marked as favorites. The Favorites tab and the main gallery may be easily switched between by users. This feature makes it possible to quickly access your favorite photos without having to look through the full gallery, which enhances user pleasure and usefulness.

Image Deletion from Favorite

With regard to photographs kept in the Favorites tab only, the Delete photographs from Favorites function mimics the general deletion procedure. The system asks for approval before allowing the user to remove a picture from their favorites. Once the user confirms, the picture is either removed entirely (if saved separately in a dedicated folder) or not tagged as a favorite (if a tag is used). After then, reload the gallery to see the most recent favorites list. With the help of this function, users may maintain complete control over their favorite photographs and make sure they are current and relevant favorites.

Chapter 5

System Implementation

5.1 Hardware Setup

The hardware that will be used in the development showed in the table below

Description	**Specifications**
Model	MSI
Processor	Intel Core i9-13900H
Operating System	Window 11
Graphic	NVIDIA® GeForce RTX™ 4070
Memory	16 GB RAM
Storage	1TB

Table 5.1.1 laptop for development specification

Description	**Specifications**
Model	-
Processor	AMD Ryzen 9 3900X 23-Core Processor
Operating System	Ubuntu 22.04.4 LTS(Jammy Jellyfish)
Graphic	NVIDIA GeForce RTX 2080
Memory	64GB
Storage	1TB

Table 5.1.2 remote machine for training StyleGAN2

5.2 Software

Software that will be used in the development :

1. For the frontend development the software that will be used for create the User Interface is Android Studio.(version 2023.2.1)
2. For the backend development the software that will be used for handle request and integrate with the styleGAN2 model is python flask.

5.3 Setting and Configuration

5.3.1 Setting and Configuration for access remote machine

First section is about setting up machine for remote using machine to train the StyleGan2 model , to facilitate a seamless interface with remote computer , we set up X2GO, a remote desktop software that support efficient and user-friendly connection to remote machine.

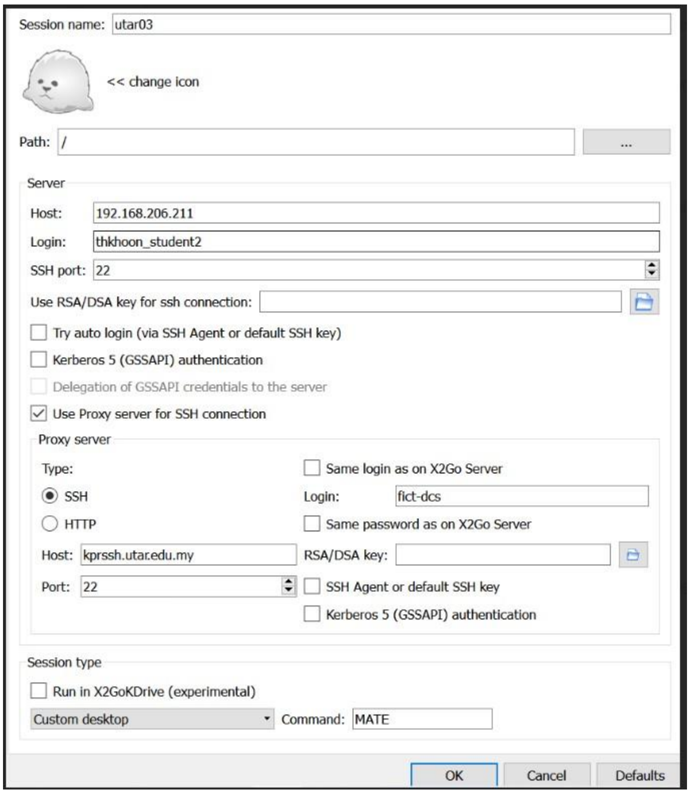


Figure 5.3.1 Setting up the X2GO for remote access to machine

First , we need to create a new session , and name the session name as our machine name (utar02 or utar03). After that ,input the IP address for the machine in the Host , user name in the Login section and port number for the SSH port section.

5.3.2 Setting and Configuration in Visual Studio Code for Flask

In this section two will showcase the setup and configuration process for creating an environment to run the flask API.

Figure 5.3.2 Install Python Extension

Figure 5.3.2 show the step to install python extension in Visual Studio Code. After installation completed, next open the project folder and select the file from the main menu as show in Figure 5.3.3 and Figure 5.3.4

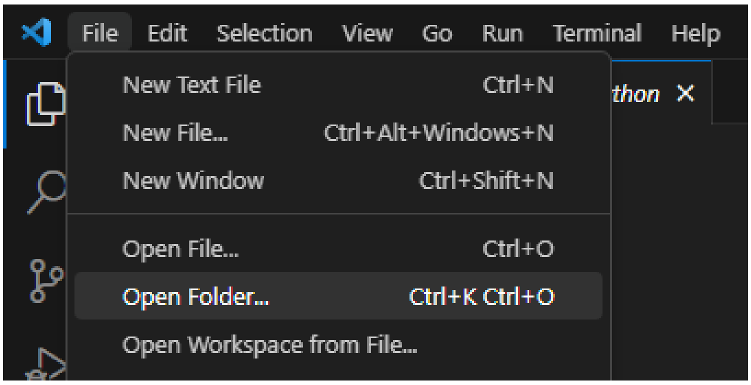


Figure 5.3.3 Open folder from menu bar

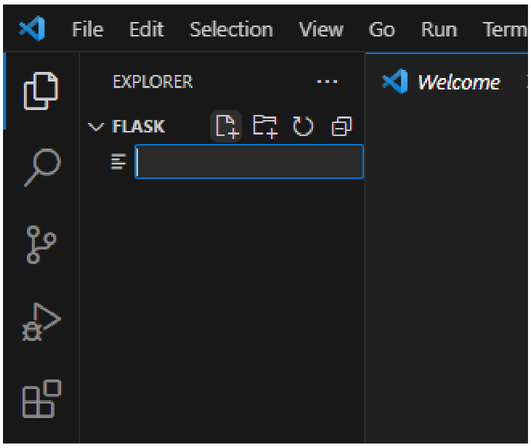


Figure 5.3.4 Select File from main menu

The Command Palette is a multifunctional feature in Visual Studio Code that allows a variety of commands to be executed. In this step, call the command palette and type in Python: Create Environment, as shown in Figure 5.3.5. As shown in Figure 5.3.6, this action prompts the system to generate a new virtual environment, which is a critical step in maintaining dependency integrity.

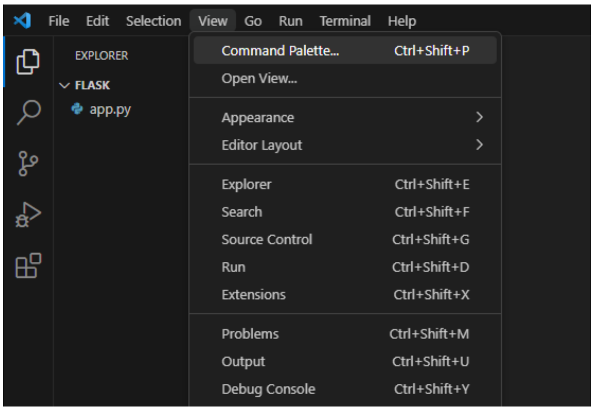


Figure 5.3.5 Click Command Palette

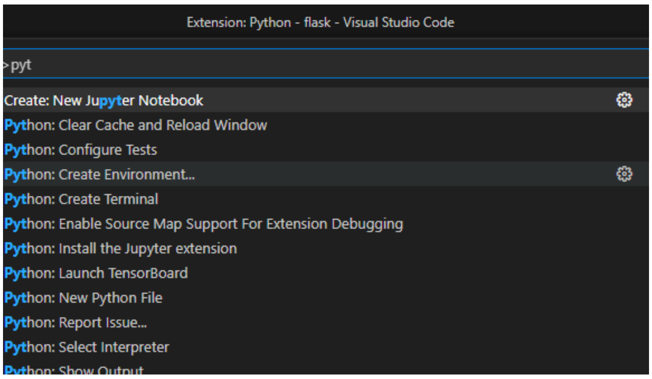


Figure 5.3.5 Select Python:Create Environment

After selecting the appropriate commands for environment creation, Visual Studio Code continues to build a new Python virtual environment, as shown in Figure 5.3.7. This encapsulated environment will install Flask and other project-specific dependencies, leaving the global Python workspace intact and free of potential conflicts between project requirements.

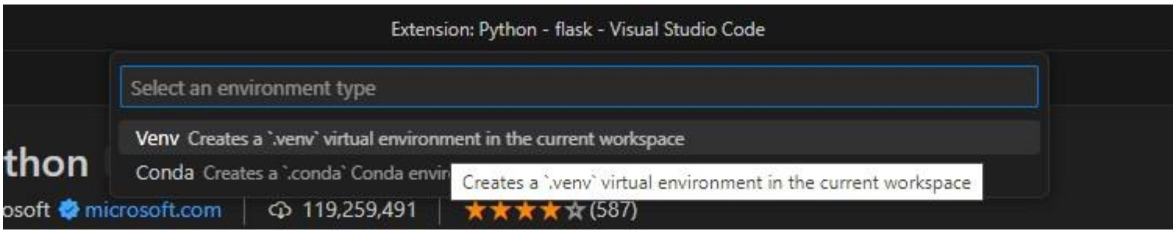


Figure 5.3.6 Create Virtual environment

Post-environment creation is the next logical step, as shown in Figure 5.3.8 .As shown in 8, it involves selecting a newly created Python interpreter. This is critical because it instructs the code editor to use Python binaries and libraries in a virtual environment, including the Flask framework and its dependencies, to tailor the development experience to the needs of the project.

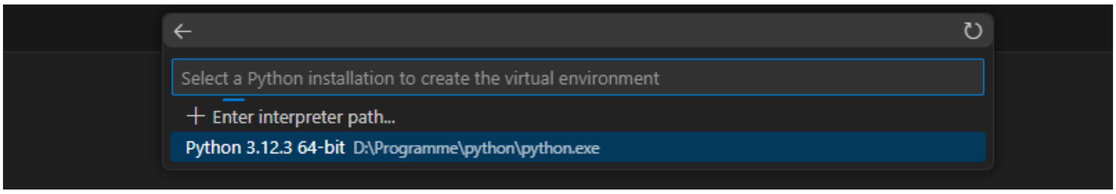


Figure 5.3.8 Choose interpreter that corresponds to created environment

The final step of the installation is shown in Figure 5.3.9.As shown in , install the Flask framework itself. Using the integrated terminal in Visual Studio Code, execute the pip install Flask command. This will install Flask in a virtual environment so that it can be used for project import and use. With Flask installed, the environment is now ready for the development of web applications and APIs, which will be discussed in the **section 5.4.1.**

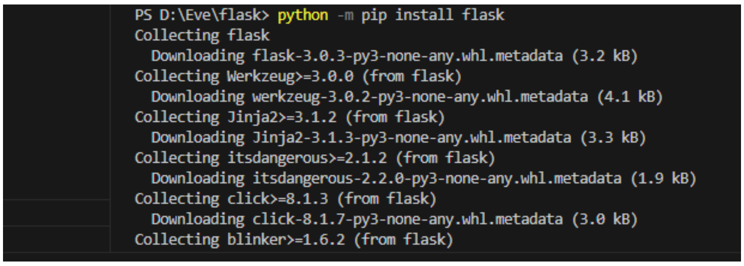


Figure 5.3.9 Install Flask Framework

5.3.3 Setting and Configuration for Debugging In Android Device

This section outlines the initial settings and configurations needed to run the application. First, navigate to the device settings and enable the developer options. Within the developer options, activate both USB debugging and installation via USB under the debugging section. Refer to Figure 5.3.10.

With these options enabled, users can connect mobile devices via USB to build and run applications for testing and implementation.

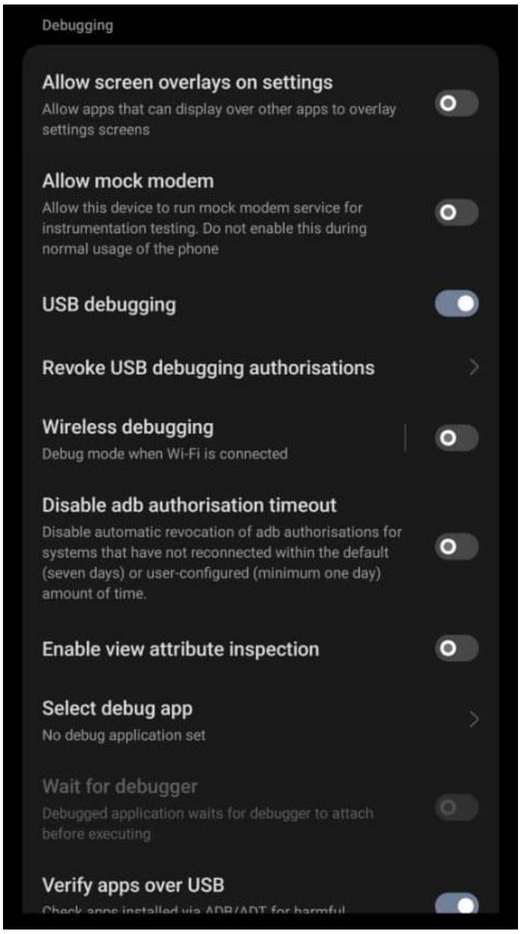


Figure 5.3.10 Set developer mode in Mobile device

Declare Permission in AndroidManifest.xml

It's also available on AndroidManifest. Declare in the xml file the permissions required for the mobile application to function properly. Here is AndroidManifest. A list of permissions that must be included in the xml file to ensure that the application works as expected. Refer to Figure 5.3.11

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"
        tools:ignore="ScopedStorage" />
    <uses-permission android:name="android.permission.INTERNET" />
```

Figure 5.3.11 Declare Permission in AndroidManifest.xml

Add related dependencies into gradle file in Android Studio

There are some related dependencies needed to be added to Gradle. Bundle for using necessary functions that other library provided. Refer to 5.3.12

```
dependencies {
    implementation project(':sdk4') // Reference to the OpenCV SDK module
    implementation 'com.squareup.retrofit2:retrofit:2.9.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
    implementation 'com.github.bumptech.glide:glide:4.15.1'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.15.1'
    implementation 'com.google.android.material:material:1.9.0'
    implementation 'com.squareup.retrofit2:converter-scalars:2.9.0'
    implementation 'com.github.yukuku:ambilwarna:2.0.1'
    implementation 'androidx.cardview:cardview:1.0.0'
    implementation 'com.github.bumptech.glide:glide:4.12.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0'


}
```

Figure 5.3.12 Add related dependencies into gradle file in Android Studio

5.4 System Operation and Implementation (with Screen shot)

This section will be divided into three subsection , **first section** showcase the practical operation for **training the styleGAN2** and install the package for train with our own dataset. The **second section** showcase the practical operation for **connect and get response back from the Flask server** that run in Ubuntu environment. The **last section** will showcase the **developed application "Batik Creator" application**.

5.4.1 Training Process in remote machine

First , after login to the remote machine we need to pip install the package using the command provided by the author.

```
(base) thkhoon_students@utar02:~$ pip install stylegan2_pytorch
```

Figure 5.4.1 Screenshot of type the pip install command in the terminal

Next, to train with our own dataset , we need to unzip and place the dataset at the directory that we want and using this command for start the training process.

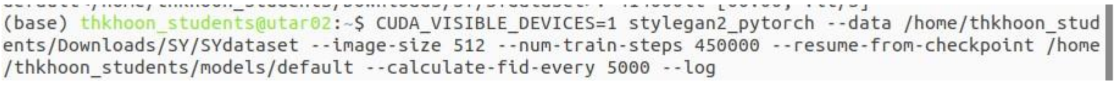


Figure 5.4.2 Screenshot of training StyleGAN2 command in terminal

This command indicates we use the GPU1 to train the model due to the limited memory constraints of the GPU0. **- -data / home / thkhoon_student / Download / SY / Stdataset** is the path where we place our dataset for training purpose. After we had trained the styleGAN2 , the checkpoint will automatically saved every 1000 iteration under the default file named as the default – model. When the training process is cut off it will automatically resume from the checkpoint. When we start the training we also can specify the calculate fid every 5000 iteration , or any number of iteration that we prefer. Below show the training process start after keyed in the command in the terminal.

Figure 5.4.3 Example of training start from checkpoint

Monitor the learning process and the loss of the models

The training of our generative adversarial network, especially the StyleGAN2 model, was carefully planned and executed. Over several epochs, we diligently monitored the evolution of generator and discriminator losses, which are key indicators of model performance. As shown in the attached loss graphs in **Figure 5.4.4**, the training showed variations characteristic of the competitive training. These fluctuations are not just minor fluctuations; they show a dynamic negotiation between the generator and the discriminator as they continuously improve during training.

In this adversarial dance, the discriminator is tasked with distinguishing between real and generated batik patterns, improving its accuracy over time. At the same time, the generator tries to produce more and more convincing batik patterns by learning from the suggestions provided by Separator. The observed oscillatory loss patterns are evidence of this learning and are a valuable measure of the stability and convergence of the training process.

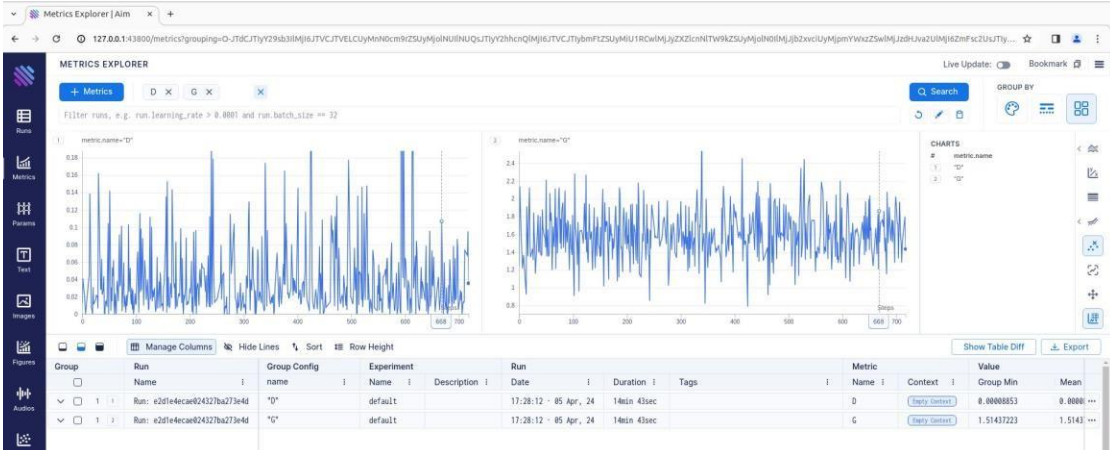


Figure 5.4.4 metric of loss of the generator and discriminator

The training result will be saved under a default file named as default-result.

Figure 5.4.5 One of the result generated during training

After we had successfully trained the model , we used the command stylegan2_pytorch - - generate to generate the result. This generated result will be stored in the same file as the training result in **Figure 5.4.5.**

5.4.2 Hosting API and Run trained model in Ubuntu Environment in local machine

After the training process is complete, the focus shifts to the deployment of the Flask API. This section discusses the process of writing APIs, setting up APIs in an Ubuntu environment on a local computer, and interacting with them to display the results generated by the training model.

The general concept is , on the server side (Flask), there are two paths: /generate/ and /image/. /generate/route Listen to the request from client and use GAN model for generate a new image in ubuntu environment. Therefore before the client side can connect to the server Flask that hosted in ubuntu environment , it is necessary to execute and host the server Flask with this line of code **python app.py** as showed in Figure 5.4.6.

```
(base) solomon@MSI:~$ python app.py
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.29.167.147:5000
Press CTRL+C to quit
```

Figure 5.4.6 Execute python app.py in Ubuntu Terminal

The process to generate content on the server side via client-initiated operations is encapsulated in the generate() function in the Flask application showed in Figure 5.4.8. When a client makes a request to the /generate/ endpoint, this function is usually triggered by a user interface element, such as a button the code snapshot in android studio is showed in Figure 5.4.9. After activation, the Flask server performs subprocess.run(command, shell=True) where command is the predefined string variable stylegan2_pytorch --generate. This is similar to manually entering a command into a terminal, which instructs the server to run Style's specified build process. The GAN model as showed in Figure 5.4.7.

```
(base) solomon@MSI:~$ python app.py
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.29.167.147:5000
Press CTRL+C to quit
continuing from previous epoch - 412
loading from version 1.8.9
100%|████████████████████████████████████████████████████████████| 1/1 [00:01<00:00,  1.40s/it]
sample images generated at ./results/default/generated-04-17-2024_10-01-11
172.29.160.1 - - [17/Apr/2024 10:01:13] "GET /generate/ HTTP/1.1" 200 -
```

Figure 5.4.7 Command generate is executed in Ubuntu terminal

```python
@app.route('/generate/', methods=['GET', 'POST'])
def generate():
    subprocess.run(command, shell=True)
    return jsonify({"Status": "Success"})
```

Figure 5.4.8 Code for generate image in terminal

```java
private void generateImage() {
    Retrofit retrofit = new Retrofit.Builder()
            .baseUrl("http://172.29.167.147:5000/")
            .addConverterFactory(GsonConverterFactory.create())
            .build();

    GenerateApiService service = retrofit.create(GenerateApiService.class);
    Call<Void> call = service.triggerGenerate();

    call.enqueue(new Callback<Void>() {
        @Override
        public void onResponse(Call<Void> call, Response<Void> response) {
            if (response.isSuccessful()) {
                Log.d("API Call", "Successfully triggered the Flask endpoint.");
            } else {
                Log.e("API Error", "Failed with response code: " + response.code());
            }
        }

        @Override
        public void onFailure(Call<Void> call, Throwable t) {
            Log.e("API Failure", "Error calling Flask endpoint: " + t.getMessage());
```

Figure 5.4.9 Code for make a network call to Flask API to trigger command generate

On the client side (Android), the displayImage function in Android code makes network calls to the /image/route of the Flask API the code snapshot is showed in Figure 5.4.10 and Figure 5.4.11. It uses a type-safe HTTP client for Retrofit, Android, and Java to handle network requests. Upon receiving a successful response, it decodes the base64 string back to the image bitmap and sets this bitmap as the source for ImageView. This ImageView is then added to the layout container to display on the screen.

```java
private void displayImage() {
    Retrofit retrofit = new Retrofit.Builder()
            .baseUrl("http://192.168.1.32:5001/") // Ensure this matches your Flask server URL
            .addConverterFactory(GsonConverterFactory.create())
            .build();

    StyleGANApiService service = retrofit.create(StyleGANApiService.class);
    service.getImageResponse().enqueue(new Callback<ImageResponse>() {
        @Override
        public void onResponse(Call<ImageResponse> call, Response<ImageResponse> response) {
            if (response.isSuccessful() && response.body() != null) {
                Log.d( tag: "APIResponse",  msg: "Raw JSON: " + new Gson().toJson(response.body()));

                LinearLayout container = findViewById(R.id.imageContainer);
                container.removeAllViews(); // Clear the container before adding new images

                String base64Image = response.body().getImageFiles();
                byte[] decodedString = Base64.decode(base64Image, Base64.DEFAULT);
                Bitmap decodedByte = BitmapFactory.decodeByteArray(decodedString,  offset: 0, decodedString.length);

                ImageView imageView = new ImageView( context: CreationActivity.this);
                imageView.setImageBitmap(decodedByte);

                LinearLayout.LayoutParams layoutParams = new LinearLayout.LayoutParams(
                        LinearLayout.LayoutParams.WRAP_CONTENT,
                        LinearLayout.LayoutParams.WRAP_CONTENT
                );
                layoutParams.gravity = Gravity.CENTER_HORIZONTAL;
                imageView.setLayoutParams(layoutParams);
```

Figure 5.4.10 Function in Android Studio make network call to Flask API's image route

```python
@app.route('/image/', methods=['GET', 'POST'])
def return_result():
    #folder_path = 'C:/Users/solomonphua/Documents/img'

    # Getting headers from the request
    user_agent = request.headers.get('User-Agent')
    content_type = request.headers.get('Content-Type')

    # Creating custom headers for the response
    custom_headers = {
        'X-Custom-Header': 'image Value',
        'Another-Header': 'image Value'
    }

    try:

        folder_path = '//wsl.localhost/Ubuntu/home/solomon/results/default'
        image_files = get_image_files(folder_path)

        # Prepare a list to hold encoded image strings
        #encoded_images = []
        response_data = {}
        #imagename = ""
        #i = 0
        # Encode each image and add it to the list
        for img_file in image_files:
            with open(img_file, "rb") as image_file:
                encoded_string = base64.b64encode(image_file.read()).decode('utf-8')
                #encoded_images.append(encoded_string)
                #imagename = "image_files" + str([i])
                response_data = {"image_files" : encoded_string}
```

Figure 5.4.11 Code for pass back image result from Flask API to Android Studio

5.4.3 System Operation of the application

This section will show the operation of the Batik Creator Application.

Application Launch and Navigation

In the initial screen of the application, as depicted in Figure 5.4.12, users are greeted with a vibrant and visually engaging interface. The header prominently features the app's name "BATIK," styled creatively to match the traditional batik theme. Below, the user is presented with several options to navigate through the app, such as "Batik Creation," "Virtual Try-On," and "Batik Customization." These options provide quick access to various features of the application. Additionally, a section labeled "My Collection" displays a grid of the user's saved or favorite batik patterns. This personalized section allows users to view and manage their custom designs or creations, with an option to add new ones by clicking on the "+" icon. The overall layout is user-friendly, aiming to provide a seamless experience for exploring and interacting with batik patterns.

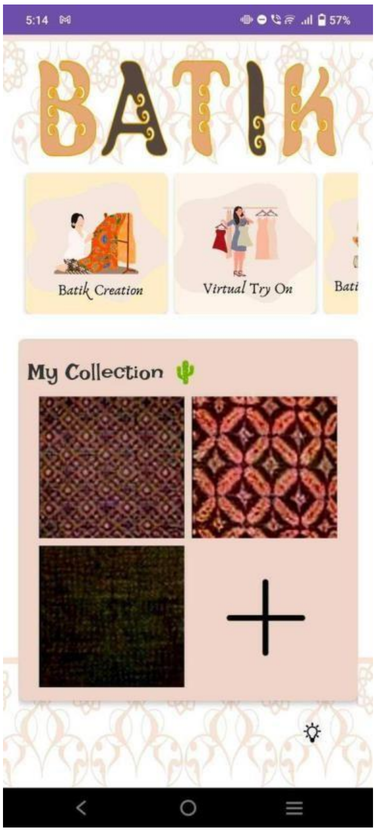


Figure 5.4.12 Main Menu Page

Creation Page Batik Patterns

The application's create pages are intended to give users a dynamic interface via which they can create and alter Batik patterns. Users may get a preview of the Batik pattern they are currently working on at the top of the page. The interface has many sliders with labels such as "complexity," "saturation," "contrast," and "brightness," which enable the user to instantly modify the pattern's aesthetic qualities. Users may experiment with different combinations by pressing the Generate Mode button, which starts the creation process depending on the specified parameters. To save the finished design, click the "Save Image" button located beneath the preview. Additionally, users may browse the gallery of pre-existing designs, choose a favorite, and utilize the tools at their disposal to customize it to their liking. By flipping the Try Mode switch, the user may create a customized and interactive pattern creation experience by seeing how the chosen design looks on the cloth right away.

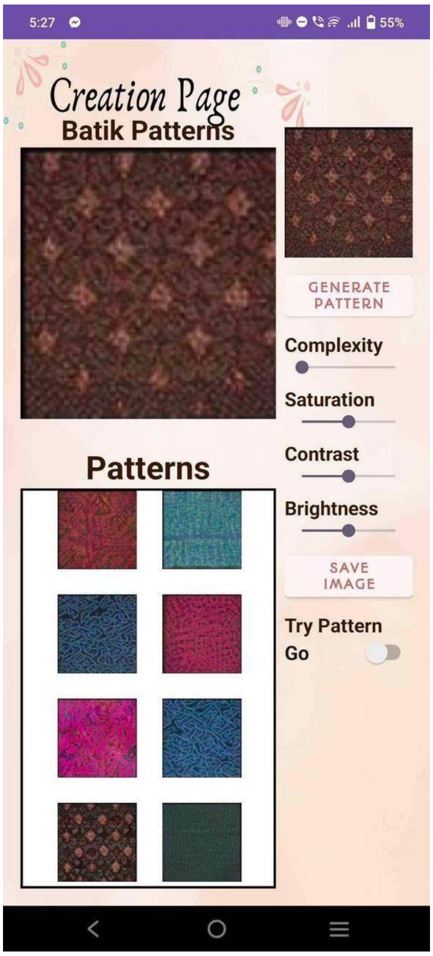


Figure 5.4.13 Creation Page

Batik Cloth Transfer Page

Users may add their own or chosen Batik patterns to shirts and jackets, among other clothing items, by visiting the Batik Cloth Transfer Page. You may choose from a variety of wardrobe options, each of which is represented by a picture, at the top of the page. After selecting the desired article of apparel, users can submit their own design or select from a selection of Batik patterns. The Clear button restores the option for more experimentation, while the Generate button shows the user the finished model of the garment with the applied pattern. This interactive portal connects traditional Bartik art with contemporary fashion by allowing users to visualize and personalize their garment designs in an entertaining and engaging way.

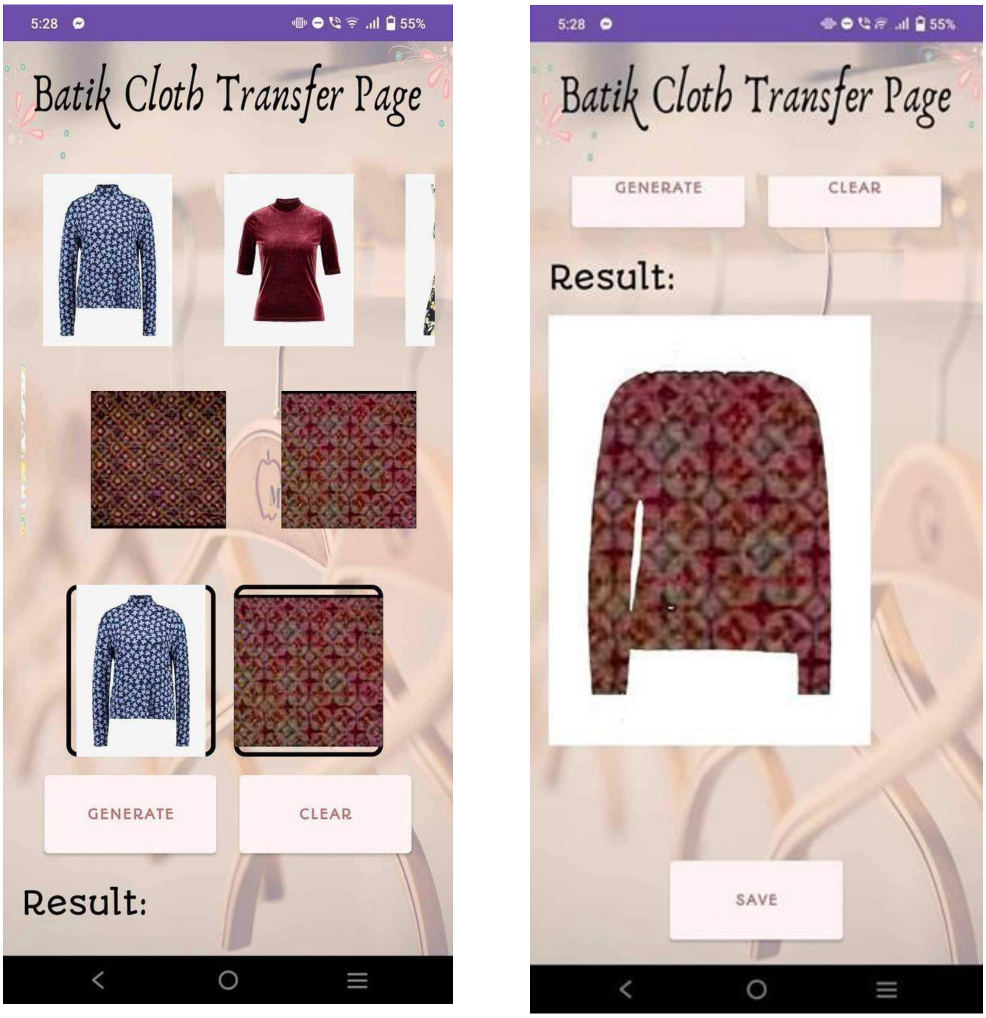


Figure 5.4.14 Batik Cloth Transfer Page and result

Customization of Batik Pattern

The app's custom batik patterns are intended to provide users a rich, interactive experience that lets them add color to uploaded batik designs. according to Figure 5.4. 15 The basic template for personalizing the procedure is the black and white batik design, as seen in Figure 14 of the first interface. Users may alter different portions of the design by selecting from a range of colors and tools. To fill in the color, users can select a paint bucket, pencil, or brush from the toolbar at the bottom. This set of tools is intended to provide precise coloring or sketching in a particular region of the pattern.

The second portion of Figure 5.4.16 which displays the batik pattern after the user applies the color, demonstrates how the technology precisely fills the chosen region while maintaining the intricate features of the batik design. The user's capacity to imaginatively alter the cloth design is demonstrated by the dark red backdrop and the filling of various flower motifs with white and green. The application's objective of enabling users to create original and customized batik designs depends on this personalization. Users have freedom and control over their creations because to the interactive nature of the color filling process, which makes for an interesting and imaginative experience.

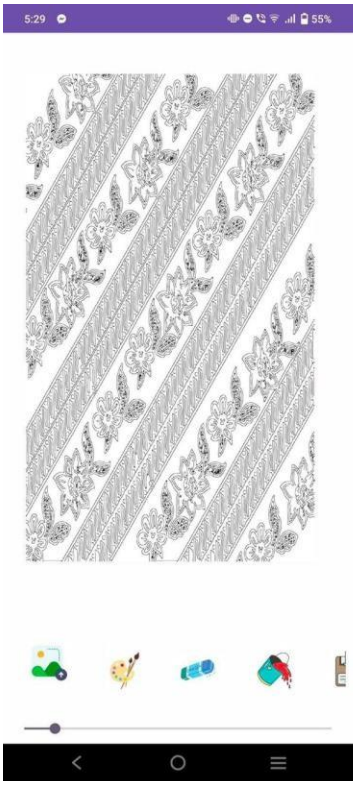


Figure 5.4.15 Customization of Batik Pattern

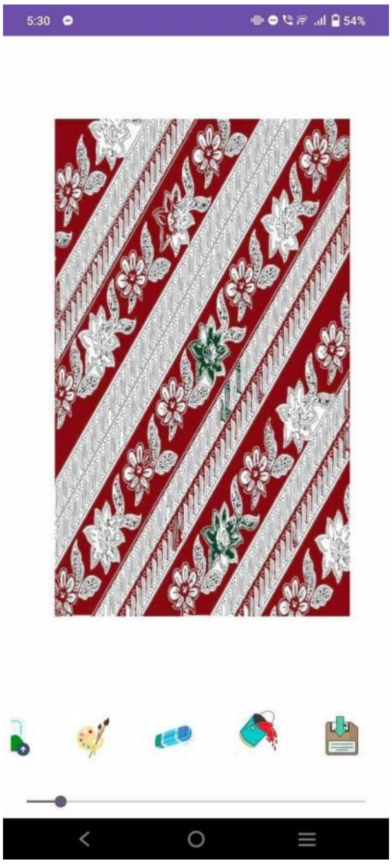


Figure 5.4.16 Customization of Batik Pattern – Filling Color on the uploaded image

Virtual Try On

The "virtual fitting page" lets users see how the chosen clothes will look on them. A range of outfit alternatives are shown on the left side of the screen for users to select from. After choosing a dress, users may see the virtual model's choices of clothing in the center display area by pressing the Generate button. They can test several alternatives and reset by pressing the "Clear" button. To improve the personalization experience, this function lets users try out different clothing styles and see how patterns are presented in actual environments. The second image shows the outcome of using the virtual fitting function on a particular article of clothing, showcasing the features' efficacy and genuineness.

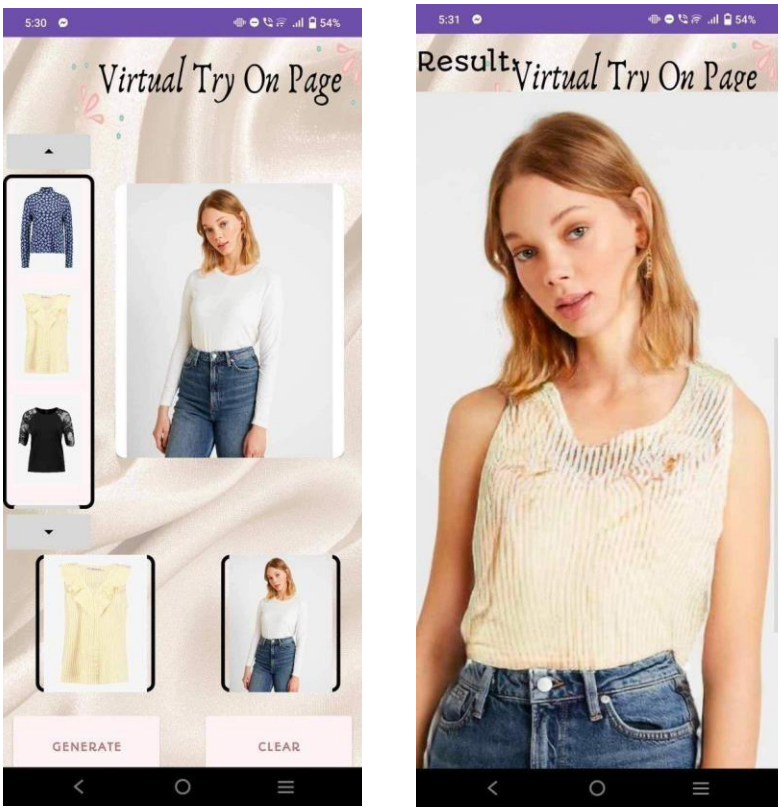


Figure 5.4.17 Virtual Try On and Generated Result

Gallery

According to Figure 5.4. 18 illustrates how the gallery's features are separated into two primary tabs: "All photos" and "Favorites." A complete collection of all photographs, including stored designs, garment fittings, and created batik patterns, may be seen in the "All Photos" page. Users may peruse this assortment, pick pictures for a larger view, and carry out other operations like sharing, removing, or adding to favorites. Users may quickly access and see photographs they have marked as favorites by using the Favorites tab. Users may keep track of the designs they find most attractive and manage their chosen designs with this page.

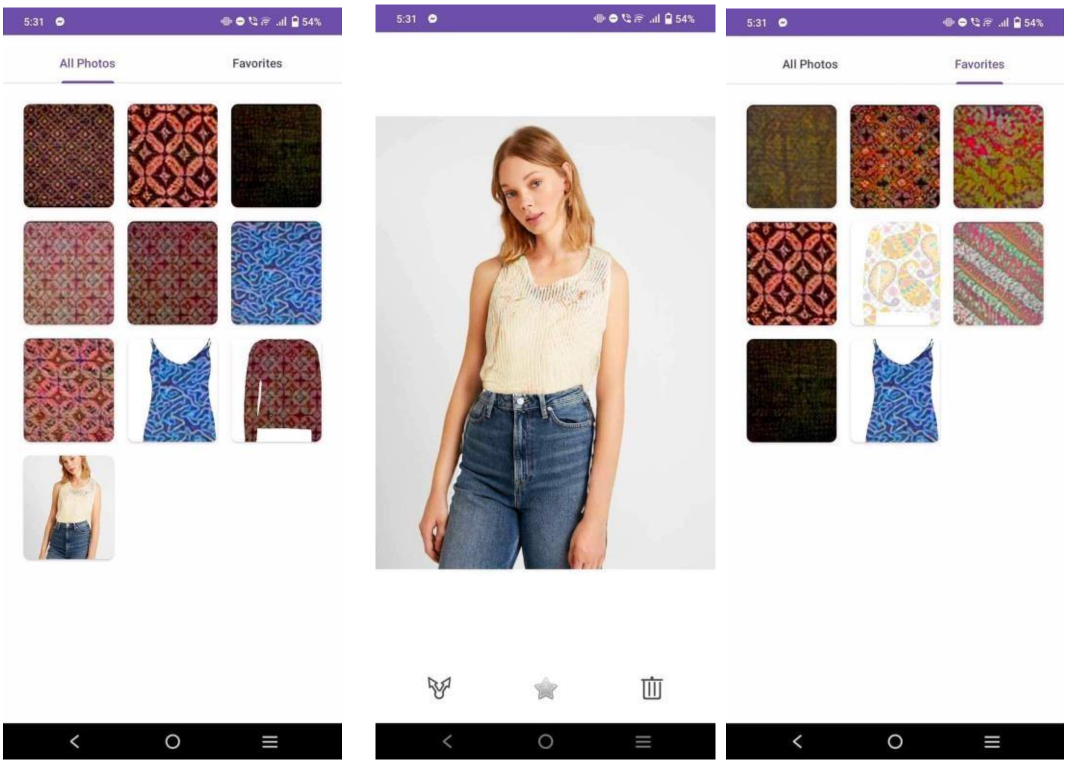


Figure 5.4.18 Gallery – All Image and Favorite

Tutorial Video Lists

Users may get guided instructions on a variety of software functionalities, including customization, virtual try-on, batik creation, and cloth transfer, through the Tutorial List feature. These choices are displayed in a dialog box when users access the lesson list, allowing them to choose a particular tutorial according to their needs or areas of interest. Every tutorial is made to assist users in comprehending and using the app's capabilities, guaranteeing a seamless and user-friendly experience.

After choosing a tutorial, a video dialog box opens with a step-by-step instruction on the specified subject. For instance, choosing "Creation Batik" will launch a movie that walks through the complete process of creating batik.

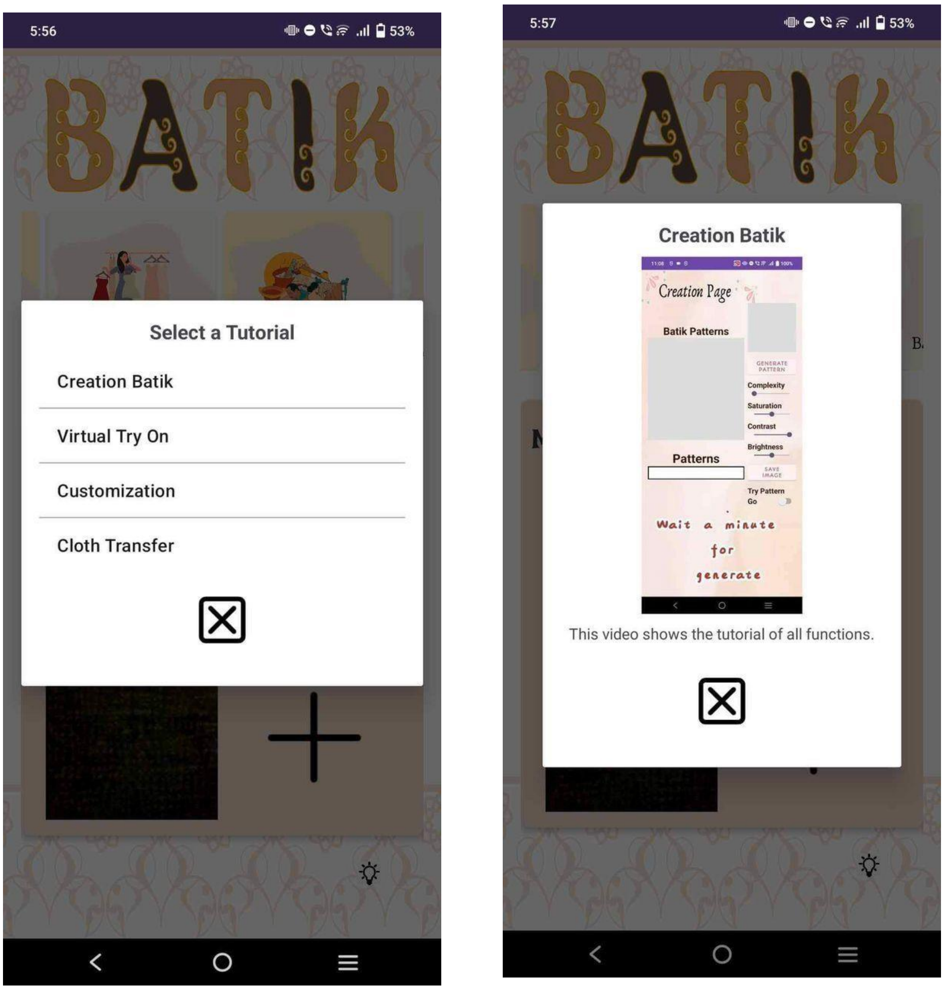


Figure 5.4.19 Tutorial List (right- example of video)

5.5 Implementation issues and Challenges

During StyleGAN2 training, significant challenges arise with the batik dataset, which are mainly due to the heterogeneity and complexity of the dataset itself. Although the dataset is rich and varied, it includes images that go beyond just batik patterns to models wearing batik clothing and batik patterns with complex backgrounds as show in **Figure 5.5.1** . This diversity, which reflects the real applications of batik, complicates the training process of a model that requires high purity and specificity of information. Foreign elements such as human figures and different backgrounds make the model more difficult to navigate. As a result, the model may inadvertently learn these pale features, reducing their ability to create targeted and authentic batik patterns . This mismatch between the composition of the dataset and the learning objectives of the model greatly hinders the full potential of StyleGAN2 to generate batik motifs. In addition, the presence of partial batik images and images with important parts of non-template elements connect the challenge. Such images can mislead the model's perception of batik patterns, resulting in incomplete or distorted patterns that do not accurately represent the traditional or desired batik aesthetic. This question highlights the critical importance of dataset curation and preparation in the context of AI-based reproductive art. This highlights a common challenge in the fields of computer vision and generative modeling, where the quality and composition of the training dataset plays a key role in determining the success and accuracy of the generated results.

Figure 5.5.1 Show the result(affected by the dataset) of the batik that generated

Figure 5.5.2 Dataset that included unrelated element

Chapter 6

System Evaluation and Discussion

6.1 System testing and Performance Metrics

To make sure each component in the Batik pattern design program is dependable and functional, unit tests were carried out. During this approach, each feature is isolated and tested in order to find and fix any problems early in the development cycle and increase the overall resilience of the program. The primary goal is to validate essential elements—like user-defined choices, virtual trial features, and Batik schema generation—to make sure they function properly in various contexts.

Building an extensive collection of test cases is a key component of a unit testing approach, which assesses the functioning of different functionalities. Tests for pattern generating components, for instance, are made to make that the system understands user inputs correctly, produces patterns with accuracy, and presents them as intended. Furthermore, elements that allow for customization—like changing the color, brightness, and complexity—are put through a thorough testing process to ensure that user changes are applied and preserved appropriately. To make sure that all potential outcomes are taken into account, each test case is created to cover a variety of user behaviors, including edge use cases.

A thorough analysis is performed on the unit test phase data to find any errors or discrepancies. To make sure the application achieves the anticipated performance criteria, any problems are promptly communicated to the development team. This methodology contributes to the upkeep of a superior application that operates consistently throughout all functionalities, finally offering users a smooth and captivating experience in the creation and visualization of personalized Batik patterns.

6.2 Tetsing Setup and Result

The checklist will be used to run the unit tests in this section. The checklist will guarantee that all application components that need to be reviewed are done so and that no important information is overlooked. There will be many parts in the test section, each of which will represent a significant application piece.

Main Menu Testing

When user open and launch the Batik Creator applictaion the system should be able to show the splash screen that showing the app name , and after that the application will enter the first page of the application which is the main menu . In the main menu there contains several selection for user to click and it will navigate user to repective page.

No.	Test Case	Expected Result	Status
1.	Launch App	Show the app logo in splash screen	Passed
2.	Main Menu	Show page selection for user in card view	Passed
3.	Click on tutorial icon	Show a list of tutorials for user	Passed
4.	Select “Batik Creation” Card	Navigate to Batik Creation Page	Passed
5.	Select “Virtual Try-On” Card	Navigate to Virtual Try On page	Passed
6.	Select “batik Customization” Card	Navigate to Batik Customization Page	Passed
7.	Select “batik Cloth Transfer “Page	Navigate to Batik Cloth Transfer Page	Passed
8.	Click “+” in the my collection	Navigate to Gallery Page	Passed

Table 6.2.1 Test Result for Main Menu Page

Generate Batik Patterns Page Testing

After that , when user click on the “Batik Creation” selection , the system should nevugate them to this page for them to click on the generate button , adjust complexity and performing filtering on the generated images. In addition , they also can save the images.

No.	Test Case	Expected Result	Status
1.	Click on Generate Button	Progress bar showed and	Passed

		generated image displayed	
2.	Slide the Slider	Complexity of images need to be changed	Passed
3.	Adjust the Saturation	Saturation of current selected image need to be updated	Passed
4.	Adjust the Contrast	Contrast of the current selected image need to be updated	Passed
5.	Adjust Brightness	Brightness of the current selected image need to be updated	Passed
6.	Click “Save image”	The current selected and displayed image need to be saved to gallery	Passed
7.	Click on “Try-on” switch	Navigate to Virtual Try On Page	Passed

Table 6.2.2 Test Result for Generate Batik Patterns Page

Virtual Try On Page Testing

When user navigate to Virtual Try -On page , the system should able to show variety of cloth for user to select and person model .

No.	Test Case	Expected Result	Status
1.	Select cloth pattern	Selected cloth showed in cloth view	Passed
2.	Select person Model	Selected Person Model showed in model view	Passed
3.	Click “Generate “button	The selected person will wearing the cloth that selected just now	Passed
4.	Click on “ Clear “button	Cloth view and person model view will be cleared	Passed

5.	Click on “Save”	The final result will be stored to mobile device and internal app gallery	Passed

Table 6.2.3 Test Result for Virtual Try On Page

Batik Customization Page Testing

In this page , system should be allowed user to upload the image from their own device gallery , and the selected image will be acted as a base for manpulation and coloring. User will be able to erase , save , filling color on the area they tap on and painting.

No.	Test Case	Expected Result	Status
1.	Click on “Upload Image” icon	Prompt user to allow permision for access gallery in device .	Passed
2.	Select image from gallery	The drawing of the selected image will be showed on the screen with black and white color .	Passed
3.	Swping on screen	The path under finger touched will be erased.	Passed
4.	Select painting icon and swiped	New path based on user input will be drawed on screen	Passed
5.	Select filling color icon and tab on area of drawing	The area user selected will be filled with color they choosed	Passed
6.	Select “Save” icon	The current drawing showed on screen will saved to device gallery.	Passed

Table 6.2.4 Test Result for Batik Customization Page

Batik cloth Transfer Page Testing

When user navigated to this page , the system should allowed user to perform the cloth transfer , which is the pattern they selected will be applied onto the cloth they selected .

No.	Test Case	Expected Result	Status
1.	Select “Cloth Pattern”	Selected Cloth Pattern showed in the cloth view	Passed
2.	Select “ Batik Pattern”	Selected Batik pattern showed in batik view	Passed
3.	Click on “Generate” button	Image of pattern transfer to selected cloth will be showed.	Passed
4.	Click on “Clear” button	All selection will be cleared	Passed
5.	Click on “Save” button.	Current result will be saved to in app gallery and device gallery.	Passed

Table 6.2.5 Test Result for Batik Cloth Transfer Page

Tutorial Video Lists testing

The system should able to show user a list of tutorial video title and allow user to click on and navigate to correct pages.

No.	Test Case	Expected Result	Status
1.	Click on “Tips” icon	List of tutorial video title show for select	Passed
2.	Select “ Batik Creation” tutorial	Show video dialog of the batik Creation	Passed
3.	Select “ Virtual Try-On” tutorial	Show video dialog of the Virtual Try-On	Passed
4.	Select “Customization Batik” tutorial	Show video dialog of the batik Customization batik”	Passed
5.	Select “Batik Cloth Transfer “ tutorial	Show video dialog of the Batik Cloth Transfer”	Passed
6.	Click on “X” close icon button	Close the video dialog that currently showing	Passed
7.	Click on the video	Pause or Stop the current playing	Passed

Table 6.2.6 Test Result for Tutorial Video List

Gallery Page Testing
In Gallery Page , the system should able to show all the images saved from all page , and also the favorite image . Futhermore it should allowed user to add to Favorite and share or delete.

No.	Test Case	Expected Result	Status
1.	Click on image	Image will be in viewed mode and enlarge for viewing	Passed
2.	Click on “Shared” icon	List of Media Platform showed for selection.	Passed
3.	Click on “Star” ticon o add to favorite	Current viewed image added to favorite images .	Passed
4.	Click on “Dustbin” icon to delete image	Current viewed image will be deleted	Passed

Table 6.2.7 Test Result for Gallery Page

6.3 Project Challenges

This section are the challenged that encountered during the development and implementation of process in the project.

1. **Reaction time for creating Batik designs:** One major issue is the slower reaction time that servers provide while creating Batik patterns. Even with the modifications to cut this time down to less than a minute, delays are still deemed excessive and negatively impact the user experience in general. The style's high processing needs are the primary cause of this issue. The GAN2 model, which creates patterns based on user input, needs a lot of computing power. Additional improvements are required to tackle this issue, such as enhancing server capability, utilizing more effective algorithms, or putting parallel processing strategies into practice. Furthermore, investigating techniques like pre-generated patterns at times of low server traffic might aid in giving consumers speedier replies.

2. **The standard of style-generated Batik patterns**, both in terms of variation and quality, is a significant obstacle. The GAN2 framework. Although models can generate distinctive patterns, they frequently fall short in terms of quality and diversity. This problem arises mostly from low variety and insufficient training data sets, which hinder the model's ability to capture all the subtleties and variances included in traditional Batik designs. More time is required to train models with larger and more diverse datasets in order to enhance quality. To accurately represent the cultural and artistic diversity of batik, this dataset should incorporate a variety of regionally-specific Batik styles, colors, and patterns. Enhancing the amount and variety of the dataset can greatly enhance the model's capacity to produce realistic, varied, and high-quality patterns.

6.4 Objective Evaluation

The project's primary objective is to develop a mobile application that, by fusing cutting-edge technologies like virtual fitting tools and Generative Adversarial Networks (GANs) with Batik's pattern design process, would improve creativity and decision-making for fabric fans, designers, and artists. The software effectively accomplishes this by offering a distinctive platform that enables users to dynamically and interactively explore, create, and view Batik designs.

The project's primary objective is to use StyleGAN2, a cutting-edge machine learning model, to create a variety of distinctive Batik designs. This goal has been partially accomplished. The program offers users a multitude of creative options by efficiently producing a range of Batik designs using the style GAN2. Notwithstanding, several obstacles persist, including the want for more model training and a wider range of datasets to enhance the caliber and genuineness of the patterns. Notwithstanding these difficulties, the study shows how machine learning may be used to improve traditional fabric design, increasing Batik's appeal and engagement in the digital era.

The second objective is to enable virtual visualization for customized Batik modes using VITON-HD technology. The app's realistic, model-based visuals enable users to view how their selected designs would look on various articles of clothing, demonstrating the effective achievement of this aim. This feature allows for real-time experimentation and modification, which enhances the creative process by helping users make more informed decisions regarding pattern selection and application. Preliminary feedback from users demonstrates that the virtual trial feature significantly raises user happiness and engagement, demonstrating its worth as a tool for decision help.

The third objective is to incorporate sophisticated machine learning models into intuitive user interfaces so that non-technical consumers can engage with machine-generated designs. This goal has been accomplished with success. With a single, user-friendly interface, the app allows users to easily experiment with various modes, view them in real time, and tailor them to their tastes. Feedback indicates that the platform is user-friendly and accessible to people with less experience in digital design, indicating that the project has been successful in democratizing complex design tools.

All things considered, the project has successfully closed the gap between traditional Batik artistry and contemporary digital innovation, providing a complete platform that fosters creativity, facilitates decision-making, and offers consumers an exclusive, interactive fabric creation experience. Even though there are still areas that need improvement, such model performance and dataset variety, the project has built a solid framework for improvements into the road and wider use.

Chapter 7
Conclusion

7.1 Conclusion

All things considered, this endeavor has effectively used Style's potential. An inventive tool created by GAN2 makes the difficult process of creating Batik designs simpler. The project not only solves the conventional problems with pattern visualization, but it also shows how AI has the ability to go beyond these constraints by giving amateurs and seasoned designers a straightforward way to get involved in Batik's pattern design process. The application's creation is in keeping with the project's objective of conserving Batik Art's cultural value in the digital age while democratizing it through the integration of sophisticated generative models with an intuitive interface.

The project's remarkable results show how well traditional art and contemporary technology may be combined. Major development goals have been met, including the employment of Style. GAN2 creates a variety of distinctive Batik designs and combines VITON-HD technology to enable virtual depiction of these patterns on various articles of clothing. The model's early learning phase is shown in the first Frechet Inception Distance (FID) score of 139.281, which also serves as a strong foundation for the application's goals. This score guarantees consumers that the created patterns maintain the distinctive qualities of traditional Batik art and serves as a foundation for future development. By emphasizing the useful implementation of created patterns and user interaction, the project offers a forward-looking approach that connects rich heritage and innovative technologies.

This development has several prospects for further growth and development in the future. The model may be further improved as artificial intelligence technology advances by adding more diverse Batik pattern datasets to it, which would raise the output's quality. Additionally, the app may develop to incorporate features that let designers submit their own ideas or sketches, which the model might subsequently interpret as Batik patterns. This feature allows for a stronger relationship between digital products and their traditional inspiration, while also expanding the dataset and offering consumers a tailored experience.

7.2 Recommendation

The project created a strong foundation to combine traditional batik art with cutting-edge generative artificial intelligence. There are many ways to improve the functionality and user experience of the app in the future. Material enrichment emerges as a primary recommendation, where expanding the collection to include a wider range of batik designs from different cultures and eras can significantly improve design diversity and authenticity. Collaboration with museums, cultural institutions and batik artisans can facilitate the use of rare and historically important patterns, enrich the offer of the program and strengthen its role in cultural preservation.

Customization stands out as an important next step to increase user engagement with the application. Introducing features that allow users to add their own designs or modify existing designs can make the app an interactive creative platform. This level of customization would not only improve the user experience by allowing a more hands-on approach to design creation, but would also be a unique opportunity for users to explore their creativity within the batik tradition. Encouraging user-generated content can further diversify the model's learning dataset, leading to richer and more diverse pattern results.

Finally, exploring collaborations with fabric manufacturers and fashion designers can extend the utility of the application beyond digital pattern creation, bridging the gap between traditional patterns and modern fashion. By allowing users to see how their custom batik designs can be realized in various textiles and garments, the application can provide a complete workflow from design to production. This can potentially open up new markets for batik art and promote sustainable and culturally rich fashion in the global market. Such partnerships can also give users a hands-on opportunity to bring their virtual creations into the real world, furthering the program's mission to innovate within traditional art forms.

REFERENCES

[1] B. H. Gürcüm, “Conceptual Design Method and creativity in Textile Design,” *Journal of Textile Engineering & Fashion Technology*, vol. 3, no. 1, 2017. doi:10.15406/jteft.2017.03.00086

[2] M. Mahyoub, S. H. Abdulhussain, F. Natalia, S. Sudirman, and B. M. Mahmmod, “Abstract pattern image generation using generative adversarial networks,” *2023 15th International Conference on Developments in eSystems Engineering (DeSE)*, 2023. doi:10.1109/dese58274.2023.10099871

[3] G. N. Yar *et al.*, “Texgan: Textile pattern generation using deep convolutional generative Adversarial Network (DCGAN),” *2023 IEEE International Conference on Emerging Trends in Engineering, Sciences and Technology (ICES&T)*, 2023. doi:10.1109/icest56843.2023.10138848

[4] W.-T. Chu and L.-Y. Ko, “Batikgan: A Generative Adversarial Network for batik creation,” *Proceedings of the 2020 Joint Workshop on Multimedia Artworks Analysis and Attractiveness Computing in Multimedia*, 2020. doi:10.1145/3379173.3393710

[5] H. M. Razali, M. Ibrahim, M. Omar, and S. F. Hashim, “Current challenges of the batik industry in Malaysia and proposed solutions,” *PROCEEDINGS OF 8TH INTERNATIONAL CONFERENCE ON ADVANCED MATERIALS ENGINEERING & TECHNOLOGY (ICAMET 2020)*, 2021. doi:10.1063/5.0055651

[6] H. M. Razali, M. Ibrahim, M. Omar, and S. F. Hashim, “Current challenges of the batik industry in Malaysia and proposed solutions,” *PROCEEDINGS OF 8TH INTERNATIONAL CONFERENCE ON ADVANCED MATERIALS ENGINEERING & TECHNOLOGY (ICAMET 2020)*, 2021. doi:10.1063/5.0055651

[7] Z. Liu, “Fashion designer application based on Generative Adversarial Network,” *Highlights in Science, Engineering and Technology*, vol. 34, pp. 127–135, 2023. doi:10.54097/hset.v34i.5434

[8] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings, pp. 1–16, 2016.

[9] A. E. Minarno, Moch. C. Mustaqim, Y. Azhar, W. A. Kusuma, and Y. Munarko, “Deep convolutional generative adversarial network application in Batik Pattern Generator,” *2021 9th International Conference on Information and Communication Technology (ICoICT)*, 2021. doi:10.1109/icoict52021.2021.9527514

[10] The history of batik and its use in modern times - prestige online, https://www.prestigeonline.com/my/lifestyle/art-plus-design/the-history-of-batik-and-its-use-in-modern-times/ (accessed Aug. 28, 2023).

[11] M. Luib, “How indonesian batik became a UNESCO Intangible World heritage halaman all,” KOMPAS.com, https://go.kompas.com/read/2020/10/28/202254174/how-indonesian-batik-became-a-unesco-intangible-world-heritage?page=all (accessed Aug. 28, 2023).

[12] S. Khalil, M. Taha, R. A. Shah, H. Mehmood, and H. Dilpazir, "Generation of Textile Patterns Through Generative Adversarial Networks," in Proceedings of 1st International Conference on Software Engineering and Computing Disciplines (ICSECD) 2019, Pakistan, March 2020.

[13] M. Giles, “The ganfather: The man who’s given machines the gift of imagination,” MIT Technology Review, https://www.technologyreview.com/2018/02/21/145289/the-ganfather-the-man-whos-given-machines-the-gift-of-imagination/ (accessed Sep. 4, 2023).

[14] O. Octadion, N. Yudistira, and D. Kurnianingtyas, "Synthesis of Batik Motifs using a Diffusion -- Generative Adversarial Network," 2023. [Online]. Available: arXiv:2307.12122

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.:3**
Student Name & ID: Chen Sin Yee 20ACB05680	
Supervisor: Dr Tan Hung Khoon	
Project Title: Batik Pattern Synthesis For Virtual Try On Application	

1. WORK DONE

1. Discuss and Refine objective with Teo Zi Ning

2. WORK TO BE DONE

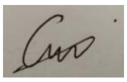
1. Search and reviewed existing system that related to Batik Creation App.
2. Discuss and draft the UI design

3. PROBLEMS ENCOUNTERED

1. So far no problem faced .

4. SELF EVALUATION OF THE PROGRESS

1. No progress has been made in the development
2. Focus is on draft the UI that needed to be developed.



Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.:6**
Student Name & ID: Chen Sin Yee 20ACB05680	
Supervisor: Dr Tan Hung Khoon	
Project Title: Batik Pattern Synthesis For Virtual Try On Application	

1. WORK DONE

1. Done for 3 objective
2. Came out the initial prototype for the UI design

2. WORK TO BE DONE

1. Make the function workable which is allow user to adjust complexity of images
2. Design own part interface UI and functions

3. PROBLEMS ENCOUNTERED

1. The problem encountered is the images passed back from server to UI is too large and unable to pass back in one.

4. SELF EVALUATION OF THE PROGRESS

1. Progress is moderate and development in progress.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	Study week no.:9
Student Name & ID: Chen Sin Yee 20ACB05680	
Supervisor: Dr Tan Hung Khoon	
Project Title: Batik Pattern Synthesis For Virtual Try On Application	

1. WORK DONE

1. Some simple UI such as Buton and image view placeholder and grid are done.
2. Done Chapter 2 until 3.

2. WORK TO BE DONE

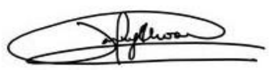
1. Continue for doing the adjusting complexity function
2. Create new features allow user to customize any batik they like.

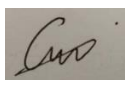
3. PROBLEMS ENCOUNTERED

1. **The problem faced is more related to handling the server response as the response passed back is images.**

4. SELF EVALUATION OF THE PROGRESS

1. **Progress is smooth but slow as to debug the response failure**





Supervisor's signature   Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.:11**
Student Name & ID: Chen Sin Yee 20ACB05680	
Supervisor: Dr Tan Hung Khoon	
Project Title: Batik Pattern Synthesis For Virtual Try On Application	

1. WORK DONE

1. Done for Batik Generate UI and Customization Batik Features
2. Solve for adjusting complexity and able to pass back the image without any response failure.
3. Solve for server timeout

2. WORK TO BE DONE

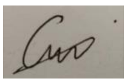
1. Integrate batik synthesis function with Teo Zi Ning part.
2. Decide the final application theme and design like background and button.
3. Start to do report.

3. PROBLEMS ENCOUNTERED

1. So far no problem encountered

4. SELF EVALUATION OF THE PROGRESS

1. Progress is more faster and less problem encountered.



Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S3	**Study week no.:13**
Student Name & ID: Chen Sin Yee 20ACB05680	
Supervisor: Dr Tan Hung Khoon	
Project Title: Batik Pattern Synthesis For Virtual Try On Application	

1. WORK DONE

1. Successfully integrated the batik creation with Teo Zi Ning part
2. Designed and finalize the final UI design.
3. Done for all Chapter in the report

2.WORK TO BE DONE

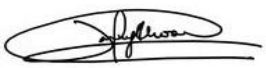
NA

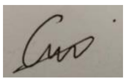
3. PROBLEMS ENCOUNTERED

1. **No problem Encounter**

4. SELF EVALUATION OF THE PROGRESS

1. **Progress of development and writing report are smooth and faster.**





Supervisor's signature

Student's signature

POSTER

Batik Pattern Synthesis For Virtual Try On Application

INTRODUCTION

The project aims to integrate Generative Adversarial Networks (GANs) with virtual try-on technology to innovate the Batik pattern design process. By leveraging StyleGAN2, a state-of-the-art machine learning model, and VITON-HD, an advanced virtual try-on tool, the application provides users with a dynamic platform for creating, exploring, and visualizing fabric designs, merging traditional Batik artistry with modern digital innovation.

OBJECTIVE

- Enhance Creativity and Decision-Making: Utilize StyleGAN2 to generate diverse and high-quality Batik patterns, enabling users to explore unique designs.
- Virtual Visualization: Implement VITON-HD technology for a virtual try-on feature, allowing users to visualize Batik patterns on various garments.
- Seamless Integration of Machine Learning: Develop an accessible platform for creative exploration using advanced AI models, promoting wider engagement in Batik design.

METHODOLOGY

- **Model Training: Train the StyleGAN2 model on a curated Batik dataset to generate unique patterns.**
- **Application Development: Develop a mobile application using Android Studio, integrating the StyleGAN2 model for real-time pattern generation and customization.**
- **Virtual Try-On Implementation: Incorporate VITON-HD for virtual fitting functionalities, allowing users to visualize designs on different clothing items.**

RESULT

FID : 139.28

CONCLUSION

- This project demonstrates the potential of AI in transforming traditional Batik design by offering an innovative and accessible platform for both designers and fabric enthusiasts. While initial results are promising, further improvements in model training, dataset diversity, and application features will enhance the user experience and the quality of generated patterns, contributing to the modernization and preservation of Batik art.

PLAGIARISM CHECK RESULT

Sin Yee Chen

ORIGINALITY REPORT

2%	1%	0%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	eprints.utar.edu.my Internet Source	1%
2	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
3	Submitted to King's College Student Paper	<1%
4	theses.liacs.nl Internet Source	<1%
5	Submitted to New Jersey Institute of Technology Student Paper	<1%
6	Submitted to University of Western Sydney Student Paper	<1%
7	Submitted to Purdue University Student Paper	<1%
8	Meredian Alam. "Biophilic architecture and designs for mental well-being", IOP Conference Series: Earth and Environmental Science, 2023 Publication	<1%

9	Kumaar Swamy Reddy Bapathi, Mostafa F. Abdelbar, Wipakorn Jevasuwan, Pramod H. Borse, Sushmee Badhulika, Naoki Fukata. "Cumulative effect of spectral downshifting, anti-reflection and space-charge region formation in enhancing the spectral response of self-powered silicon photodetectors on sensitisation with CdZnS/ZnS core-shell quantum dots", Nano Energy, 2024 Publication	<1 %
10	Masahiko Higashino, Shin Okamoto. "Response Control and Seismic Isolation of Buildings", Taylor & Francis, 2006 Publication	<1 %
11	www.ncbi.nlm.nih.gov Internet Source	<1 %
12	Mike Tooley, Mike Tooley, Lloyd Dingle, Lloyd Dingle. "Higher National Engineering", Routledge, 2019 Publication	<1 %
13	ia802202.us.archive.org Internet Source	<1 %
14	umpir.ump.edu.my Internet Source	<1 %
15	www.isteams.net Internet Source	<1 %
16	www.itu.int Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 8 words
Exclude bibliography	On		

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Chen Sin Yee
ID Number(s)	20ACB05680
Programme / Course	Computer Science
Title of Final Year Project	Batik Pattern Synthesis For Virtual Try On Application

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 2 **%** **Similarity by source** Internet Sources: 1 % Publications: 0 % Student Papers: 1 %	
Number of individual sources listed of more than 3% similarity: 0	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor	- Signature of Co-Supervisor
Name: Tan Hung Khoon	Name: -
Date: 11/9/2024	Date: -

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

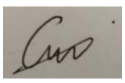
CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	20ACB05680
Student Name	Chen Sin Yee
Supervisor Name	Dr Tan Hung Khoon

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.



____

(Signature of Student)

Date: 10/09/2024