UNIVERSITY OF CALIFORNIA
SANTA CRUZ

GENERATING IMAGES WITH FREE-FORM TEXT GRIDS

A thesis submitted in partial satisfaction of the
requirements for the degree of

Master of Science

in

COMPUTATIONAL MEDIA

by

Wilson Mui

June 2021

The Dissertation of Wilson Mui
is approved:

Professor Sri Kurniawan, Chair

Assistant Professor Adam M. Smith

Dean Quentin Williams
Interim Vice Provost of Graduate Studies

Table of Contents

List of Figures

Abstract

Generating Images with Free-form Text Grids

by

Wilson Mui

This project demonstrates a new kind of drawing tool with a novel interface. Recent research in machine learning for image generation has mainly been focused on producing high-quality results and its efficiency in doing so. There was little consideration for how these systems should be interacted with by their actual intended users. My project offers an interface allowing for users to generate images by specifying details in a composition using high-level textual descriptions. To validate this design, I conducted two studies involving expert artists, designers, and UX practitioners.

Acknowledgments

I want to thank Adam Smith for putting up with me wandering aimlessly these past two years.

Chapter 1

Introduction

1.1 Overview

There is a wide variety of machine learning methods for the fast generation of life-like images. What would be a "good" user interface or user experience (UI / UX) for this type of machine learning based image generation? Specifically, I can focus on image generation in the context of machine learning tools. A "good" interface should be a usable interface that empowers users to interact with the tool and allows them to generate high quality images intuitively.

The motivation for this research came from frustrations in my past design work. It was, and still is, difficult to search for specifically composed images to use on my personal work. Searching for images with plants on some corner and other details in other parts of the image can take an endless amount of time because such an image may not even exist. This situation occurs often, such as when empty space is needed

in some parts of an image in order to place text. Creating a high-fidelity image using common tools such as Adobe Photoshop is time-consuming and also requires sufficient experience to generate something useful.

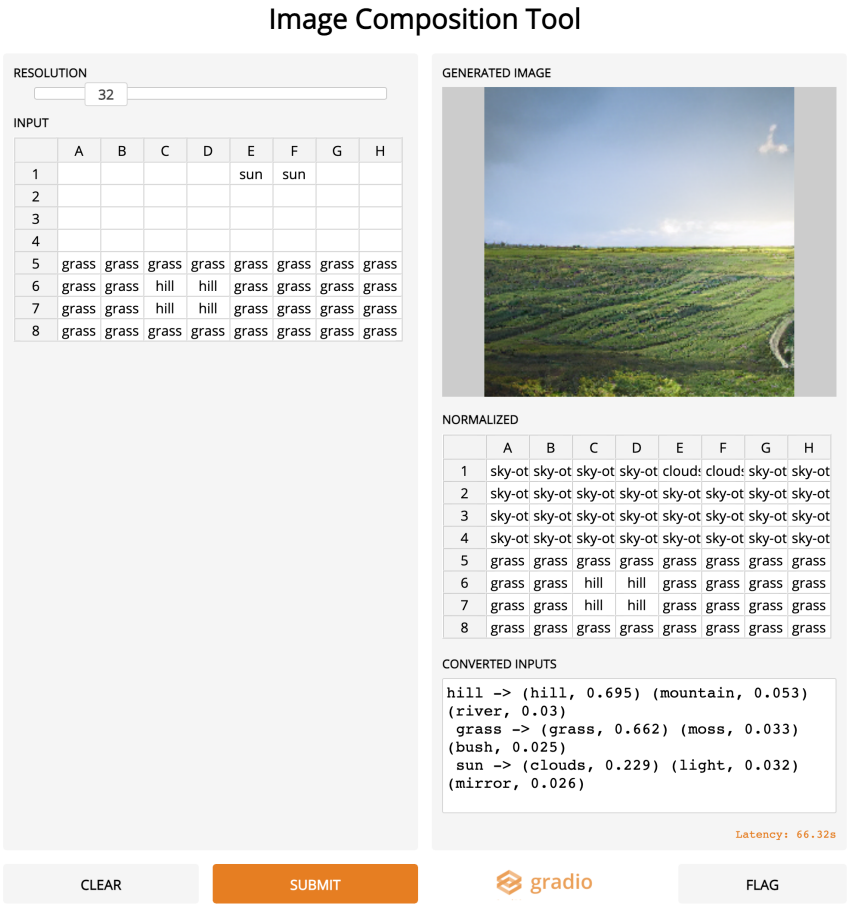

Figure 1.1: Screenshot of image generation tool being used to create image of landscape specified in free-form grid input.

The new tool contributed in my research (see in Figure 1.1) allows artists to create highly detailed images by specifying image features in a grid. If an artist wanted clouds on the top left of the image, they can simply type "clouds" in the top left cells of the input grid. The user would not have to worry about their input being one of

many special keywords, as the tool will attempt to translate any words from user input into something the underlying model understands. The generated image will often be a lifelike representation of what the user proposed. More examples of images generated from detailed composition ideas can be found in Appendix B

The main contributions of this project are the introduction of free-form text grids as an input method for image generation, the validation and analysis of this interface, and showcasing the use of transformer-based models for practical image generation.

1.2 Background

While there has been research done in image generating interfaces and in machine learning image tools, there has been a lack of study in where the two meet. Some novel examples of interfaces can be seen in studies that experimented with voice interfaces like VoiceDraw [6]. However, interfaces like those were designed to work around users with disabilities [3] or provide additional assistance to power users willing to accept a higher learning curve [8].

Examples of machine learning tools that emphasized the user experience to some degree are GauGan [13] and DALL-E [14]. GauGan hoped to allow "user control of semantics and style." They attempted this with a conventional paintbrush-oriented experience. Users were expected to "paint" various features in areas of the board and then pressed a button to generate the image. On the other hand, DALL-E attempted a much different user experience. With DALL-E, users simply entered a sentence defining

the image they had in mind. This incorporated natural language processing into the design.

My project can be thought of as a middle ground between GauGan and DALL-E. While the paintbrush functionality is intuitive in applications like Illustrator that mimicked the physical act of painting, it was less intuitive in GauGan's use case of designating features and their placement. DALL-E's input was intuitive, but lacked ability in defining specifics of image features and their composition. In my design, I allow users to specify image details in a spatial composition grid. Users provide feature words of their choosing to place in the areas they want on the grid. The tool will understand the words and generate a user-defined image.

Chapter 2

System Design

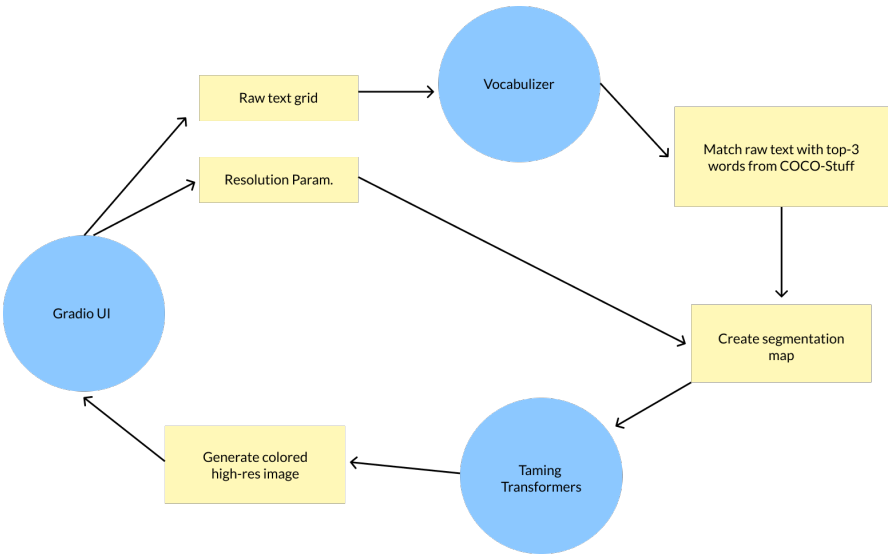

Figure 2.1: System architecture for my tool.

2.1 Components

This tool was implemented using three main pre-existing components: Gradio [1], Vocabulizer, and Taming Transformers [5].

Gradio is a Python library for rapidly building web interfaces. It is normally used for machine learning applications. Structured input and output fields can be displayed for the user.

Vocabulizer was a small script provided by my advisor (detailed in Appendix D) for matching user input text to my system's known vocabulary. It works by finding which existing term has the nearest word vector to the average word vector of the user-specified input. Vocabulizer is based on Wiki-words-250 [9], a word2vec model trained on Wikipedia text.

Taming Transformers is the system adapted from Esser and Rombach. I utilize the model they provide in their example Google Colab notebook.[1] Their model uses their novel VQGAN and transformers to synthesize high resolution images, potentially conditioned on semantic information such as a semantic label map.

2.2 Integration

With Gradio, I implemented a compositional input-based interface for users (using the DataFrame input type). This allowed me to retrieve a raw text grid defined by the user. Empty cells are automatically converted to the "sky" keyword. The grid

[1] `https://colab.research.google.com/github/CompVis/taming-transformers/blob/master/scripts/taming-transformers.ipynb`

was stored as a 2D array.

```
sample_input = [
                ["", "", "", "", "", "", "", ""],
                ["", "", "", "cloud", "", "cloud", "", ""],
                ["", "", "", "", "", "", "", ""],
                ["", "", "", "", "", "", "", ""],
                ["", "", "", "", "", "", "", ""],
                ["", "", "", "mountain", "mountain", "", "", ""],
                ["", "", "", "mountain", "cat", "mountain", "", ""],
                ["ocean", "ocean", "ocean", "sand", "sand", "sand", "ocean", "ocean"],

]
```

Figure 2.2: An example of how the grid is represented.

Gradio's Slider input type was used to retrieve the resolution parameter. This parameter was used to store a value to be used in setting up the resolution of the segmentation map later on.

The Vocabulizer serves to match each word from the 2D array of raw text to the nearest word from COCO-Stuff [2] set of labels (supported by Taming Transformers). I modified Vocabulizer to retrieve the nearest three keywords. COCO-Stuff is the vocabulary understood by the Taming Transformers model. A new 2D array of the closest mappings is created.

The initial code provided by Taming Transformers did not take a 2D array of labels as input (or any GUI), so additional code was written to encode the 2D text array into the segmentation map we needed. I generated the segmentation map by interpreting the user-provided text as a low-resolution grid. After finding the index of which COCO-stuff word best matched their input, I created a higher-resolution grid where each input grid cell became a 32x32 pixel block of a solid color (or a smaller or larger size of block based on the resolution slider).

The transformer iterates though the segmentation map in a "sliding window

Figure 2.3: Resulting segmentation map generated with the sample input show earlier and a low resolution.

manner" to update the segmentation map sequentially. This generates a lifelike high-resolution image. The image generation process takes about 1.5 minutes with the default block resolution of 32.

When the image generation is done, the image is returned to the Gradio interface. The generated image is shown through Gradio's Image output. Along with the image, my tool also displays additional info for transparency about the text translation (using Text output type).

Chapter 3

User Studies

3.1 Study Designs

I designed and executed two studies to test the user experience of my design and find usability issues. The first was a task-oriented study and the second was a heuristic evaluation. The goal of my study was to prove that this method of interaction is intuitive and useful for artists using machine learning based art tools.

3.1.1 Task Analysis

The task analysis is intended to observe an expected audience for my tool use it in a common scenario. By doing this, I hoped to see if the interface design aided the participant in the task or if it proved cumbersome and hindered them. They were asked to also use the tool they were most familiar with so they can better speak about the differences in using the more traditional tools compared to my novel tool when carrying

out the task. I recruited 3 participants with strong backgrounds in digital art. They practiced art as either a hobbyist, semi-professional, or professional.

- Participant A is a professional artist who creates art as a career. They have a formal background in the arts, specifically in computational art. Their tool-of-choice is Adobe Illustrator in which they are an expert at using.
- Participant B is a semi professional artist who used art as supplemental income. They also use Illustrator primarily for digital art. Their background is in computational art, and they are a graduate student in that field.
- Participant C uses art as a hobby. They are most proficient in Adobe Photoshop. They generate art for shows and personal enjoyment. Their background is also in computational art.

This study was conducted over Zoom, and it was recorded with the consent of each participant.

3.1.1.1 Task analysis protocol

1. **Pre-survey questions** (5 min)
   (a) What tools do you normally use to create images?
   (b) In what settings do you use these tools? (Professional, hobbyist, etc.)
   (c) How would you describe your skill level with those tools?

2. **Introduction to my tool** (1 min): The tool is introduced to the participant. They are given a demonstration of how the interface works and how to use it to generate images.

3. **Participant asked to familiarize themselves with tool** (5 min): I allow the participants to try out the tool and ask any questions about how to use it.

4. **Task** (25 min): The participant is asked to perform a given task twice, once with the tool they are most familiar with and once with my tool. The task was: "Generate an image of a landscape with a beach, mountains, and light clouds in the sky. Please try to generate a similar image with both tools." They are asked to speak out loud their thoughts while performing this task.

5. **Post-survey questions** (10 min)

   (a) Do you see yourself using this tool in the future?

   (b) What situations can you envision this tool being used in?

   (c) Would you say the tool was intuitive to use?

3.1.2 Heuristic Evaluation

For this study, I used heuristic evaluation and competitive analysis to validate my novel UI while comparing it to the existing approach (based on the metaphor of traditional painting practices) implemented in GauGan. Nielsen discovered that it only takes 5 expert evaluators to uncover over 80% of usability issues and almost all severe

ones [11]. I decided to recruit 5 expert UX practitioners to perform an heuristic evaluation on GauGan and my free-form composition tool. The heuristics used were the ten general usability heuristics for user interfaces [4].

My 5 experts were from academia and industry. Some currently perform UX research at TikTok, Adobe, and other companies. Others are graduate-level student researchers with UI / UX and human-computer interaction backgrounds. As both tools were essentially still prototypes, it was expected that there would be substantial usability issues. I want to focus on those that specifically relate to the manner of input (paintbrush and free-form text composition). The experts were asked to use both tools and then note the issues found as well as rate the severity of the issues from on a 1-5 scale (5 being most severe).

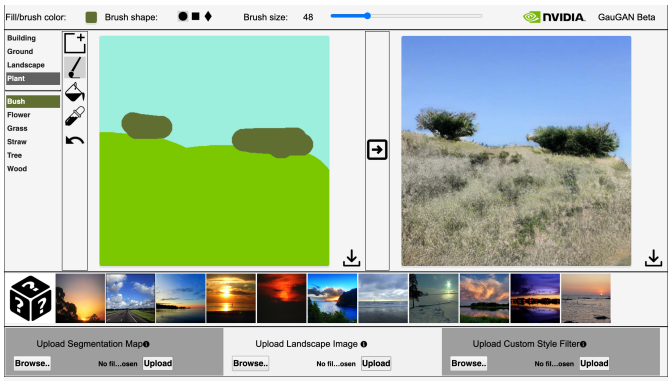

Figure 3.1: Interface for GauGan tool relying on more conventional "paint brush" interaction.

Chapter 4

Results

4.1 Task analysis results

- Participant A was observed to be an instant power user of the tool. For example, the artist was immediately manipulating entire rows and columns to efficiently enter and edit cells in the grid. They claimed the interface was "very intuitive" and were able to compose images quickly. They liked how little time and effort was required to produce a high-resolution image because there was no need to consider colors and other fine details. They could see this tool being used to generate unpredictable results for ideation/inspiration, collaging, creating story sets, and also making quick images for gifting purposes.

- Participant B also believed my interface to be intuitive and "so clear and easy to understand." They said it was apparent how the tool works at first glance. They also believed it to be better suited for the task than Illustrator. The transparency

of the system was valuable to them as well. They could see themself using this tool for auto-generating images as well as for teaching others about machine learning. They would be more inclined to use it if there was more control over the output.

- Participant C also believed that my tool would be more intuitive to use for new users compared to the tool they are most familiar with. However, they personally found Photoshop to be more intuitive for the task because it offered more direct control to the user. They could see themself using this tool to aid them in generating images within Photoshop. They mentioned how it could be helpful for creating samples to use in Photoshop when they have to perform tasks similar to the one they had just done. It could also be used for generating 2D worlds for video games because of its grid-based nature.

4.2 Heuristic evaluation results

A total of 43 usability issues were noted for GauGan and 29 for my free-form compositional tool (detailed in Appendix C). The usability issues found can be organized and summarized like so:

- GauGan
  - Lack of documentation indicating proper usage
  - Difficult to undo application placement of features
  - Reliance on recognition of paintbrush functionality seen in other tools leads to confusion for less-experienced artists

  - Lack of transparency of underlying system
  - Interface too busy / lack of visual hierarchy
  - Connection between feature being applied and paintbrush color can be confusing
  - Terminology used on interface may not make sense to those not involved in ML
  - Lack of error prevention or mitigation

- Free-Form Composition Tool
  - Not immediately intuitive due to lack of match between real world interactions.
  - Lack of documentation indicating proper usage
  - Lack of error prevention or mitigation
  - No way to quickly clear entire grid
  - Not immediately intuitive due to lack of match between real world interactions.
  - System status unclear while waiting for image to generate
  - Resolution feature may be confusing to many

Chapter 5

Findings

The study with the artists showed potential with this tool. The artists interviewed believe that this tool is approachable due to having an intuitive interface with a very low learning curve. They all believe that it is more intuitive than Adobe's Illustrator and Photoshop for new users. The artists had also revealed interesting potential use cases that are empowered with this unique interface design.

The free-form composition tool may be well-suited for generating high-resolution images where composition is important but specifics of the features themselves are not. Some scenarios for this is in collaging, particularly when done in conjunction with tools like Photoshop. Two of the artists have suggested that it could be useful for generating samples for further processing in other applications. When artist C was creating their image on Photoshop, they noted how it would be handy to use the composition tool to generate images they can graft onto their project.

The results from the heuristic evaluation were mostly minor usability issues

due to the early nature of both tools. Issues like those were mainly categorized as issues due to lack of documentation indicating proper usage of certain features. The vast majority of issues found in both interfaces could be resolved with more descriptive text.

My focus for the heuristic evaluation was to observe and compare the usability of the paint-brush functionality and the free-form grid. Neither were stated to be intuitive by the UX practitioners. With the paint brush, it was difficult to connect the color to the correlated feature. As a result, it was difficult to recall what features are on the board once many are being applied. This issue does not occur with my interface because each feature is clearly labeled with text.

The paint brush relies heavily on users recognizing the functions related to it (paint bucket, eye dropper, etc). When the user is not familiar with the paint brush, the interface may become severely less usable and the learning curve is significant. Compared to my tool, GauGan's usage of the paint brush can be less intuitive than the free-form grid for inexperienced users as the grid has less of a learning curve.

It is interesting to note that many of the UX practitioners stated that the free-form interface is not intuitive when the artists claim the opposite. This may be due to the artists already being well-acquainted with using similar interfaces such as digital spreadsheets. It may be useful to carry out more testing with artists from non-computational art backgrounds.

Chapter 6

Conclusion

The novel interface for machine learning image generation tools demonstrated by my project can empower artists and creators in their work. With more proper indicators and interface documentation, the interface can be even more intuitive and usable. This method of interaction may be more usable and suitable than the conventional paint brush interaction typically used by non-machine learning tools.

With a free-form grid, artists can very quickly synthesize new images with the compositions they desire for use in storyboarding, collaging, ideation, and creating samples for further processing.

Appendix A

Link to Google Colab project

https://colab.research.google.com/drive/1SoCq1Hjc4VxDc7DBeu--1vw4UQ-iIt6s?usp=sharing

Appendix B

Project Screenshots

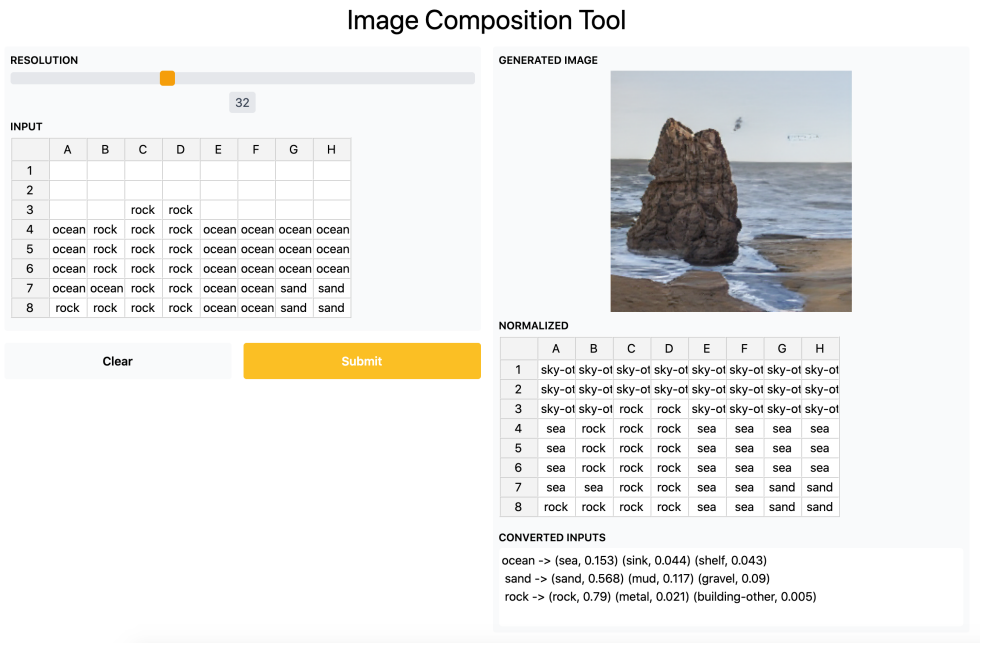

Figure B.1: Image of ocean landscape with sand in bottom left and rock formation on center-left of image.

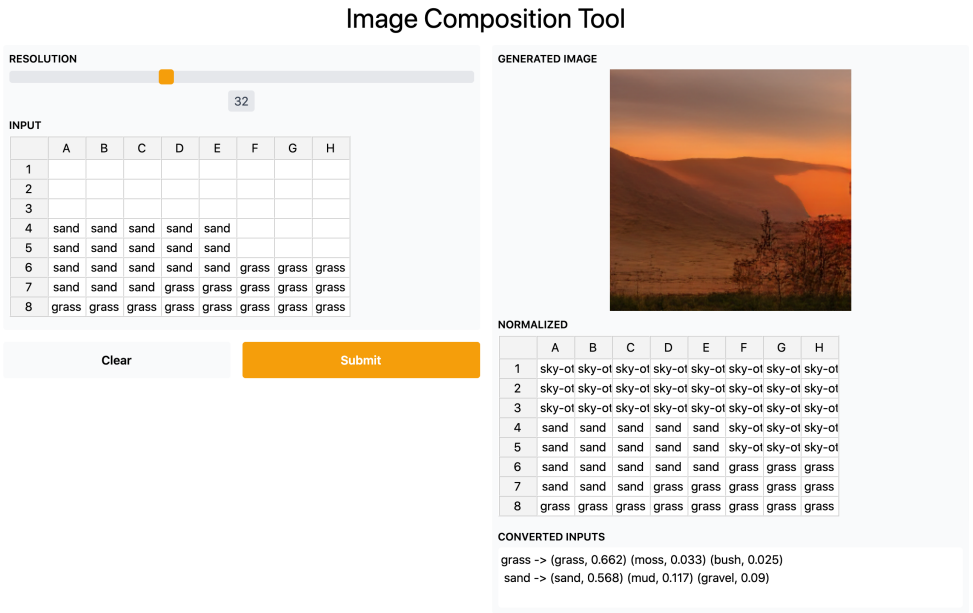

Figure B.2: Image of grassy and sandy landscape.

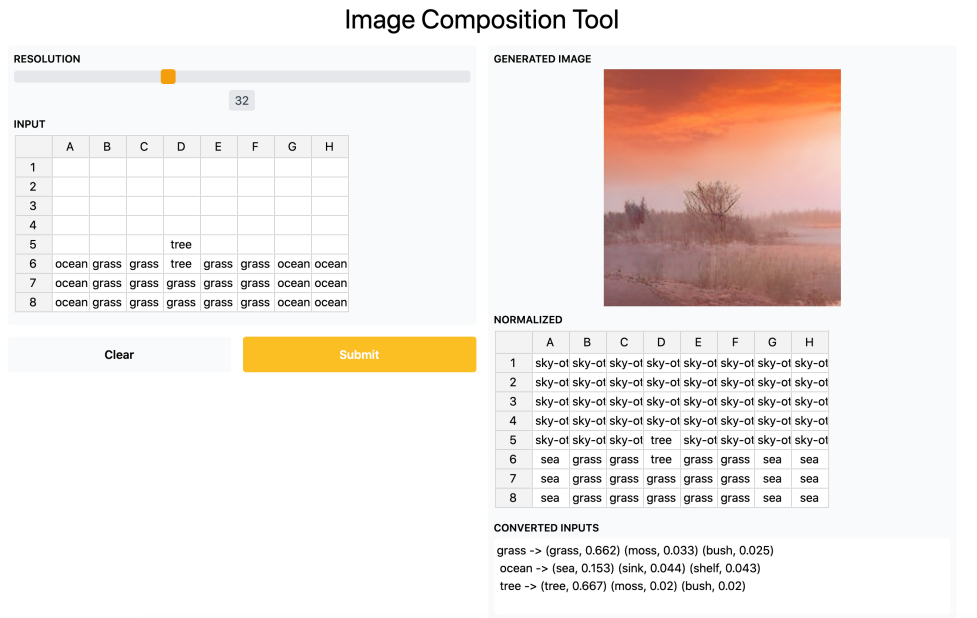

Figure B.3: Image of complex landscape featuring water, grass, and trees.

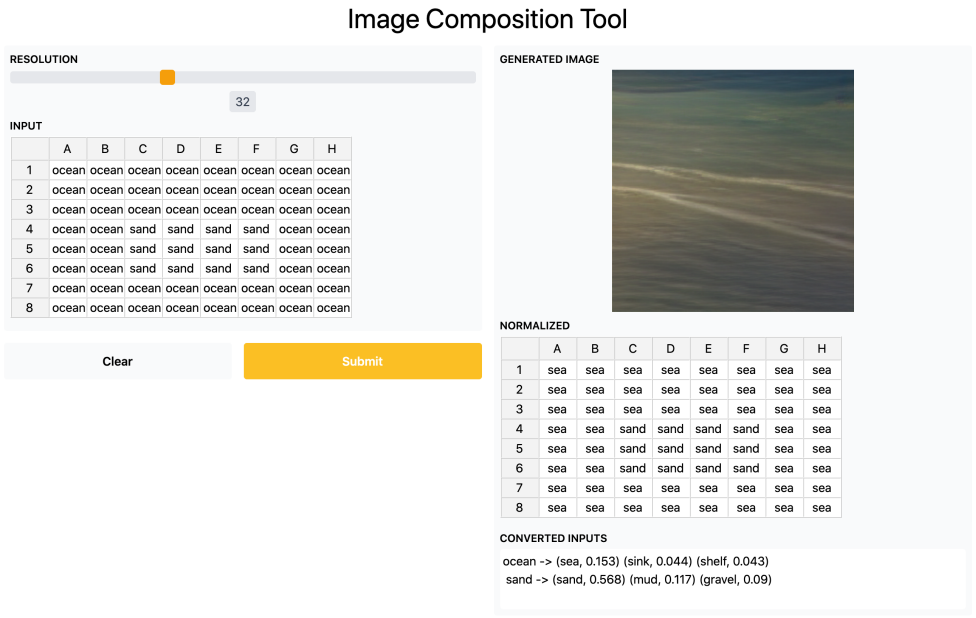

Figure B.4: Image of a body of wanted with some sand in center area.

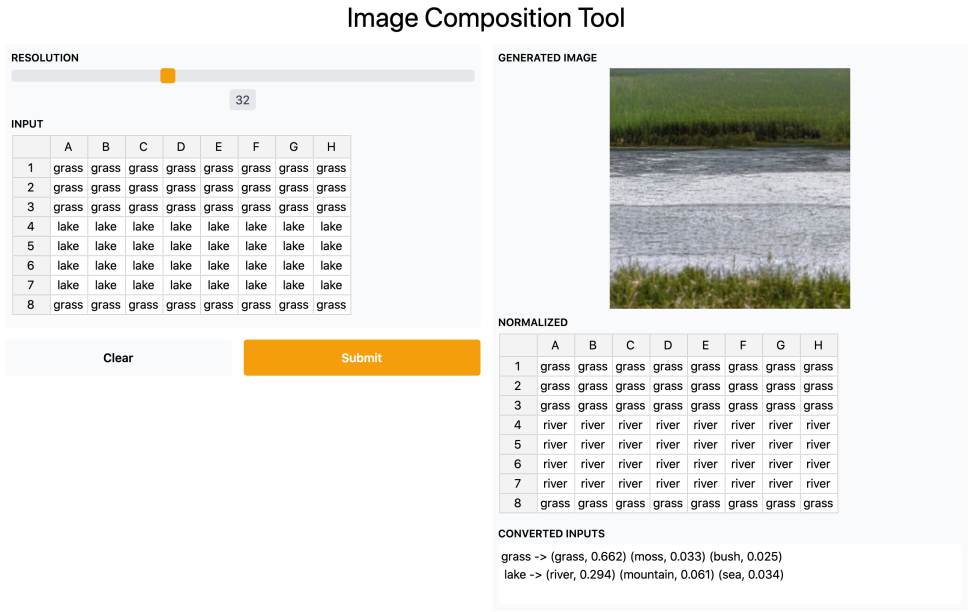

Figure B.5: Image of lake bordered by grass in front and back.

Appendix C

Raw Heuristic Evaluation Results

C.1 Expert 1

C.1.1 Nvidia's GauGan

Severity	Issue Found
2	Visibility of System Status: there's nothing showing me the progress of the loaded image or how long it's going to take to show up.
5	Match between system and real: originally missed the words and the terms and conditions checkbox below. Didn't know there was a tutorial video until after I'd tested out the tool. *Recommendation: maybe the bottom portion can be moved to the top so people can notice that first and to minimize confusion over the usage of the tool.*

5	User control and freedom: There is an undo button as well as a clear board button, which is nice but the undo button only allows me to undo the most recent line, shape, etc. that I've created. In other words, if I create 3 separate lines, the only button only works for the most recent line that I made. To clear the rest i have to click the clear board button.
4	Recognition rather than recall: The clear/add board button was a bit confusing at first, as well as the dropper button for picking up the colors on the board. It took me a bit to understand what those were for.
3	Aesthetic and minimalist design: There's a lot going on at once. I had no idea what to look at or click first for a while.
5	Help and documentation: same as number 2.

C.1.2 Free-Form Composition Tool

Severity	**Issue Found**
4	Visibility of System Status: there's nothing telling me how far along the image is as it's processing. Not sure how long I'm supposed to wait for the image to pop up.
5	Match between system and real world: no instructions on what to do or how to use the tool. Thought I was supposed to select the boxes and then generate something at first - didn't realize that I had to type words in. The cursor doesn't pop up in the boxes until you start typing.
1	User control and freedom: when I wanted to delete a bunch of inputs in particular cells or select a batch of cells to add in the same input for all of them, I couldn't do that.

4	Error prevention: The word error popped up whenever an action couldn't be performed, but there was no message telling me whether or not I was making an error and what error I was possibly making or if it was just an error with the system.
4	Help users recognize, diagnose, and recover from errors: same as number 4.
5	Help and documentation: same as number 2.

C.2 Expert 2

C.2.1 Nvidia's GauGan

Severity	**Issue Found**
2	Feedback for clicking the brush. Needs a visual indicator that the brush is working
2	It's not clear that the different menus on the left expands
5	Undo button only works once and is still highlighted
2	Different cursors should match the paint brush we're using
1	The highlighting the type of art (color) isn't as intuitive in the start. Didn't notice that it represented the pain color
2	Reset button needs to be a little clearer. Not fully intuitive that it restarts the image
1	Probably highlight which image we're using for the right side as a reminder

3	General feedback on the clicks (either highlighting, or button pushes)

C.2.2 Free-Form Composition Tool

Severity	Issue Found
4	Tell us what percentage the image is at in terms of processing so we're not waiting
5	At a smaller resolution screen, I could actually scroll further to the right
2	Maybe highlight the data points that were changed and normalize
4	Some of the texts bleeds into the other cells making it hard to read
5	If you copy and paste fields and it reaches too far to the right, then it creates new cells and makes an error
5	BE Clear with capitalizations and how it affects
5	ALSO PRESSING TAB will add new cells

C.3 Expert 3

C.3.1 Nvidia's GauGan

Severity	Issue Found
5	No user journey to get started; instructions not clear or easily findable to get started. Without understanding what to do and how to use it, users will churn

N/A	The shortcuts should be connected to the tools. For example, if you rollover a tool in Microsoft, you can see what shortcut it connects to. This encourages user play, discovery, and learning
5	Video should come before landing on the page to encourage Learn
N/A	About the Tool should also come before the actual Tool to give context of what this is. For example, 5 with the paper. Perhaps a summary of the context and why this came to be, and the possible uses.
N/A	6 & 7 should be a footnote. Any additional information is unnecessary because it distracts the user and is a cognitive load. Should focus on tool
N/A	The icons are confusing to use, unclear what they do
N/A	Bad user flow
N/A	No transparency in how the system works
N/A	Style transfer is powerful and cool
N/A	The colors are often very similar between different features,
N/A	How can users edit the image or improve on the image after it is generated?
N/A	AI can only get users 80 of the way there, how can we give users power to pick up after that?
N/A	The power to undo is important
N/A	Editing capability is important
N/A	Clearly defining the options of features to paint is helpful in understanding the limitations

C.3.2 Free-Form Composition Tool

Severity	Issue Found
N/A	No user journey in how to use the tool (same issue with Gaugan)
N/A	Should be ways to clear whole grid input
N/A	Screenshot and GIF are confusing
N/A	It's too unclear how to use
N/A	A lot of the issues experienced are due to the limitations of the underlying models
N/A	Prefer paint brush because it's easier to visualize the outcome of the image
N/A	Requires user to come up with composition has higher cognitive load

C.4 Expert 4

C.4.1 Nvidia's GauGan

Severity	Issue Found
1	Categorization - not always sure where to look for certain objects ex: Thought grass would fall under landscape
2	Cue after action is hard to see - when using the dropper tool I was not sure where to look when I chose a color
2	How to create - not sure if I have to draw out a whole hill or tree or if dots suffice

3	The button that clears the board doesn't look like it is supposed to clear board looks more like "add new board". I was surprised when it cleared.
2	When using the fill tool the arrow was confusing just because I am used to seeing a paint can (other software have that)
1	Assumed that "roof" would be included with "house"
3	No visuals next to the options (i.e. no cloud next to cloud)

C.4.2 Free-Form Composition Tool

Severity	Issue Found
1	Initially unsure if every cell has to be filled out
3	Not immediately sure of what can be typed in boxes
2	Image loading icon was not immediately noticeable was not sure if it grid was submitted
1	Unsure of how number ties back to resolution

C.5 Expert 5

C.5.1 Nvidia's GauGan

Severity	Issue Found
5	Poor visibility of system status. From the start, there aren't clear indicators on the amps to show the user where they are or what steps to go next

5	Wording could be more intuitive and recognizable to the user. Segmentation Map & Custom style filter in particular may not mean much to the general user even if they are an artist
4	Poor user control because again there isn't any guidance. The initial checkbox makes it limiting & confusing to start with. It puts a level of constraints on the user.
5	Consistency on the page is fine, but the overall interface could be more modern.
5	Poor error prevention from start to finish. On my first attempt, it completely did not load the image after I drew. There was no guidance or alert either that something had gone wrong
5	Busy interface with tons of texts, not a clear information architecture, and calls to action are not evident to the user.
3	Very little documentation to help user when roading into roadblocks and not intuitive on how to start.

C.5.2 Free-Form Composition Tool

Severity	**Issue Found**
4	Should have had some initial documentation or instruction to help guide user who is unfamiliar with the interface. This would have been helpful to also include some sort of key with valid inputs.
3	Resolution & Input aren't necessarily intuitive to me as a user
2	The interface is a lot more modern learning and not as cluttered. It is minimal which is helpful & more easy to navigate
4	Poor error prevention, could provide an indicator on the type of error and how the user should prevent it

2	Consistency on the interface is fine.

Appendix D

Vocabulizer

In this notebook I develop a simple method for semantically projecting (almost) any short English phrase into its most similar item in a tiny given vocabulary. You might use this to allow open-vocabulary inputs to a neural model that was only trained to handle a small collection of specific terms.

In [ ]:

```
import tensorflow as tf
import tensorflow_hub as hub
```

In [ ]:

```
%%time
embed = hub.load("https://tfhub.dev/google/Wiki-words-250/2")
```

```
CPU times: user 9.55 s, sys: 3.24 s, total: 12.8 s
Wall time: 21.1 s
```

In [ ]:

```
class Vocabulizer(tf.keras.layers.Layer):
  def __init__(self, embed, terms, gamma=5.0, **kwargs):
    super().__init__(**kwargs)
    self.embed = embed
    self.terms = tf.constant(terms)
    self.embedded_terms = embed(self.terms)
    self.gamma = gamma

  def call(self, input, mode='hard'):
    input = tf.identity(input)
    embedded_inputs = self.embed(tf.reshape(input, [-1]))
    distances = tf.reduce_sum((tf.expand_dims(embedded_inputs,1) - self.embedded_terms)**2,-1)
    if mode == 'hard':
      indexes = tf.argmin(distances, -1)
      return tf.reshape(indexes, input.shape)
    elif mode == 'soft':
      return tf.reshape(tf.nn.softmax(-self.gamma*distances,-1), input.shape+(self.terms.shape[0],))
    else:
      raise ValueError("Unknown mode " + mode)

  def lookup(self, indexes):
    return tf.gather(self.terms, indexes)
```

In [ ]:

```
standard_terms = [
  "airplane",
  "cat",
  "table",
  "guitar",
  "waste basket",
]

strange_terms = [
  "plane",
  "fighter jet",
  "airliner",
  "feline pet",
  "kitten",
  "kitty",
  "kittycat",          # likely to be out-of-vocabulary for example embedding
  "adorable kittycat", # recoverable, maybe?
  "lion",
  "puma",
  "cougar",
  "desk",
```

```
  "office chair",
  "plate",
  "easel",
  "lute",
  "erhu",
  "singing voice",
  "xylophone",
  "recycle bin",
  "dumpster",
  "trash",
]

v = Vocabulizer(embed, standard_terms)
best_indexes = v(strange_terms)
distributions = v(strange_terms, mode='soft')
best_terms = v.lookup(best_indexes)

for a, i, b, dist in zip(strange_terms, best_indexes, best_terms, distributions):
  print(f'{a:>20} -> {b.numpy().decode("utf8"):15} ({i}) ~ {dist.numpy().round(3)}')
```

```
               plane -> airplane        (0) ~ [0.926 0.024 0.039 0.002 0.009]
         fighter jet -> airplane        (0) ~ [0.999 0.    0.    0.    0.   ]
            airliner -> airplane        (0) ~ [0.999 0.001 0.    0.    0.   ]
          feline pet -> cat             (1) ~ [0.    0.999 0.    0.    0.   ]
              kitten -> cat             (1) ~ [0.003 0.991 0.001 0.001 0.003]
               kitty -> cat             (1) ~ [0.027 0.448 0.359 0.012 0.153]
            kittycat -> airplane        (0) ~ [0.229 0.229 0.229 0.229 0.083]
   adorable kittycat -> cat             (1) ~ [0.04  0.757 0.035 0.154 0.015]
                lion -> cat             (1) ~ [0.01  0.972 0.009 0.003 0.006]
                puma -> cat             (1) ~ [0.013 0.97  0.007 0.004 0.006]
              cougar -> cat             (1) ~ [0.013 0.98  0.003 0.002 0.003]
                desk -> table           (2) ~ [0.242 0.112 0.461 0.068 0.116]
        office chair -> table           (2) ~ [0.297 0.033 0.381 0.046 0.242]
               plate -> table           (2) ~ [0.087 0.065 0.575 0.009 0.263]
               easel -> table           (2) ~ [0.183 0.191 0.378 0.108 0.14 ]
                lute -> guitar          (3) ~ [0.003 0.022 0.016 0.956 0.002]
                erhu -> guitar          (3) ~ [0.005 0.012 0.011 0.969 0.003]
       singing voice -> guitar          (3) ~ [0.005 0.05  0.005 0.939 0.   ]
           xylophone -> guitar          (3) ~ [0.003 0.011 0.005 0.98  0.002]
         recycle bin -> waste basket    (4) ~ [0.035 0.064 0.069 0.01  0.822]
            dumpster -> waste basket    (4) ~ [0.109 0.353 0.059 0.029 0.449]
               trash -> waste basket    (4) ~ [0.017 0.053 0.018 0.006 0.906]
```

Bibliography

[1] Abubakar Abid, Ali Abdalla, Ali Abid, Dawood Khan, Abdulrahman Alfozan, and James Zou. Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild. *arXiv:1906.02569 [cs, stat]*, June 2019. arXiv: 1906.02569.

[2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and Stuff Classes in Context. *arXiv:1612.03716 [cs]*, March 2018. arXiv: 1612.03716.

[3] Chris Creed. Assistive technology for disabled visual artists: exploring the impact of digital technologies on artistic practice. *Disability & Society*, 33(7):1103–1119, August 2018.

[4] Mahmut Ekşioğlu, Esin Kiris, Burak Çapar, Murat N. Selçuk, and Selen Ouzeir. Heuristic Evaluation and Usability Testing: Case Study. In P. L. Patrick Rau, editor, *Internationalization, Design and Global Development*, volume 6775, pages 143–151. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. Series Title: Lecture Notes in Computer Science.

[5] Patrick Esser, Robin Rombach, and Björn Ommer. Taming Transformers for

High-Resolution Image Synthesis. *arXiv:2012.09841 [cs]*, February 2021. arXiv: 2012.09841.

[6] Susumu Harada, Jacob O. Wobbrock, and James A. Landay. Voicedraw: a hands-free voice-driven drawing application for people with motor impairments. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility - Assets '07*, page 27, Tempe, Arizona, USA, 2007. ACM Press.

[7] Robin Jeffries, James R. Miller, Cathleen Wharton, and Kathy Uyeda. User interface evaluation in the real world: a comparison of four techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, pages 119–124, New York, NY, USA, March 1991. Association for Computing Machinery.

[8] Yea-Seul Kim, Mira Dontcheva, Eytan Adar, and Jessica Hullman. Vocal Shortcuts for Creative Experts. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–14, Glasgow Scotland Uk, May 2019. ACM.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*, September 2013. arXiv: 1301.3781.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*, September 2013. arXiv: 1301.3781.

[11] Jakob Nielsen. Finding usability problems through heuristic evaluation. In *Pro-*

ceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '92, pages 373–380, New York, NY, USA, June 1992. Association for Computing Machinery.

[12] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 249–256, New York, NY, USA, March 1990. Association for Computing Machinery.

[13] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic Image Synthesis with Spatially-Adaptive Normalization. *arXiv:1903.07291 [cs]*, November 2019. arXiv: 1903.07291.

[14] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. *arXiv:2102.12092 [cs]*, February 2021. arXiv: 2102.12092.