**A PROJET REPORT ON**

**“Face2Comic: Pix2Pix cGAN-Based Approach to Cartoonify Human Portraits”**

*Submitted in partial fulfilment of the requirements for the award of degree in Master of Computer Application (MCA) under Manipur University in the Academic session July 2021 – June 2024*

SUBMITTED BY

**Keithellakpam Oken**

MCA 6th Semester

Roll No. 2182



**राष्ट्रीय इलेक्ट्रॉनिकी एवं सूचना प्रौद्योगिकी संस्थान (रा.इ.सू.प्रौ.सं) ईम्फाल**

**National Institute of Electronics & Information Technology, Imphal**

An Autonomous Scientific Society of Ministry of Electronics & Information Technology, Govt of India,

**अकांपत्त पोस्ट बोक्स नं १०४: इंफाल ७९५००१ मणिपुर**

***Akampat* P.O.Box-104, Imphal-795001 Manipur**

Phone (0385)-2454109, 2452231 Fax (0385)-2449578 E-mail : dir-imphal@nielit.gov.in, Website : nielit.gov.in/imphal

**A PROJET REPORT ON**

**“Face2Comic: Pix2Pix cGAN-Based Approach to Cartoonify Human Portraits”**

*Submitted in partial fulfilment of the requirements for the award of degree in Master of Computer Application (MCA) under Manipur University in the Academic session July 2021 – June 2024*

SUBMITTED BY

**Keithellakpam Oken**

MCA 6th Semester

Roll No. 2182

UNDER THE GUIDANCE OF

(Internal Guide) (External Guide)

**Laimujam Joy Singh Dr. Khelchandra Thongam**

Guest Faculty Associate Professor (CSE Dept.)

NIELIT IMPHAL NIT Manipur, Langol



**राष्ट्रीय इलेक्ट्रॉनिकी एवं सूचना प्रौद्योगिकी संस्थान (रा.इ.सू.प्रौ.सं) ईम्फाल**

**National Institute of Electronics & Information Technology, Imphal**

An Autonomous Scientific Society of Ministry of Electronics & Information Technology, Govt of India,

**अकांपत्त पोस्ट बोक्स नं १०४: इंफाल ७९५००१ मणिपुर**

***Akampat* P.O.Box-104, Imphal-795001 Manipur**

Phone (0385)-2454109, 2452231 Fax (0385)-2449578 E-mail : dir-imphal@nielit.gov.in, Website : nielit.gov.in/imphal

A logo with blue text

Description automatically generated

**Certificate**

This is to certify that the project work entitled **“Face2Comic: Pix2Pix cGAN-Based Approach to Cartoonify Human Portraits”** is the comprehensive work done by **Keithellakpam Oken,** bearing Roll No. 2182 at National Institute of Technology, Langol, Manipur, under the guidance and supervision of **Dr. Khelchandra Thongam**, Associate Professor, NIT Manipur, Langol as External guide and **Laimujam Joy Singh, Guest Faculty, NIELIT IMPHAL, AKAMPAT** as internal Guide, for the partial fulfillments of the requirements for the award of the degree in Master’s of Computer Application (MCA)**, 6th Semester** Examination under Manipur University in the Academic session, **July 2021** to **June 2024**

This project is the bonafide work carried out by him. It has neither been published in any journal nor submitted in any academic institution before. This project is truly authentic and original.

Date: N. Debachandra Singh

Place: Director

NIELIT Imphal



A logo with blue text

Description automatically generatedA close up of a sign

Description automatically generated

**Certificate**

This is to certify that the project work entitled **“Face2Comic: Pix2Pix cGAN-Based Approach to Cartoonify Human Portraits”** is the comprehensive work done by **Keithellakpam Oken,** bearing Roll No. 2182 at National Institute of Technology, Langol, Manipur, under the guidance and supervision of **Dr. Khelchandra Thongam**, Associate Professor, NIT Manipur, Langol as External guide and **Laimujam Joy Singh, Guest Faculty, NIELIT IMPHAL, AKAMPAT** as internal Guide, for the partial fulfillments of the requirements for the award of the degree in Master’s of Computer Application (MCA)**, 6th Semester** Examination under Manipur University in the Academic session, **July 2021** to **June 2024**

This project is the bonafide work carried out by him. It has neither been published in any journal nor submitted in any academic institution before. This project is truly authentic and original.

Date: Laimujam Joy Singh

Place: Guest Faculty

NIELIT Imphal



***Department of Computer Science and Engineering***

**राष्ट्रीय प्रौद्योगिकी संस्थान, मणिपुर**

**NATIONAL INSTITUTE OF TECHNOLOGY, MANIPUR**

**Langol, Imphal, Manipur-795004, Ph. (0385) 2445812**

Email: khel@nitmanipur.ac.in, Website:- [www.nitmanipur.ac.in](http://www.nitmanipur.ac.in)

(An Autonomous Institute under MHRD, Govt. of India)

 **Certificate**

This is to certify that the project work entitled **“Face2Comic: Pix2Pix cGAN-Based Approach to Cartoonify Human Portraits”** is the comprehensive work done by **Keithellakpam Oken,** bearing Roll No. 2182 at National Institute of Technology, Langol, Manipur, under the guidance and supervision of **Dr. Khelchandra Thongam**, Associate Professor, NIT Manipur, Langol as External guide and **Laimujam Joy Singh, Guest Faculty, NIELIT IMPHAL, AKAMPAT** as internal Guide, for the partial fulfillments of the requirements for the award of the degree in Master’s of Computer Application (MCA)**, 6th Semester** Examination under Manipur University in the Academic session, **July 2021** to **June 2024**

This project is the bonafide work carried out by him. It has neither been published in any journal nor submitted in any academic institution before. This project is truly authentic and original.

Date: Dr. Khelchandra Thongam

Place: Associate Professor (CSE Dept.)

NIT Manipur, Langol

**DECLARATION**

I hereby declare that the project work entitled “**Face2Comic: Pix2Pix cGAN-Based Approach to Cartoonify Human Portraits”** is an authentic work developer by me at NIELIT, Imphal under the guidance and supervision of **Dr. Khelchandra Thongam**, Associate Professor, NIT Manipur, Langol as External guide and **Laimujam Joy Singh, Guest Faculty, NIELIT IMPHAL, AKAMPAT** as internal Guide, for the partial fulfillments of the requirements for the award of the degree in Master’s of Computer Application (MCA)**, 6th Semester** Examination under Manipur University in the Academic session, **July 2021** to **June 2024**

I also declare that all contents incorporated in this report has not been submitted in any form for the award of any other degree or diploma of any other Institutions or University

Date: …../…../….. (Signature)

Place:: …../…../….. **Keithellakpam Oken 18560205 (2018)**

**ACKNOWLEDGEMENT**

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and the institute. I would like to extend my sincere thanks to all of them.

I am highly thankful to my supervisor **Laimujam Joy Singh, Guest Faculty**, NIELIT Imphal for his guidance, constant supervision and for providing necessary information regarding the project and for conducting weekly progress review kept me going. I would like to express my gratitude towards my family & faculty members of the institute for their kind co-operation and encouragement which helped me in completion of this project

My thanks and appreciation also goes to my friends in developing the project and people who have willingly helped me out with their abilities.

Date: …../…../….. (Signature)

Place:: …../…../….. **Keithellakpam Oken 18560205 (2018)**

**ABSTRACT**

This project presents a novel application of Pix2Pix based on conditional Generative Adversarial Networks (cGAN) to translate human facial images into comic-style illustrations. By leveraging the Pix2Pix framework, we aim to generate high-quality, realistic comic portraits from real photographs. The dataset consists of paired images, enabling supervised learning to map input photos to their corresponding comic renditions. The training process involves optimizing the generator and discriminator to achieve visually appealing and accurate outputs. This approach demonstrates the potential of cGANs in artistic domains, offering a unique blend of technology and creativity to produce compelling visual art.

PROJECT

ON

**Face2Comic: Pix2Pix cGAN-Based Approach to Cartoonify Human Portraits**

CONTENTS

Chapter 1: **Introduction Page No.**

* 1. Introduction ………………………………………………… 2
  2. Objectives …………………………………………………... 3
  3. Limitations …………………………………………………. 3

Chapter 2: **Literature Review** …………………………………………… 4

2.1 Literature Review ……………………… …………………. 5-6

2.2 Merits of the proposed system …………………………....…. 6

Chapter 3: **Methodology** ………………………………………………… 7

3.1 Methodology …………………………………..………..…... 8-10

3.2 Generator Architecture …………. …………………………. 10-12

3.3 Discriminator Architecture …………………………………. 13-14

3.4 GAN Architecture ………………....…………………………15

3.5 Dataset Details and Pre Processing …… ……………………15-16

Chapter 4: **System Development** ……………………… ……………….. 17

4.1 Flowchart …....………………………………………………. 18

4.2 Programming Language Used ……………… ……………... 19-20

4.3 Libraries ……………………………………………………... 20-21

4.4 Hardware …………………………………………………….. 21

Chapter 5: **Model Implementation** ………………………..………… … 22

5.1 Dataset Used ……………………………………...…………. 23

5.2 Dataset Preprocessing ………………………………...……... 23-24

5.3 Model Compilation ………………………………………..… 24-25

5.4 Screenshots ……………………………………………..…… 26

5.5 Source Code …………………………………………………. 27-28

Chapter 6: **Conclusion** ………...…………………………………...… 29

6.1 Future scope …………...…………………………………… 30

6.2 Conclusion …………...…………………………..………… 30

Chapter 7: **Reference..** …..………………………………………………. 31

7.1 Reference ……………………………………………………. 32

CHAPTER 1

Introduction

**1.1 INTRODUCTION**

The advent of Generative Adversarial Networks (GANs) has revolutionized the field of image synthesis, enabling the generation of highly realistic images. Among the various GAN architectures, Pix2Pix stands out for its effectiveness in paired image-to-image translation tasks. This project leverages the Pix2Pix framework, based on conditional GANs (cGAN), to transform human facial images into comic-style illustrations.

The core idea of Pix2Pix is to use paired training data to learn the mapping from input images to output images, as well as a loss function to train this mapping. In our context, the input consists of real human facial photographs, and the target output is their corresponding comic-style renditions. This project aims to explore and demonstrate the capabilities of Pix2Pix cGANs in generating high-quality and aesthetically pleasing comic portraits from real photographs.

The dataset used in this project comprises paired images of real faces and their comic-style counterparts, allowing for supervised learning. The generator in the Pix2Pix framework learns to create comic images that are indistinguishable from the hand-drawn versions, while the discriminator evaluates the authenticity of the generated images against real comic images. The training process involves a delicate balance between these two components, optimizing the generator to produce images that can deceive the discriminator.

This approach not only showcases the artistic potential of GANs but also provides insights into the application of deep learning techniques in creative domains. By blending technology with creativity, this project contributes to the growing field of AI-generated art, offering a unique perspective on how machine learning can enhance and expand artistic expression. Through rigorous experimentation and fine-tuning, we aim to achieve a model capable of producing visually striking and contextually accurate comic portraits from real-world images.

**1.2 OBJECTIVES**

The primary objective of this project is to develop and optimize a deep learning model based on the Pix2Pix framework with conditional Generative Adversarial Networks (cGAN) to effectively translate human facial images into comic-style illustrations. By leveraging paired datasets of real and comic-style faces, we aim to train a robust model capable of generating high-quality, visually appealing comic portraits. This project seeks to demonstrate the feasibility and artistic potential of using advanced machine learning techniques in creative domains, providing a novel approach to digital art generation. Additionally, we aim to explore the practical applications of cGANs in transforming realistic images into stylistic renditions, thereby contributing to the broader field of AI-driven art and image synthesis.

**1.3 LIMITATION**

* Dataset Size: The project uses a limited dataset of 1000 image pairs. This relatively small sample size may limit the model's ability to generalize to a wide variety of face types and styles.
* Image Resolution: The model processes images at a fixed size of 128x128 pixels. This low resolution may result in loss of fine details in both input and output images.
* Training Duration: With only 30 epochs of training, the model may not have reached its full potential. GANs often require extensive training to produce high-quality results.
* Limited Style Variety: The model is trained on a single comic style. It may not be able to generate diverse comic styles or adapt to different artistic preferences.
* Computational Resources: The model's architecture, while effective, may be computationally intensive, potentially limiting its use in real-time applications or on devices with limited processing power.
* Lack of User Control: The current implementation doesn't allow users to influence specific aspects of the comic style conversion, limiting customization options.

CHAPTER 2

Literature Review

**2.1 LITERATURE REVIEW**

In this project, a model is built using a Generative Adversarial Network (GAN) with the goal of converting realistic face images into comic-style illustrations. The model consists of a generator and a discriminator, both implemented as convolutional neural networks.

An image-to-image translation method based on deep learning techniques is discussed. Using the Kaggle dataset "Comic Faces Paired Synthetic," the model is trained on pairs of realistic face images and their corresponding comic-style versions. The generator uses a U-Net-like architecture with skip connections, while the discriminator follows a traditional convolutional structure.

The proposed prediction model uses a combination of convolutional and transposed convolutional layers, along with batch normalization and LeakyReLU activations. Moreover, the result of the system is not only a generated comic-style image but also a discriminator output that attempts to distinguish between real and fake images. To test the system's efficiency, they set up an experiment by using 1000 image pairs from the dataset. The training process involves alternating between training the discriminator and the generator.

In this project, "Face to Comic Conversion using Generative Adversarial Networks" - used deep learning techniques to construct a model which transforms realistic face images into comic-style illustrations, solving an image-to-image translation problem which is an effective approach for style transfer tasks. The model uses a combination of binary cross-entropy loss for the adversarial component and mean absolute error for image reconstruction. As a result, the GAN model shows promising performance in generating comic-style images from realistic face inputs.

They evaluate the importance of the features that can be used to generate comic-style images, after examining key aspects of the input face images. Results suggest that these attributes contribute substantially to the final generated image, such as facial features, expressions, and overall composition.

In this project, the model was develop such that there are initial preprocessing steps combined with data augmentation techniques:

1. Image resizing to ensure consistent input dimensions (128x128 pixels).

2. Normalization of pixel values to the range [0, 1].

3. Horizontal flipping for data augmentation.

4. Use of RGB color channels for both input and output images.

**2.2 MERITS OF THE PROPOSED SYSTEM**

1. Easily identifies visual patterns: The GAN model can review large volumes of image data and discover specific patterns that would not be apparent to humans. For instance, it can learn to identify and transform key facial features into comic-style representations.

2. No human intervention needed (automation): With implementation of the GAN model, there is no need to have any manual intervention in the image conversion process. The model learns to generate comic-style images from realistic face inputs automatically.

3. Continuous Improvement: As the GAN model trains, it keeps improving in accuracy and efficiency. This lets it generate more convincing and stylistically consistent comic-style images over time.

4. Handling complex image data: The model is good at handling image data with multiple channels and complex features, and it can do this in a way that preserves important visual information while applying the desired style transfer.

5. Wide Applications: This technology could be applied in various fields such as entertainment (creating comic avatars), digital art (assisting artists in style conversion), or even in developing unique visual filters for social media platforms.

CHAPTER 3

Methodology

**3.1 METHODOLOGY**

The methodology for this face-to-comic conversion project involves several key steps, leveraging Generative Adversarial Networks (GANs) for image-to-image translation. Here's a detailed breakdown of the process:

1. **Data Acquisition and Preprocessing**
   * Dataset: The project uses the "Comic Faces Paired Synthetic" dataset from Kaggle.
   * Image Processing:
     + Images are read using OpenCV (cv2) and resized to 128x128 pixels.
     + Color space is converted from BGR to RGB.
     + Pixel values are normalized to the range [0, 1].
   * Data Augmentation: Horizontal flipping is applied to increase dataset diversity.
2. **Model Architecture** a. Generator:
   * Follows a U-Net-like architecture with encoder-decoder structure.
   * Encoder: Uses convolutional layers with LeakyReLU activation.
   * Decoder: Uses transposed convolutional layers with ReLU activation.
   * Skip connections between encoder and decoder layers.
   * Input shape: (128, 128, 3), Output shape: (128, 128, 3)

b. Discriminator:

* + Convolutional network structure.
  + Uses strided convolutions with LeakyReLU activation.
  + Batch normalization applied after most convolutional layers.
  + Input shape: Two (128, 128, 3) images concatenated.
  + Output: Single channel for binary classification (real/fake).

1. **Loss Functions**
   * Discriminator: Binary cross-entropy loss.
   * Generator: Combination of binary cross-entropy (adversarial loss) and mean absolute error (pixel-wise loss).
2. **Training Process**
   * Batch size: 5
   * Epochs: 30
   * Optimizer: Adam (learning rate: 0.002, beta\_1: 0.5)
   * Training loop: a. Generate fake images using the generator. b. Train discriminator on real and fake image batches separately. c. Train the combined GAN model, with the discriminator's weights frozen.
3. **Evaluation and Visualization**
   * Visual inspection of generated images compared to input images.
   * Plotting of discriminator loss and generator loss over epochs.
4. **Model Saving and Loading**
   * The trained generator, discriminator, and combined model are saved separately.
   * Capability to load pre-trained models for inference or further training.
5. **Inference**
   * Function to process single images through the trained generator.
   * Visualization of input image alongside the generated comic-style image.

**3.2 GENERATOR ARCHITECTURE**

The generator in the Pix2Pix framework is designed to translate input images into desired output images, in this case, from real human faces to comic-style illustrations. It follows an encoder-decoder structure with skip connections (U-Net architecture).

**Encoder**

* **Input Layer**: Takes an image of shape (128, 128, 3).
* **Layer 1**: Convolutional layer with 64 filters, kernel size 4x4, stride 2, padding, followed by LeakyReLU activation.
* **Layer 2**: Convolutional layer with 128 filters, kernel size 4x4, stride 2, batch normalization, LeakyReLU activation.
* **Layer 3**: Convolutional layer with 256 filters, kernel size 4x4, stride 2, batch normalization, LeakyReLU activation.
* **Layer 4**: Convolutional layer with 512 filters, kernel size 4x4, stride 2, batch normalization, LeakyReLU activation.
* **Layer 5**: Convolutional layer with 512 filters, kernel size 4x4, stride 2, batch normalization, LeakyReLU activation.
* **Layer 6**: Convolutional layer with 512 filters, kernel size 4x4, stride 2, batch normalization, LeakyReLU activation.
* **Layer 7**: Convolutional layer with 512 filters, kernel size 4x4, stride 2, batch normalization, LeakyReLU activation.
* **Layer 8**: Convolutional layer with 512 filters, kernel size 4x4, stride 2, batch normalization, LeakyReLU activation.

**Decoder**

* **Layer 1**: Transposed convolutional layer with 512 filters, kernel size 4x4, stride 2, batch normalization, dropout, ReLU activation, concatenated with corresponding encoder layer output.
* **Layer 2**: Transposed convolutional layer with 512 filters, kernel size 4x4, stride 2, batch normalization, dropout, ReLU activation, concatenated with corresponding encoder layer output.
* **Layer 3**: Transposed convolutional layer with 512 filters, kernel size 4x4, stride 2, batch normalization, dropout, ReLU activation, concatenated with corresponding encoder layer output.
* **Layer 4**: Transposed convolutional layer with 512 filters, kernel size 4x4, stride 2, batch normalization, ReLU activation, concatenated with corresponding encoder layer output.
* **Layer 5**: Transposed convolutional layer with 256 filters, kernel size 4x4, stride 2, batch normalization, ReLU activation, concatenated with corresponding encoder layer output.
* **Layer 6**: Transposed convolutional layer with 128 filters, kernel size 4x4, stride 2, batch normalization, ReLU activation, concatenated with corresponding encoder layer output.
* **Layer 7**: Transposed convolutional layer with 64 filters, kernel size 4x4, stride 2, batch normalization, ReLU activation, concatenated with corresponding encoder layer output.
* **Output Layer**: Transposed convolutional layer with 3 filters, kernel size 4x4, stride 2, activation function tanh to produce output image.

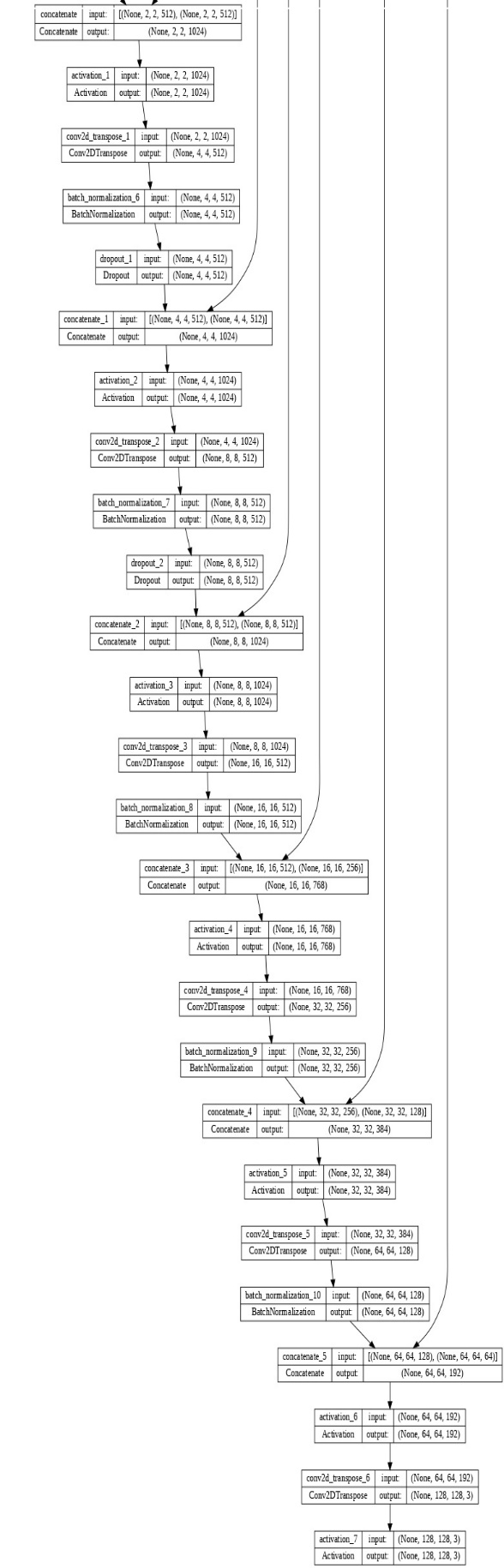
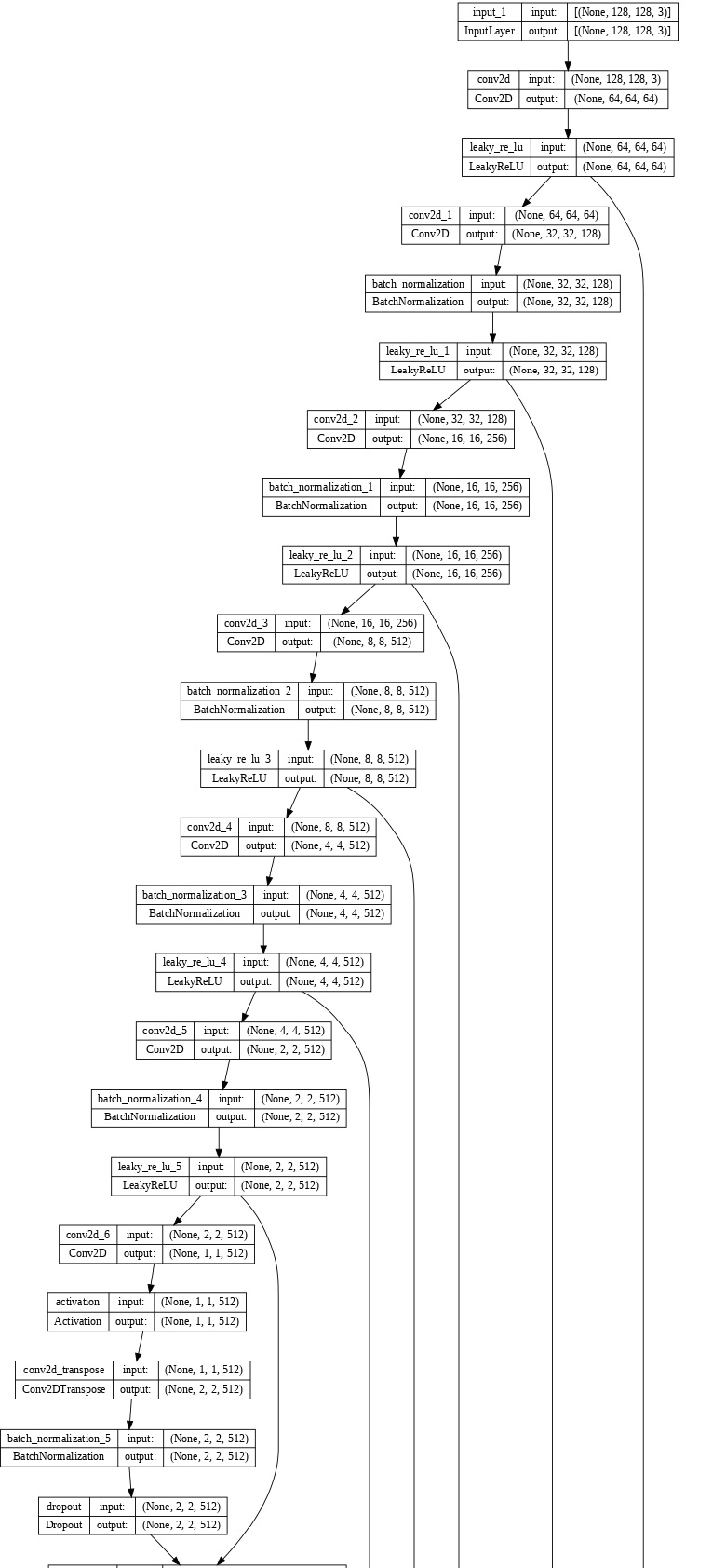


Fig. The Generator Architecture

**3.3 DISCRIMINATOR ARCHITECTURE**

The discriminator in the Pix2Pix framework is designed to distinguish between real and generated images. It uses a PatchGAN classifier that classifies 70x70 image patches as real or fake.

* **Input Layer: Takes two images (the input image and the target/comic image) concatenated along the channels axis, resulting in a shape of (128, 128, 6).**
* **Layer 1: Convolutional layer with 64 filters, kernel size 4x4, stride 2, LeakyReLU activation.**
* **Layer 2: Convolutional layer with 128 filters, kernel size 4x4, stride 2, batch normalization, LeakyReLU activation.**
* **Layer 3: Convolutional layer with 256 filters, kernel size 4x4, stride 2, batch normalization, LeakyReLU activation.**
* **Layer 4: Convolutional layer with 512 filters, kernel size 4x4, stride 2, batch normalization, LeakyReLU activation.**
* **Output Layer: Convolutional layer with 1 filter, kernel size 4x4, stride 1, activation function sigmoid to produce a probability map.**

**A close-up of a line of text

Description automatically generated**

**Fig. Discriminator Architecture**

**3.4 GAN ARCHITECTURE**

The GAN model combines the generator and discriminator:

* **Generator**: Takes an input image and generates a comic-style image.
* **Discriminator**: Takes the input image and the generated image, and produces a probability map indicating whether the patches are real or fake.
* The discriminator is not trainable during the training of the GAN model to ensure that only the generator is optimized during this phase.
* The loss function used is a combination of binary cross-entropy loss for the discriminator and mean absolute error (MAE) for the generator.

**Model Compilation**

* **Optimizer**: Adam optimizer with learning rate 0.0002 and beta\_1 0.5.
* **Loss Functions**: Binary cross-entropy for the discriminator, and a weighted sum of binary cross-entropy and MAE for the GAN.

**3.5 DATASET DETAILS AND PRE PROCESSING**

**Dataset:** The dataset consists of paired images of real human faces and their corresponding comic-style illustrations. Each pair includes a photograph of a human face and its hand-drawn comic equivalent, providing the necessary supervision for training the Pix2Pix model.

**Preprocessing:**

1. **Image Reading:**
   * Images are read from the dataset using OpenCV.
   * Each image is resized to 128x128 pixels to maintain consistency.
   * Color conversion is applied to change images from BGR to RGB format.
2. **Normalization:**
   * Pixel values are normalized to the range [0, 1] by dividing by 255.0.
3. **Data Loading:**
   * Paired images are loaded into memory.
   * The real face images are stored as input data, and the corresponding comic images are stored as target data.
   * The data is then converted into NumPy arrays for efficient processing.

CHAPTER 4

System Development

**4.1 FLOWCHART**

A black rectangle with white text

Description automatically generated

**4.2 PROGRAMMING LANGUAGE USED**

**PYTHON**

The programming language used or out project is Python. There are various versions of python till now, we used Python version 3.10.10.

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural. object oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Python was conceived in the late 1980s as a successor to the ABC language.

Python has a wide range of libraries and frameworks that make it a powerful tool for many different applications. Some of the most popular libraries and frameworks include NumPy, Pandas, Matplotlib, Tensor Flow, and PyTorch. These libraries make it possible to data perform complex data analysis, build machine learning models, and create data visualizations.

Another advantage of Python is its portability, Python code can run on a wide range of platforms, including Windows, macOS, and Linux. This makes it easy for developers to write code on one platform and deploy it on another. Python also has a large and active community of developers, who contribute to the language's development and help to create new libraries and frameworks.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

Python's popularity has also led to its widespread adoption in the industry. The use of Python in the industry has led to a high demand for Python developers, making it a lucrative career path for those with Python skills.

**4.3 IMPORTED LIBRARIES**

**Flask**

Flask is a micro web framework that has gained popularity in recent years due to its simplicity, flexibility, and ease of use. Developed by Armin Ronacher, Flask is a Python-based framework that enables rapid development of web applications with a minimalistic approach. Unlike other frameworks that bundle a plethora of features, Flask takes a modular approach, providing only the essential components required for building a web application.

Flask's lightweight architecture makes it an attractive choice for developers who value agility and simplicity. The framework's core features include a modular design, a flexible routing system, and a robust templating engine. Flask's extensibility is further enhanced by its vast array of extensions, which provide additional functionality for tasks such as authentication, caching, and database integration.

**Tensorflow**

TensorFlow is an open-source software library developed by Google Brain Team. It is used for machine learning, artificial intelligence, and neural networks. TensorFlow allows developers to create large-scale neural networks and deep learning models, and to run those models on a variety of platforms including desktop, mobile, and web environments.

TensorFlow provides a flexible platform for defining and running computations using data flow graphs. It also has a large community of developers and researchers, and it is constantly being updated with new features and improvements. TensorFlow is widely used in both academia and industry for a variety of applications, including image and speech recognition, natural language processing, and predictive analytics

**Pandas**

Pandas is an open-source data analysis and manipulation library for the Python programming language. It is designed to work with structured data, and is built on top of the NumPy library. Pandas provides data structures and functions needed to manipulate structured data, including functions for reading and writing data in a variety of formats (e.g. CSV, Excel, SQL, etc.), as well as tools for filtering, sorting, and aggregating data.

Pandas provides two primary data structures: Series (1-dimensional) and DataFrame (2-dimensional). A Series is similar to an array in Python, but it can have labeled indices, while a DataFrame is a 2-dimensional table with rows and labeled columns, similar to a spreadsheet or SQL table.

**Numpy**

NumPy (Numerical Python) is an open-source library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

NumPy provides an efficient and convenient way to perform operations on arrays and matrices, such as element-wise operations, matrix and vector multiplications, and various mathematical transformations. It is widely used in scientific computing, data analysis, and machine learning. NumPy arrays are also the underlying data structures for many other libraries in the Python data stack, such as Pandas, Matplotlib, and Scikit-learn.

**4.5 TECHNICAL REQUIREMENTS**

The model is trained on Google Colab T4 GPU making it highly efficient and good enough for training on a medium scale dataset for the Face2Comic Image translation. And since it’s on the cloud, it can be accessed and used remotely.

CHAPTER 5

**Model Implementation**

**5.1 DATASET USED**

The dataset for this project consists of paired images, where each pair contains a real human facial photograph and its corresponding comic-style illustration. These images are carefully curated to ensure a diverse and representative sample of different facial features, expressions, and artistic styles. The dataset provides the necessary supervision for training the Pix2Pix model, enabling it to learn the mapping between real and comic-style images.

**5.2 DATA PREPROCESSING**

Data preprocessing involves several critical steps to prepare the images for training:

1. Image Reading:
   * Images are loaded from the dataset using OpenCV.
   * Each image is read and resized to a uniform size of 128x128 pixels to maintain consistency across the dataset.
2. Color Conversion:
   * Images are converted from BGR (Blue, Green, Red) format to RGB (Red, Green, Blue) format, which is the standard color format for most deep learning frameworks.
3. Normalization:
   * Pixel values are scaled to the range [0, 1] by dividing by 255.0. This normalization step helps in speeding up the convergence during the training process.
4. Data Loading:
   * Paired images (real face and comic face) are loaded into memory.
   * The real face images are stored as input data, and the corresponding comic images are stored as target data.
   * The data is then converted into NumPy arrays for efficient processing and fed into the Pix2Pix model.

**5.3 MODEL COMPILATION**

The Pix2Pix model comprises two main components: the generator and the discriminator, which are compiled as follows:

1. Generator (U-Net Architecture):
   * Encoder: The encoder consists of several convolutional layers with increasing filter sizes and down-sampling using strided convolutions. Each layer applies batch normalization and LeakyReLU activation, capturing various levels of abstraction from the input image.
   * Decoder: The decoder consists of transposed convolutional layers that up-sample the encoded features back to the original image size. Skip connections from the encoder layers to the corresponding decoder layers are used to preserve spatial information, forming a U-Net architecture.
2. Discriminator (PatchGAN Architecture):
   * The discriminator is designed to classify 70x70 image patches as real or fake. It consists of several convolutional layers with increasing filter sizes, followed by batch normalization and LeakyReLU activation. The final layer uses a sigmoid activation function to output a probability map indicating the authenticity of the patches.
3. Loss Functions:
   * Generator Loss: A combination of binary cross-entropy loss and L1 loss (mean absolute error). The binary cross-entropy loss measures how well the generator can fool the discriminator, while the L1 loss ensures that the generated image is close to the ground truth in pixel space.
   * Discriminator Loss: Binary cross-entropy loss is used to measure the discriminator's ability to distinguish between real and fake images.
4. Optimizers:
   * Adam optimizer is used for both the generator and discriminator, with a learning rate of 0.0002 and beta\_1 set to 0.5. This configuration is chosen to ensure stable training.
5. Training:
   * During training, the generator creates comic-style images from real facial photographs, which are then evaluated by the discriminator. The discriminator's feedback helps to iteratively improve the generator's performance.
   * The model is trained over several epochs, with both the generator and discriminator weights updated based on their respective loss functions.

**5.4 SCREENSHOTS**

A white background with black text

Description automatically generated

**Fig. Homepage**

**A screenshot of a video game

Description automatically generated**

**Fig. Generated Output**

**5.5 SOURCE CODE**

from flask import Flask, request, render\_template, send\_file

import tensorflow as tf

import numpy as np

from PIL import Image

import io

app = Flask(\_\_name\_\_)

# Load the model

model = tf.keras.models.load\_model('./')

def preprocess\_image(image):

# Resize and normalize the image as expected by your model

image = image.resize((128, 128)) # Adjust size based on your model's requirement

image = np.array(image) / 255.0

image = np.expand\_dims(image, axis=0)

return image

def generate\_comic(image):

preprocessed\_image = preprocess\_image(image)

comic\_image = model.predict(preprocessed\_image)

comic\_image = (comic\_image[0] \* 255).astype(np.uint8)

comic\_image = Image.fromarray(comic\_image)

return comic\_image

@app.route('/')

def home():

return render\_template('index.html')

@app.route('/upload', methods=['POST'])

def upload\_file():

if 'file' not in request.files:

return 'No file part'

file = request.files['file']

if file.filename == '':

return 'No selected file'

if file:

image = Image.open(file.stream)

comic\_image = generate\_comic(image)

img\_io = io.BytesIO()

comic\_image.save(img\_io, 'PNG')

img\_io.seek(0)

return send\_file(img\_io, mimetype='image/png')

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

CHAPTER 6

**Conclusion**

**6.1 FUTURE SCOPE**

The application of Pix2Pix-based cGANs in image-to-image translation opens numerous avenues for future exploration and development. One potential direction is to expand the dataset to include a wider variety of artistic styles, such as watercolor, pencil sketches, and other comic genres, to make the model more versatile and adaptable to different artistic preferences. Additionally, integrating advanced techniques like StyleGAN could further enhance the quality and diversity of the generated images.

Another promising area is the incorporation of interactive user interfaces, allowing users to customize aspects of the generated comic images, such as adjusting facial expressions, hairstyles, or background elements. This can be achieved through fine-tuning and extending the existing model to incorporate user input and control.

**6.2 CONCLUSION**

This project demonstrates the efficacy of Pix2Pix cGANs in translating real human facial images into high-quality comic-style illustrations. By leveraging a dataset of paired images and employing a carefully designed generator and discriminator architecture, the model successfully learns the intricate mapping between real and comic images. The results showcase the artistic potential of GANs, highlighting their ability to blend technology with creativity to produce visually striking and contextually accurate outputs.

The comprehensive data preprocessing steps ensure that the input images are uniformly formatted and normalized, facilitating efficient and effective training of the model. The use of specialized loss functions and optimizers further enhances the model's performance, enabling it to generate realistic and appealing comic portraits.

Overall, this project not only contributes to the growing field of AI-generated art but also offers a novel approach to digital image transformation. The future scope of this work holds significant promise, with opportunities to expand and refine the model, integrate interactive features, and explore cross-domain applications, thereby pushing the boundaries of what is possible with AI-driven artistic expression.

CHAPTER 7

**Reference**

**7.1 REFERENCE**

1. Tensorflow Pix2Pix tutorial

https://www.tensorflow.org/tutorials/generative/pix2pix

2. Comic Face Dataset

https://www.kaggle.com/datasets/defileroff/comic-faces-paired-synthetic

3. General Adversarial Networks

https://arxiv.org/abs/1406.2661

4. Conditional Adversarial Networks

https://arxiv.org/abs/1411.1784

5. Image-to-Image Translation with Conditional Adversarial Networks

https://arxiv.org/abs/1611.07004

6. U-Net Architecture

https://www.geeksforgeeks.org/u-net-architecture-explained/