**ANKARA UNIVERSITY**

**FACULTY OF ENGINEERING**

**DEPARTMENT OF COMPUTER ENGINEERING**

****

**COM 4062 PROJECT REPORT**

**Text to Image for Children Books with Stable Diffusion and LoRA**

**Mehmet Bayram Alpay**

**20290310**

**Doç. Dr. Gazi Erkan Bostancı**

**May 2024**

# ABSTRACT

The purpose of this graduation project report is to gain experience in the latest technologies used to produce images and to create a beautiful application for people interested in children's books to produce their own images. The report aims to provide an overview of natural language processing and deep learning techniques such as stable diffusion and chatgpt and use them to develop a text to image project that anyone can use. The main purpose of this report is to increase our knowledge and create a user-friendly application by using the latest technologies in deep learning and NLP. The following tools were used in the project: Google Colab, Jupyter Notebook, HuggingFace, LoRA models, ChatGPT, Gradio Interface, as well as the following libraries: Torch, Diffusers, Safetensors, os, Gradio. The report includes informative explanations about stable diffusion and LoRA models. The report also includes detailed explanations of terms that are often used in the wrong context and create confusion in people's minds. Within the scope of the project, advanced applications developed in the field of visual creation were analyzed, the interface and project scope process were detailed, and a project example was created using natural language processing techniques. As a result of this project, with the help of my consultant, we took the approach to this technology, which has high hardware and usage costs, to the next level and developed an application that can be adapted to languages ​​other than English. We now have an application that can compete with other studies in this field or direct future studies. This project also helped me discover the potential of the technologies used, and I began to frequently include these technologies in my other works. I was encouraged to implement new projects and my self-confidence increased. At the end of the project, all content, articles, codes, and reports related to the subject have been filed by us and can be shared publicly in the future.

# TABLE OF CONTENTS

[ABSTRACT ii](#_Toc39681004)

[TABLE OF CONTENTS iii](#_Toc39681005)

[1. INTRODUCTION 1](#_Toc39681006)

[1.1. **Background and Subjects Declarations** 1](#_Toc39681007)

[2. **Visual Content Generation using Stable Diffusion and Gradio InterFace** 2](#_Toc39681009)

[2.1. **Definitions and Declarations about Visual Content Generation** 2](#_Toc39681010)

[2.2. **Tools and Platforms for Visual Content Generation** 3](#_Toc39681011)

[2.2.1. **Data Collection Tools**  3](#_Toc39681012)

[2.2.2. **Model Search Sites**  3](#_Toc39681012)

[2.2.3. **Visualization Tools for Models and Interface**  3](#_Toc39681012)

[2.2.4. **Visual Content Generation, NLP Community and Resources**  4](#_Toc39681012)

[2.3. **How Are Visual Generator Applications Transforming Our Lives?** 4](#_Toc39681011)

[2.3.1. **How Visual Generators are Conducted?**  5](#_Toc39681012)

[2.4. **OpenAI, Prompt Design, Scheduler, LoRA** 5](#_Toc39681011)

[2.5. **StableDiffusionPipeline, Gradio Interface and XL Models** 7](#_Toc39681011)

[2.6. **Hands-on calling Libraries, Functions and Creating Interface for Our Project** 8](#_Toc39681011)

[3. **CONCLUSION** 14](#_Toc39681013)

[**Bibliography 19**](#__RefHeading___Toc149033957)

# INTRODUCTION

**1. 1. Background and Subjects Declarations**

Throughout this project, we will learn the underlying logic of models that create images, what to consider when creating a text to image project, how to create the code structure and compatibility between models, how to solve optimization problems and how to apply various developed applications. I had the opportunity to research and learn. This opportunity allowed me to increase my knowledge of deep learning and architectures and learn new technologies. In short, the project includes the processes of providing the code structure, guiding the user, and creating meaningful visual outputs after the given prompt, which makes the purpose of the project important.

• **Stable Diffusion** is a deep learning-based visual synthesis technique. It aims to create realistic and high-quality visuals by using complex relationships learned from large-scale data sets. This technique is particularly notable for its photorealism and richness of detail.

NLP developers and staff use programming languages, NLP methods, computer science and current tools to understand and analyze current technologies and produce logical output accordingly. It combines a variety of skills including: Every stage, from conducting current research and development to making meaningful inferences and applying the methods, can sometimes become so difficult and inextricable that this problem is tried to be solved by dividing it into as few parts as possible. Thus, such projects become one of the most important tools that have the potential to use and transform information to find solutions to people's problems, transfer the right information as simply as possible and improve the user experience. For example,

• Technologies that we can transfer as **Natural Language Processing (NLP)** and **Large Language Models (LLM)** applications such as ChatGPT, LoRA models and Stable Diffusion are very important for visual production and prompt creation. ChatGPT helps generate the desired prompt using a ready-made template, Stable Diffusion helps us get models based on the generated image, and LoRa provides a subject-specific mechanism. The diffusers library, which is also offered with the contribution of HuggingFace, makes a great contribution to the reduction and dissemination of this technology into the Python language.

• **Google Colab, HuggingFace and Python** offer a financial-independent opportunity to access and use technology. Especially when creating such a beautiful interface, Colab provides CUDA support to the user, and HuggingFace brings stable diffusion to us. Seamlessly integrated with Python, both models demonstrate their impact in the field of machine learning applications by offering the interface's ability to produce visuals specific to children's books.

**2. Visual Content Generation using Stable Diffusion and Gradio InterFace**

**2.1. Definitions and Declarations about Visual Content Generation**

Visual producing tools gained importance with examples such as Midjourney and DALL-E and gained their independence with stable diffusion. We need visuals in many fields such as art and design, education, advertising, game development, health, and engineering, and it is of great importance that we analyze the tools that produce visual content well and understand what value they can add to our lives in the future. Many visual quality enhancing, or visual matching applications have been made in the past, but none of them were as effective as the technologies we use now. Considering today's data environment, open-source data and individual interests, the number of sources continues to increase day by day. We can create our own LoRA models even using low-quality photos we take in daily life. For this very reason, this technology has entered our lives as one of the important breakthroughs that has become widespread and used for individual use and continues to fascinate humanity every day. This technology, classified as an NLP application, provides the service of understanding the transmitted text and producing the appropriate visual by giving certain weights to the words. For example, when we give the keywords in the children's book as a whole to the model, the model has no difficulty in producing photographs that capture the context. Today, it is a source of inspiration and a key technology for many applications, from art galleries to artists' works, to artificial intelligence concerts. For this very reason, it has become a fashion to create visual-producing applications with easily understandable interfaces for the purpose of increasing our perception and personal experience. LoRA models are frequently preferred to focus on visuals specific to each field, and this technology, which can also be used commercially, can consequently be adapted to new works. The application we made is a beautiful project created for books and offers a detailed interface.

## 2.2. Tools and Platforms for Visual Content Generation

**2.2.1. Data Collection Tools**

Görsel üretme projeleri için görsel ve text verileri gerekebilir. LoRA için ise sadece görsel veriler ile de eğitim yapılabilir ama biz projemiz için sadece hazır modelleri kullanacağız. Veri setinin büyük olması ise faydalı olacaktır fakat en az 10-15 arası görsel bile model eğitmek için yeterli olabilir.

• ImageNet, COCO OpenImages

• Hugging Face Hub, TensorFlow Hub

**2.2.2. Model Search Sites**

Models like LoRA and Stable Diffusion represent significant and innovative advancements in the field of artificial intelligence. These models utilize deep learning techniques to generate visual content based on text-based inputs. Platforms hosting these technologies, such as Civitai, provide users with the ability to discover, access, and utilize various artificial intelligence models. These sites offer a range of advantages, including access to pre-trained models, community engagement, developer support, and the sharing of innovative projects. Specifically, platforms featuring models like LoRA and Stable Diffusion serve as valuable resources for artificial intelligence researchers and practitioners.

• Civitai, Hugging Face Model Hub, Lexica.art

**2.2.3. Visualization Tools for Models and Interface**

Visualization tools play a crucial role in understanding and interpreting the outputs of models like LoRA and Stable Diffusion. These tools provide users with intuitive interfaces to interact with the models and explore their results. Additionally, they enable the visualization of complex data structures and model outputs, making it easier for researchers and practitioners to gain insights from the models. There are several libraries and frameworks available for building visualization tools for models and interfaces.

• Mediapy for Images

• Streamlit, Gradio for Interface

**2.2.4. Visual Content Generation, NLP Community and Resources**

Visual content generation and Natural Language Processing (NLP) development rely on a diverse array of tools, each offering distinctive features and capabilities. In this ecosystem, platforms like Hugging Face and OpenAI stand as pillars of support for developers and researchers alike. Hugging Face, renowned for its Model Hub, serves as a central hub for the NLP community, providing access to a vast array of pre-trained models, collaborative spaces, and resources. Similarly, OpenAI offers cutting-edge NLP models and resources, empowering developers to create innovative solutions for natural language understanding and generation tasks. These resources form the foundation of the visual content generation and NLP landscape, equipping developers with the necessary tools and information to drive innovation in text-based contact experiences and content creation.

## 2.3. How Are Visual Generator Applications Transforming Our Lives?

Visual generator applications, powered by innovative technologies such as Stable Diffusion and LoRA, are revolutionizing our interaction with visual content, using in a new era of creativity and accessibility across various domains. These advanced tools enable users to generate high-quality visual content seamlessly, transcending traditional boundaries and unlocking new possibilities in art, education, and beyond.

In the field of art, visual creator applications contribute. This process allows artists of all levels to unleash their creativity without restriction. With tools like Stable Diffusion and LoRA, artists can explore new artistic styles, experiment with a variety of visual elements, and push the boundaries of art.

Beyond the art field, visual creator applications are reshaping the educational landscape by providing immersive and interactive learning experiences. Educators can leverage these tools to create engaging visual materials, personalized learning resources, and visuals that appeal to their preferences. Additionally, visual creator apps increase accessibility to education by eliminating language barriers and providing visual representations of complex concepts, thus promoting inclusivity and equity in education. We already aim to increase visual interaction in education by making applications on behalf of children's books.

In general, technologies such as Stable Diffusion and LoRA, supported by visual creator applications, transform our lives by improving learning experiences and revolutionizing education life. As these technologies continue to evolve, they are empowering individuals to drive innovation across a variety of industries and unleash their imaginations.

**2.3.1 How Visual Generators are Conducted?**

Developing visual generators requires a systematic and iterative process. The design is created with the target audience in mind. During the design phase, importance is given to user interfaces that facilitate the creation of visual content. The application involves integrating advanced technologies such as Stable Diffusion and LoRA to enable the creation of high-quality images based on text inputs. Training the visual generator involves exposing it to various datasets and fine-tuning its algorithms to produce images. It tries to produce visuals that will satisfy the user, depending on the given prompt and context. These images can then be used for any purpose the user wishes.

**2.4. OpenAI, Prompt Design, Scheduler, LoRA**

Natural Language Processing (NLP) is a field of artificial intelligence that revolves around the interaction between computers and human language. NLP plays a crucial role in enabling machines to understand and process human language. LLM models are an important breakthrough that are trained with large data sets and can perform multiple tasks. We use the ChatGPT technology developed by OpenAI for the ready-made templates it provides us, because ChatGPT provides support for multiple languages ​​and allows us to produce the prompt we want for our Stable Diffusion project.

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 2.1. Prompt Design Example for OpenAI

Some of the reasons why using ready-made templates and using ChatGPT are important are as follows:

Other open-source language models do not have a wide template capability like ChatGPT, and with this capability, they contribute to many languages. It provides us with the opportunity to create a template specific to each model and to create a special prompt for this template based on the book scenarios we provide and leads us to develop an application that is interactive with the user. The prompt it receives gives a special output with a certain randomness and then applies it to its own formula.

Schedulers are critical components for managing and optimizing the working processes of deep learning models during inference. These components make various adjustments during inference to improve the performance of the model, reduce computation time, and use resources more effectively. Schedulers are particularly important to increase the stability and reliability of the model during inference. For example, they enable efficient management of the resources used in the inference process (e.g., memory and processing power). They also perform low-level adjustments necessary to optimize the model's performance in the inference process. In this way, schedulers improve the model's performance during inference, optimize calculation time, and provide an overall more efficient inference experience. As a result, schedulers are critical tools to optimize the performance of deep learning models in the inference process and achieve better results.

Stable Diffusion is an innovative artificial intelligence technology used to create visual content. LoRA models also have an important place in the use of this technology. LoRA models are known for their ability to analyze complex data structures and detect patterns in time series. Thanks to these features, Stable Diffusion can also use LoRA models to create realistic and creative visuals from text-based inputs.

LoRA uses learned patterns and relationships to understand textual information and transform this information into realistic and diverse visuals during the visual content creation process. In this way, Stable Diffusion allows users to create complex and engaging visuals even with simple text-based inputs.

LoRA models help Stable Diffusion create realistic visuals quickly and effectively, while also increasing diversity. In this way, users can create images in different styles and concepts and express their creativity freely.

**2.5. StableDiffusionPipeline, Gradio Interface and XL Models**

StableDiffusionPipeline is an AI modeling workflow used for text-based visual creation. This workflow takes text-based inputs and creates realistic and diverse visuals based on those inputs. It automatically converts textual descriptions into visual content using a large pre-trained language model and visual generation algorithms.

In my own project, I used a pre-trained model called "SG161222/Realistic\_Vision\_V2.0" in StableDiffusionPipeline. This model is an artificial intelligence model that has been trained to create realistic and detailed visuals. So, StableDiffusionPipeline calls this pre-trained model and takes a specific text-based input, allowing it to create realistic visuals.

This workflow can be used to create creative visual content across many industries, including art, design, film, and video games. It also facilitates the production of visual content quickly and effectively by enabling users to produce visual content with text-based explanations.

Gradio Interface is a user-friendly library that is used to reflect the artificial intelligence model and the images we produce to the user and enables creating a user-friendly website. Thanks to the easy-to-use and rapid development opportunities offered by Gradio, users can set the model's login information, the results can be visualized, and the display of the model can be monitored instantly. This enables AI model developers and researchers to quickly prototype and share results. Gradio Interface can be operated with many types of data such as text, images, audio and more, allowing users to create custom buildable enhancements for a variety of projects.

XL models is a term used to refer to large-sized and complex AI models. The term “XL” stands for “extra large” and generally describes models trained on large data sets and designed to accomplish more complex tasks. These models typically require more parameters, deeper network structures, and more computational power. XL models can perform complex analysis on many types of data, including text, images, audio, and more, and can form the basis for artificial intelligence applications used across many industries.

**2.6. Hands-on calling Libraries, Functions and Creating Interface for Our Project**

**1. Install Required Libraries**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 2.2. Install Libraries

Google Colab is an online platform that provides GPU support up to a certain limit. Therefore, this site was used to run the project code, so the libraries required for the project should be installed beforehand. As seen in Figure 2.2, other libraries that form the basis of our project, such as transformers, diffusers, accelerate, safetensors, and gradio, are being installed. The Transformer library is a tool developed by HuggingFace and widely used in the NLP field. It has a collection of pre-trained models and allows users to use these models. The Diffusers library was also developed by HuggingFace and produces high-quality images using advanced generative models. The Accelerate library was developed by Hugging Face to accelerate and scale PyTorch-based models. Safetensors is a library used specifically for safe and fast storage and loading of large models and tensor data. Gradio is a library that enables creating fast and user-friendly web interfaces for machine learning models. With a few lines of code, users can create interactive interfaces where they can test, demo, and share their models with others. All of these libraries play a key role in our project.

**2. Import Required Libraries**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 2.3. Import Libraries

In the import libraries section shown in Figure 2.3, we perform some import operations to use the libraries we have downloaded and are required for our project. Torch is an open source library widely used for building and training deep learning models. Torch autocast is a tool used for mixed precision training, but we make inference in our model. It is also very useful to use it for the Inference part to reduce memory usage and increase calculation speed. MediaPy library is a very useful library if we do not have an interface from the Gradio library and want to see the images produced directly in Python. We used this library frequently when testing the model. As we mentioned before, Gradio is the library I use to provide a user-friendly interface. StableDiffusionPipeline is a tool developed by Hugging Face and is specifically optimized for working with Stable Diffusion models. This pipeline is a diffusion model used to produce high quality and realistic images. EulerDiscreteScheduler is a scheduler used for diffusion models. It works based on Euler's method and helps optimize the performance of diffusion models. Safetensors is a library used specifically for safe and fast storage and loading of large models and tensor data. Numpy is a basic library for scientific computing in Python. It offers multidimensional arrays (ndarray) and functions that enable fast mathematical operations on these arrays. The OS module is a standard library in Python that allows interacting with the operating system. It provides functions to perform various operating system tasks such as file and directory operations, environment variables management, process management, etc.

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 2.4. OpenAI key and Function to get answer from ChatGPT

Figure 2.4. We can quote the following for the OpenAI key and Function to get answer from ChatGPT image. First of all, we need to install the openai library on Google Colab, then we import it and create the openai.api\_key variable because we have to use this key to connect to chatgpt. It is a function that gives the prompt given in the openai\_chat function to the gpt-3.5-turbo-16k model and returns a single result with a temperature value of 0.5, and returns this result from chatgpt at the end of the function. The reason why we connect to this type of function is, of course, that we need chatgpt. ChatGPT is a very powerful tool that can generate answers for templates and offer different language support. We will get help from chatgpt to produce the prompt we need for our stable diffusion model. Therefore, this function is very important.

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 2.5. Give a Template to ChatGPT and Create Stable Diffusion Prompt

Figure 2.5. Give a Template to ChatGPT and Create Stable Diffusion Prompt visual has a formula, this formula is called template and is the format given to get the desired output from ChatGPT. The prompt given by the user is added to chat\_prompt, and at the end, ChatGPT is asked to provide output containing only English output and the formula. This is the main function that produces the Stable Diffusion prompt, stripped of extra details, for us and is used to edit the prompt to be used in the image creation section.

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 2.6. The Beginning of Image Generator function and Seeds

The image\_generator function is the function we use to create the interface. It takes prompt, x\_dimension, y\_dimension, seed\_enabled, seed, enable\_inference\_steps, num\_inference\_steps and guidance parameters as parameters. Prompt is the parameter entered by the user that we will send to chatgpt for editing. x\_dimension and y\_dimension are the parameters that give the width and height values ​​of the image to be created, that is, the dimension values. Thanks to the interactive interface, the user can choose one of the options 512, 640, 768, 1024, depending on the user's wishes. It is a parameter set for torch.Generator based on seed\_enabled and seed. The -1 value is used to constantly produce different images, but if we want to constantly produce the same image, we need to give the same seed number. If we click on the seed\_enabled button and then define the seed value in the interface, torch.Generator will generate images according to this value. enable\_inference\_steps and num\_inference\_steps are a very important part. If the user has enabled enable\_inference\_steps, it is generally recommended to give a value between 20-40 in the num\_inference\_steps section. This value indicates how many times our model will work for that image while producing the image, and giving a low value may lead to poor quality or distorted images, so it would be logical to keep the value high, but the value Entering a value that is too high will also waste time, as setting it too high will take too much time. The Guidance parameter is used in Stable Diffusion models to adjust the effect of text annotation on the rendered image. The higher the Guidance scale, the closer the images the model creates to the given text description. This parameter plays an important role in achieving the desired results by giving the user more control in the creative processes.  
First, we need to assign defaultSeed and inference\_steps, because even if these values ​​are not assigned by the user, they must have a basic value, otherwise we cannot use our functions, and then they will be updated according to user input. The user must enter the prompt to produce the image and select the x\_dimension and y\_dimension values, otherwise we cannot enter the if condition and the code will stop working. Since the size information selected by the user interface is received as string, we first convert them to integer values. We chose to use the Realistic\_Vision\_V2.0 model, which is pre-trained on HuggingFace using StableDiffusionPipeline, but different models can also be preferred here. The pipe to "cuda" section has been set to use the GPU support that colab provides us. If the user wants to enter seed and num\_inference\_steps values, these are also checked as an if condition and the variables created at the beginning are updated with new values.

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 2.7. EulerScheduler, LoRA and Image Creation Part

Figure 2.7. The section includes EulerDiscreteScheduler, LoRA, torch.generator, negative prompt and image creation section. EulerDiscreteScheduler is a scheduler for diffusion models and uses an algorithm based on the Euler method to determine and manage steps in the diffusion process. This scheduler helps optimize the quality and stability of the model's output, especially in image synthesis and editing tasks. LoRA (Low-Rank Adaptation) is a technique that makes the training and adaptation of large language models and other deep learning models more efficient. Instead of updating all parameters of the model, LoRA adapts only certain layers with low-rank matrices. As the LoRA model, some of the models created specifically for children's books from the civitai.com website were tried and then we started to use the appropriate one through huggingface. Torch.Generator is a class used as a random number generator in PyTorch. This class provides various functions for generating and checking random numbers. Negative Prompt is used especially in text-based rendering processes to ensure that the model avoids certain undesirable features. In the image section, we use the pipe and enter parameters into our model to produce an image using the information we obtain. Then, since we will produce a single image, we return the 0th element and this becomes our image.

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Figure 2.8. Gradio Interface

In Figure 2.8, we see the code of the Gradio Interface we created for our project. As we mentioned before, the dimensions variable shows 4 different options that can be used for the dimensions of the image. Our first variable in gr.Interface is the function we will use. We give our image\_generator function here. The inputs parameter corresponds to the parameters in the image\_generator function, so we need to enter 8 different inputs in total. gr.Textbox is a box where the user can write, set to give a prompt. gr.Radio is the section that has different options and where the user can choose the dimension they want. gr.Checkbox is a function that can be given a checkmark and returns a True or False value. It returns a boolean type. gr.Number is a box set to 0 by default, where the user can enter the number he wants. gr.Slider is a good option for the part where the desired number value can be set with a slider and the inference\_steps value is selected. The outputs section contains what our function returns. Since we return the image we produced in our function, we use the gr.Image feature here and Gradio prints the image produced for us. title is the title of our project and appears at the top of the interface. There are different themes in the theme section and we are using a soft theme here. The description section provides an area under the title that can be used for description. The examples section is an area at the bottom of our project that contains sample input data. We designed this section to be compatible with our model in Civitai, and people who have never used this interface before can start producing visuals by selecting examples from here. In the last part, we launch it and set debug=True, and this part warns us if any errors occur.

**3. CONCLUSION**

It is very important to be able to visualize the stories of the books we read and think about the topics. In this project, we aimed to create a user interactive interface that can produce images for children's books. While doing this project, stable diffusion and LoRA models were the most frequently used and gave the best results. In order to document these processes, which can normally be installed directly on the computer, and to use the prompt section more actively, we wrote and designed the interface ourselves through Google Colab. One of the important shortcomings of text to image projects or applications is that they only work with sensitivity to the English language. One of the aims of this project was to save the prompt section from being only sensitive to English. ChatGPT currently provides support in more than 80 different languages ​​and can also be used as a large language model that can work with templates that can be formulated. Thanks to ChatGPT and the template used, the prompt section in our application produces a prompt given by the user, ready for our LoRA model, and then produces an image for the user. Thanks to these features, our project becomes a tool that can provide visual production services for children's books in more than one language.

In the project interface in Figure 3.1, there are text prompt, x dimension, y dimension, seed, inference\_number, guidance and some previously tried examples at the bottom. The user has to select prompt, width and height variables from this interface and after selecting, he can produce images and can also change other parameters according to his wishes. As output, a visual suitable for the prompt is produced using the Realistic Vision V2.0 model and the KIDS ILLUSTRATION LoRA on the Civitai website and can be evaluated by the user.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.1. Project Interface

In Figure 3.2, we see the interface where prompt, x dimension, y dimension and other parameters are set and an image is produced. These values ​​were created based on the prompt and other parameter values ​​of previously trained models available on the civit.ai website. The image created with the "closeup, a family portrait, momn, dad, kitchen in the background" prompt is printed as output on the right side of the screen, and the user can save this image to the directory he is working in by pressing the Flag button. In the production of model images, 5 is generally used as guidance and 40 as inference steps. At lower values, poor visual quality may occur, or at higher values, excessive time loss may occur. The submit button starts the image production, while clear resets all entries. As we learn to use this interface, we can constantly produce different images by trial and error, and we can easily produce images that we like or are useful in a short time.

metin, insan yüzü, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Figure 3.2. Usage of Interface and an Image

The image in Figure 3.3 was trained to have the order of the parameters "a Wolf is camping, kid, Desert Scrub", "1024", "1024", True, 3490148812, True, 40, 5. As can be seen, the image created at the end of this training is suitable for use in children's books and does not contain any risky images. At the same time, it looks like an image produced entirely based on the prompt. The base model we use in this project is SD 1.5, and depending on the type of these models, changing the height and width values ​​may lead to significant changes in image quality.

In the image in Figure 3.4, we see an image similar to the first image, but created with different characters and a different ambience. The most important point in which the style and structure of the images created here is related is the data set on which the LoRA model is trained. Depending on the data set on which the LoRA model is trained, the positive and negative aspects of the model begin to emerge, and the model chosen has great importance in managing these issues. Especially when producing images for children's books, we want these images not to contain anything negative or for parents to not be disturbed by the images when buying a book, because otherwise the book may become unsellable. Therefore, it may be important to first check the data set on which the model we use is trained when choosing the data set for the trained model, or if we are using a ready-made model. Another one is the negative prompt part. It is the part that indicates the features we do not want to be in the image in models where negative prompt is used. If there are issues that bother us or if we cannot find suitable data, we can easily get rid of them by using a negative prompt, but since there is no serious problem in the data set of the KIDS ILLUSTRATION model we use, "ngtvH, ngtv, ngtvB" is used as the negative prompt and this part is not presented to the user. No need has been seen in this regard.



Figure 3.3. First Created Image Figure 3.4. Second Created Image

The images in Figure 3.5 and Figure 3.6 reflect some negative features related to our model. The first negative feature of the project is that it cannot produce very good images for size values ​​such as 512, 640. Although SD 1.5 models generally performed well for 512x512 images, we could not obtain quality images for small size images. That's why images of 1024x1024 size are always produced in the examples section. Another feature of the model is that it constantly produces more than one character, depending on the prompt. This may also be a connection with the SD 1.5 model, but when we produced 1024x1024 images, we witnessed that more than one character was frequently produced. Even if we say "a Meerkat" or "a Chilean girl" representing a single character to the given prompt, we still have a project that insists on producing more than one character. It may sometimes be more useful to use "one" instead of the "a" prefix. Since there is more than one character in the images produced, each of them can focus on a different feature and an appearance can be created as if the characters are in independent situations. Another point that may cause a problem is that the guidance value must be between 3.5 and 7. Although these values ​​vary from model to model, if we give large values, serious errors or breaks may occur in the images. One of the biggest problems of image-generating models is that they have problems focusing on more detailed limbs such as hands, so unexpected missing or redundant images may occur in the hands of female or male characters from time to time.



Figure 3.5. Meerkat-based Created Image



Figure 3.6. Chilean Girl in a Wasteland

Newly used models do not need very detailed prompts like old models. Compared to old models, the weight of the prompts has been increased and now very beautiful images can be produced even with short prompts. Since we use the ChatGPT model as a Large Language Model that produces a prompt, attention should be paid to the template given to this model because long prompts may produce images that deviate from the desired visual or fail to achieve some desired things. In this respect, although new models offer convenience in some respects, their behavior in long prompts cannot be fully predicted. In addition, people who want to work on this project locally rather than on sites that offer GPU support such as Google Colab should have good hardware. Otherwise, depending on the graphics card, the image may not be produced, or the visual production time may be excessively long. Unlike Midjourney and DALL-E, Stable Diffusion offers people an independent structure. We can use the visuals we produce for our own benefit. Nowadays, AI influencers or artificial intelligence concert applications are frequently encountered, and these are thanks to Stable Diffusion, but it should not be forgotten that if we want to do more detailed work with these programs, we need very powerful hardware and the project we have created here is limited in number. It allows us to set parameters because we created this project using HuggingFace diffusers and transformers and we do not have access to all features.

**BIBLIOGRAPHY**

Wang, Jianfeng, et al. ”Git: A generative image-to-text transformer for vision and language.” arXiv preprint arXiv:2205.14100 (2022).

Hu, Edward J., et al. ”Lora: Low-rank adaptation of large language models.” arXiv preprint arXiv:2106.09685 (2021).

Wang, Lufeng, et al. ”Constructing a personalized AI assistant for shear wall layout using Stable Diffusion.” arXiv preprint arXiv:2305.10830 (2023).

Wang, Chenghao, and Jeanhun Chung. ”Research on AI Painting Generation Technology Based on the [Stable Diffusion].” International journal of advanced smart convergence 12.2 (2023): 90-95.

Datta, Siddhartha, et al. ”Prompt expansion for adaptive text-to-image generation.” arXiv preprint arXiv:2312.16720 (2023).

Wu, Ziheng, et al. ”EasyPhoto: Your Smart AI Photo Generator.” arXiv preprint arXiv:2310.04672 (2023).

Luo, Simian, et al. ”Lcm-lora: A universal stable-diffusion acceleration module.” arXiv preprint arXiv:2311.05556 (2023).

Levin, Artyom O., and Yuri S. Belov. ”A Study on the Application of Using Hypernetwork and Low Rank Adaptation for Text-to-Image Generation Based on Diffusion Models.” 2024 6th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE). IEEE, 2024. 8

Kim, Seonuk, et al. ”Designing interfaces for text-to-image prompt engineering using stable diffusion models: a human-AI interaction approach.” (2023).

Chang, Huiwen, et al. ”Muse: Text-to-image generation via masked generative transformers.” arXiv preprint arXiv:2301.00704 (2023).

Maerten, Anne-Sofie, and Derya Soydaner. ”From paintbrush to pixel: A review of deep neural networks in AI-generated art.” arXiv preprint arXiv:2302.10913 (2023).

Lee, Seongmin, et al. ”Diffusion explainer: Visual explanation for text-toimage stable diffusion.” arXiv preprint arXiv:2305.03509 (2023).

AUTOMATIC1111/stable-diffusion-webui: Stable Diffusion web UI (n.d.). https://github.com/AUTOMATIC1111/stable-diffusion-webui/tree/master (2023).

Medium, ”Build A Text-To-Image Application with Python and Open Journey in A Few Lines of Code” (<https://ifeanyidiaye.medium.com/build-a-text-to-image-application-with-python-andopen-journey-in-a-few-lines-of-code-9fd8b40af89a>)

Medium, Local Text-to-Image Generation using older Nvidia GPU and Gradio Interface (<https://medium.com/@daniela.vorkel/local-text-to-image-generation-using-older-nvidia-gpu-and-gradio-interface-a41aff359f2a>)