Veggie World!

Rohan Mathur, Ryan Tabrizi, DALLE

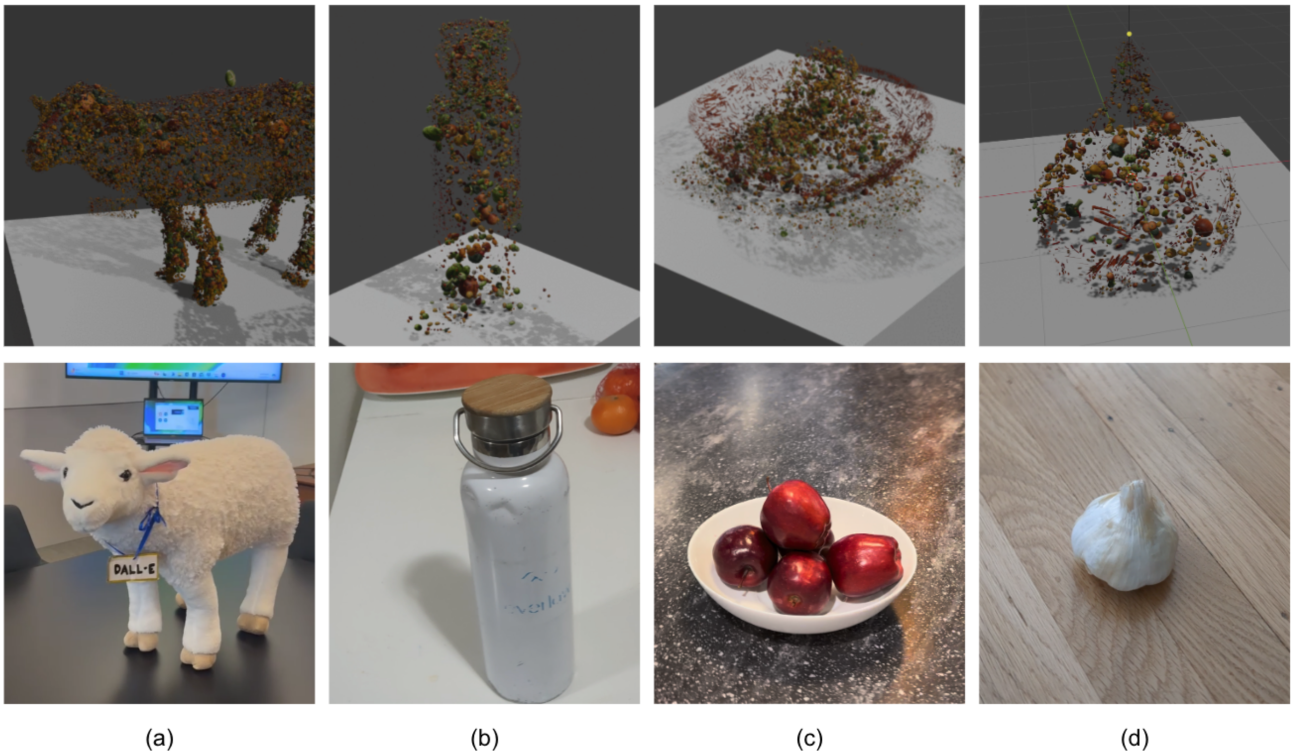

Figure 1. **Veggie World.** Veggie World lets you Veggify 3D scenes using Gaussian Splatting and without any diffusion or neural-based extensions. Examples of the 'Veggifying' method applied to various 3D scenes. Please refer to the corresponding file submissions for the animations. (a) DALL-E, the crown jewel of BAIR. (b) Rohan's water bottle. (c) The bowl of apples in BWW8, surprisingly with apples in it. (d) A bulb of garlic moments before Rohan's housemates made pasta with it.

Abstract

We present Veggie World, a method for rasterizing 3D scenes using any set of 3D assets, in our case, vegetables. Veggie World builds off Nerfstudio's Gaussian Splatting library to Veggify 3D reconstructions through our Veggie Regularization techniques during training, and render our Veggie Worlds using Blender without the need for diffusion or other neural-based extensions. In addition to visually pleasing Veggified renders of real-world scenes, Veggie World also poses an interesting research question on the importance of texture and 3D structure in image classification.

1. Introduction

3D reconstruction has seen considerable advancements in recent years, enabling seamless 3D renders of scenes with as little as 10 seconds of footage. Most recently, neural radiance fields (NeRFs) [6] and Gaussian Splatting [3] have produced markedly high-fidelity 3D reconstructions that outperform classical 3D rendering techniques. Open source frameworks like Nerfstudio [8] have enabled users to seamlessly use such 3D reconstruction approaches, empowering academic and online communities to creatively build their own extensions as they wish.

Veggie World utilizes the Guassian Splatting implementation within Nerfstudio to create visually-pleasing 'Veggified' 3D reconstructions of scenes. Our method involves optimizing 3D Gaussians to take on vegetable shapes through our own 'Veggie' loss. Additionally, we implement our own 'Veggie' dropout that prevents an excessive number of Guassians from becoming the same vegetable type. We explore the effect of this dropout in our rendered results, as well as how Veggification affects image classification and reveal several interesting insights.

2. Related Work

Neural Radiance Fields (NeRFs) In recent years, NeRFs have emerged as the defacto method for 3D scene reconstruction from 2D images. NeRFs perform volumetric scene rendering by using a differentiable ray tracing procedure: the associated 3D coordinate $\{x, y, z\}$ and viewing direction $\{\phi, \theta\}$ for all samples along the ray are predicted using an MLP and accumulated to predict the RGB for the pixel corresponding to this ray, which is then used in a reconstruction loss to measure the difference between the predicted pixel RGB and that of the training image.

Although NeRFs yield high fidelity scene reconstructions compared to prior works, the training and inference of these networks are prohibitively expensive, as an MLP forward pass must be computed for every sample along a ray and for all rays in a training batch. Furthermore, NeRFs remain relatively less interpretable due to its reliance on an MLP for predicting samples, making downstream tasks like neural style transfer and object editing difficult.

3D Gaussian Splatting Instead of representing a scene using NeRF's ray-tracing sampling technique, (kerbl et al) reconstruct scenes using 3D Gaussians and yield considerable rendering speedup and fidelity improvement. 3D Gaussians are initialized from a sparse point cloud produced by SfM [7] and have an associated mean (position), covariance (axis lengths), and opacity. They demonstrate that Gaussians are a suitable representation for 3D reconstructions as each Gaussian can be used to represent both high and low-level scene features, and can be coalesced into even more Guassians as well as combined with other Gaussians. Each 3D Gauassian is projected into 2D then, using a tile-based rasterization technique that is differentiable, Gaussians are sorted by distance and undergo alphaα-blending to render any arbitrary view. Gaussian Splatting yields quicker training and inference than NeRFs due to the tile-based rasterization used as opposed to NeRF's ray tracing approach. Notably, because each 3D Gaussian is associated with a position in world coordinates, such scene representations are conducive to downstream geometric tasks that involve using pre-training or jointly training Gaussian Splats. For the same reason that 3D Gaussians are an optimal representation for 3D reconstructions as discussed in the original paper, they are also a natural representation for Veggified 3D worlds, as we can associate each Gaussian with a vegetable and effectively Veggify the world.

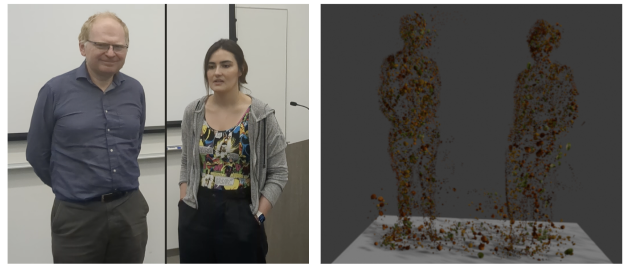

Figure 2. **Veggified Pixel Enjoyers.** Among other items, Veggie World can Veggify humans! In our analysis with human evaluators, humans are still able to gather who the people are even after Veggification. Please refer to the corresponding file submission for the animation.

Stylized 3D Scene Reconstructions Because of their high-fidelity scene reconstructions, NeRFs and Gaussian Splatting have enabled many creative applications that generate visually-pleasing alteration to these 3D representations. InstructNerf2Nerf [2] enables instruction-based scene editing using iterative diffusion-based dataset editing during training. Although such an approach can be used to 'Veggify' a 3D scene, this would in practice require an expensive forward pass for each image through a diffusion network and does not guarantee consistency across all training views. Additionally, such an approach remains reliant on the slow MLP-based rendering of NeRFs, failing to provide the rich and immersive experience that Veggie World strives to provide. Instead of iterative dataset updating, StyleRF [4] makes transformations directly to the feature space of the radiance field through their Deferred Style Transformation (DST) that enables multi-view consistency during style transfer. Nonetheless, this approach still requires a neural extension to stylize the 3D scene, which can be prohibitively expensive depending on the application.

As for Gaussian-based scene stylization, StyleGaussian [5] proposes a method for style transfer of 3D reconstructions with Gaussian Splatting. The method involves taking a pre-trained 3D Gaussian Splat and 1) embedding 2D VGG image into the scene and assigning each feature embedding an associated learnable feature parameter $f_p \in \mathbb{R}^D$, 2) using the input syle image I^S to further transform the transformed feature f_p, and 3) RGB decoding in which the transformed image features of the 3D Gaussians are converted back to RGB. In practice, such an approach could be used to veggie scenes as we do, although our method only involves a simple modification to the training loss and does not involve additional training of feature embeddings after training the 3D reconstruction.

3. Methodology

The veggification process involves two steps. First, we train a Gaussian splat using a regularization so the Gaussians optimize to have similar scales to their closest vegetable. We refer to a Gaussian's standard deviation as its scale because visually, the majority of visible Gaussian is within 1 standard deviation from its center. Second, we run a Blender script that takes each Gaussian and replaces it at its position with its closest vegetable at the correct scale and rotation.

3.1. Veggified Gaussian Splatting Training

We augment Gaussian splatting by adding a component to the loss function in order to make the Gaussians more elegantly match vegetables.

3.1.1 Veggie Regularization and Loss

In order to regularize a Gaussian to its nearest vegetable, we first need a metric to determine how close a Gaussian is to a vegetable. The metric we choose, $D_{g,v}$, is the norm of the difference between each ratio of the standard deviations of the Gaussian and the ratio of the scales of each vegetable. For the Gaussian, let σ be the scale/standard deviation and let R^g be the scale ratio. For the vegetable, let S be the scale and let R^v be the scale ratio.

$$R^g_{xy} = \frac{\sigma_x}{\sigma_y},\ R^g_{xz} = \frac{\sigma_x}{\sigma_z},\ R^g_{yz} = \frac{\sigma_y}{\sigma_z} \tag{1}$$

$$R^v_{xy} = \frac{S_x}{S_y},\ R^v_{xz} = \frac{S_x}{S_z},\ R^v_{yz} = \frac{S_y}{S_z} \tag{2}$$

Then D is calculated as follows:

$$R^g = \begin{bmatrix} R^g_{xy} \\ R^g_{xz} \\ R^g_{yz} \end{bmatrix},\ R^v = \begin{bmatrix} R^v_{xy} \\ R^v_{xz} \\ R^v_{yz} \end{bmatrix} \tag{3}$$

$$D_{g,v} = ||R^g - R^v||_2 \tag{4}$$

To get the closest vegetable to each Gaussian, v_g^*, we simply choose the vegetable with the minimum distance:

$$v_g^* = \text{argmin}_v\{D_{g,v}\} \tag{5}$$

From here, we add the distance from each Gaussian to its closest vegetable to the loss function as the scale loss.

$$\mathcal{L}_{scale} = ||D_{g,v_g^*}||_2 \tag{6}$$

We only add the veggie loss to the loss function after 100 iterations of training to allow the Gaussians to take some shape on their own.

3.1.2 Rotation Invariance

One issue we faced is that the vegetable scales are not rotation invariant. For example, if we have a long and skinny Gaussian:

$$\sigma = \begin{bmatrix} 1 \\ 2 \\ 10 \end{bmatrix},\ R^g = \begin{bmatrix} 0.5 \\ 0.1 \\ 0.2 \end{bmatrix} \tag{7}$$

we'd probably want that to match to a carrot as carrots are long and skinny vegetables. What if, however, the carrot asset we have has scales and ratios as follows:

$$S = \begin{bmatrix} 3 \\ 15 \\ 1.5 \end{bmatrix},\ R^v = \begin{bmatrix} 0.2 \\ 2 \\ 10 \end{bmatrix} \tag{8}$$

The scale ratios are totally different even though if we had just permuted the scales of the vegetable to [1.5, 3, 15] the ratios would be identical. So, we do exactly this! For each vegetable, we get all 6 permutations of the scale to achieve this rotational invariance.

3.1.3 Veggie Dropout

Another issue we faced is that often, we get a very sparse distribution of what vegetables are used. In particular, we found that there are way more carrots than other vegetables because high frequency features are often represented with lots of long and thin Gaussians. While we do want to optimize the splat to represent the 3D structure well, we also want to have a wide assortment of vegetables in the result. To achieve this we implement veggie dropout. If more than $\frac{1}{6}$ of the Gaussians are assigned to a single vegetable, v', we set $\forall g,\ D_{g,v'} = \infty$ which forces all the Gaussians to regularize to a different, slightly less close-in-scale vegetable. This makes our veggie distribution way more uniform (as shown in ablations).

3.2. Veggified Gaussian Splatting Rendering

We export the splat as a pointcloud where each Gaussian has a point with attributes of location, scale, rotation, opacity, and veggie index. Then, in a blender script we take each Gaussian and place the correct vegetable at the location rotated at the direction that the Gaussian is rotated. We scale the location so that the vegetable is able to capture space that the Gaussian did at points further than one standard deviation from it's mean, effectively enlargening the hull of the object.

4. Results

In addition to Veggifying different scenes, we also perform experiments to observe the extent to which texture affects an ImageNet classifier's outputs, as well as ablations to observe the effect of Veggie Dropout. On average, we observe roughly 300 images per scene used for 3D reconstruction.

4.1. Veggie World Renders

Figure 1 displays the main results of our work across a variety of scenes. Upon applying COLMAP, training in Nerfstudio, and rendering the splat export in Blender, we successfully 'Veggify' scenes. In our experience, we observed

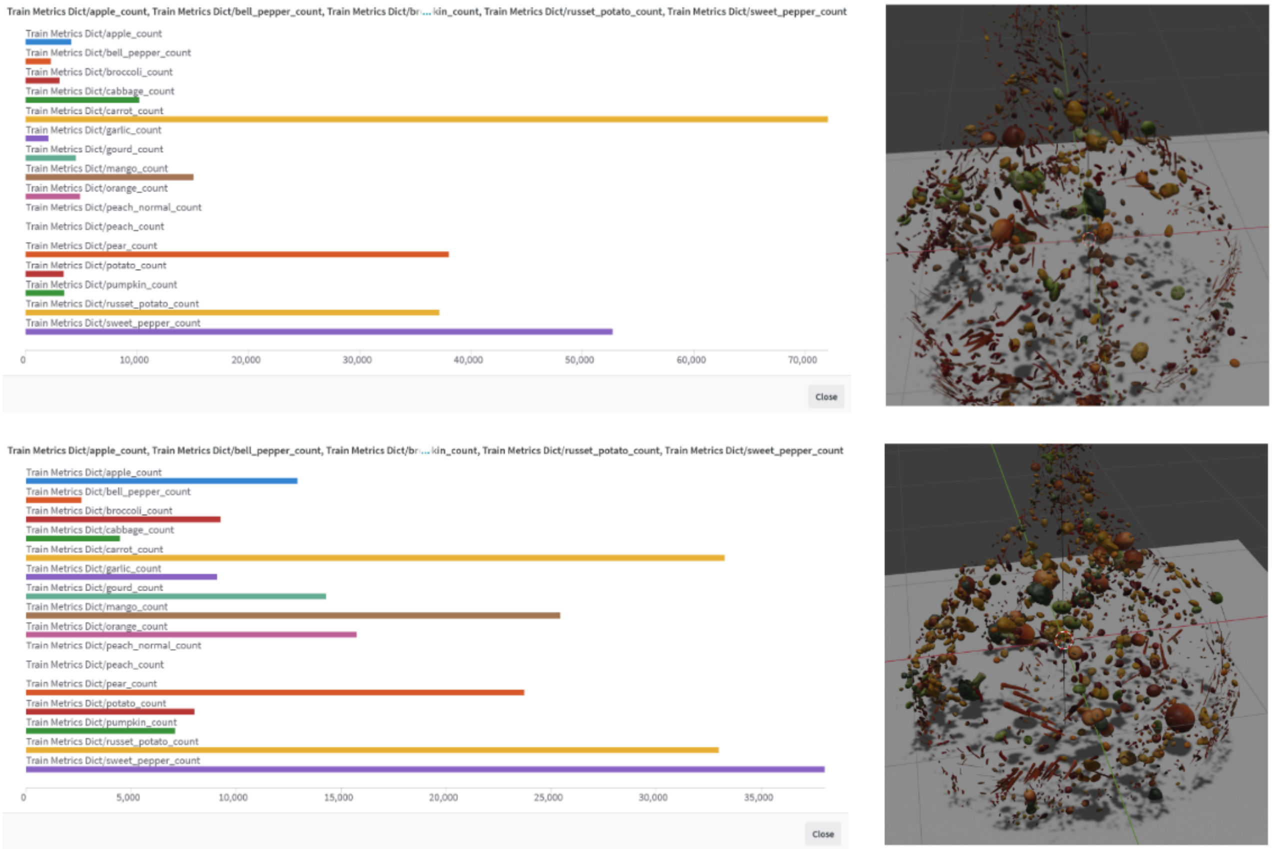

Figure 3. **Veggie Dropout Impact.** Top: garlic results without Veggie dropout. We see that carrots and sweet peppers dominate with over 70,000 and 50,000 occurances respectively. We attribute this to the many high-frequency features needed to model the edges of the garlic bulb, resulting in many Gaussians transforming to these vegetables during training. Bottom: results with a Veggie dropout ratio of $\frac{1}{6}$. Upon applying Veggie dropout, carrots and sweet peppers no longer dominate the distribution of vegetables and the distribution of vegetable counts becomes less more uniform. The diverse distribution of vegetables yields object surfaces that are more covered/dense, as the small high-frequency carrots and sweet-peppers no longer dominate the distribution, allowing other larger vegetables to populate the scene. Note: we felt that the peach asset was distorted and manually opt to exclude peaches from our Veggified outputs.

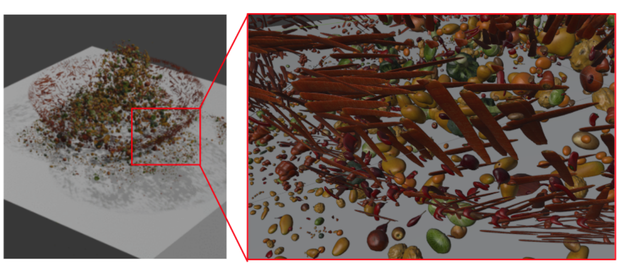

Figure 4. **Veggie World Models High Frequencies.** In order to model high-frequency features within scenes, Veggie World optimizes Gaussians into carrots and sweet peppers due to their shape. In this scene, the thin, flat sides of the bowl are modeled with an abundance of carrots and sweet peppers to accurately capture the high-frequency features.

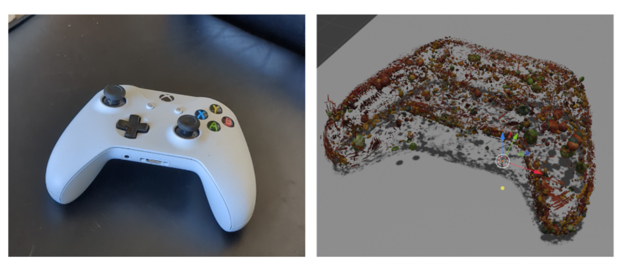

Figure 5. **Veggified Controller.** A successful Veggification of a video game controller. We observe a higher concentration of vegetables around the edges of the object, preserving the object's overall structure.

the peach assets to be heavily distorted and opted to automatically exclude these from our rendered outputs.

4.2. Classification Analysis

Inspired by the work presented in [1] which asserts that ImageNet classifiers are biased towards textures rather than image content or other signals, we experiment applying an ImageNet-pretrained ResNet50 and observing changes in prediction scores upon Veggifying the scene. In this experiment, we naively perform classification on an object before and after Veggification, paying attention to top-3 output scores as well as the score associated with the actual object (if applicable). Because most of the in-the-wild objects we record do not correspond to actual labels in ImageNet, we use text embedding comparison via HuggingFace SentenceTransformer's 'all-MiniLM-L6-v2' model. In doing so, we are able to fetch the 3-closeset labels based on text, and then observe the ResNet50 prediction scores for these labels before and after Veggification. Our initial results are mixed. As shown in the project presentation, in some cases, the classifier struggles to classify the original input image, let alone the veggified version of this object. Another interesting observation is how, because of the Veggification process, we often lose many important details about the main object itself, and the classifier instead attends to the structure of the scene. For instance, the Veggified bowl of apples loses virtually all notion of 'apples' and yet the classifier is still able to retrieve 'mixing bowl' in its top 3 prediction labels. More concretely, when looking at table 1, we see that the score associated with 'soup bowl' and 'mixing bowl' increases substantially upon Veggification. Whereas the ImageNet labels for the un-Veggified image are associated with the fruit in the bowl, we see that the model's outputs on the Veggified scene are far more concerned with the broader semantics of the scene, that being the presence of a bowl. As for the other results, we see that the model still manages to allocate non-zero probability to the labels associated with the original object, which is to say that the model is perhaps still able to make sense of the Veggified scene, albeit with very poor performance.

Veggie World raises an interesting question on the importance of structure and texture for image classification. To the human eye, our Veggified scenes are ones such that human evaluators are typically able to predict what the original object was. In contrast, we observe that classifiers largely struggle to make sense of our Veggified outputs, which is to say that there is still a large gap to fill with respect to getting models to make sense of the world as we do. It also speaks to how limited supervised systems are in that they fail to make sense of out-of-distribution samples (e.g. our Veggified outputs) and are constrained to their training set distribution. With more time, we'd like to explore how self-supervised classifiers and their associated features make sense of our Veggified outputs relative to their supervised counterparts.

Object	1-NN Label	2-NN Label	3-NN Label
Bowl	Soup Bowl	Mixing Bowl	Microwave
Original	0.0000	0.0000	0.0000
Veggified	0.0097	0.0870	0.0008
Bottle	Beer Bottle	Bottle Cap	Water Bottle
Original	0.0004	0.0001	0.0419
Veggified	0.0004	0.0007	0.0014
Sheep	Old English Sheepdog	Wool	Shetland Sheepdog
Original	0.0028	0.0022	0.0001
Veggified	0.0001	0.0006	0.0000
Human	Gorilla	Organ	Chimpanzee
Original	0.0000	0.0000	0.0004
Veggified	0.0000	0.0001	0.0000

Table 1. **Veggification Effect on Image Classification.** For each object, we generate our own text label (e.g. "sheep" for DALLE) and fetch the 3 nearest labels using the 'all-MiniLM-L6-v2' SentenceTransformer from HuggingFace. We then compute the ResNet50 probability score for each of these labels before and after Veggification. For instance, 1-NN refers to the closest text label based, and 2-NN refers to the second-closest label.

4.3. Ablation Studies

For one of our ablation studies, we examine the effect of Veggie Dropout on a scene. We originally added Veggie Dropout upon noticing the trend in which thin vegetables, namely carrots and sweet peppers, dominate the count distribution for many virtually all scenes. Upon further analysis, we determined this is because many Gaussians are needed to model the high-frequency edges of our scenes, and these vegetables are a natural choice for representing high-level features. To this end, we observe results before and after veggie dropout shown in figure 3. Most notably, the scene with dropout appears much less sparse than the scene without dropout, presumably because the scenes without dropout have much more high-frequency vegetables which do not occupy much area in the 3D reconstruction. Upon applying dropout, the distribution of vegetables more even and we see a much more diverse set of vegetable counts, using vegetables with higher area to model the low-frequency parts of the scene.

4.4. Limitations

In order to selectively Veggify only the object within a scene and not the background, we must create a bounding box in the Nerfstudio viewer that is sufficiently small enough so-as-to not include any background point cloud elements. Future extensions may include support within Nerfstudio to remove undesired Gaussians using a lasso tool of sorts.

We are also compute-constrained when rendering our Veggified worlds in blender. Because of this, most of our results are object-centric and we are unable to Veggify entire rooms with backgrounds included. With more compute, we would love to see what a true 'Veggie World' would look like. This is also why we are unable to render the gifs for all shown scenes.

Additionally, our method relies on scaling the Gaussians in Blender (both the vegetable scales and the coordinates) which varies from scene to scene. We'd like to come up with a heuristic of automating this process to further streamline our pipeline.

Lastly, we anticipate some difficulty applying our approach to objects other than vegetables: the selection of 3D assets must be diverse in shape and color in order to properly represent scenes without compromising the geometry of the underlying scene. For instance, a 3D pastry collection not only lacks diversity in color and texture, but there is likely not a pastry that can reasonably represent high-frequency features without the need to squish/distort the pastry (and its associated Gaussian) into a weird shape.

5. Conclusion

In this work we presented Veggie World, an non-neural augmentation to Gaussian Splatting that allows users to Veggify their 3D worlds as a fun twist. By using Nerfstudio's Gaussian Splatting framework, Veggie World optimizes Gaussians to take on the shape of a variety of vegetables which are then rendered in Blender to yield visually-pleasing results while maintaining the structural integrity of the scene. Veggie World demonstrates the versatility of Gaussian Splatting and the ability to create stylisitic 3D reconstructions without the use of extensive diffusion models or iterative dataset updating. Additionally, we share preliminary analysis on how texture-augmentation like Veggie World can affect image classification, and propose future directions of further exploring the extent to which structure and geometry vs texture affect classifiers commonly seen in benchmark leaderboards and everyday applications. We hope Veggie World brings the reader as much enjoyment as it did to us!

References

[1] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness, 2022. 5

[2] Ayaan Haque, Matthew Tancik, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions, 2023. 2

[3] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. 1

[4] Kunhao Liu, Fangneng Zhan, Yiwen Chen, Jiahui Zhang, Yingchen Yu, Abdulmotaleb El Saddik, Shijian Lu, and Eric Xing. Stylerf: Zero-shot 3d style transfer of neural radiance fields, 2023. 2

[5] Kunhao Liu, Fangneng Zhan, Muyu Xu, Christian Theobalt, Ling Shao, and Shijian Lu. Stylegaussian: Instant 3d style transfer with gaussian splatting, 2024. 2

[6] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 1

[7] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *Proceedings of SIGGRAPH*, 2006. 2

[8] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David Mcallister, Justin Kerr, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*. ACM, 2023. 1

Figure 6. **Veggie World Up Close.** DALL-E the beloved BWW8 sheep Veggified. Please refer to the corresponding file submission for the animation.