MAGICK: A Large-scale Captioned Dataset from Matting Generated Images using Chroma Keying

Ryan D. Burgert
Stony Brook University
rburgert@cs.stonybrook.edu

Brian L. Price
Adobe Research
bprice@adobe.com

Jason Kuen
Adobe Research
jkuen@adobe.com

Yijun Li
Adobe Research
yijli@adobe.com

Michael S. Ryoo
Stony Brook University
mryoo@cs.stonybrook.edu

Abstract

We introduce MAGICK, a large-scale dataset of generated objects with high-quality alpha mattes. While image generation methods have produced segmentations, they cannot generate alpha mattes with accurate details in hair, fur, and transparencies. This is likely due to the small size of current alpha matting datasets and the difficulty in obtaining ground-truth alpha. We propose a scalable method for synthesizing images of objects with high-quality alpha that can be used as a ground-truth dataset. A key idea is to generate objects on a single-colored background so chroma keying approaches can be used to extract the alpha. However, this faces several challenges, including that current text-to-image generation methods cannot create images that can be easily chroma keyed and that chroma keying is an underconstrained problem that generally requires manual intervention for high-quality results. We address this using a combination of generation and alpha extraction methods. Using our method, we generate a dataset of 150,000 objects with alpha. We show the utility of our dataset by training an alpha-to-rgb generation method that outperforms baselines. Please see our project website at https://ryanndagreat.github.io/MAGICK/.

1. Introduction

Recent breakthroughs in diffusion models have led to an abundance of new research in text-to-image generation [22, 25, 27, 29]. Given a short text prompt, an entire image can quickly be generated. These images can contain complex foreground objects against complex backgrounds. However, they do not handle use cases in which a user may want an isolated object with an accurate alpha channel.

For example, a user may want to guide the generation of an image with not only a text prompt but also with an accurate alpha mask. While segmentation maps have been used to guide diffusion models [47], these maps are rough and do not contain precise details like human hair or transparencies like in a wine glass. Our attempts to use alpha to condition the off-the-shelf ControlNet v1.1 yielded results that did not follow the alpha channel (Fig. 1). Other applications could include generating objects to insert into images and generating training data for matting datasets.

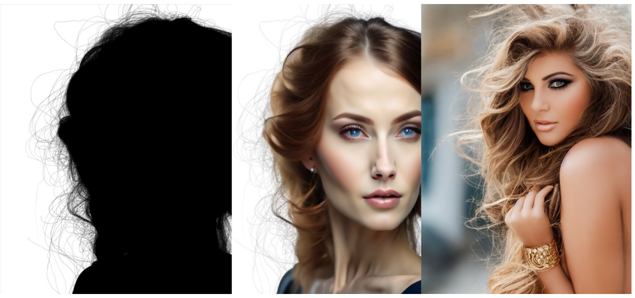


Figure 1. An alpha matte (inverted for visibility) is used to generate an rgb image using ControlNet trained by our dataset (center) verses the original (right). Note our version trained with our data closely follows the alpha matte.

The inability of current methods to address generation involving alpha may be due to the lack of training data. While many large-scale segmentation datasets exist [6, 12, 18, 23, 50], they do not contain accurate soft boundaries, usually because they were segmented manually using boundary-tracing tools. Matting datasets contain high-quality alpha ground-truth but are too small for training generation methods due to the difficulty in obtaining ground-truth alpha. For example, the alphamatting.com dataset [26] only contains 27 images, the Adobe Deep Image Matting (DIM) dataset [44] contains 431 objects, and the Semantic Matting dataset [35] expands DIM to 726 objects. Without a suitable large-scale alpha dataset, training models with accurate boundaries will remain difficult.

To address this lack of data, we proposed MAGICK, a large-scale dataset of generated objects with high-quality alpha mattes for use in training future generation models. MAGICK contains 150,000 generated objects across a wide range of object types and includes masks with significant soft edges or transparencies. To create this dataset, we generate objects on green screen (or other solid colored) backgrounds and extract the objects using chroma keying. Several significant challenges complicate this approach: 1) an appropriate background color must be selected (generating a green leaf on a green background will yield a poor result), 2) current diffusion-based methods are poor at generating objects on a green screen, producing images with little detail around the edges or producing background that are not constant colored (Fig. 5), and 3) chroma keying is an underconstrained problem, often failing to produce high-quality alpha without manual post-processing.

We overcome these challenges with the following approach. First, for a given prompt, we generate a sample object to determine its primary colors, then select the color least present in the object for the background color. We then use two generation models to produce a suitable image for chroma keying: the text-to-image generation method DeepFloyd [1] which generates solid colored backgrounds but foregrounds with poor alpha detail, then the image-to-image generation method proposed in [20] with SDXL [22] that can generate a good foreground with alpha detail given the output of DeepFloyd. Finally, we use a combination of chroma-based and deep-learning based keyers and segmenters to extract the object from the image accurately. This process is shown in Fig. 3.

Given this approach, we produced a dataset of 150,000 images (to be released upon publication), a sample of which is shown in Fig. 2. To show the utility of MAGICK, we train a model on one of the many applicable tasks, alpha-to-image generation. We fine-tune ControlNet [47] with our dataset to generate RGB images given an input alpha and show improved performance over using the pretrained ControlNet (Fig. 1), showing the utility of our dataset.

2. Related Work

Synthetic segmentation data generation: Many methods have recently been proposed to synthetically generate segmentation data. Early methods utilize GANs for data synthesis. DatasetGAN [13] proposes to decode GAN latent codes to generate segmentation data, primarily of specifc object parts or of limited scenes like bedrooms. BigDatasetGAN [49] produces masks for single primary objects by training a GAN on Imagenet [28].

Diffusion-based model typically take a text prompt as input, then simultaneously synthesize an image along with a mask. The mask may be of a single primary object [16, 42] or a semantic segmentation within the domain of an existing hand-labeled dataset [21, 41]. Variations of this theme include generating multiple images and object masks at once [43] or generating the masks first and then the image [45]. Peekaboo [3] generates single objects, the most closely related method to our own, but the results lack details like hair and fur. While arguments can be made against training with generated data [2, 33], these works show that training with synthetic data, either alone or in conjunction with real data, yields results that are on par with or surpass the state-of-the-art set by methods trained with real data.

A drawback of these methods is they all focus on binary masks and lack transparencies and fine details such as hair or fur. Our method specifically targets generating images with alpha mattes that exhibit fine details.

Alpha matting datasets: While our method is the first to generate images with alpha masks, many alpha matting datasets already exist. Generating alpha mattes is a difficult task, requiring complicated image capture methods and/or excessive user interaction, resulting in the existing matting datasets being small in size. The early matting dataset from alphamatting.com [26] was generated using triangulation matting, a tedious process of photographing the same object against multiple backgrounds, resulting in a dataset of 27 training images and 8 test images. This was extended to video to produce a small number of frames using stop-motion photography [8].

Manual extraction of the alpha matte from photographs using existing matting methods has been used to create several datasets [32, 35, 38, 44]. However, this approach is very time consuming and prone to error, generally resulting in only a few hundred objects.

Video matting datasets often use chroma keying [34] to generate alpha data [17, 48]. However, high quality chroma keying requires both careful setup of the green (blue) screen and lighting as well as manual post-processing to tweak parameters and manually correct or mask out mistakes.

While these methods can produce high-quality alpha mattes, they are difficult and tedious to collect. This contrasts our approach that can produce large numbers of accurate alpha mattes with the corresponding images with minimal user interaction.

Segmentation and Matting: Arguably, an alternative method of producing a dataset like ours would be to simply extract objects from generated images with standard segmentation/matting methods without bothering to produce them on green screens. Such generated images would contain background details and potentially other foreground objects that would need to be separated from the object.

While many segmentation datasets exist [6, 12, 18, 50], as do many methods for segmenting salient objects [14, 15, 24] or multiple objects [9, 11, 23] from images, these methods would not yield the accurate alpha mattes that our method produces. Matting methods [4, 7, 19, 30, 36, 44], despite significant progress recently, are still imperfect and would include errors or artifacts in the alpha mattes. Indeed, we hope our dataset will be used to improve matting methods in future works.

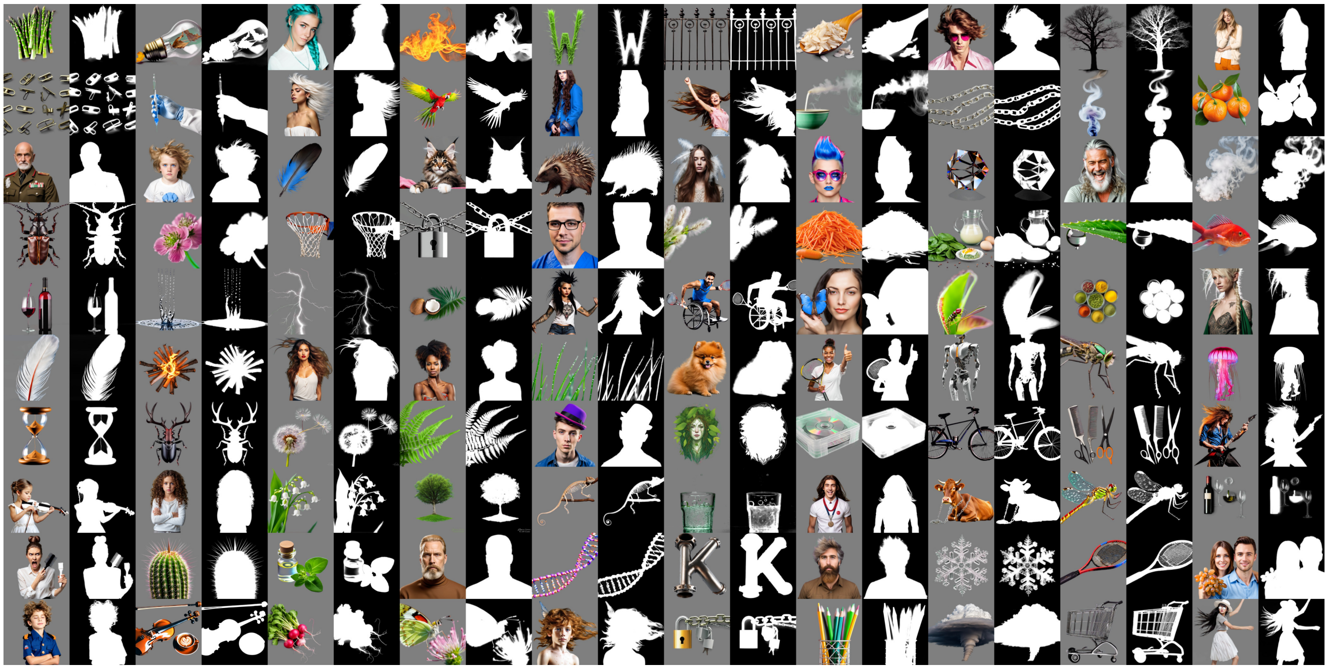


Figure 2. 100 image and alpha samples from our dataset. Please zoom in to see all the details. Please see the appendix for more examples.

3. Dataset

The MAGICK dataset is a collection of 150,000 objects extracted from generated images. Each object consists of 1) an image of the object with pure foreground colors (i.e. no background color is mixed into edge and transparent pixels), 2) the alpha matte of the object, and 3) the caption used to generate the object. The wide variety of objects is not restricted to any small set of given object classes. Many of the objects exhibit details that require a detailed matted such as hair, fur, thin parts, or transparencies.

Fig. 2 shows 100 examples from our dataset. As can be seen, the object types vary widely and the alpha mattes contain accurate details.

Generating a dataset like MAGICK is non-trivial. No existing generation model can produce accurate images with alphas. Datasets with accurate alpha are limited, making it difficult to train such a method. While existing segmentation or matting methods could be applied to generated images to extract objects, such methods are imperfect and would not yield results suitable for representing ground-truth for training. To accomplish this, we needed to use a combination of generation and alpha extraction methods.

3.1. Dataset Creation

To create MAGICK, we generate objects on a green screen (or other constant colored screen) so that ground-truth quality results can be extracted. While chroma keying from green screen footage is common, doing this at scale with generation faces several challenges. In addition to requiring a large set of prompts, we must have a way to select the background color automatically, to generate a object on a colored background that is suitable for chroma keying, and to extract the alpha automatically. We propose a method to overcome these challenges.

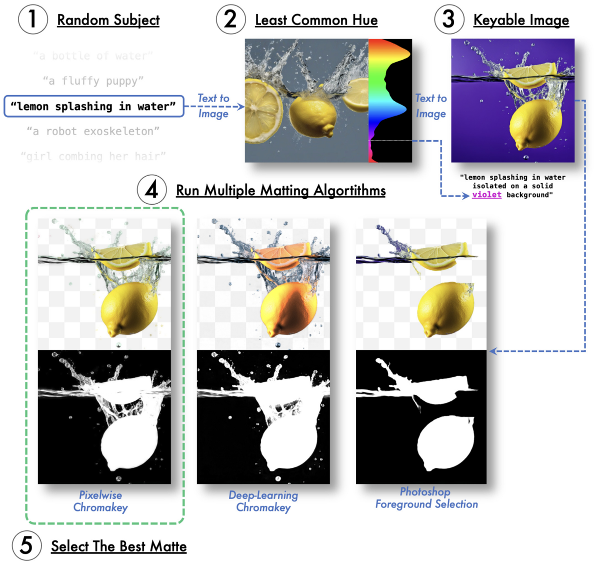


Figure 3. A general overview of our pipeline for image generation.

The overview is shown in Fig. 3. First, a prompt is chosen. Then, a suitable color for the background is chosen by generating an image from that prompt and analysing its distribution of hues. Next, a *keyable* image, or image that is suitable for chroma keying, is created by chaining together two diffusion models, DeepFloyd [1] and SDEdit [20] with SDXL [22]. Finally, multiple alpha extraction methods are run and the best matte is chosen.

3.1.1 Selecting Prompts

The first step in generating our large synthetic image dataset is to come up with a list of prompts. As we will be engineering the prompts with additional descriptions in later steps, we will refer to the part of the prompt that describes the object as the "subject". The subject must highlight and meaningfully describe a single subject in each image without mentioning other objects or background details so that it can be easily isolated from the background.

We obtain the subjects in our dataset from three sources: 1) Outputs from LLMs such as GPT-4 and ChatGPT, 2) Procedurally generated subjects for humans, and 3) Captions from an existing image-caption datasets

LLM-Generated Prompts Recent large language models (LLMs) have proven capable of generating text given a prompt. We leverage this by using ChatGPT and GPT4 to create subject prompts. We instruct the LLMs to write a descriptive caption of an object in an image without describing the background or other objects. We provide LLMs with a list of object categories, both general objects and objects with details that require complex mattes like hair, fur, or transparent parts, but also allow the LLMs to extrapolate their own categories for additional variety.

Examples of captions created using LLMs include "`A detailed macro shot of a butterfly wing`" and "`A piece of amber glass reflecting sunlight.`" Additional examples and the LLM prompt we use can be found in the appendix.

Procedurally Generated Prompts Because humans are an important part of our dataset, we created a template mechanism for procedurally constructing descriptions of humans. We focused on diversity, attempting to capture many different professions, ethnicities, clothing, accessories, genders and hairstyles. Some example subjects are "`lawyer woman diamond earrings`", "`person wearing gown`", and "`hispanic barista man with black flowing hair`". More examples can be found in the appendix.

Image Captions While many image captioning datasets exist [5, 31, 46], their captions typically describe the scene and potentially multiple objects, such as the caption "`A large bus sitting next to a very tall building`" from [5]. Such captions are not suitable for our needs as identifying the subject is difficult.

We instead extract subjects from a proprietary image-caption dataset. In addition to the full scenes present in prior datasets, this dataset also includes images of isolated objects. The captions of such objects often contain identifying words such as "clipping path", "green-screen", or "on a white background". We search for such tags and remove them from the caption. This yields descriptive subjects such as "'`Close-up of a new basketball ball`" and "`White and brown chicken wings`". More examples can be found in the appendix.

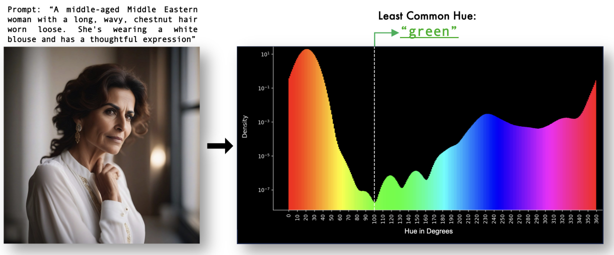


Figure 4. Finding the least common hue of a given subject. In this example, "green" is least common.

3.1.2 Finding the Least Common Hue

For our approach, we must generate images with solid colored backgrounds for chroma keying. To do this, we must choose a background color that does not conflict with the subject, as that would render any chroma keying algorithm useless. It is well-known that wearing a green shirt against a green screen will causes your torso disappear in the output.

To find an appropriate background color for a given subject, we follow the procedure depicted in Fig. 4. First, for a given subject, we generate an image using SDXL with the subject (unmodified) as the prompt. While this generated image will not be on a solid colored background as we need for alpha extraction, it will typically show the color distribution for that given subject. We create a histogram of the hues of the pixels in the generated image weighted by saturation, then smooth the hue histogram with a Gaussian kernel with $\sigma = 10°$. We quantize the histogram into regions representing named colors (e.g. green, blue, etc) and return the color name as a string. This string representing the hue will be used to generate the image in the next step.

We found that "green" and "blue" are by far the most common background colors in our dataset, totaling over 90% of images. This matches practical experience where objects to be chroma keyed are almost always shot against a green or blue screen. This is also partially due to the wide band of hues that these colors cover.

3.1.3 Generating *Keyable* Images

Give a subject and a background color, we can now generate an image from which we will extract the alpha. For this to work, the image must be *keyable*. For an image to be

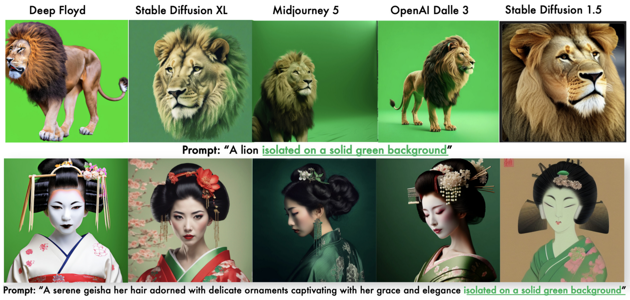

Deep Floyd
Stable Diffusion XL
Midjourney 5
OpenAI Dalle 3
Stable Diffusion 1.5
Prompt: "A lion isolated on a solid green background"
Prompt: "A serene geisha her hair adorned with delicate ornaments captivating with her grace and elegance isolated on a solid green background"

Figure 5. We compare results from multiple diffusion models. Each algorithm produces images that are not *keyable*, either the foreground has no fine detail or is tinted green or the background is not suitable for chroma keying. Please zoom in for details.

keyable, it 1) must have a constant, bright, saturated background color, 2) must not have any objects or gradients in the background, 3) must have fine details like hair, fur, and transparencies in the object when appropriate, and 4) must not have color spill, or background colors tinting the foreground.

Unfortunately, all the publicly available methods we tested were incapable of consistently creating *keyable* images. As shown in Fig. 5, SDXL, Midjourney, Dalle, and Stable Diffusion often produce backgrounds that are not suitable for extraction (being too dark, desaturated, or having gradients or objects) or have color spill. DeepFloyd is quite good at generating vibrant clean backgrounds and foreground with no color spill but fails to produce soft edges (see the lion's mane in Fig. 5).

Because of these shortcomings, we propose combining multiple generation methods (namely DeepFloyd and SDEdit) with the goal of overcoming these weaknesses through the combination. Our process is described below.

Prompt engineering We first augment the subject to indicate the background color. For example, if the background color were "green", we augment the subject with the phrase "`isolated on a solid green background`".

Image generation To create a *keyable* image, we use both the text-to-image generation method DeepFloyd followed by the image-to-image generation methods SDEdit. This process is illustrated in Fig. 6.

First, we use our prompt with DeepFloyd to generate an initial image. We found that DeepFloyd could consistently produce suitable backgrounds but the foregrounds were lacking detail in the alpha. However, the backgrounds are often not as bright and saturated needed for SDEdit. To address this, we perform an initial extraction of the object and composite it onto a brighter, more saturated background with the approximately same hue. We use Photoshop's Subject Selection feature, a method that uses deep-learning-based segmentation and matting to compute masks for primary objects in an image and can be run in batch mode. This method does not perfectly extract the object and may leave green pixels in regions like hair, but these typically do not cause a problem as the object is immediately composited onto a similar colored background.

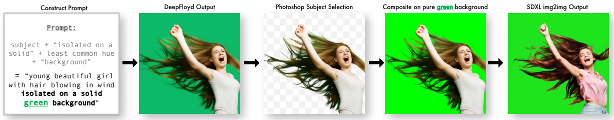

Construct Prompt
Prompt:
subject + "isolated on a solid" + least common hue + "background"
= "young beautiful girl with hair blowing in wind isolated on a solid green background"
DeepFloyd Output
Photoshop Subject Selection
Composite on pure green background
SDXL img2img Output

Figure 6. We generate the RGB images in our dataset using a combination of both DeepFloyd, Photoshop's Subject Selection feature, and Stable Diffusion XL. The resulting images have near-perfect, high saturation backgrounds that are ideal for chroma keying. Please zoom in for details.

We then use SDEdit [20], the image-to-image version of SDXL implemented by Huggingface [10], to regenerate the final image. In this step, we need to guarantee that the object has fine details, that it has no color spill, and that the background is solid, bright, and saturated enough for chroma keying. To produce fine details, we set SDEdit's strength parameter to .95. This parameter, ranging from 0 to 1, determines how closely the generation follows the input image by modulating the amount of noise added to the image. This allows enough freedom for SDEdit to generate a new version of the same object with better details, not only at the edges but often in the interior of the object as well. To prevent color spill, we add the background color as a negative prompt. This suppresses that color in the foreground object, but because the background is so bright and saturated from the previous step it fails to suppress the color in the background.

Fig. 7 shows the impact of this process. The images in the column "Before Img2Img" are the outputs from DeepFloyd after being matted by Photoshop Subject Selection and composited onto a solid background, and the "After-Img2Img" column are the outputs of SDEdit. Note the fidelity of the image increases and any color spill corrected due to the negative prompt. (e.g. the submarine's green tint or the girl's green dress). Also note the mistakes made by Photoshop Subject Selection are corrected as well - the dandelions look poorly extracted before SDEdit but look natural after.

Despite these efforts to make the images as *keyable* as possible, our process is imperfect, sometimes yielding results with gradients in the background color or tinting of the foreground object. To deal with this, we require a robust alpha computation method.

3.1.4 Alpha extraction

Once we have a *keyable* image, we must extract the alpha from the image. Unfortunately, this is also a difficult pro-

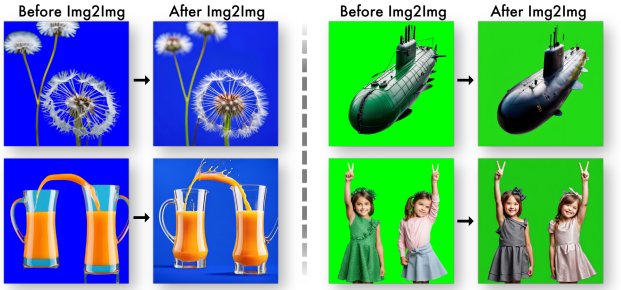


Figure 7. Examples of image before and after applying image-to-image SDEdit. The before images are outputs of DeepFloyd after being extracted by Photoshop Subject Selection and composited onto a new background. Please zoom in for details.

cess. Chroma keying is an underconstrained problem. Even with a solid background color, chroma keying methods can fail to correct compute the alpha, and in practice users must manually change parameters or correct the mattes for a high quality result. This time-consuming process does not scale to large datasets.

To address this, we generate three alpha mattes and choose between them. These three methods operate differently and in difficult cases one algorithm may compute accurate results when the other cannot. The algorithms we use are:

1. *A pixel-based chroma key method*. We modified a traditional color difference chroma key algorithm [40] that takes a single rgb color representing the background as input to instead take a background rgb color per pixel. We conservatively delete the foreground object and inpaint the background using [37] to provide the background color at each pixel. This allows us to better handle subtle gradients in the background color. Note that this also performs color decontamination in the same step.

2. *A deep-learning based chromakey model* that was trained on an internal dataset and takes in an input RGB image and a background RGB image and returns the alpha and foreground color.

3. *Photoshop's Subject Selection* which uses proprietary segmentation and matting algorithms to select the primary object in an image. It works well on images with simple backgrounds and is robust to color spill.

Fig. 8 compares the three methods on three examples, highlighting cases where the methods have inconsistent results. In such cases, one of the methods is able to compute an accurate alpha.

Recalling the image matting equation:

$$I = \alpha F + (1 - \alpha)B, \quad (1)$$

we require not just the alpha but also F, the pure foreground color of the pixel with any background color B removed. Each of our three alpha extraction methods generate a predicted F. However, the method that predicts the best alpha does not necessarily also produce the best F. Experimentally, we chose the best seven combinations as possible choices for the final alpha and F.

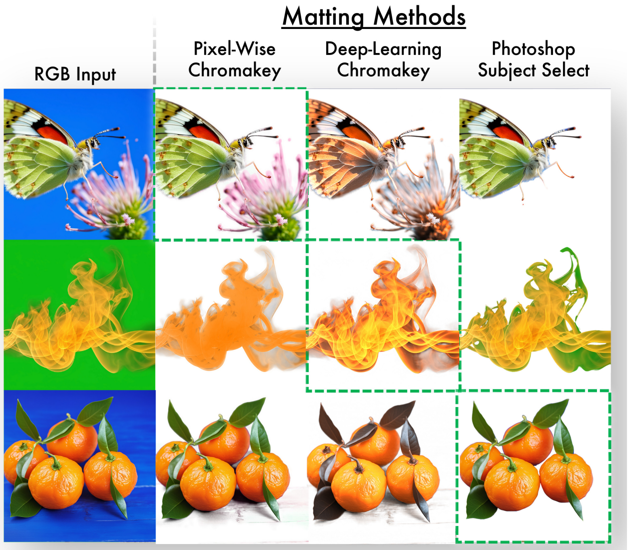


Figure 8. A comparison of our three matting methods on different input images. The green dotted rectangles indicate the best result for each example.

3.1.5 Image Selection Process

The last step in our dataset creation pipeline is to select the best matte. Each subject will have multiple matted results, and the goal of this step is to choose the best one. We propose a simple automatic process that can approve a large number of the images. For those failing the automatic process, we fall back to having humans select the best option.

Automatic Selection Process Each of our alpha extraction methods operate differently, with one being color-based, one being trained for computing alpha from green-screen images, and one being designed for general object extraction. Because of this, they tend to make different mistakes. However, for cases where our method successfully produced a *keyable* image, the three alpha extraction methods often produce nearly identical results. We use this as an indication that the alpha computation was successful and each of the resulting alphas are good.

To measure the similarity, we devise a similarity score metric taking into account both their alpha and RGB values - but not penalizing differences in RGB values if the alpha values are both low. To do this, we compare both RGBA images composited on different backgrounds (white and black) and take the mean similarity between these two composite images. We measure this similarity using MSS-SIM [39] (multi-scale structural image similarity) as we

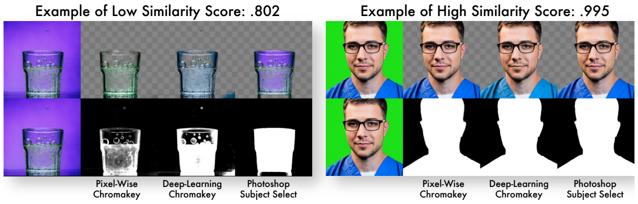


Figure 9. A qualitative comparison of a high and low similarity score. Note how the alpha masks and colors are different between samples on the one with low similarity, but nearly identical on the one with high similarity.

found it gave the best results empirically.

MSSSIM is a metric that measures similarity between images on a scale from 0 to 1, assuming the pixel values are also between 0 and 1. Given our three RGBA images I_0, I_1, and I_2, a white image W, a black image B, the composition function $\mathcal{C}$ and the MSSSIM function $\mathcal{M}$, we compute our similarity score $\mathcal{S}$ as:

$$\begin{aligned} \mathcal{S} &= min_{(a,b)}\mathcal{F}(I_a, I_b) \quad \text{where } a,b \in \{0,1,2\} \\ \mathcal{F}(I_a, I_b) &= \frac{1}{2}(\mathcal{M}[\mathcal{C}(I_a, W), \mathcal{C}(I_b, W)] \\ &\quad + \mathcal{M}[\mathcal{C}(I_a, B), \mathcal{C}(I_b, B)]) \end{aligned} \tag{2}$$

Fig. 9 shows a visual comparison between images with a high and a low similarity score.

Fig. 10 shows the distribution of similarity scores. Most images have very high similarity scores, indicating our process to make the images *keyable* was largely successful. The median score is 0.984. We found that the top 50% of samples (measured by similarity score) yield decent results, resulting in 110,000 images in our dataset being automatically selected. These objects tend to be solid objects including objects with hair or fur, and tend to not be objects with significant transparencies. For these automatically-selected images, we choose the pixel-wise chroma keying algorithm as it often gives the highest detail.

Manual Selection Process For images that do not fall above the threshold of automatic selection, we rely on humans to select the best alpha for us. These tend to be subjects that contain difficult transparencies such as glass, water, smoke, or fire. We acquired 40,000 images using manual selection of the computed mattes.

We've created a program that will be released to the public along with our dataset to aid in manual selection. It presents multiple combinations of alpha and F and allows changing the background colors and zooming for accurate assessment of details, as well as additional features such as tagging images. It also serves as an efficient way to quickly view and audit the dataset. See the Appendix for details.

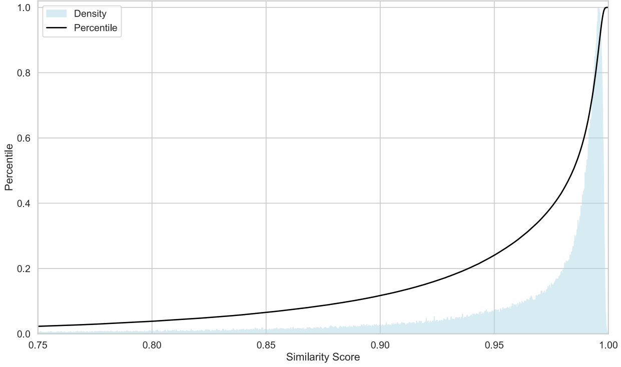


Figure 10. The distribution of similarity scores in the dataset.

4. Application: Alpha-to-RGB Generation

This dataset has many potential applications It could be used to train a direct RGBA generation method or for training improved matting methods. To illustrate the usefulness of the dataset, we investigated the application of alpha-to-RGB generation. Given an alpha matte and a prompt as input, we train a model to generate an accompanying RGB image.

Method We trained ControlNet [47] with Stable Diffusion 1.5 (SD1.5) on our dataset. As the output is RGB, we must decide on a background for our target images. We chose to composite our objects onto gray backgrounds. The pure foreground color F can easily be derived with:

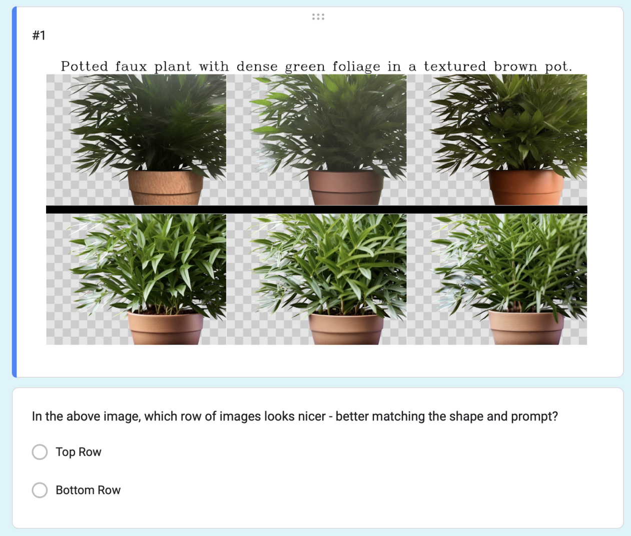
$$F = \frac{1}{\alpha}(I - G) + G \tag{3}$$

where G is the color of the gray background. Only G needs to be estimated to compute Eq. (3) which is trivial as the G is nearly constant. Because gray is a neutral color, any small errors in estimating G will not shift the hue of F.

We used ControlNet's default settings for training. For testing, we also use the default settings except we set our guidance scale to 7.5 and our control strength to 1.2 as we found empirically this generates better results.

Baselines We are unaware of any baselines that directly take alpha and produce a corresponding RGB with adherence to detailed edges. SegGen [45] proposes a mask-to-image model, but it assumes multiple objects and does not produce matted details. While ControlNet v1.1 can take segmentation masks as guidance, it was trained on ADE20K [50] and so has a limited set of classes it covers and its adherence to the matted details is too poor to provide a meaningful comparison (see Fig. 1 top).

Instead, we convert our mask into edges and compare to ControlNet using canny edges and sketch edges as guidance as proposed in [47]. For fair comparison, we use SD1.5, the same base model we trained on our dataset.



#1

Potted faux plant with dense green foliage in a textured brown pot.

In the above image, which row of images looks nicer - better matching the shape and prompt?

○ Top Row

○ Bottom Row

Figure 11. Interface for user study.

SD guidance	User preference SD1.5	 Ours
Canny Edges	.16	**.82**
Sketch Edges	.23	**.77**

Table 1. Results of user study on alpha-to-rgb generation.

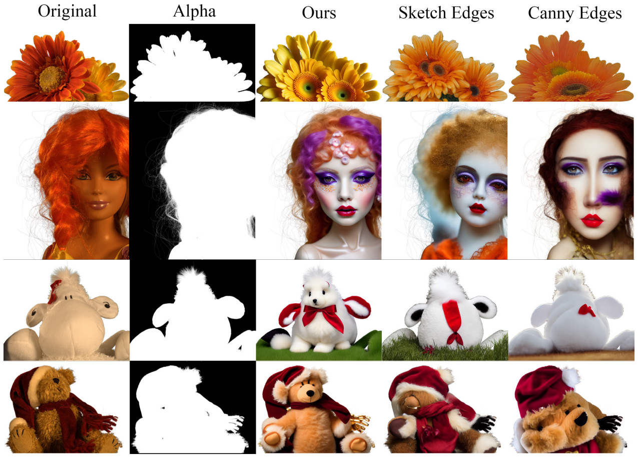


Figure 12. Generation results from our user study. The original images were the original images from [26], shown as reference and not used in the study. Given the alpha values and captions (not shown), images were generated using SD1.5 trained with our dataset, SD1.5 Sketch Edges, and SD1.5 Canny edges.

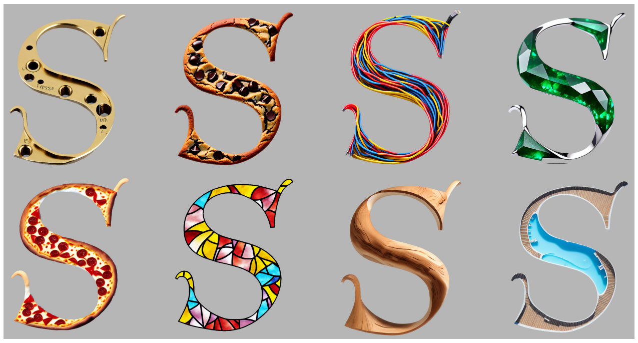


Figure 13. Example of alpha-to-rgb generation. The letter "S" is generated using different prompts to generate stylized text.

Experiment We used the 27 alpha masks from the alphamatting.com training set [26] as the test set for our experiment. This dataset contains interesting, complex alpha mattes that were not seen by any of the algorithms before testing. Prompts were generated for the images using GPT4. Each method generated three RGB images for each of the 27 examples given their alpha mattes and prompts.

Results We asked 52 participants to rate our results verses those from SD1.5 Canny edges and sketch edges. As shown in Fig. 11, the users were presented with the prompt and two row of images, with each row showing results from one randomly chosen method. The users were asked to select the better row of images according to their appearance and adherence to the prompt. As shown in Table 1, our results were preferred 82% of the time over SD1.5 Canny Edges and 77% of the time over SD1.5 Sketch edges. Fig. 12 shows example results from our experiment. Despite differences from the original image (shown for reference and not used in the study), the model trained with our dataset is able to create aesthetically pleasing objects that follow the given alpha. The captions and results from all three methods are shown in the Appendix.

Fig. 13 shows examples of our alpha-to-rgb generation. The mask of the letter "S" was given to our model along with eight different captions to create a number stylized glyphs. The shapes of the generated letters conform to the input mask. Despite not being explicitly trained for glyph generation, the model produced a variety of aesthetically pleasing results. Surprisingly, several results show consistent 3d effects such as realistic extrusion or shadowing (e.g. the top left and bottom right examples). Semantic features also emerged, such as the cookie and pizza examples both showing overcooking along the edges of the glyphs but not the interiors as can happen with real food.

5. Conclusion

We present MAGICK, a novel dataset consisting of 150,000 generated objects with accurate alpha mattes. The dataset covers a wide variety of objects and has high quality mattes with details such as hair, fur, thin parts, and transparencies. We hope this dataset will be useful for future research, such as for training rgba generation, alpha-to-rgb, or natural image matting networks.

References

[1] Deepfloyd. https://github.com/deep-floyd/if. 2, 4

[2] Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard G. Baraniuk. Self-consuming generative models go mad, 2023. 2

[3] Ryan Burgert, Kanchana Ranasinghe, Xiang Li, and Michael S. Ryoo. Peekaboo: Text to image diffusion models are zero-shot segmentors, 2023. 2

[4] Shaofan Cai, Xiaoshuai Zhang, Haoqiang Fan, Haibin Huang, Jiangyu Liu, Jiaming Liu, Jiaying Liu, Jue Wang, and Jian Sun. Disentangled image matting. In *ICCV*, 2019. 2

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server, 2015. 4

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 1, 2

[7] Yutong Dai, Brian Price, He Zhang, and Chunhua Shen. Boosting robustness of image matting with context assembling and strong data augmentation. In *CVPR*, 2022. 2

[8] Mikhail Erofeev, Yury Gitman, Dmitriy Vatolin, Alexey Fedorov, and Jue Wang. Perceptually motivated benchmark for video matting. In *BMVC*, 2015. 2

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2

[10] Hugging Face. Image-to-image - stable diffusion. https://huggingface.co/docs/diffusers/api/pipelines/stable_diffusion/img2img, 2023. Accessed: 2023-11-16. 5

[11] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 2

[12] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 1, 2

[13] Daiqing Li, Huan Ling, Seung Wook Kim, Karsten Kreis, Sanja Fidler, , Antonio Torralba Torralba, and Sanja Fidler. Datasetgan: Efficient labeled data factory with minimal human effort. In *CVPR*, 2021. 2

[14] Guanbin Li, Yuan Xie, Liang Lin, and Yizhou Yu. Instance-level salient object segmentation. In *CVPR*, 2017. 2

[15] Yin Li, Xiaodi Hou, Christof Koch, James M. Rehg, and Alan L. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014. 2

[16] Ziyi Li, Qinye Zhou, Xiaoyun Zhang, Ya Zhang, and Yanfeng Wang. Open-vocabulary object segmentation with diffusion models. In *ICCV*, 2023. 2

[17] Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L Curless, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Real-time high-resolution background matting. In *CVPR*, 2021. 2

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Peronaand Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. *ECCV*, 2014. 1, 2

[19] Hao Lu, Yutong Dai, Chunhua Shen, and Songcen Xu. Indices matter: Learning to index for deep image matting. In *ICCV*, 2019. 2

[20] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 2, 4, 5

[21] Quang Nguyen, Truong Vu, Anh Tran, and Khoi Nguyen. Dataset diffusion: Diffusion-based synthetic dataset generation for pixel-level semantic segmentation, 2023. 2

[22] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv*, 2023. 1, 2, 4

[23] Lu Qi, Jason Kuen, Tiancheng Shen, Jiuxiang Gu, Wenbo Li, Weidong Guo, Jiaya Jia, Zhe Lin, and Ming-Hsuan Yang. High quality entity segmentation. In *ICCV*, 2023. 1, 2

[24] Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *ECCV*, 2022. 2

[25] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 1

[26] Christoph Rhemann, Carsten Rother, Jue Wang, Margrit Gelautz, Pushmeet Kohli, and Pamela Rott. A perceptually motivated online benchmark for image matting. In *CVPR*, 2009. 1, 2, 8

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bj¨orn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1

[28] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 2

[29] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 2022. 1

[30] Soumyadip Sengupta, Vivek Jayaram, Brian Curless, Steve Seitz, and Ira Kemelmacher-Shlizerman. Background matting: The world is your green screen. In *CVPR*, 2020. 2

[31] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018. 4

[32] Xiaoyong Shen, Xin Tao, Hongyun Gao, Chao Zhou, and Jiaya Jia. Deep automatic portrait matting. In *BMVC*, 2016. 2

[33] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget, 2023. 2

[34] Alvy Ray Smith and James F Blinn. Blue screen matting. In *Siggraph*, 1996. 2

[35] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. Semantic image matting. In *CVPR*, 2021. 1, 2

[36] Jingwei Tang, Yagiz Aksoy, Cengiz Oztireli, Markus Gross, and Tunc Ozan Aydin. Learning-based sampling for natural image matting. In *CVPR*, 2019. 2

[37] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9(1): 23–34, 2004. 6

[38] Tiantian Wang, Sifei Liu, Yapeng Tian, Kai Li, and Ming-Hsuan Yang. Video matting via consistency-regularized graph neural networks. In *ICCV*, 2021. 2

[39] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, 2003. 6

[40] Steve Wright. *Digital compositing for film and video*. Taylor Francis, 2010. 6

[41] Weijia Wu, Yuzhong Zhao, Hao Chen, Yuchao Gu, Rui Zhao, Yefei He, Hong Zhou, Mike Zheng Shou, and Chunhua Shen. Datasetdm: Synthesizing data with perception annotations using diffusion models. In *NeurIPS*, 2023. 2

[42] Weijia Wu, Yuzhong Zhao, Mike Zheng Shou, Hong Zhou, and Chunhua Shen. Diffumask: Synthesizing images with pixel-level annotations for semantic segmentation using diffusion models. In *ICCV*, 2023. 2

[43] Jiahao Xie, Wei Li, Xiangtai Li, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Mosaicfusion: Diffusion models as data augmenters for large vocabulary instance segmentation, 2023. 2

[44] Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In *CVPR*, 2017. 1, 2

[45] Hanrong Ye, Jason Kuen, Qing Liu, Zhe Lin, Brian Price, and Dan Xu. Seggen: Supercharging segmentation models with text2mask and mask2img synthesis, 2023. 2, 7

[46] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2014. 4

[47] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 1, 2, 7

[48] Yunke Zhang, Chi Wang, Miaomiao Cui, Peiran Ren, Xuansong Xie, Xian-Sheng Hua, Hujun Bao, Qixing Huang, and Weiwei Xu. Attention-guided temporally coherent video object matting. In *ACM MM*, 2021. 2

[49] Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations. In *CVPR*, 2022. 2

[50] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. *CVPR*, 2017. 1, 2, 7