Top Text

Tom Jacobs and James Boogaard

December 15th 2022
Module: SEAR
Venlo, Limburg, Netherlands

Abstract

This is the abstract.

Contents

List of Figures

1 Introduction

1.1 Context and background

During the year of 2022 there has been a massive increase in the popularity and use of text-to-image generators, an artificial intelligence technique that translates human text into a computer-generated image. You will likely have heard of at least one of these if you frequent the internet. This explosion in popularity has led to many different text-to-image applications being made, which on the one hand has given many more options to artists, (people who use a text-to-image generator to create and refine an image), but it has also made the landscape more difficult to navigate because of its abundance of choice. This paper makes an effort to try and alleviate this problem by picking four of the most used text-to-image generators and comparing them based on their accuracy and process so we can help people gain some perspective on which text-to-image generator is best for their needs.

A research paper was published on the nineteenth of February 2020 called: "A survey and taxonomy of adversarial neural networks for text-to-image synthesis" that makes use of surveys to compare models for text-to-image generations. This paper aimed to show which text-to-image technique yielded the most realistic results in a wide variety of categories. One issue with this paper was that is was written before the explosion of consumer text-to-image generators happened, which is why we will instead focus on modern consumer-grade generators which are available today. Another issue is that they don't allow average inexperienced artists to use their tools, instead their results are created by the researchers themselves. Because of this their results do not reflect what the average inexperienced artist can expect when using these text-to-image techniques.

1.2 Problem and hypothesis

Our goal during the course of this research paper is to find out what the best text-to-image generator is out of the four we chose, in the case of an artist who is not experienced in the world of text-to-image generation. Our hope is that artists who are new to text-to-image generation will be able to make an educated decision on which generator is best for them on the basis of this paper.

Another question we take a look at is which applications have access to the most tools. More tools allow the artist more freedom to improve their results in a multitude of ways. Take for example inpainting, both Dall-E and stable diffusion give the artist access to this. With inpainting you can select, or paint over, a part of an image, you can then tell the generator what you would like to have generated in the selected area. More tools may sound great, however in practice inexperienced artists may struggle to get the hang of these kinds of tools.

The text-to-image generators we will be testing during the course of this paper are: Stable diffusion, Midjourney, DallE and Dream by Wombo. We came to these models after looking at several applications in google trends. This gave us a good idea of what the landscape looked like, and which applications artists would be likely to choose. You may notice below in Figure 1 that Dream by Wombo is not that popular anymore, however it was one of the first openly accessible text-to-image applications out there.

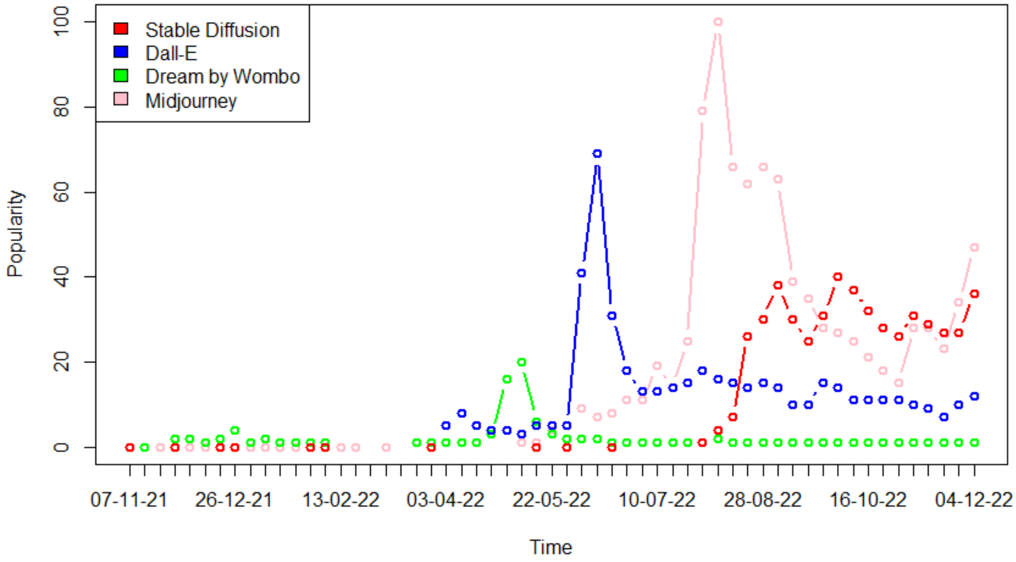

Figure 1: Results from Google Trends showing the popularity of different text-to-image applications

We expect the best text-to-image generator out of the four we chose to be stable diffusion. This is due to the fact that stable diffusion offers its users more tools and techniques like inpainting (allows you to regenerate a specific area of an image without regenerating the whole image), this allows the user more freedom to fine-tune their result. Given enough time this will yield a more accurate result. This however only applies for artists who are already experienced, because it can take quite some practice to get used to all the tools that are available in stable diffusion. We expect inexperienced artists to struggle finding the right settings because of the amount of freedom they get. In the case of a completely inexperienced artist, they might find Dall-E easier to use because it can create photo-realistic results without having to fine tune any settings apart from the prompt.

2 Methods

2.1 Data collection

Our goal during the course of this research paper is to find out what the best text to image generator is, out of the four we chose, for artists who are experienced as well as artists who are not experienced in AI art generation.

In order to find out which is the best one, we will look at two things, namely: their accuracy in recreating an image, and their ease of use for inexperienced artists.

Our x value is the array of different applications that we use and the y value is the result of that (the distance) for each iteration on the road to trying to recreate an image using these applications. The intervening variable in this case would be the amount of experience using these applications.

2.1.1 Generating images

We have chosen an image that we find sufficiently covers all the challenging parts of recreating an image using ai art generation such as reflections and complex composition. We then use each of the four artificial intelligence applications to try and recreate that image within a given time frame (15 minutes). After each iteration we take the generated image (iteration) and add it to a list of the iterations that we created while recreating that image for that specific application (each application will thus have their own history of the process of the participants tying to recreate the image).

2.1.2 Scoring generated images

With those sets of data, we then plug the images (each iteration) into an AI application, developed by DeepAI, that can give you a numerical value for how close one image is to another and see how close each iteration is to the image we are trying to recreate. this then gives us insight into how fast the application is (how efficient its technique is) and how many iterations you have to do to achieve a certain result.

In order to compare the image generation applications effectively we observe a multitude of factors that will indicate the strengths and weaknesses of that application. These factors are: The amount of iterations each participant can generate while trying to create the image within the 15 minute time limit, how close each iteration is to the image we are trying to replicate using the ai image comparing program's unit of measurement (the unit is called distance. The closer to zero the distance is, the closer the images are together. zero means that the two images that are being compared are exactly the same), how easily we were able to access the application and its functionality, how much functionality each application offers and their effects on the final results of our experiment.

When choosing an image to compare each result to, we chose one that sufficiently covers all the hard parts of recreating an image using ai art generation such as reflections, complex composition, lighting differences and multiple objects within the image. We then use each of the four artificial intelligence applications to try and recreate that image within a given time frame (15 minutes). After each iteration we take that image and compile it into a list of the iterations for the process of recreating that image for that specific application.

2.2 Visualizing data

Once the data is gathered, we can create a graph to compare the applications more easily. The people that carried out this experiment also consisted of one novice and one experienced individual. This is to see which is the best application for each group of people and thus, the results will be conveyed on two separate graphs.

In terms of our secondary research question this method also gives us insight if we are aware of the techniques each of them use we can notice if there is a pattern of success or failure with each.

3 Results

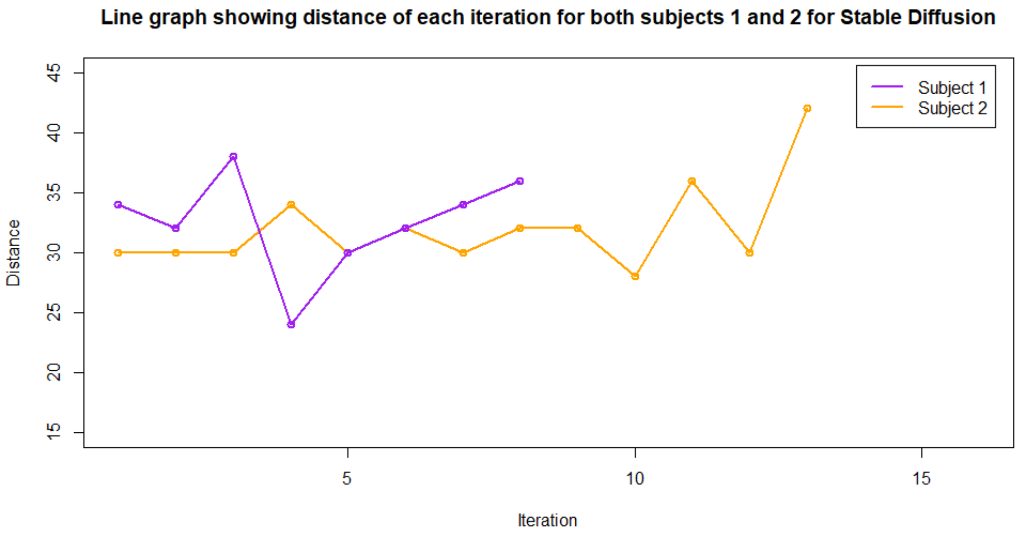

Figure 2

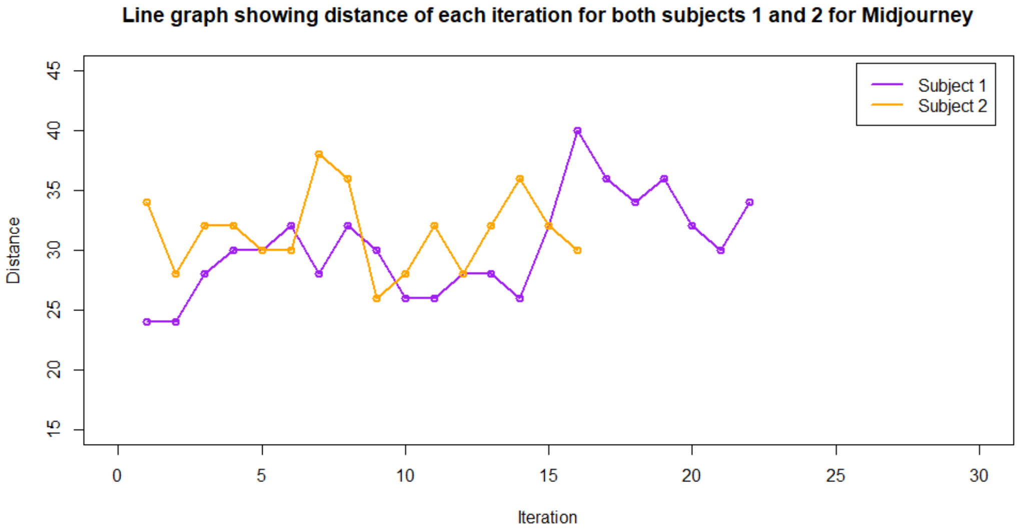

Figure 3

3.1 figure 2 and 3

What is interesting in both figure one and two is that the discrepancy between the two subjects in both applications is larger than that of any of the other applications we tested. This is seen with Subject two having nearly double the amount of iterations when compared to Subject one in figure one and then Subject 1 having more iterations than Subject two in figure two. Another point of interest is that the application in figure one has as a small amount of iterations when compared to the other applications. (-make reference to the fact that subject 2 has a bit of exper with sd and subject 1 has expr with midJ-).

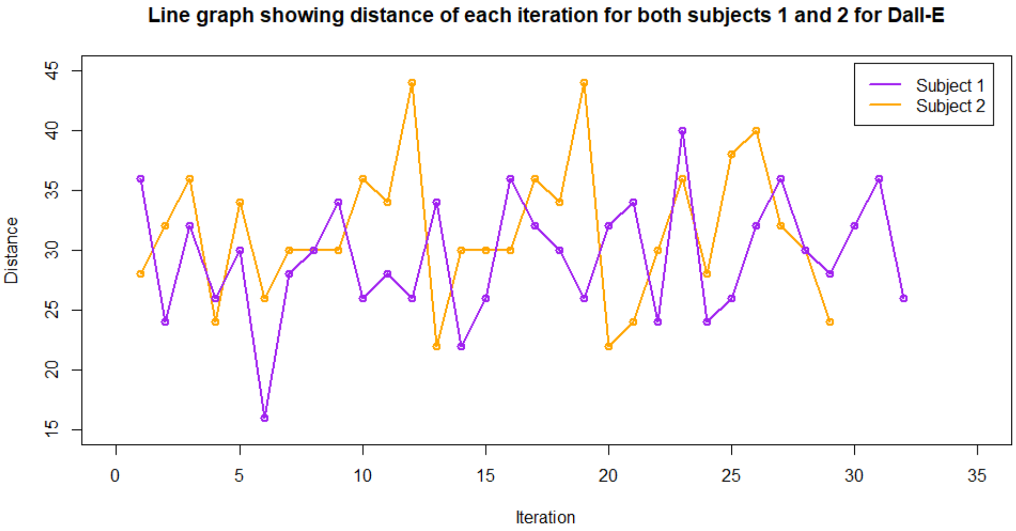

Figure 4

3.2 figure 4 and 5

In figure two we see it has the most iterations out of any of the other applications. The spread of the distances is also the largest out of any of the other applications. When comparing the amount of iterations that each subject has in both figure on and figure two, these two applications are very close.(-This can be attributed to the fact that the amount of experience with these graphs is very similar for both subjects-)

3.3 figure 6

Figure 5 shows that the average distances of all the applications from broth subjects one and two are very close.(this is due to the random nature of t-i-g and shows that a good application is more judged by the functionality it offers rather than its algorithm when it comes to new comers). When comparing the applications Dall-E has the least average distance and stable diffusion has the most average distance but again only by a slight margin.

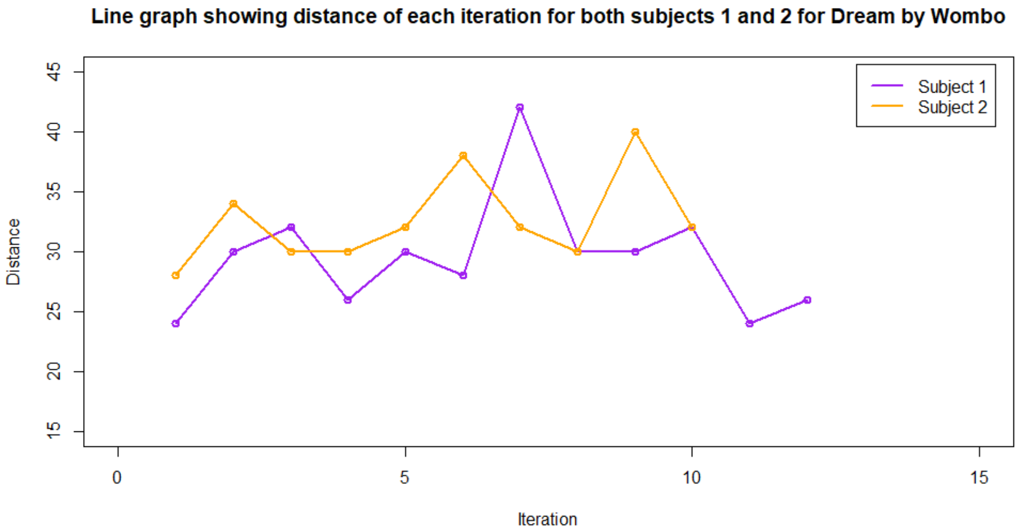

Figure 5

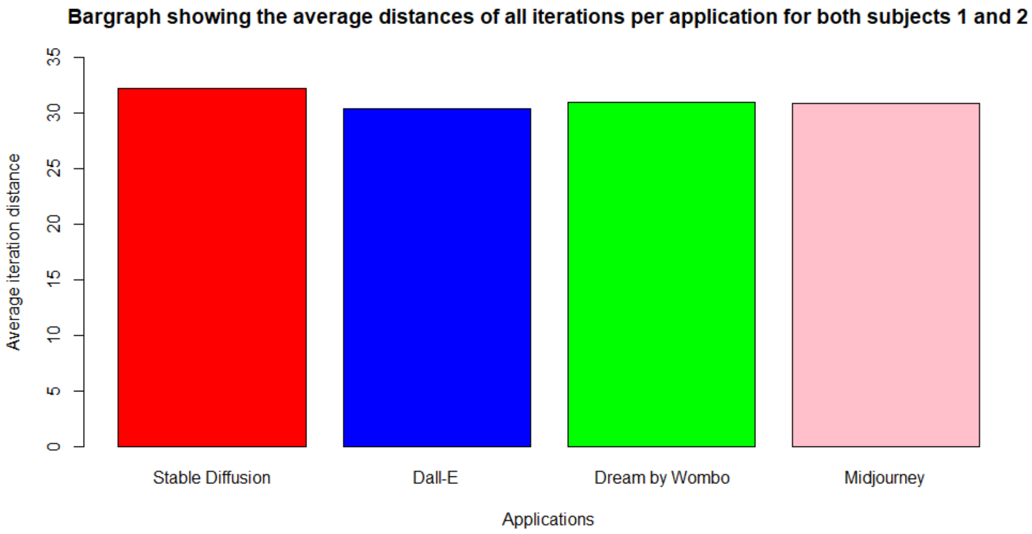

Figure 6

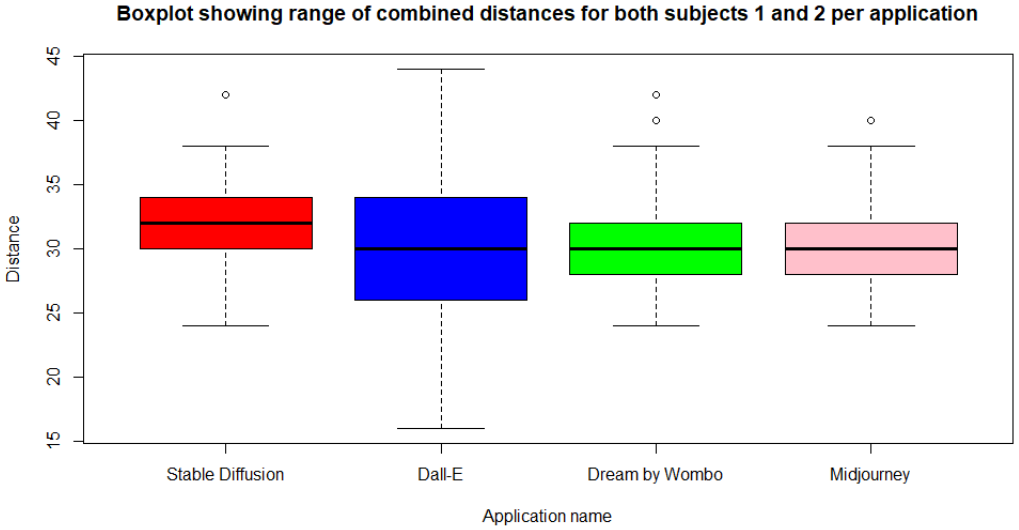

Figure 7

3.4 figure 7

In Figure 6 there are two important points of interest. Firstly is that Dall-E has a massive range between the minimum and maximum distances when compared to the other applications as well as having the largest space between q2 and q3. Second is that Stable Diffusion has the largest median distance. Mention something about the outliers.

4 Discussion

4.1 Title

During the course of this research we were able to prove the legitimacy of our secondary research question. This was done by a analysing figures two through to six. The applications shown in figure two and three are applications that both subjects have experience in(Subject 1 has experience with Midjouney and Subject 2 has experience with Stable diffusion). What makes figure two and three important is that both the graphs show the difference that having experience in an application makes, and that difference is only in the amount of iterations the user is able to generate.
In figure three and four we then see two applications that neither Subject 1 nor Subject 2 have any prior experience in. using comparative analysis we can then compare both the graphs that the users have experience in (being figure two and three) and the graphs that the subjects don't have experience in (being figure four and five) and we can see more clearly that when both subjects have the same amount of experience with an applications the amount of iterations the subjects are able to generate is noticeably closer together.
When we take a look at figure six we can see that even though even though it is the combinations of the averages of all iterations across both Subject 1 and 2 per application, the differences between the bars in very small. This can be attributed to the nature of text-to-image generation being that of a random series of attempt until the user reaches a point where they are happy with the picture. This is further proven when you look at any of the patterns for figures 2 through 5 and how the distance for each iterations follows no solid pattern but rather jumps around sometimes even ending up further away from the original image than some of the earlier iterations.

4.2 Secondary research question justification

With all this information we can now say that an application's value to an inexperienced user is not about how good the AI generations technique is because AI text-to-image generation is very random and relies a lot on luck even for users that are experienced with applications. What we do know is that if all the applications rely on luck to some degree it is then better if the application is able to give many iterations because the more attempts you are able to make the faster you are more likely to get the result you are looking for.