National College of Ireland

TEXT-TO-VIDEO GENERATION USING DCGAN

MSc Research Project
Data Analytics

Aniruddha Nandanwar
Student ID: x23104741

School of Computing
National College of Ireland

Supervisor: Syed Muhammed Raza Abidi

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Aniruddha Nandanwar
Student ID:	x23104741
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Syed Muhammed Raza Abidi
Submission Due Date:	16/09/2024
Project Title:	TEXT-TO-VIDEO GENERATION USING DCGAN
Word Count:	6023
Page Count:	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Aniruddha Nandanwar
Date:	16th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	✓
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	✓
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	✓

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

TEXT-TO-VIDEO GENERATION USING DCGAN

Aniruddha Nandanwar
x23104741

Abstract

Generating the videos from text has actually been a growing technology and proven to important and good type of challenge for generative types of models. This study will of course explore the development and evaluation of a Deep Convolutional Generative Adversarial Network (DCGAN) for text-to-video generation, mainly focusing on floral imagery. Using the 102-category flowers dataset which consist over 7,000 annotated image-caption pairs, this study has been trained the model to convert textual descriptions into video sequences of flowers. The combination of 300-dimensional GloVe embeddings gave accurate representation of textual inputs. Training was obviously conducted on low-resolution (64x64 pixels) images due to resource constraints which is optimizing model performance with a 16 GB RAM setup. The model used and faced 438 epochs, with each epoch averaging 25.77 seconds. Results shows that the model's capability to generate good flower videos from textual descriptions, achieving a generator loss of 1.3285 and discriminator loss of 1.2313. An interactive web application was of course been developed to showcase practical usage, enabling users to input flower descriptions and generate corresponding videos. Some types of challenges included managing computational resources and optimizing model hyperparameters for definitely good and efficient video synthesis.

Keywords: Text-to-Video Generation, Deep Learning, Generative Adversarial Networks (GANs), Deep Convolutional Generative Adversarial Networks (DCGANs)

1 Introduction

1.1 Background

Text-to-video generation is a very popular and growing type of technology that converts textual descriptions into corresponding video sequences automatically (Namratha et al.; 2024). This process mainly involves advanced machine learning models, such as deep neural networks, which of course interpret and transform textual input into visual content. The workflow of text-to-video generation starts with natural language processing (NLP) techniques to understand the meaning and context of the provided text. Next, the system will of course identify key elements such as objects, actions, and scenes described in the text. These elements are then mapped onto visual representations, which definitely include selecting or generating appropriate images, animations, or video clips. Techniques like image retrieval from large datasets or generative models like GANs (Generative Adversarial Networks) may be used for this purpose (Song et al.; 2018). One of the most and of course important challenge in text-to-video generation is maintaining constancy and things to the original text while obviously ensuring the generated video

is visually engaging and realistic. Achieving this balance mainly requires combining and using multiple AI components, such as language models for text understanding and computer vision models for visual synthesis things. Applications of text-to-video generation having so many types of domains which includes content creation, education, entertainment, and virtual reality (Zhou et al.; 2024). As technology advances, text-to-video systems are expected to become more complex.

1.2 Aim of the Study

The primary aim of this study is to of course develop and evaluate a deep learning model capable of generating videos from textual descriptions using the power of Deep Convolutional Generative Adversarial Networks (DCGANs). This research looks to bridge the gap between natural language processing and computer vision by obviously converting detailed text inputs into dynamic visual sequences, focusing specifically on floral imagery. By using the 102-category flowers dataset, which includes over 7,000 image- caption pairs, the study aims to actually create a robust model that can accurately interpret and visualize text descriptions. A key objective is to of course evaluate the performance of such a model when trained on low- resolution images (64x64 pixels), given the things of computational resources. This includes exploring the combination of GloVe embeddings for text representation and evaluating the model's capability to generate high and good quality images and videos under these conditions. The study also aims to develop a user-friendly type of web application to show the practical application of the model which actually allow users to input textual descriptions and generate corresponding videos.

1.3 Research Objectives

There are some research objectives in this report are:

1. To create a Deep Convolutional Generative Adversarial Network (DCGAN) capable of generating videos from textual descriptions mainly focus on floral imagery.

2. To use the 102-category flowers dataset which consist over 7,000 image-caption pairs, to train and validate the model. Ensure that each image-caption pair is effectively used to enhance the model's learning and output quality.

3. To have 300-dimensional GloVe embeddings to represent text descriptions accurately which is having the conversion of textual data into meaningful visual representations.

4. To train the model using low-resolution images (64x64 pixels) to manage computational resource things while maintaining the quality of generated videos.

1.4 Research Questions

There are some research questions in this report are:

1. How effective is the integration of GloVe embeddings in converting textual descriptions of flowers into meaningful visual representations?

2. What are the computational challenges and resource requirements associated with training a DCGAN model to generate videos from low-resolution (64x64) flower images and text descriptions?

3. How does the quality of generated videos vary with the complexity and specificity of textual inputs describing flower characteristics such as color, shape, and species?

4. What are the optimal hyperparameters and architecture configurations for the generator and discriminator networks to achieve high-quality video generation from textual inputs?

1.5 Research Gaps

Research gaps in text-to-video generation using DCGANs include the need for optimized methods to handle high-dimensional text embeddings effectively within the generator framework. Additionally, further exploration is necessary into enhancing model robustness for diverse and nuanced textual descriptions of flowers. Another gap lies in evaluating the scalability of the current approach to higher image resolutions while maintaining computational efficiency. Moreover, there is a gap in understanding the potential biases or limitations in generated videos and addressing user interaction challenges in the context of generating videos from user-provided textual inputs.

The rest of the project is then organized as follows: Section 2 introduces the methodology of CRISP-DM and its applicability to the project. This section covers an in-depth review of related literature about GANs, especially Deep Convolutional GANs, and their application in the tasks of text-to-image and video generation. In Section 3, we talk about the dataset used, namely the 102 Flower Dataset, and various preprocessing techniques that were used to set the data in place for model training. Section 4 presents the architecture and implementation of the DCGAN model with regard to the generator and discriminator networks. We explain the evaluation metrics and methodologies for assessing the quality and accuracy of generated videos in Section 5. Section 6 presents the experimental results, the evaluation of performance of the DCGAN model, and analysis of several performance metrics. Section 7 finally concludes the project by discussing the findings that fill the research gaps and point out areas for future work in text-to-video generation using DCGANs.

2 Literature Review

2.1 Text-to-Video Generation using Text Embeddings

In text-to-video generation, text embeddings are very very important for converting textual descriptions into numerical representations that models can use to generate corresponding video content. Key models that actually include text embeddings include Word2Vec, GloVe, and FastText, which transform words into vectors that capture semantic relationships.

In their paper, (Soares and Barrére; 2019) addresses the challenge of effectively navigating video lectures by obviously proposing an optimization model for temporal segmentation. Video lectures, integral to daily learning and exploration, regularly give some kind of difficulties in finding specific content due to their extensive nature covering multiple topics, not all of which may be relevant to every viewer. This issue results in users spending excessive time searching within irrelevant type of content. To mitigate this, (Soares and Barrére; 2019) have been suggested a type of solution with the help of temporal segmentation, enabling non-linear navigation across different topics within a single video lecture. Their proposed approach uses features extracted from audio transcripts using Word2Vec representations and combined low-level acoustic characteristics.

In their study, (Dong et al.; 2018) introduces an innovative type of approach to content description retrieval from images and videos which of course focusing exclusively on the visual domain rather than using a joint subspace approach. Their proposed method centers on Word2VisualVec, a novel deep neural network architecture designed to predict visual feature representations directly from textual inputs. Unlike existing methods, which mainly combine image and video caption retrieval in a shared subspace, (Dong et al.; 2018)'s approach uses multi-scale sentence vectorization to encode example captions into textual embeddings. These embeddings are of course then transformed into visual features using a simple multi- layer perceptron, giving efficient retrieval of relevant type of content descriptors from visual inputs. Experimental evaluations has been conducted on some kind of benchmarks such as Flickr8k, Flickr30k, the Microsoft Video Description dataset.

In their research, (Miech et al.; 2019) presented a novel kind of approach to learning text-video embeddings using a large-scale dataset called HowTo100M which of course consist 136 million video clips extracted from 1.22 million narrated instructional web videos. Unlike traditional methods that trust on manually annotated type of captions, (Miech et al., 2019) used automatically transcribed narrations associated with these videos, making the dataset creation process faster, scalable, and cost-effective.

The main and primary contributions of their work are kind of triple. Firstly, they introduce HowTo100M as a valuable resource for training text-video embeddings, addressing the challenge of dataset scalability and accessibility in this domain. Secondly, they have of course showed performance of embeddings trained on this dataset with the help of state-of-the-art results in tasks such as text-to-video retrieval and action localization, mainly on instructional video benchmarks like YouCook2 and CrossTask. To preprocess the transcribed narrations they applied standard techniques such as discarding common English stop-words and usse pre-trained word embeddings from GoogleNews Word2Vec model to represent textual content effectively.

In their project, (Hindocha et al.; 2019) propose using GloVe (Global Vectors for Word Representation), a popular word embedding model developed by Stanford University, to generate word vectors that capture semantic information. Unlike other models like Word2Vec, GloVe constructs word vectors which is of course based on the global statistical type of information of word which allows it to encode semantic type of similarities and relationships effectively. One of the main problem or kind of challenge which have been addressed by (Hindocha et al.; 2019) is evaluating the performance and effectiveness of GloVe embeddings in semantic comparison tasks compared to alternative models like Word2Vec.

(Singgalen and Abdurrahman; 2024) proposes combining and having the GloVe model into Social Network Analysis (SNA) to extract semantic relationships from content reviews of "Wonderland Indonesia," mainly focusing on a video by Alffy Rev ft. Novia Bachmid (Chapter 1) within the YouTube community. The study have been identifies a gap in this research on the performance of this integration mainly for Wonderland Indonesia's content reviews. Using the CRoss-Industry Standard Process for Data Mining (CRISP-DM), the study have been of course used topic analysis and SNA methodologies to analyze the reception and impact of the video content. The study have also conducts sentiment analysis with the help of Vader and TextBlob on a subset of 2,204 posts. (Ali et al., 2020) proposes an automated type of approach for generating rule-based co-speech gesture mappings to of course enhance interactions by using human-like co-speech gestures which have been derived from large publicly available video datasets without human

expert intervention. In their approach, word embeddings, mainly using the GloVe model which have been used at runtime for semantic-aware rule searching.

Table 1: Summary of Studies on Video and Image Caption Retrieval

Study	**Main Focus**	**Methodology**	**Approach**	**Key Findings**
Soares and Barrere, 2019	Video and image caption retrieval	Word2VisualVec neural network	Proposes a visual-only approach for text-to-video retrieval using neural networks.	Achieved state-of-the-art results for text-to-video retrieval and action localization across datasets.
Dong et al., 2018	Image and video caption retrieval	Word2VisualVec	Develops a method for predicting visual features from textual input, extending to video caption retrieval.	Demonstrated effectiveness across multiple datasets and domains, outperforming traditional models.
Miech et al., 2019	Learning text-video embeddings	Word2Vec	Trains embeddings from transcribed video narrations, improves text-to-video retrieval and action localization.	Effective transfer of embeddings to diverse domains, showing superior performance on benchmark datasets.
Hindocha et al., 2019	Word embeddings for semantic comparisons	GloVe word embeddings	Uses GloVe embeddings to compare semantic differences between phrases, evaluates against Word2Vec.	Aims to enhance semantic comparison accuracy, suggests GloVe as a viable alternative to Word2Vec.
Singgalen, 2024	Social network analysis of video content reception	GloVe word embeddings	Integrates GloVe with SNA for analyzing YouTube video reception, evaluates sentiment and engagement metrics.	Highlights storytelling effectiveness, community engagement, and sentiment analysis insights from video content.
Ali et al., 2020	Automated generation of co-speech gesture mappings	GloVe word embeddings	Automates rule-based co-speech gesture mapping using GloVe embeddings, compares with Levenshtein distance.	Achieved comparable performance to human-generated mappings, enhanced variety of activated gestures.

2.2 Text-to-Video Generation using LSTM and GAN

In their paper, (Yang et al.; 2018) propose a good type of approach to enhance video captioning with a combination of adversarial learning and LSTM networks. The main and primary motivation behind their novel approach is to of course address the limitations of LSTM-based methods in video captioning, which struggle with error things over extended sequences. While LSTM networks is good in capturing the temporal type of dynamics of video data for caption generation, they mostly found some types of challenges with maintaining accuracy over longer sequences. To handle these problems and challenges, (Yang et al.; 2018) introduced a hybrid architecture known as LSTM-GAN (Generative

Adversarial Network). This architecture will actually include two key components: a "generator" responsible for producing textual descriptions based on the video content, and a "discriminator" tasked with evaluating generated captions. The discriminator operates as an adversary to the generator which will then guide it towards producing more accurate and relevant type of captions by providing feedback on the generated outputs.

In their study, (Gupta et al.; 2022) presented an innovative and good type of approach to handle the challenging task of unconditional video generation using a recurrent Generative Adversarial Network (GAN) architecture. While generative models have achieved good type of success in high image synthesis from noise. The proposed solution introduces a Recurrent Variational GAN (RV-GAN) architecture designed to model the distribution of video data. Central to their approach is the development of a novel LSTM variant named TransConv LSTM (TC-LSTM). Unlike traditional ConvLSTM units which are mainly used for video processing but obviously struggle with unconditional video generation, TC-LSTM combine and use a transpose convolutional structure in its input-to-state transitions. Also there is and another good approach which have been given by (Islam et al.; 2019) which have used for Bangla text generation using deep learning techniques, mainly focusing on Long Short-Term Memory (LSTM) networks—a type of Recurrent Neural Network (RNN). The main and primary objective of their research is to automate the process of generating Bangla text sequences, which is a very very important type of task in natural language processing and can find applications in so so many fields such as machine translation, speech recognition, and image captioning. The proposed approach uses the capabilities of LSTM networks, which are very very good and well-suited for modeling sequential type of data like text due to their ability to capture long-term dependencies. An IRC-GAN (Introspective Recurrent Convolutional GAN) approach has been used by (Deng et al.; 2019) which mainly aimed at addressing the problems and challenges of automatically generating high-quality videos that maintain semantic type of consistency with given textual descriptions. The main and primary issues include the lack of effective types of methods to measure semantic alignment between generated videos and text. The proposed IRC-GAN approach contain and have two key innovations. First, they present a recurrent transconvolutional generator architecture. This generator combined LSTM cells with 2D transconvolutional layers, which are of course designed to focus more on the details of individual video frames compared to traditional 3D convolutional layers. Experimental evaluations conducted by (Deng et al.; 2019) who validated the performance of IRC-GAN across three types of datasets.

In their research, (Bin et al.; 2018) introduced a novel type of framework for video captioning that aims to overcome limitations in existing approaches, which mostly fail to accurately describe dynamic motions and global temporal relationships in videos. The proposed framework combines Bidirectional Long Short-Term Memory (BiLSTM) networks and a soft attention mechanism to enhance the generation of basic and accurate captions for videos. By using BiLSTM, the framework addresses the problems of previous methods that mainly trust mainly on local temporal knowledge, mainly limited to short sequences of frames.

In their research, (Balaji et al.; 2019) introduces the Text-Filter conditioning Generative Adversarial Network (TFGAN), a novel type of approach aimed at facing the complex task of text-to-video synthesis using conditional generative models. The main and primary objective of their work is to improve the association between textual descriptions and generated videos.

Table 2: Generated Flower Video from Text Input

Study	Proposed Approach	Key Challenges Addressed	Results
Yang et al., 2018	LSTM-GAN for video captioning. Uses adversarial learning and LSTM networks.	Addressing deficiencies of LSTM in video captioning, mitigating error accumulation.	Outperforms existing methods in video captioning tasks on standard datasets. Significant improvements in caption accuracy and relevance.
Gupta et al., 2022	RV-GAN with TC-LSTM for unconditional video generation.	Challenges in unconditional video generation, leveraging LSTM and trans convolutional layers.	Improved video generation quality, handles longer sequences, applies to class-conditional and text-to-video synthesis tasks.
Islam et al., 2019	Bangla text generation using LSTM for deep learning approach.	Early-stage development for Bangla language, validating model accuracy.	Successfully generates Bangla text with satisfactory accuracy rates.
Deng et al., 2019	IRC-GAN integrating recurrent trans convolutional generator and mutual-information introspection.	Issues in global temporal understanding and semantic consistency in video generation.	Demonstrates superior video generation quality and semantic alignment compared to state-of-the-art methods on multiple datasets.
Bin et al., 2018	BiLSTM and soft attention mechanism for video captioning, focusing on global temporal information.	Lack of accurate global motion representation and semantic alignment in video captioning.	Significant improvements in describing motions accurately and maintaining semantic consistency in generated captions. Outperforms existing methods on benchmark datasets.
Balaji et al., 2019	TFGAN with multi-scale text-conditioning for text-to-video synthesis.	Improving text-video associations, and handling novel category generation.	Generates high-quality videos from text descriptions, and surpasses existing approaches in visual quality, semantic coherence, and diversity of generated videos.

Therefore, the existing research on text-to-video generation discusses promising directions and Fig. 2: Architecture of a couple of text-to-video generation methodologies such as text embeddings, LSTM, and GAN architectures. Text embeddings such as Word2Vec, GloVe and Fasttext have been important in converting text descriptions of videos into vectors for generation and retrieval of videos. Thus, methods such as Word2VisualVec have enhanced the content search and retrieval in video databases and temporal segmentation and HowTo100M addressing the scalability problem by using large-scale video narrations. Although LSTM networks can have problems with working with long sequences, they must be augmented with GANs to provide higher accuracy in video captioning as in LSTM-GAN and IRC-GAN. The next-level RV-GAN and TFGAN architectures have also considered unconditional video generation and the text-video synthesis introducing the multi-scale text conditioning and transconvolutional layers semantically and visually improved. In different research works, the major concern has been the increase in the quality, relevance, and semantic similarity of the produced videos to the input texts. These approaches remain relevant, able to solve the issues in the representation of global motion, the management of long-term dependencies, and the integration of visual and textual content, which indicates the promising future of further study in text-to-video generation.

3 Methodology

3.1 Dataset Description

The dataset used for this project on text-to-video generation using Deep Convolutional Generative Adversarial Networks (DCGAN) actually contains images and corresponding captions from the 102 Flowers Dataset. This data `https://www.robots.ox.ac.uk/~vgg/data/flowers/102/index.html`, contains a huge collection of flower images categorized into 102 classes, with each category containing between 40 to 258 images. In total, the dataset consists of over 7,000 annotated image-description pairs, where each image is of course accompanied by a descriptive caption. The images in the dataset are annotated with detailed captions that describe the appearance and characteristics of each flower. These captions provide rich textual descriptions that are very very important for training the text-to-video generation model. Due to resource constraints, the images are resized to a low-resolution format of 64x64 pixels for training purposes, which helps manage computational demands while still enabling meaningful model training. During data preprocessing, the images are converted into NumPy arrays, which are very good for handling and processing within the neural network framework. Similarly, the captions are processed into embeddings with the help of pre-trained 300-dimensional GloVe embeddings. These embeddings capture the semantic meaning of words in the captions which actually gives the model's understanding of the textual input and enhancing the generation of coherent and relevant video sequences.

3.2 CRISP-DM

The CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology is a structured type of framework for data mining projects which consist of six phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. This framework actually have and ensures a systematic type of approach

Figure 1: CRISP-DM Flow Diagram

to data-driven projects for obviously enhancing clarity and efficiency. In this project, the goal of the CRISP-DM framework is to develop a model that generates videos of flowers from text descriptions using Deep Convolutional Generative Adversarial Networks (DCGANs). The process begins with Business Understanding to define objectives and success things. Data Understanding mainly includes collecting and analyzing the 102 Flower Dataset and corresponding text descriptions. Data Preparation includes converting images to 64x64 pixel arrays and transforming captions into GloVe embeddings. The Modeling phase actually focuses on building and training the DCGAN, creating discriminator and generator networks, and setting loss functions and optimizers. Evaluation assesses the model's performance in generating accurate and realistic videos. Finally, the Deployment phase implements a Flask-based GUI, enabling users to input text and receive generated videos.

3.2.1 Business Understanding

The Business Understanding phase in the CRISP-DM methodology mainly focuses on clearly defining the project's objectives and success things to of course ensure alignment with business goals. For this project, the primary objective is to develop a model capable of generating realistic type of videos of flowers based on text descriptions using Deep Convolutional Generative Adversarial Networks (DCGANs). The motivation behind this project is to obviously explore the potential of advanced generative models in transforming textual descriptions into visual content, which can have applications in various fields such as digital content creation, education, and virtual reality. There are some kind of key success criteria which of course include the model's ability to generate high-quality, visually accurate videos that accurately reflect the input descriptions. Understanding the constraints, such as limited computational resources leading to the decision to use 64x64 pixel images is very very important. Additionally, the project aims to provide a user-friendly type of interface which has been of course developed using Flask, where users can input text descriptions and receive generated videos as outputs.

3.2.2 Data Understanding

The Data Understanding phase in the CRISP-DM methodology includes collecting and exploring the dataset to gain data or things that will guide the steps of the project. For this text-to-video generation project, the primary and main type of data sources are the 102 Flower Dataset and its associated captions. This dataset contains over 7,000 images of flowers across 102 categories, each annotated with descriptive text. The images and captions provide a good and of course rich type of source of paired visual and textual data necessary for training the DCGAN model. During this phase, the images are analyzed for quality and consistency, with a decision to resize them to 64x64 pixels due to resource constraints. The text descriptions are examined to ensure they are detailed and relevant which is suitable for generating meaningful video content. The captions are then processed into 300-dimensional GloVe embeddings, which capture the semantic meaning of the words. This phase also includes identifying any kind of potential issues with the data, such as missing or inconsistent entries, and addressing them to ensure a clean and reliable dataset.

3.2.3 Data Preparation

The Data Preparation phase in the CRISP-DM methodology is actually used for transforming raw data into a format suitable for modelling. For this text-to-video generation project, the first step includes obviously pre-processing the flower images from the 102 Flower Dataset. Each image is resized to 64x64 pixels to align with our computational resource things, and then converted into NumPy arrays for good and efficient processing during model training. To ensure data consistency, I will store the processed images and their corresponding embeddings in structured formats, such as CSV files for the captions and binary files for the images. This organization will of course gives easy loading and manipulation during the modeling phase. Also, the data preparation phase includes combining the image and text data into a good type of dataset, ensuring each image is correctly paired with its caption embedding. Any missing or corrupted type of data points are identified and handled appropriately to maintain dataset integrity.

3.2.4 Modelling

The Modeling phase in the CRISP-DM methodology mainly focuses on developing and training the Deep Convolutional Generative Adversarial Network (DCGAN) to generate videos from text descriptions. The first step actually include to create the architecture for the DCGAN, which consists of two neural networks: the discriminator and the generator. The discriminator's role is to distinguish between real and generated images, while the generator creates images from the text embeddings. I will begin by defining the structure and parameters of these networks, ensuring they are capable of handling the 64x64 pixel images and the 300-dimensional GloVe embeddings. The generator network is of course designed to take text embeddings as input and generate corresponding flower images. The discriminator network is trained to differentiate between real images from the dataset and fake images produced by the generator. Loss functions are actually defined for both networks, with the generator's loss to produce realistic type of images and the discriminator's loss ensuring it can accurately classify real versus fake images. This phase is actually concluding a trained DCGAN capable of generating high-quality flower images from textual descriptions.

3.2.5 Evaluation

The Evaluation phase in this project using the CRISP-DM methodology mainly focuses on evaluating the performance and quality of the generated videos produced by the trained Deep Convolutional Generative Adversarial Network (DCGAN). The primary goal is to measure how well the model translates text descriptions into visually accurate and realistic type of videos of flowers.

3.2.6 Deployment

The Deployment phase in this project will make the generated video generation model accessible and usable for end-users. After successful evaluation of the DCGAN model's performance in generating videos from text descriptions of flowers, the next step is to of course deploy the model into a practical application. Firstly, the model is integrated into a user-friendly kind of interface using Flask, a web framework in Python. This interface will obviously allows users to input textual descriptions of flowers and receive corresponding generated videos as outputs. The Flask application is designed to handle user interactions smoothly.

3.3 Model Training

'train_step' function using TensorFlow for training a Generative Adversarial Network (GAN). The function is with '@tf.function', which compiles it into a TensorFlow graph for optimized performance. During each training step, the function first generates a batch of random seed vectors and then will use these with real and fake captions to produce images with the generator network. Also, it evaluates the discriminator's responses to real images paired with real and fake captions, as well as to generated images with real captions. The 'generator_loss' function will of course compute the generator's loss based on how well it fools the discriminator, aiming to generate images that the discriminator identifies as real. The 'discriminator_loss' function calculates the discriminator's loss, which includes separate components for correctly classifying real images and identifying fake images, whether they are paired with real or fake captions. Within the 'train_step', TensorFlow's gradient tapes ('gen_tape' and 'disc_tape') are of course used to actually calculate gradients of the generator and discriminator losses respectively, with respect to their trainable variables. These gradients are then obviously applied using respective optimizers ('generator_optimizer' and 'discriminator_optimizer') to update the network weights, optimizing them towards better performance in generating and discriminating images. This iterative type of process of adversarial kind of training continues will iteratively improving the generator's ability to produce realistic types of images and the discriminator's ability to accurately distinguish between real and generated images. 'train' function which shows the training process of a Generative Adversarial Network (GAN) for generating images from textual descriptions. It starts by initializing a fixed set of random seed vectors and embeddings used to generate preview images throughout training. Each epoch iterates through the training dataset, consisting of batches of images and corresponding embeddings. For each batch, a fake set of embeddings is generated by of course shuffling the original embeddings, which serves as a negative example for the discriminator. The 'train_step' function is to compute and apply gradients for both the generator and discriminator networks based on their respective losses. These losses,

averaged across batches, are printed at the end of each epoch, along with the elapsed type of time.

3.4 Model Building

The 'build_generator_func' function has been constructed a generator model for of course generating images based on inputs of a seed vector ('input_seed') and an embedding vector ('input_embed'). The model begins by having the seed vector with an embedding vector that is initially processed through a dense layer and LeakyReLU activation to enhance semantic understanding. This input is reshaped and upsampled with the help of a series of 'UpSampling2D' layers to gradually increase spatial dimensions. Each upsampled layer is been followed by a 'Conv2DTranspose' layer to learn feature maps from the upsampled data, with batch normalization and LeakyReLU activation applied for stabilization and non-linearity. The final layer will obviously uses 'Conv2DTranspose' to produce an output image with specified channels, and a 'tanh' activation function ensures pixel values range from -1 to 1, suitable for image generation tasks. The 'build_discriminator_func' function constructs a discriminator model which have been designed to classify images as real or generated by the generator in a Generative Adversarial Network (GAN). It takes two inputs: an image ('input_shape') and an embedding vector ('input_embed') that represents textual information associated with the image. The image input undergoes a series of convolutional layers ('Conv2D') with LeakyReLU activations to of course extract features and downsample the spatial dimensions. Dropout layers are used for regularization to prevent overfitting. Batch normalization layers stabilize training by obviously normalizing the input to each layer. The embedding vector is processed with the help of a dense layer and LeakyReLU activation, reshaped, and concatenated with the feature maps from the convolutional layers to combine image and text information. This model architecture will definitely enables the discriminator to easily differentiate between real images from the dataset and synthetic images generated by the generator which is very very important for the adversarial training process in GANs. Functions for computing the loss functions used in training a Generative Adversarial Network (GAN) for image generation. The 'cross_entropy' function initializes a Binary Cross-Entropy loss object from TensorFlow, which is commonly used in binary classification tasks such as distinguishing real from fake images in GANs. The 'discriminator_loss' function calculates the loss for the discriminator network. It of course computes two things: 'real_loss', which measures how well the discriminator correctly classifies real images with real text descriptions (targeting values between 0.8 to 1.0 to encourage realistic discrimination), and 'fake_loss', which evaluates how well the discriminator identifies fake images, both with real and fake text descriptions (using target values between 0.0 to 0.2 for adversarial type of training). The average of these losses forms the total discriminator loss, which of course guides the network in improving its ability to distinguish between real and generated images. The 'generator_loss' function actually computes the loss for the generator network. It evaluates how effectively the generator fools the discriminator by maximizing the probability that generated images are classified as real. Here, 'tf.ones_like(fake_output)' generates a tensor of ones as the target which shows the generator's objective to produce images that resemble real images as closely as possible. In GAN training, these loss functions are very very important for optimizing the networks' parameters. The optimizer, such as Adam or RMSprop, would be used to of course minimize these losses iteratively during training.

4 Design Specification and Implementation

This project focuses on the generation of videos from text descriptions using Deep Convolutional Generative Adversarial Networks (DCGANs). The workflow which has been shown in Figure 4.1, details each step from dataset collection to the final output displayed in a web application. The design is structured to ensure a systematic type of approach to text-to-video generation which is having data preprocessing, model building, training, and evaluation phases. The process begins with the Flower Dataset, which contains images of flowers along with their respective captions. This dataset is actually sourced from Oxford Flowers, includes 102 flower categories with each category having 40 to 258 images. The total number of image-description pairs exceeds 7000 which actually providing a robust and good type for training the models. In the Data Preprocessing stage, images are converted into numpy arrays of a pre-set pixel size (64x64) to manage computational resources effectively. Also, the textual descriptions are transformed into embeddings using 300-dimensional GloVe embeddings. This step actually ensures that both images and text data are in a compatible format for the subsequent stages. Model Building includes the construction of two key components: the Generator and the Discriminator. The generator network takes in random noise and text embeddings to create images, while the discriminator network shows the authenticity of these images by of course distinguishing between real and generated images. Both networks are important thing to the adversarial training process that DCGANs use. During Model Training, the generator and discriminator are trained iteratively. The generator aims to create realistic type of images that can fool the discriminator, while the discriminator improves its accuracy in identifying real versus fake images. The training process involves calculating losses for both networks, adjusting gradients, and optimizing the networks' parameters. After training, the models will definitely do and evaluation thing so that it can easily ensure that they generate high-quality outputs. The Evaluation phase evaluate the performance of the models based on their ability to create convincing images from textual descriptions. Finally, the project will create this in a Web Application displaying Generated Video. Implemented using Flask, this application allows users to input text descriptions and view the corresponding generated videos.

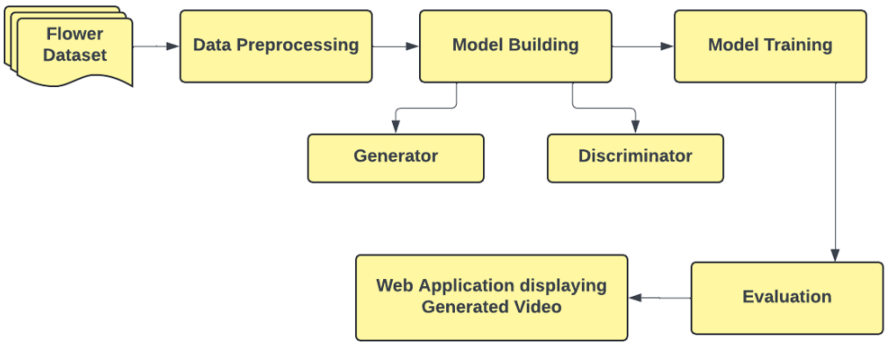

Figure 2: Workflow Diagram

5 Evaluation

During the evaluation phase, the performance of the DCGAN model was closely monitored and recorded. After 438 epochs of training, the generator loss was 1.3285 and the discriminator loss was 1.2313, with each epoch taking approximately 25.77 seconds. The model successfully generated 28 images for each input text, which were subsequently converted into a video, demonstrating the effective synthesis of visual content from textual descriptions. Training the model was resource-intensive, requiring 12 hours to complete. This high demand for computational power arises from the necessity to merge both text and image data. The images were trained at a resolution of 64x64 pixels to manage the computational load, as increasing the resolution significantly raises the memory requirements. The chosen resolution is a compromise, balancing quality and resource availability. In this case, a system with 16 GB of RAM was utilized, which was sufficient for handling the combined high-dimensional text vectors and image data.

5.1 Performance Evaluation

Figure 3 shows that the performance evaluation of the text-to-image generation model, actually showing the generator's capability. The code snippet begins by loading preprocessed caption embeddings from a binary file, which contains the semantic representations of the textual descriptions used to guide the image generation process. These embeddings are very very important because they encode the information that the generator uses to create images. Next, the generator model is initialised using the 'build_generator_func' function, mainly the dimensions for the seed vector, embedding size, and the number of image channels. The generator's pre-trained weights are then loaded from a specified file path which of course ensures that the model is ready to generate images based on prior training. To generate a sample output, a random noise vector of size 100 is created, which serves as the initial input for the generator. Along with this noise vector, a specific caption embedding (selected from the pre-loaded embeddings) is fed into the generator. In this case, the 25th caption embedding is actually been used. The generator processes these inputs and produces a generated image.

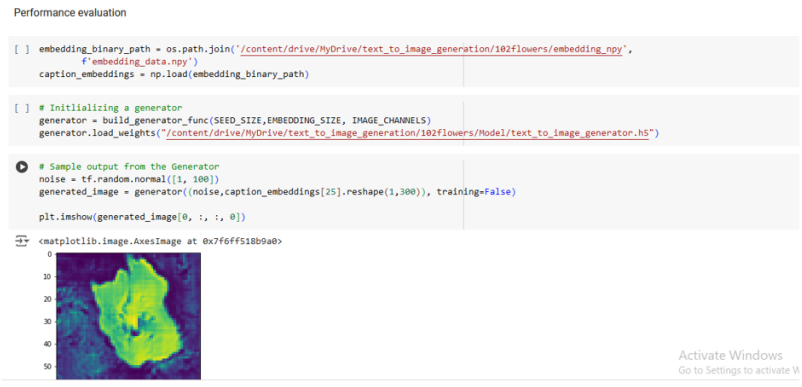

Figure 3: Sample Output from the Text-to-Image Generator

Figure 4 shows the process of generating images from a textual description using the trained DCGAN model. The function 'test_image' is designed to convert an input text,

in this case, "this flower is purple in color with oval shaped petals," into a format that the generator can use to produce corresponding images. The function starts by of course initializing an empty numpy array, 'test_embeddings', to store the GloVe embeddings of the input text. The text is processed by converting it to lowercase and removing spaces to ensure consistency. Each character in the processed text is then mapped to its corresponding GloVe embedding, and these embeddings into the 'test_embeddings' array. The embeddings are averaged by dividing by the count of valid characters, resulting in a single embedding vector representing the entire input text. To match the input requirements of the generator, this single embedding vector is repeated 28 times, creating a batch of embeddings. A noise vector of size 100 is also generated for each of the 28 embeddings which of course providing the random component which is important for the generator to produce different outputs. The function then calls 'save_result_images', which uses the generator model to create images from the combined noise and text embeddings. The resulting images reflect the model's interpretation of the input description.

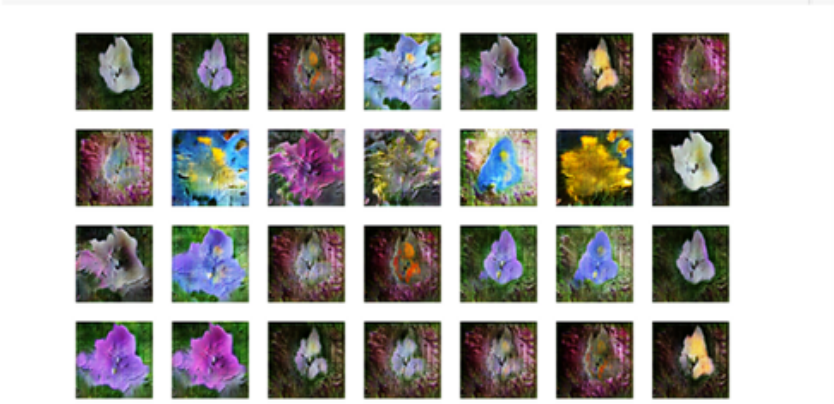

Figure 4: Purple Flower having Oval-Shaped Petals

Figure 5 displays the generated images showed and of course produced by the DCGAN model when given the textual description, "this flower is yellow in color with oval shaped petals." The function 'test_image' processes this text to create GloVe embeddings, which are then have into the generator along with random noise vectors. The resulting images reflect the model's description interpretation, showing yellow flowers with oval-shaped petals.

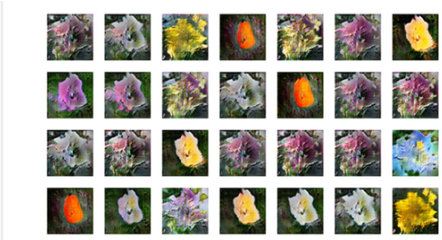

Figure 5: Yellow Flower having Oval-Shaped Petals

5.2 Text to Image Generator (GUI Result)

Figure 6 showcases the web application designed to generate videos from textual descriptions using a DCGAN-based AI model. This user-friendly interface allows users to input multiple sentences, separated by dots (.), to create a sequence of images that are compiled into a video. The central feature of this app is the text input field where users can type or paste their flower descriptions, following the prompt "enter flower caption text". Upon entering the text, users click on the "Generate Text to Video" button, which initiates the process. The backend of the web application processes the input text, converting each sentence into GloVe embeddings that the DCGAN model uses to generate images. These images are actually showing and representing various stages of the described flowers are then stitched together to create a video. This web app using Flask for the frontend and TensorFlow for the model's backend operations for of course seamlessly having advanced AI capabilities into a simple and good type of user interface.

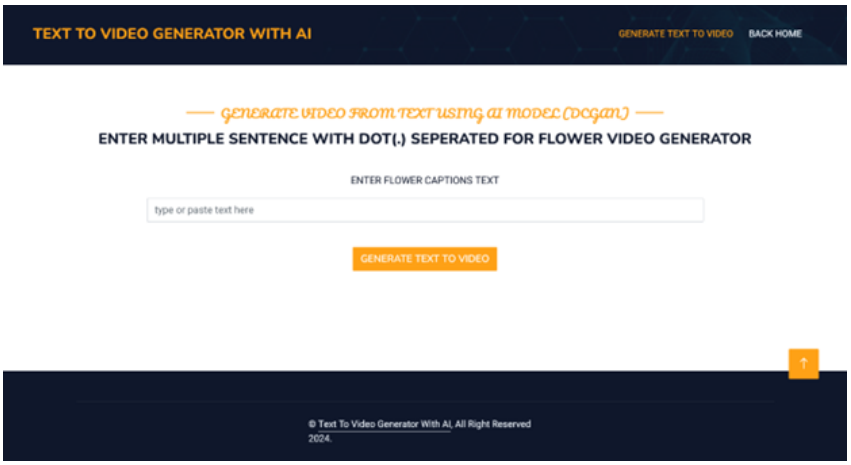

Figure 6: Web Application Interface for Text-to-Video Generation Using AI

Figure 7 shows the final output of the web application, which is a video generated from the text input describing flowers. Users enter textual descriptions of flowers into the application, which are processed by the DCGAN model to create a sequence of images. These images are then compiled into a video that visually represents the described flowers.

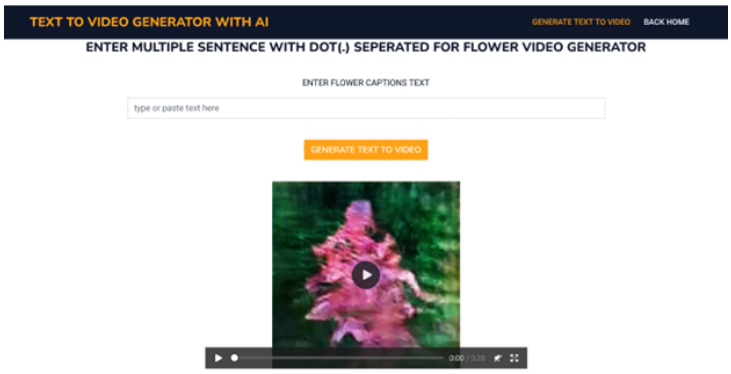

Figure 7: Generated Flower Video from Text Input

5.3 Discussion of Results

The experiment shows the possibility of creating a realistic video based on the input textual description of flowers using the DCGAN model. Nevertheless, it was able to capture the main ideas and themes of the text where for each input the model generated 28 images which can be combined to generate videos. We also noted that using 300-dimensional GloVe embeddings was essential in capturing the real semantic meaning of the text and thereby producing excellent visualization results. Fig 4 illustrates both the generator loss (1. 3285) and the discriminator loss (1. 2313) to exhibit a balance in the adversarial networks and show the ability of model to generate realistic videos in low-resolution (64 x 64 pixels) images.

The results were quite encouraging but some issues were noted that include; great amount of computational power required and video resolution restraint. Training was about 12 hours with the specified 16 GB RAM and all we could go up to was 480 x 480 pixels. However, there are some limitations to this model and in its current form, which has to be considered when evaluating the importance of this field for future application such as content generation or educational purposes. The developed web application proved that the model can be utilized practically as an example, letting people create videos from written data; it provides a valuable advancement in the area of text-to-video generation.

6 Conclusion and Future Work

6.1 Conclusions

This study successfully developed and evaluated a Deep Convolutional Generative Adversarial Network (DCGAN) for generating videos from textual descriptions of flowers. Using the 102-category flowers dataset, the model showed the ability to convert detailed textual inputs into visual type of sequences, using 300-dimensional GloVe embeddings for accurate text representation. Training on low-resolution (64x64 pixels) images was very very important to manage computational resources effectively, with the model achieving satisfactory performance metrics after 438 epochs. Key findings shows that the model can generate high-quality flower videos.

6.2 Limitations

There are so many limitations which have been identified during this study on text-to-video generation using DCGANs for floral imagery. Firstly, the use of low-resolution (64x64 pixels) images limited the visual quality of generated videos. This limitation could affect the detail of the generated videos mainly when depicting complex type of floral features. Secondly, the computational resources required for training the model were important with training times extending over 12 hours on a system with 16 GB of RAM. Scaling the model to higher resolutions or larger datasets would demand even greater computational power, which actually showing a problem to broader application in high-definition video generation.

6.3 Future Works

Moving forward, there are so many things for enhancing the text-to-video generation system for floral imagery using DCGANs can be explored. Firstly, upgrading computational resources, such as using higher RAM capacities and more powerful GPUs, could of course give training at higher resolutions (e.g., 128x128 or 256x256 pixels). This type of enhancement would definitely improve the visual quality and detail of generated videos having more complex and realistic floral representations. Also optimization of the model architecture and hyperparameters could also enhance performance and efficiency. Exploring advanced techniques in deep learning, such as attention mechanisms or transformer architectures will of course improve the model's ability to capture complex details from textual descriptions and generate more diverse and visual outputs.

References

Balaji, Y., Min, M. R., Bai, B., Chellappa, R. and Graf, H.-P. (2019). Conditional gan with discriminative filter generation for text-to-video synthesis, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 1, p. 2.

Bin, Y., Yang, Y., Shen, F., Xie, N., Shen, H. T. and Li, X. (2018). Describing video with attention-based bidirectional lstm, *IEEE Transactions on Cybernetics* **49**(7): 2631–2641.

Deng, K., Fei, T., Huang, X. and Peng, Y. (2019). Irc-gan: Introspective recurrent convolutional gan for text-to-video generation, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2216–2222.

Dong, J., Li, X. and Snoek, C. G. M. (2018). Predicting visual features from text for image and video caption retrieval, *IEEE Transactions on Multimedia* **20**(12): 3377–3388.

Gupta, S., Keshari, A. and Das, S. (2022). Rv-gan: Recurrent gan for unconditional video generation, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2024–2033.

Hindocha, E., Yazhiny, V., Arunkumar, A. and Boobalan, P. (2019). Short-text semantic similarity using glove word embedding, *International Research Journal of Engineering and Technology* **6**(4).

Islam, M. S., Mousumi, S. S. S., Abujar, S. and Hossain, S. A. (2019). Sequence-to-sequence bangla sentence generation with lstm recurrent neural networks, *Procedia Computer Science* **152**: 51–58.

Miech, A., Zhukov, D., Alayrac, J.-B., Tapaswi, M., Laptev, I. and Sivic, J. (2019). Howto100m: Learning a text-video embedding by watching hundred million narrated video clips, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2630–2640.

Namratha, M., Kumar, M., Nisha, P. and Rakshith, R. (2024). Academicvid: Academic pdfs to video generation-exploratory literature survey, *2024 International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications (ICETCS)*, IEEE, pp. 1–5.

Singgalen, Y. A. and Abdurrahman, F. (2024). Implementation of global vectors for word representation (glove) model and social network analysis through wonderland indonesia content reviews, *Jurnal Sistem Komputer dan Informatika (JSON)* **5**(3): 559–569.

Soares, E. R. and Barrére, E. (2019). An optimization model for temporal video lecture segmentation using word2vec and acoustic features, *Proceedings of the 25th Brazilian Symposium on Multimedia and the Web*, pp. 513–520.

Song, J., He, T., Gao, L., Xu, X., Hanjalic, A. and Shen, H. T. (2018). Binary generative adversarial networks for image retrieval, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

Yang, Y., Zhou, J., Ai, J., Bin, Y., Hanjalic, A., Shen, H. T. and Ji, Y. (2018). Video captioning by adversarial lstm, *IEEE Transactions on Image Processing* **27**(11): 5600–5611.

Zhou, P., Wang, L., Liu, Z., Hao, Y., Hui, P., Tarkoma, S. and Kangasharju, J. (2024). A survey on generative ai and llm for video generation, understanding, and streaming, *arXiv preprint arXiv:2404.16038* .