CONVERTING HAND-DRAWN SKETCH TO REALISTIC IMAGE USING GANS

Capstone Project Report

Submitted by:

Prakhar Bhateja	**101903098**
Shobhit Gupta	**101903095**
Arpan Rehan	**102083014**
Umar Tariq Wani	**102083065**

BE Third Year- COE

CPG No. 100

Under the Mentorship of

Dr. Yadwinder Singh
CSED Department

Computer Science and Engineering Department
Thapar Institute of Engineering and Technology,

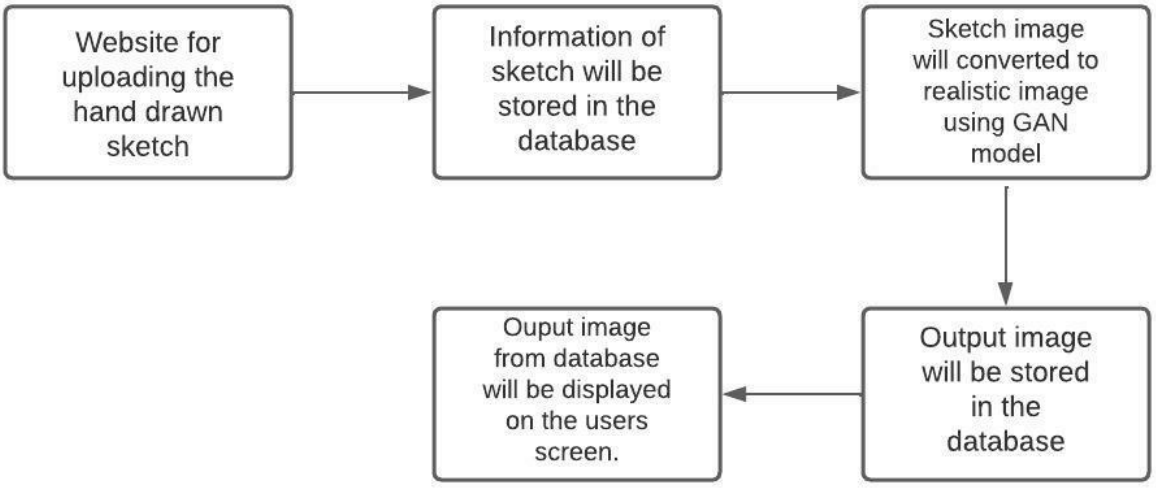
MARCH - DECEMBER 2022

Product Perspective:

The power and promise of deep generative models such as GANs lie in their ability to synthesize endless realistic, diverse, and novel content with minimal user effort. Automatically converting sketch images into realistic images has significant application value in the fields of digital entertainment, art, law enforcement, and other industries.

1. In the law enforcement industry, there is a big difference between an image drawn by the police or a medical expert according to an oral description from an eyewitness and a portrait image with low recognition. Manually generating a real color portrait requires a skilled labor force with experience in drawing and painting as well as investigation. Automatic generation of realistic facial portraits based on sketch images with generative adversarial networks (GANs) can improve the possibility of identification and enhance police efficiency in solving cases.

2. In the comic domain of the entertainment industry, cartoon sketch coloring is being highly implemented to fill colors into the black-and-white anime sketches and to obtain the colored anime images which are more visually appealing in nature. At present, cartoon sketch coloring mainly relies on the anime/comic painters. It takes a lot of time and effort to manually color the anime sketches and the coloring effect is influenced by the professional ability of the anime painters. In order to reduce the difficulties of manual coloring, it is very important to design an appropriate automatic coloring method that can be used to avoid the complicated work procedures generated by manual coloring. Recently, it has become a new research hotspot in the field of deep learning and many generative adversarial networks (GANs) have been used to design appropriate coloring methods.



Block Diagram

Design and Architecture:

In a GAN, the generator and discriminator models receive input information in the form of a vector. This information could be the label of the input image or some other property. The information is one hot encoded and sent to the generator. The generator takes this vector of information and encodes features from an image like the class (male and female if we are trying to generate images of faces) or properties like hair, nose, eyes etc. To make this Generative and Adversarial process simple, both these blocks are made from Deep Neural Network based architecture which can be trained through forward and backward propagation techniques.

GENERATOR NETWORK:

The generator part of a GAN learns to create fake data by incorporating feedback from the discriminator. It learns to make the discriminator classify its output as real. Generator training requires tighter integration between the generator and the discriminator.

The portion of the GAN that trains the generator includes:

- random input.
- generator network, which transforms the random input into a data instance.
- discriminator network, which classifies the generated data.
- discriminator output.
- generator loss, which penalizes the generator for failing to fool the discriminator.

DISCRIMINATOR NETWORK:

The discriminator in a GAN is simply a classifier. It tries to distinguish real data from the data created by the generator. It uses a simple convolutional neural network with batch normalization. To identify the real and fake images, the output layer stores the probability values for each of the images in the input layer which indicates the probability of the image being real or fake.

The discriminator's training data comes from two sources:

- Real data instances, such as real pictures of people. The discriminator uses these instances as positive examples during training.
- Fake data instances created by the generator. The discriminator uses these instances as negative examples during training.

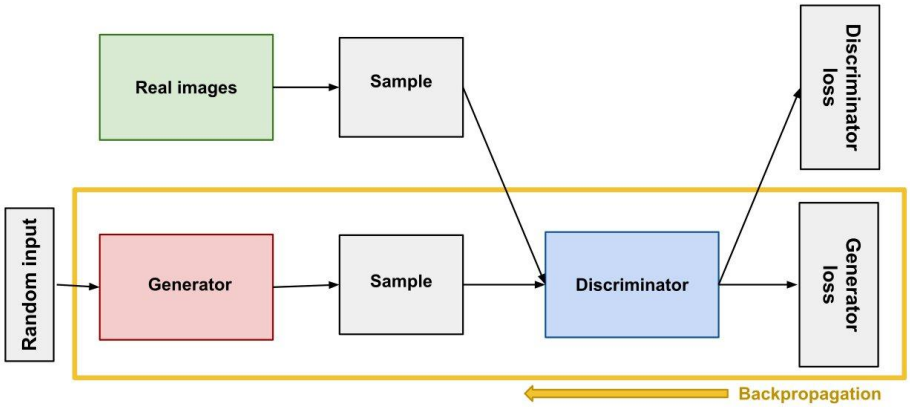


Figure showing design of generator and discriminator model

USE CASE DIAGRAM:

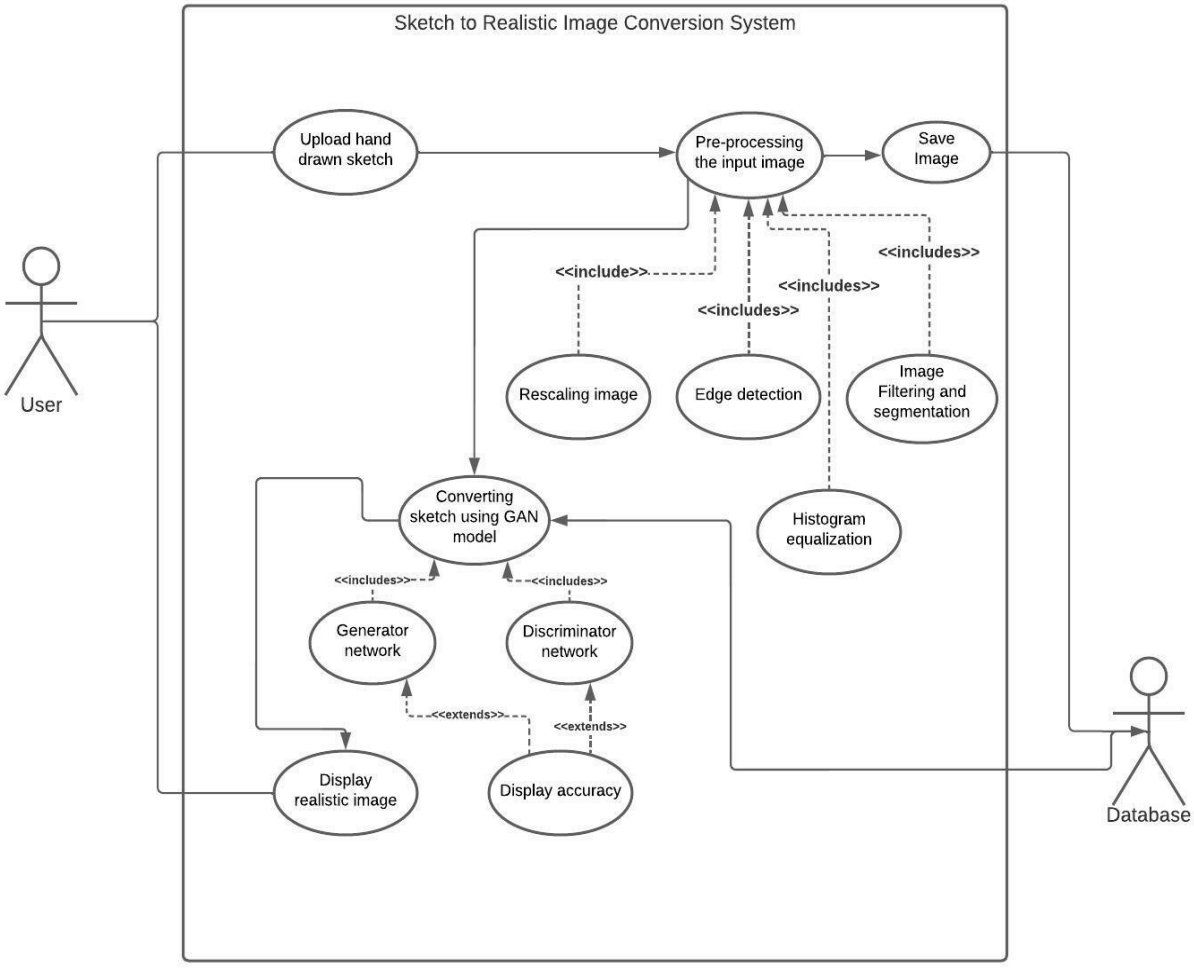


Figure showing Use Case Diagram

Use Case Templates:

Use Case Title	Upload Hand Drawn Sketch
Abbreviated Title	Upload Hand Drawn Sketch
Use Case ID	1
Actors	User , Database
Description	Users can upload hand drawn sketches
Pre-conditions	The user must have a hand drawn sketch
Task Sequence	1. Open the website 2. Select the Hand Drawn Sketch 3. Upload the Hand Drawn Sketch
Post-conditions	1. Image is Uploaded

Use Case Title	Pre-processing the input image
Abbreviated Title	Pre-processing the input image
Use Case ID	2
Actors	User , Database
Description	Processing the image provided by the user
Pre-conditions	The user must upload the image
Task Sequence	1. Rescaling the image 2. Edge Detection 3. Histogram Equalization 4. Image filtering and segmentation
Post-conditions	1. Image is Processed

Use Case Title	Save Image
Abbreviated Title	Save Image
Use Case ID	3
Actors	User , Database
Description	Image is saved in database
Post-conditions	Image saved

Tasks and Subtasks:

1. **Planning:**

1.1. Estimated cost

1.2. Estimated scope

1.3. Planned the document

2. **Research and analysis**

2.1. Prepared the Gantt chart

2.2. Made feasibility report

2.3. Made UML diagrams

3. **Requirement gathering**

3.1. Gathered external interface requirements

3.2. Gathered performance and other requirements

4. **Website Deployment**

4.1. Designing a website for the project

4.2. Incorporate Database into it

4.3. Designing wireframe for better understanding

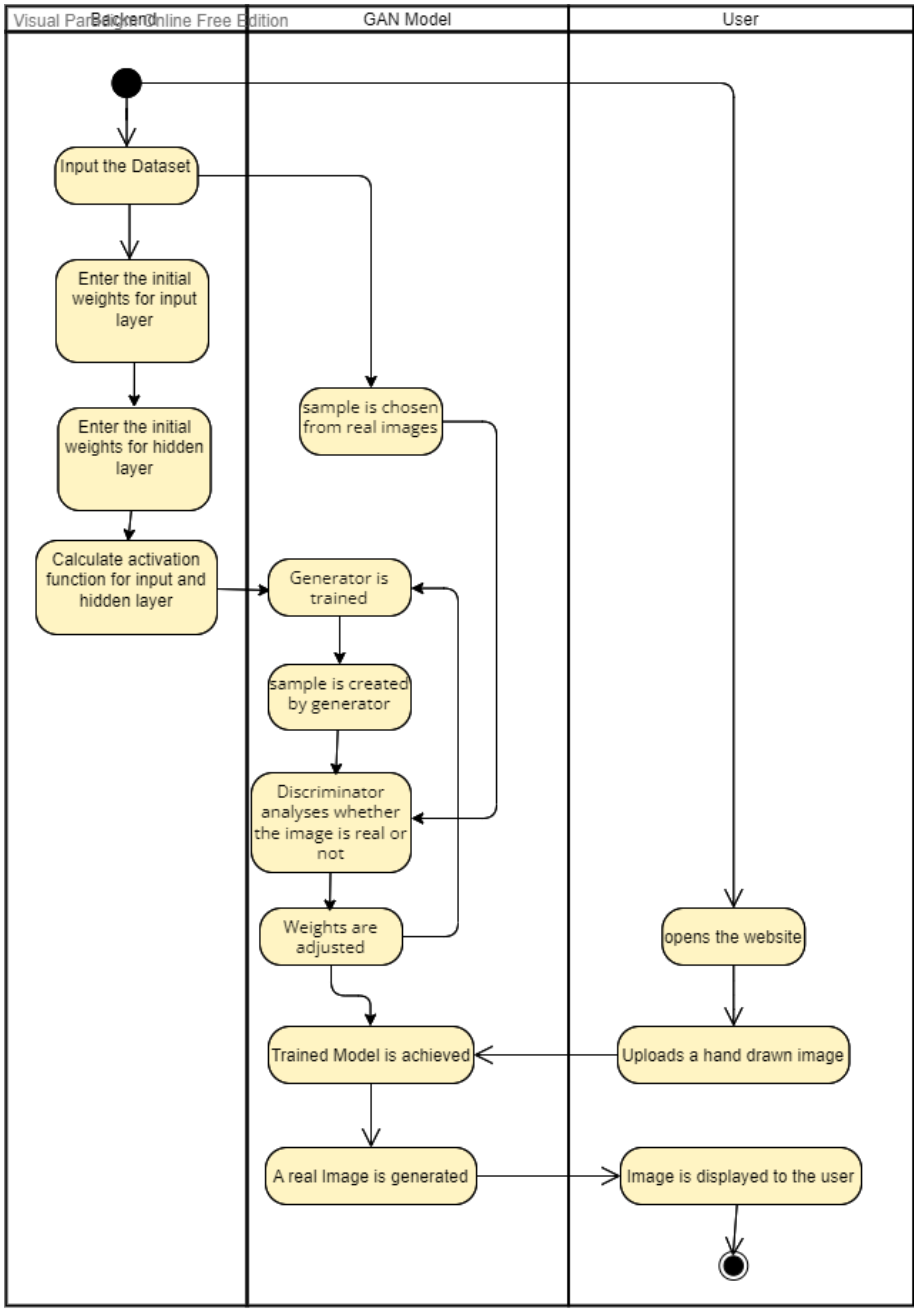
5. Testing

5.1 Usability Testing

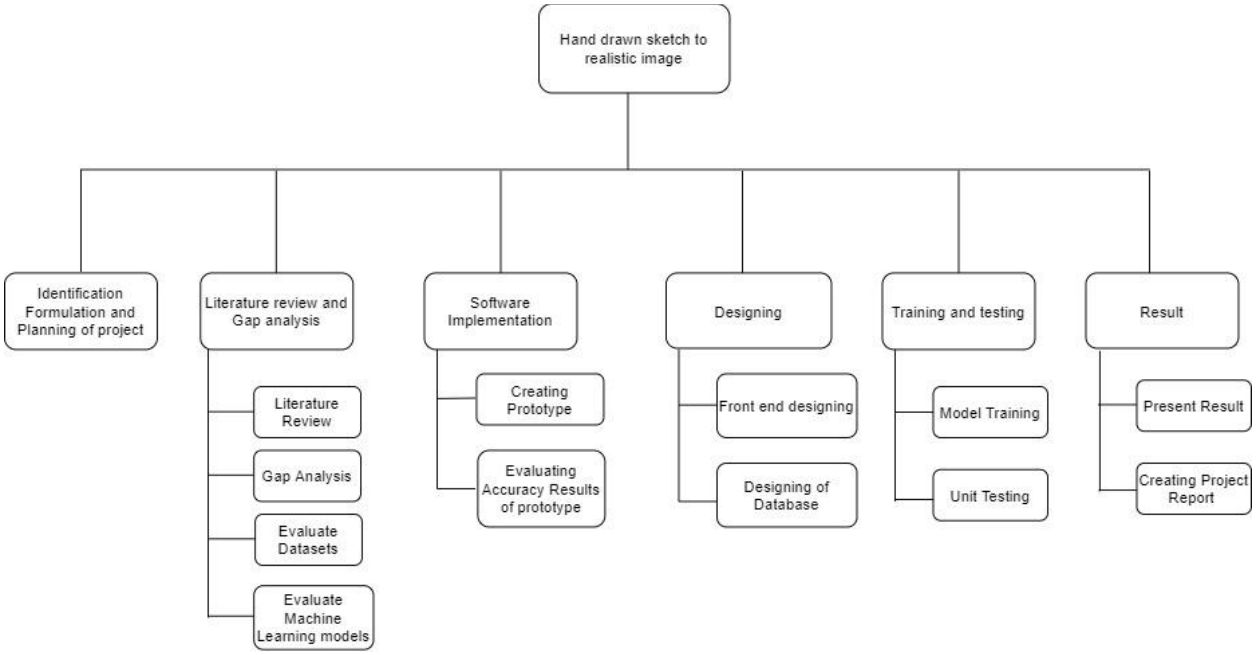
5.2 Progress Submission

5.3 Improvement of Prototype

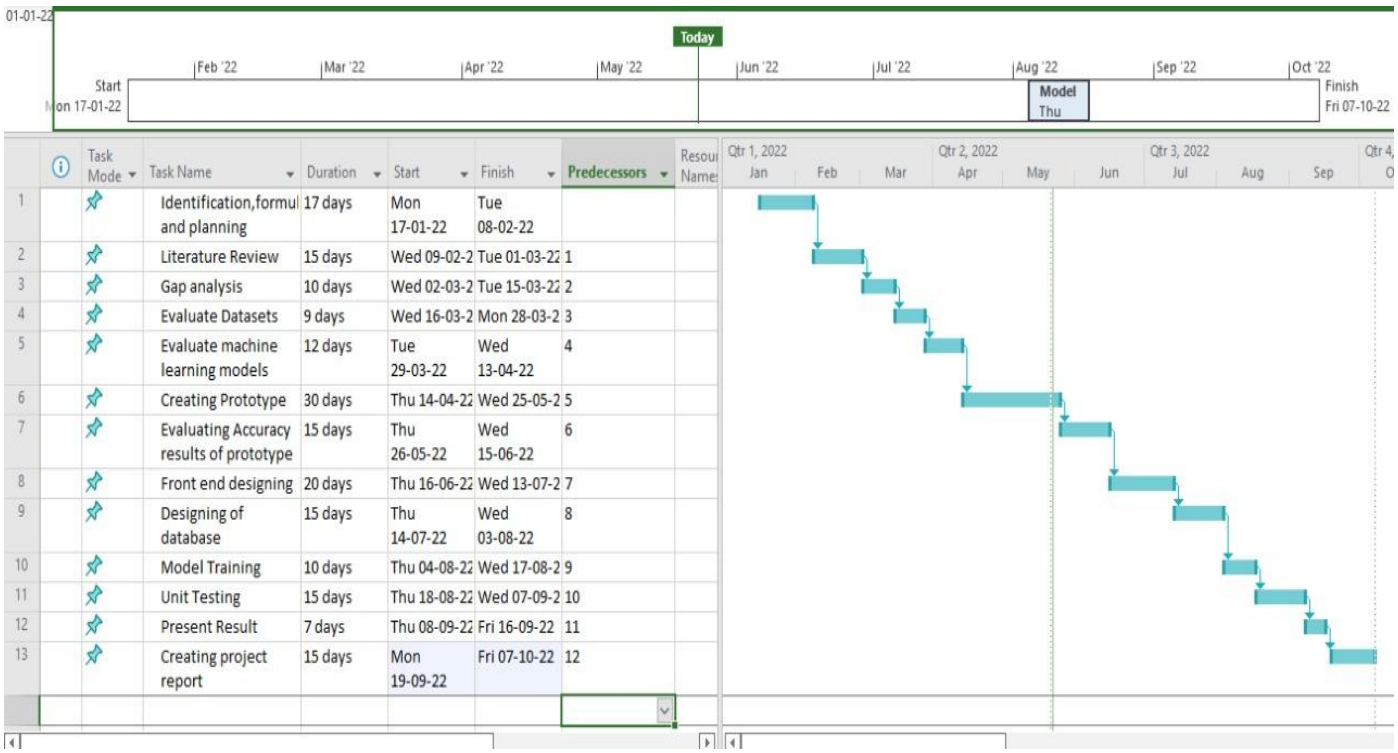
Swimlane diagrams/Activity diagrams



Work Breakdown Structure



Gantt chart



	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		Identification,formul and planning	17 days	Mon 17-01-22	Tue 08-02-22		
2		Literature Review	15 days	Wed 09-02-2	Tue 01-03-22	1	
3		Gap analysis	10 days	Wed 02-03-2	Tue 15-03-22	2	
4		Evaluate Datasets	9 days	Wed 16-03-2	Mon 28-03-2	3	
5		Evaluate machine learning models	12 days	Tue 29-03-22	Wed 13-04-22	4	
6		Creating Prototype	30 days	Thu 14-04-22	Wed 25-05-2	5	
7		Evaluating Accuracy results of prototype	15 days	Thu 26-05-22	Wed 15-06-22	6	
8		Front end designing	20 days	Thu 16-06-22	Wed 13-07-2	7	
9		Designing of database	15 days	Thu 14-07-22	Wed 03-08-22	8	
10		Model Training	10 days	Thu 04-08-22	Wed 17-08-2	9	
11		Unit Testing	15 days	Thu 18-08-22	Wed 07-09-2	10	
12		Present Result	7 days	Thu 08-09-22	Fri 16-09-22	11	
13		Creating project report	15 days	Mon 19-09-22	Fri 07-10-22	12	

Functional and Non-Functional requirements

Functional:

- User should be able to access the website easily
- The size of the image that the user will upload should be of the specified range.
- The user should be able to preview the image.
- The user should be able to download the predicted image in multiple formats. (eg. JPEG, PNG, PDF…)

Non-Functional :

- Enhancing the quality of the image produced.
- The model will be able to give more accurate results as it receives more and more data over time.
- The website will be easy to use and self-descriptive.
- System components must be built so they can handle failure of other components they depend on.