VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA",MACHHE, BELAGAVI-590018

ML Mini Project Report
on

Text to image generator

Submitted in partial fulfillment of the requirements for the VI semester

Bachelor of Engineering

in

Artificial Intelligence & Machine Learning

of

Visvesvaraya Technological University, Belagavi

by

Yashaswini S (1CD21AI061)
Pallavi RV (1CD21AI038)

Under the Guidance of
Dr.Varalatchoumy.M
Prof.Syed Hayath
Dept. of AI&ML

Department of Artificial Intelligence & Machine Learning
CAMBRIDGE INSTITUTE OF TECHNOLOGY,BANGALORE-560 036
2023-2024

CAMBRIDGE INSTITUTE OF TECHNOLOGY

K.R. Puram, Bangalore-560 036

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

CERTIFICATE

Certified that **Ms.Yashaswini S,** bearing USN **1CD21AI061 and Ms.Pallavi RV,** bearing USN **1CD21AI038,** a Bonafide students of **Cambridge Institute of Technology,** has successfully completed the ML mini project entitled"**Text to image generator"**in partial fulfillment of the requirements for VI semester **Bachelor of Engineering** in **Artificial Intelligence & Machine Learning** of **Visvesvaraya Technological University, Belagavi** during academic year 2023-24. It is certified that all Corrections/Suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements prescribed for the Bachelor of Engineering degree.

Mini Project Guides,

Dr.Varalatchoumy.M

Asst Prof.Syed Hayath
Dept. of AI&ML, CITech

Head of the Department,
Dr.Varalatchoumy.M
Dept. of AI&ML, CITech

DECLARATION

We Yashaswini S and **Pallavi R V** of VI semester BE, Artificial Intelligence & Machine Learning, Cambridge Institute of Technology, hereby declare that the ML Mini Project entitled **"Text to image generator"**has been carried out by us submitted in partial fulfillment of the course requirements of VI semester **Bachelor of Engineering** in **Artificial Intelligence & Machine Learning** as prescribed b**y Visvesvaraya Technological University, Belagavi**, during the academic year 2023-2024.

We also declare that, to the best of my knowledge and belief, the work reported here does not form part of any other report on the basis of which a degree or award was conferred on an earlier occasion on this by any other student.

Date:

Place: Bangalore

Yashaswini S

1CD21AI061

Pallavi R V

1CD21AI038

ACKNOWLEDGEMENT

We would like to place on record my deep sense of gratitude to **Shri. D. K. Mohan,** Chairman, Cambridge Group of Institutions, Bangalore, India for providing excellent Infrastructure and Academic Environment at CITech without which this work would not have been possible.

We are extremely thankful to **Dr. G.Indumathi,** Principal, CITech, Bangalore, for providing me the academic ambience and everlasting motivation to carry out this work and shaping our careers.

We express my sincere gratitude to **Dr. Varalatchoumy M.,** Prof. & Head, Dept. of Artificial Intelligence & Machine Learning, CITech, Bangalore, for her stimulating guidance, continuous encouragement and motivation throughout the course of present work.

We also wish to extend my thanks to Mini Project Guides, **Dr. Varalatchoumy M.,** Prof. & Head, **Prof.Syed Hayath,** Assistant Professors, Dept. of AI&ML, CITech, Bangalore for the critical, insightful comments, guidance and constructive suggestions to improve the quality of this work.

Finally to all my friends, classmates who always stood by me in difficult situations also helped me in some technical aspects and last but not the least, we wish to express deepest sense of gratitude to my parents who were a constant source of encouragement and stood by me as pillar of strength for completing this work successfully.

Yashaswini S

Pallavi R V

ABSTRACT

Artificial intelligence (AI) capable of generating images from a text prompt are becoming increasingly prevalent in society and design. The general public can use their computers and mobile devices to ask a complex text-to-image AI to create an image which is in some cases indistinguishable from that which a human could create using a computer graphics package. These images are shared on social media and have been used in the creation of art projects, documents and publications. This exploratory study aimed to identify if modern text-to-image AI (Midjourney, DALL-E 2, and Disco Diffusion) could be used to replace the designer in the concept generation stage of the design process. Teams of design students were asked to evaluate AI generated concepts from 15 to a final concept. The outcomes of this research are a first of its kind for the field of engineering design, in the identification of barriers in the use of current text-to-image AI for the purpose of engineering design. The discussion suggests how this can be overcome in the short term and what knowledge the research community needs to build to overcome these barriers in the long term.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Artificial intelligence (AI) has become something of a societal buzzword in recent years. However, research on AI in design has a long history and a well-established community including journals such as AI EDAM1 and conferences such as DCC2 . Researchers have explored a variety of potential roles that AI systems can play in the design process (Ayele & Juell-Skielse, 2021; Gero & Kannengiesser, 2014; Karimi et al., 2019), but a key focus has been on computational creativity. That is, the ability of computational systems to support creative thinking and the development of ideas (Sosa & Gero, 2016). This work has often treated AI as a kind of co-creative assistant. However, there has long been debate around the extent to which AI can be considered to be itself a 'creator', versus simply supporting a human's creative process (Cornock & Edmonds, 1973).

Text-to-image AI models are only aware of things that exist in the real world (based on the images they have been trained on) and can only generate images that are derivations of known things. The AI will also be biased towards the visual characteristics of frequently occurring everyday objects, because these are likely to appear more frequently in the training sets. Overall, therefore, it might be expected that concept development supported by text-to-image AI runs the risk of being highly derivative and lacking in novelty, particularly when it comes to developing new products with no analogue.

The text-to-image AI models discussed in this paper use a method of machine learning where an AI model is firstly trained on sets of millions of real-world image-text pairs, essentially learning associations between natural language and visual attributes. New images are generated with a technique named diffusion where random noise is used as a starting point to generate.

The difference between text-to-image AI systems is the method of training the systems, the database of images used to train the images, the method of image generation and the user interface.

AI in engineering is currently a novel concept as it comes to maturity, however there are pockets of the field that are more established in the development of solutions. At the 2022 Design Computing and Cognition conference held at the University of Strathclyde, Alymani et al., (2022) published on a nodebased learning approach to classify architectural building to ground relationships, Deprez et al., (2022) presented an approach to the generation of residential floorplans using neural networks, and Koh, (2022) on generative machine learning for architectural design to create automated layouts of stairs and rooms.

1.1 PROBLEM STATEMENT

"Generte image from Text that can be approached by using generative adversarial network model that achieves higher accuracy in classifying the different artificial intelligence generated images."

The AI-powered resume parsing system is designed to revolutionize the recruitment process by automating the screening of candidates. This system leverages advanced natural language processing (NLP) and machine learning algorithms to extract and analyze relevant information from resumes with high precision and speed. By doing so, it significantly reduces the time and cost associated with traditional manual resume screening.

Key Features:

- ➢ Automated Data Extraction**:** The system accurately extracts key details such as contact information, education, work experience, skills, and certifications from various resume formats.
- ➢ Enhanced Candidate Matching**:** Using sophisticated algorithms, the system matches candidate profiles against job descriptions, scoring and ranking candidates based on their fit for the role.
- ➢ Scalability and Efficiency**:** Capable of processing large volumes of resumes simultaneously, the system ensures scalability and operational efficiency for businesses of all sizes.
- ➢ User-Friendly Interface**:** A simple and intuitive user interface allows recruiters to easily upload resumes and job descriptions, and view ranked candidate profiles.
- ➢ Customizable andAdaptable**:** The system can be tailored to specific industry requirements and continuously learns from new data to improve its accuracy over time.

Benefits:

- ➢ TimeSavings**:** Automates the initial screening process, drastically reducing the time recruiters spend on reviewing resumes.
- ➢ Cost Reduction**:** Lowers recruitment costs by minimizing the need for extensive manual processing and reducing the time-to-hire.
- ➢ Improved Accuracy**:** Reduces human error in the screening process and ensures a more consistent evaluation of candidate qualifications.
- ➢ Better Candidate Experience**:** Allows recruiters to focus more on engaging with top candidates, improving the overall candidate experience.

By integrating AI technology into the recruitment process, this system provides a smarter, faster, and more efficient way to identify the best candidates for the job, transforming the way organizations approach talent acquisition.

1.2 OBJECTIVES

The primary objective of developing an AI-powered resume parsing system is to streamline and enhance the recruitment process by automating the initial screening of candidate resumes. This system aims to:

- ➢ Literature review: conduct a comprehensive review of current state of the art CNN models used for image generate.
- ➢ Data preprocessing:implementation and evaluate various data preprocessing techniques such as normalization,data argumentation,and noise reduction to improve data quality.
- ➢ Model architecture design:design and implementation several CNN architecture with varying depth,filter sizes,and layer configuration to determine the most effective struture image generation ,model.

CHAPTER 2

LITERATURE SURVEY

RESUME SCREENING APP USING AI

Description: AI in engineering is currently a novel concept as it comes to maturity, however there are pockets of the field that are more established in the development of solutions. At the 2022 Design Computing and Cognition conference held at the University of Strathclyde, Alymani et al., (2022) published on a nodebased learning approach to classify architectural building to ground relationships, Deprez et al., (2022) presented an approach to the generation of residential floorplans using neural networks, and Koh, (2022) on generative machine learning for architectural design to create automated layouts of stairs and rooms. The architecture research field appears to have a long history (at least since 1995) in creating demonstrations from AI theory that engineering design can learn from, with 2D and 3D concepts generated (Castro Pena et al., 2021) and an understanding of the role of the conceptual design phase in exploring solutions to requirements. Perhaps it is the nature of architecture researchers to be more accepting of AI to support their concept generation, or perhaps it is the nature of that which they are generating. Yet it seems conceivable that if an AI can generate configurations of building layouts, then an AI can be built that can generate configurations of components of a product e.g. a chair with three main components, legs, a seat.

AI in future projects, indicating that current AI systems are perhaps without consideration if they are appropriate or not to be used in education (Beetham & Sharpe, 2019) or in practice. Maher & Fisher, (2012) discuss an AI approach for judging creative design. The approach focused on measures of novelty, value, and surprise and was evaluated using student laptop computer concepts against those made by Apple. The method of evaluation appears appropriate in terms of a machine learning approach and the decisions made.

Develop automated scripts to scrape job postings and resumes from job portals and company websites. Ensure adherence to the terms of service of these websites. Utilize APIs provided by job portals to programmatically access structured data. Use platforms like Amazon Mechanical Turk to collect resumes and job descriptions from a diverse group of individuals.Employ a team to manually enter specific high-quality data to ensure accuracy.

CHAPTER 3

METHODOLOGY

3.1DATA COLLECTION

Image-processing techniques and applications of computer vision (CV) have grown immensely in recent years from advances made possible by artificial intelligence and deep learning's success.

Identifying Data Sources

- ➢ Job Portals: Scrape job postings and descriptions from popular job portals such as LinkedIn, Indeed, Glassdoor, and Monster.
- ➢ Company Websites: Collect job descriptions directly from the career pages of various companies.
- ➢ Datasets Public: Utilize publicly available datasets such as the Kaggle resume datasets or the Open Resume Project.
- ➢ HR Firms and Agencies: Collaborate with recruitment agencies and HR firms to access anonymized resumes and job descriptions.
- ➢ Surveys and Interviews: Conduct surveys and interviews with job seekers and employers to gather detailed job matching criteria and real-world examples.

Data Collection Techniques

- ➢ Web Scraping: Develop automated scripts to scrape job postings and resumes from job portals and company websites. Ensure adherence to the terms of service of these websites.
- ➢ APIs: Utilize APIs provided by job portals to programmatically access structured data.
- ➢ Crowdsourcing: Use platforms like Amazon Mechanical Turk to collect resumes and job descriptions from a diverse group of individuals.
- ➢ Manual Entry: Employ a team to manually enter specific high-quality data to ensure accuracy.

Data Volume and Variety

- ➢ Volume: Aim to collect a large volume of data to ensure the model can generalize well across different industries and job roles.
- ➢ Variety: Ensure the data represents various job roles, industries, experience levels, and geographical locations to minimize bias and improve the model's robustness.

Ethical Considerations

- ➢ Privacy: Anonymize all personal information and ensure compliance with data protection regulations (e.g., GDPR, CCPA).

- ➢ Diversity and Inclusion: Actively seek data from diverse sources to ensure the model does not perpetuate existing biases.

3.2 DATA PREPROCESSING

Once data is collected, it must be preprocessed to prepare it for model training. Data preprocessing involves cleaning, normalizing, and transforming the raw data into a format suitable for machine learning algorithms.

Data Cleaning

- ➢ Remove Duplicates**:** Identify and remove duplicate entries to ensure data integrity.
- ➢ Correct Errors**:** Fix any errors or inconsistencies in the data, such as misspelled words or incorrect formatting.
- ➢ Standardize Formats**:** Convert all resumes and job descriptions to a consistent format (e.g., plain text) to facilitate easier processing.

Data Annotation

- ➢ Label Key Information**:** Annotate important elements in resumes and job descriptions, such as skills, experience, education, and job requirements.
- ➢ Human-in-the-Loop**:** Involve HR professionals to manually annotate a subset of the data, ensuring high- quality labels.

Normalization

- ➢ Standardize Terminology**:** Convert different terms referring to the same concept (e.g., "software engineer" and "developer") into a standardized format.
- ➢ Numeric Conversion**:** Convert categorical data into numerical values where applicable (e.g., years of experience).

Tokenization

- ➢ Text Tokenization**:** Break down text into smaller units (tokens), such as words or phrases, for easier processing by machine learning models.
- ➢ Handling Stop Words**:** Remove common stop words (e.g., "and", "the") that do not contribute significant meaning to the data.

3.3MODEL TRAINING

It has 102 classes; each class consists of 40 to 60 images, and each of the images have 10 matching textual descriptions. In this study, we considered 8000 images for training. This dataset was used to train the model for 300 epochs.

Model Selection

- ➢ Algorithm Choice**:** Choose suitable machine learning algorithms for resume parsing and job matching, such as decision trees, random forests, support vector machines, or neural networks.
- ➢ Pretrained Models: Leverage pretrained language models from OpenAI for their ability to understand and generate human-like text.

Training Process

- ➢ Training-Validation Split**:** Split the dataset into training and validation sets to evaluate model performance during training.
- ➢ Hyperparameter Tuning**:** Optimize hyperparameters using techniques like grid search or random search to improve model performance.
- ➢ Cross**-**Validation**:** Implement cross-validation to ensure the model generalizes well to unseen data.

Model Evaluation

- ➢ Performance Metrics: Use metrics like accuracy, precision, recall, F1-score, and ROC-AUC to evaluate the model's performance.
- ➢ Benchmarking**:** Compare the model's performance against baseline models or existing solutions to assess its effectiveness.

3.4 FEATURE EXTRACTION

Feature extraction involves identifying and extracting meaningful features from the data that can be used to train the machine learning model effectively.

Textual Features

- ➢ N-grams: Extract n-grams (unigrams, bigrams, trigrams) to capture the context of words in resumes and job descriptions.

- TF-IDF: Use TF-IDF to measure the importance of words in a document relative to the entire dataset.
- Word Embeddings: Utilize word embeddings like Word2Vec, GloVe, or BERT to represent words as dense vectors in a continuous space.

Domain-Specific Features

- Skills and Certifications: Extract specific skills and certifications relevant to different job roles.
- Experience: Identify and quantify the years and types of experience mentioned in resumes.

Contextual Features

- Sentiment Analysis: Analyze the sentiment of the text to infer the candidate's tone and attitude.
- Entity Recognition: Use named entity recognition (NER) to identify entities like company names, job titles, and locations.

Feature Selection

- Filter Methods: Use statistical tests to select features with the highest correlation to the target variable.
- Wrapper Methods: Employ techniques like recursive feature elimination (RFE) to iteratively select the most important features.
- Embedded Methods: Integrate feature selection within the model training process using algorithms like LASSO (Least Absolute Shrinkage and Selection Operator).
- Combining Models: Use ensemble methods like bagging, boosting, or stacking to combine multiple models and improve overall performance.

3.5 SYSTEM ARCHITECTURE

This section describes the training details of deep learning-based generative models. Conditional GANs were used with recurrent neural networks (RNNs) and convolutional neural networks (CNNs) for generating meaningful images from a textual description. The dataset used consisted of images of flowers and their relevant textual descriptions. For generating plausible images from text using a GAN, preprocessing of textual data and image resizing was performed. We took textual descriptions from the dataset, preprocessed these caption sentences, and created a list of their vocabulary. Then, these captions were stored with their respective ids in the list. The images were loaded and resized to a fixed dimension. These data were then given as input to our proposed model. RNN was used for capturing the contextual information of text sequences by defining the relationship between words at altered time stamps. Text-to-image mapping was performed using an RNN and a CNN. The CNN recognized useful characteristics from the images without the need for human intervention. An input sequence was given to the RNN, which converted the text to image.

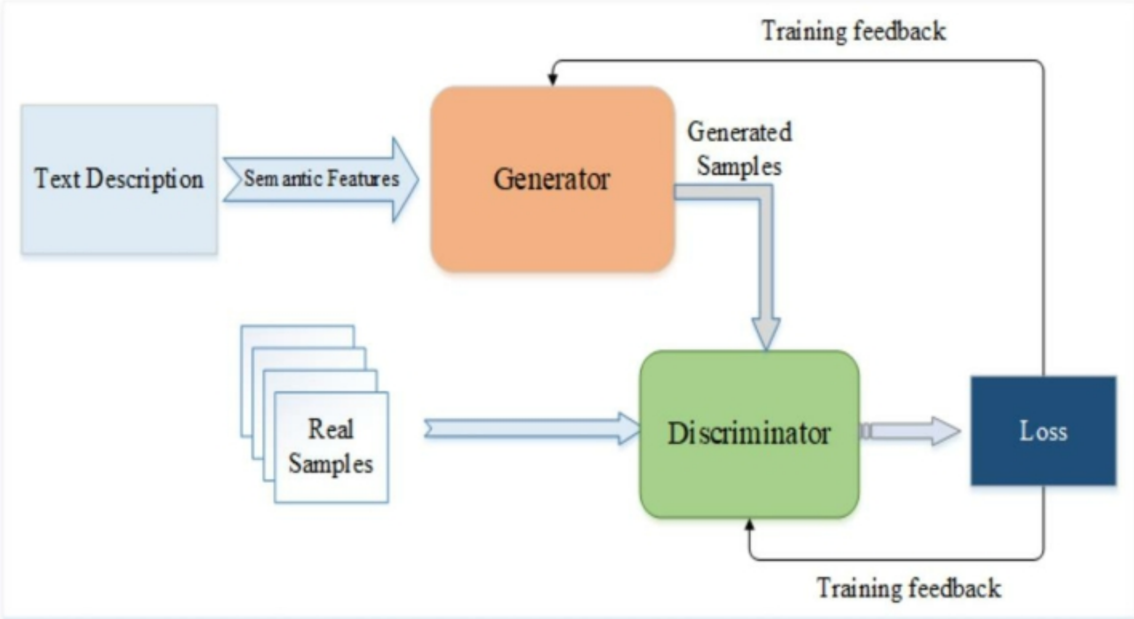

Figure1:generate image from text description

3.6 TOOLS AND TECHNOLOGIES

The motivation of this research was to understand the role that text-to-image AI could play in conceptual engineering design. All teams who took part in the research activity were successful in evaluating the 15 concepts to a final concept. However, there were complexities caused by the use of AI images that have the potential to be alleviated through change in design process and/or change in future text-to-image AI systems.

- ➢ PyMuPDF:PyMuPDF is a Python binding for MuPDF, a lightweight PDF and XPS viewer. It allows for fast and accurate text extraction from PDF documents, including those with complex layouts and formats.
- ➢ PDFMiner:PDFMiner is a tool for extracting information from PDF documents. Unlike other PDF- related tools, it focuses entirely on getting and analyzing text data. It is capable of extracting text along with its position, font, and other information.
- ➢ Apache Tika: Apache Tika is a toolkit for detecting and extracting metadata and structured text content from various documents, including PDFs. It supports a wide range of document types and provides a unified API for document processing.

- ➢ spaCy:spaCy is an open-source software library for advanced NLP in Python. It provides pre-trained models for various NLP tasks such as tokenization, part-of-speech tagging, named entity recognition (NER), and dependency parsing. spaCy is known for its speed and accuracy, making it suitable for processing large volumes of text data.
- ➢ NLTK (Natural Language Toolkit): NLTK is a platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.
- ➢ Transformers from Hugging Face: The transformers library from Hugging Face provides state-of-the- art pre-trained models for various NLP tasks, including NER. These models are based on transformer architectures like BERT, GPT-3, and GPT-4, which offer high accuracy and contextual understanding.40% of students who took part in the study were able to identify that the initial concepts were created using AI.

Machine Learning and Deep Learning

Machine Learning Frameworks:

- ➢ Scikit-learn: Scikit-learn is a Python library for machine learning. It provides simple and efficient tools for data mining and data analysis, including classification, regression, clustering, and dimensionality reduction algorithms. It is built on NumPy, SciPy, and Matplotlib.
- ➢ TensorFlow: TensorFlow is an open-source machine learning framework developed by Google. It is widely used for building and training deep learning models. TensorFlow offers high flexibility and scalability, making it suitable for both research and production environments.
- ➢ PyTorch:PyTorch is an open-source machine learning library developed by Facebook's AI Research lab. It provides a flexible and dynamic computational graph, which is particularly useful for research and experimentation. PyTorch is widely adopted for developing deep learning models due to its ease of use and integration with Python.
- ➢ BERT (Bidirectional Encoder Representations from Transformers): BERT is a transformer-based model developed by Google. It pre-trains deep bidirectional representations by jointly conditioning on both left and right context in all layers. BERT has set new benchmarks in various NLP tasks, including NER and text classification.
- ➢ GPT-3 and GPT-4 (Generative Pre-trained Transformer): GPT-3 and GPT-4 are autoregressive language models developed by OpenAI. These models use deep learning to produce human-like text

based on the input they receive. GPT-3 and GPT-4 have shown impressive performance in various NLP tasks, including text generation, summarization, and entity extraction.

Data Storage and Management

Databases:

- ➢ PostgreSQL: PostgreSQL is an open-source relational database management system. It is known for its robustness, scalability, and support for advanced data types and indexing techniques. PostgreSQL is used to store structured data extracted from resumes, including candidate details and job descriptions.
- ➢ MongoDB: MongoDB is a NoSQL database known for its flexibility and scalability. It stores data in JSON-like documents, making it suitable for handling semi-structured and unstructured data. MongoDB is often used for storing and retrieving large volumes of text data.

Data Processing Libraries:

- ➢ Pandas: Pandas is a data manipulation and analysis library for Python. It provides data structures and functions needed to manipulate structured data seamlessly. Pandas is widely used for data cleaning, transformation, and analysis in machine learning pipelines.
- ➢ NumPy: NumPy is a library for numerical operations on large, multi-dimensional arrays and matrices. It provides a high-performance, flexible array interface and is the foundation for many scientific computing libraries in Python.

Hardware Requirement

- ➢ Processor:A high-performance multi-core processor (e.g., Intel Core i7/i9 or AMD Ryzen 7/9) is recommended to handle the computational load during model training and inference.
- ➢ RAM:At least 16 GB of RAM is necessary for smooth operation. For larger datasets and more complex models, 32 GB or more is recommended.
- ➢ Storage:A solid-state drive (SSD) with at least 500 GB of storage is recommended for faster data access and processing. An HDD can be used for additional storage if needed.
- ➢ Graphics Processing Unit (GPU):A dedicated GPUis highly recommended for efficient model

- Internet Connection:A stable and fast internet connection is necessary for downloading large datasets, accessing cloud services, and integrating with other online recruitment systems.Participants did identify that a pre-defined specification for evaluation would have supported the activity.

Software Requirement

- Operating System: JobFitAI can be developed and run on major operating systems such as Windows, macOS, or Linux. Linux distributions (e.g., Ubuntu) are often preferred for their compatibility with development tools and packages.
- Programming Languages: Python is the primary programming language used for developing JobFitAI due to its extensive libraries and frameworks for machine learning and natural language processing.
- Development Environment:An Integrated Development Environment (IDE) such as PyCharm, VSCode, or Jupyter Notebook is recommended for coding and debugging.
- Machine Learning Libraries:
- TensorFlow: An open-source platform for machine learning used for building and training models.
- PyTorch: Another popular machine learning framework that offers dynamic computational graphs and is widely used in research and industry.
- Natural Language Processing (NLP) Libraries:spaCy: An open-source software library for advanced NLP in Python.
- OpenAI API:Access to OpenAI's GPT-4 API for leveraging advanced NLP capabilities in JobFitAI. An API key from OpenAI is required to use this service.
- Web Frameworks (if building a web interface):Flask or Django: Python web frameworks for developing the front-end interface and integrating the AI model with the web application.
- Database:A relational database (e.g., PostgreSQL, MySQL) or a NoSQL database (e.g., MongoDB) to store resumes, extracted features, and job descriptions.

CHAPTER 4

IMPLEMENTATION

4.1 DETAILED DESCRIPTION OF IMPLEMENTATION

Input:

- ➢ File Type: The system accepts resumes in PDF format. This is indicated by the label "File: PDF."
- ➢ Function**:** This is the entry point of the system where the user uploads a CV for processing. The system is designed to handle standard PDF files, ensuring compatibility with most resumes submitted by candidates.

Text Extraction

- ➢ Description: The primary function of this stage is to extract raw text from the uploaded PDF CV.
- ➢ Process: The text extraction process involves reading the contents of the PDF file and converting it into a format that can be further processed. This stage deals with:
- ➢ Text Elements: Words, sentences, and paragraphs are extracted from the PDF.
- ➢ Tools and Techniques**:** Common tools used for this purpose include PDF parsing libraries like PyMuPDF, PDFMiner, or Tika, which can accurately extract text from PDFs, including those with complex formatting.

Text Segmentation

- ➢ Description: This stage involves breaking down the extracted text into logical sections or blocks.
- ➢ Process: Text segmentation is critical for identifying and organizing different sections of the CV. It typically includes:
- ➢ Text Blocks: Segments such as personal information, education, experiences, and other relevant sections are identified and isolated.
- ➢ Algorithms**:** Machine learning algorithms and NLP techniques, such as tokenization and sentence segmentation, are used to accurately divide the text into meaningful blocks.

Information Extraction

- ➢ Description: This is the core stage where specific entities and details are extracted from the segmented text.

- ➢ Process: Information extraction involves identifying and retrieving key pieces of information from each text block. This stage focuses on:
- ➢ Entities: Extracting specific details such as the candidate's name, email, phone number, company names, college names, job titles, and other pertinent information.
- ➢ Techniques**:** Named Entity Recognition (NER) models, often built using libraries like spaCy, NLTK, or transformers from Hugging Face, are employed to detect and extract these entities accurately.

Output:

- ➢ Description: The final stage involves presenting the extracted information in a structured format.
- ➢ Function: The system compiles the extracted entities and details into a structured output that can be easily reviewed and utilized for further processing. This output format can include:
- ➢ Structured Data: Information is organized into a database or a standardized format such as JSON or CSV, making it easy to integrate with other recruitment systems or databases.
- ➢ User Interface: The extracted information can be displayed on a user interface for recruiters to review or directly fed into an Applicant Tracking System (ATS) for automated processing.

4.2 CODE SNIPPET

```
import torch
print(torch.version.cuda)
rom pathlib import Path
import tqdm
import torch
import pandas as pd
import numpy as np
from diffusers import StableDiffusionPipeline
from transformers import pipeline, set_seed
importmatplotlib.pyplotas plt
importmatplotlib.pyplotas plt
import cv2
import gradio as gr
rom pathlib import Path
import tqdm
import torch
import pandas as pd
import numpy as np
from diffusers import StableDiffusionPipeline
from transformers import pipeline, set_seed
importmatplotlib.pyplotas plt
importmatplotlib.pyplotas plt
import cv2
The cache for model files in Transformers v4.22.0 has been updated. Migrating your old
cache. This is a one-time only operation. You can interrupt this and resume the
migration later on by calling `transformers.utils.move_cache()`.
 0/0 [00:00<?, ?it/s]
import gradio as gr
import torch
```

```
from transformers import pipeline
from diffusers import StableDiffusionPipeline
from PIL import Image

class CFG:
    device = "cuda"
    seed = 42
    generator = torch.Generator(device).manual_seed(seed)
    image_gen_steps = 35
    image_gen_model_id = "stabilityai/stable-diffusion-2"
    image_gen_size = (400, 400)
    image_gen_guidance_scale = 9
    prompt_gen_model_id = "gpt2"
    prompt_dataset_size = 6
    prompt_max_length = 12

image_gen_model =
    StableDiffusionPipeline.from_pretrained( CFG.image_gen_model_id,
    torch_dtype=torch.float16, revision="fp16",
    use_auth_token='your_hugging_face_auth_token'
)
image_gen_model = image_gen_model.to(CFG.device)

efgenerate_image(prompt):
    image = image_gen_model(
        prompt, num_inference_steps=CFG.image_gen_steps,
        generator=CFG.generator,
        guidance_scale=CFG.image_gen_guidance_scale
    ).images[0]

    image = image.resize(CFG.image_gen_size)
    return image

defgradio_generate_image(prompt):
    image = generate_image(prompt)
    return image

interface = gr.Interface(
    fn=gradio_generate_image,
    inputs=gr.Textbox(lines=2, placeholder="Enter your prompt here..."),
    outputs=gr.Image(type="pil"), title="Stable
    Diffusion Image Generator",
    description="Generate images from text prompts using Stable Diffusion."
)
```

CHAPTER 5

RESULT

In this section, the experimental analysis and generated images of flowers are presented. Thetraining of the proposed model was performed on an Nvidia 1070 Ti GPU, 32 Gb memoryand windows 10 operating system. The weights of the generator and discriminator were optimized using an Adam optimizer, the mini-batch size was 64, andthe learning rate was 0.0003. The ground truths from the dataset and the images generated from the input textual descriptions are shown in Figure 2. For evaluating the performance of the proposed model, the inception score (IS) and PSNR values are calculated. Inception scores capture the diversity and quality of the generated images. PSNR is used for calculating the peak signal-to-noise ratio in decibels among two images. The quality of the original and produced images is compared using this ratio. The PSNR value increases as the quality of the created or synthesize.

.

Figure 2: images extracted from text

These different AI methods can help to solve different challenges in the product design process. By identifying papers in the research field, the authors report the most common combinations of these school of thought are genetic algorithm and neural networks, and genetic algorithm and fuzzy logic. This identifies that there are other combinations of AI training and executing to explore which may bring benefits for engineering design specifically. In the future, there will certainly be examples of AI engines in product/industrial design that are successful in generating highly specific concepts within a context. However, at this time it appears there is a need to manage this process and for human intuition to play a part in evaluation and making design decisions. AI may certainly support designers, but it is not yet foreseen if they can replace the creative process of designing a birds desirable form text.

In the same way that those creating AI define specific criteria tailored to the needs of the systems development, engineering designers must consider this to support the development of text-to-image AI to best support engineering design activities. The survey questions were intended to explore theories about the ways in which designers wish to interact with AI during the design process i.e., how the process of designing changes when each of the design team do not have ownership of the concepts; and which aspects of the design process do the designers want to have control over. Designers have a high influence over the design process towards design development and AI has the potential to reduce that level of control. An analogy to the inclusion of AI in the design process can come from research in collaboration. It is understood that designers experience anxiety about lack of control which is a major factor in the design process. Therefore, there is a need to consider the level of control that is appropriate when implementing AI as a 'design team member'. Perhaps this means greater control over the inputs of the text-to-image AI system or processes.

In this initial exploratory study, a simple prompt of chair was selected. This text prompt was selected with the reasonable assumption that the images used to train the AI would include many variations as it is an abundant product in the world and the images would be suitably labelled. However, in the case of disco diffusion the concepts are highly similar.

CONCLUSION

This research investigated the application of text-to-image AI within an engineering design context. Examples of designers using text-to-image AI have begun to appear in the public discourse without consideration if the outcomes are suitable and appropriate. This exploratory study aimed to answer two questions on the extent to which publicly available text-to-image AI systems can be used to generate images of concepts for concept selection in engineering design, and what are designers' perceptions of this. A literature review revealed a lack of practical examples of the development of text-to-image AI in field such as product design and industrial design where other design field such as Architecture have established fields of study. There is much knowledge that can be transferred and practices to consider adopting. However, there has been long established research in areas such as machine learning, human cognition and AI that can help to bridge the gap between these cross-discipline research efforts. The exploratory study revealed that student teams demonstrated a preference towards criteria-based evaluation techniques with this experimental setup that may be an appropriate technique for future research studies. If chosen, there are some recommendations from the participants of this study including pre-define the specification and use this in image generation and pre-selection. However, there was also an identification of improvements for the development of the text-to-image AI systems that may be used to generate these images including: clearer image (meaning defined form), improved realism and 3D. And finally, there were some suggestions to the process including: the use of the images as aesthetic inspiration, fewer concepts and comparison with a datum.

FUTURE ENHANCEMENT

In the fields of computer vision and natural language processing, text-to-image generation is a hot topic these days. For producing visually realistic and semantically consistent images, we presented a deep learning-based model (RC-GAN) and described how it works in the confluence of computer vision and natural language processing. The model was trained by text encoding and image decoding. Extensive experiments on the Oxford-102 flowers dataset demonstrated that the proposed GAN model generates better-quality images compared with existing models by providing the best recorded IS. The performance of our proposed method was compared with that of state-of-the-art methods using IS. Our model achieved an IS of 4.15 and a PSNR value of 30.12 dB on the Oxford-102 flowers dataset. We want to expand this work by training the model on a variety of datasets for image generation. Author Contributions: The authors confirm contribution to the paper as follows: study conception, design and data collection: S.R.; analysis and interpretation of results: S.R., M.M.I., T.K.; draft manuscript preparation: S.R., M.M.I. All authors have read and agreed to the published version of the manuscript. Funding: The Authors do not received funding for this research. Institutional Review Board Statement: Not applicable. Informed Consent Statement: Not applicable. Data Availability Statement: Dataset is publicly available. Conflicts of Interest: The authors are anonymously declared that they have no conflict of interest.

Integration with HR Systems and Workflows

- Seamless Integration: Deepening integration with existing HR systems and applicant tracking systems (ATS) to streamline data flow and improve efficiency.
- Automated Workflow Integration: Automating various stages of the recruitment process, such as scheduling interviews, sending offer letters, and onboarding new hires.

REFERENCES

1.Kosslyn, S.M.; Ganis, G.; Thompson, W.L. Neural foundations of imagery. Nat. Rev. Neurosci. 2001, 2, 635–642. [CrossRef] [PubMed].

2. Karpathy, A.; Fei-Fei, L. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3128–3137.

3. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3156–3164.

4. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2048–2057.

5. Albawi, S.; Mohammed, T.A.; Al-Zawi, S. Understanding of a convolutional neural network. In Proceedings of the 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017; pp. 1–6.

6. Kim, P. Convolutional neural network. In MATLAB Deep Learning; Springer: Berlin/Heidelberg, Germany, 2017; pp. 121–147.

7. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. arXiv 2014, arXiv:1406.2661. [CrossRef].