Utilizing Image Segmentation for 2.5D Background Creation in Film and Television Production

Introduction

Image segmentation technology, a critical component of artificial intelligence, exhibits substantial potential for practical applications within the film and television industry. The intricacies of on-location filming, which include challenges such as procuring suitable locations, managing equipment logistics, and navigating scheduling constraints, frequently result in substantial production costs. Furthermore, traditional solutions, such as green screen backgrounds and photo shoots, offer limited flexibility. These methods are often hindered by viewpoint restrictions, making the discovery of missing shots during the post-production stage a significant challenge.

Meanwhile, 2.5D technology offers an attractive solution by preserving depth information at a fraction of the cost of 3D modeling or reshooting. Therefore, employing 2.5D as a backdrop for supplemental shots is a promising approach. This research aims to utilize image segmentation technology to achieve 2.5D effects, thereby offering a viable alternative to on-location shooting. By exploring the practical application of these technologies, this study seeks to contribute to the ongoing discourse on efficient and cost-effective production methods in the film and television industry.

1.1Theoretical foundation

Image segmentation, a crucial aspect of deep learning, allows for the division of an image into distinct regions with varying semantics. This study explores three subdivisions within this field: semantic segmentation, instance segmentation, and depth map generation with subsequent segmentation via a clustering algorithm.

Semantic segmentation entails the partitioning of an image into diverse regions with disparate semantics. The model is trained using a dataset with images annotated with multiple labels, assisting the model in learning the features of these objects. With a sufficiently large training set, semantic segmentation models can accurately capture object features.

Instance segmentation, considered an advancement of semantic segmentation, enables precise identification of individual entities within an image, facilitating more accurate analysis of their relative positions.

Generating a depth map and segmenting it into different levels using a clustering algorithm provides a means to measure depth information based on the relative positions of objects. However, due to the lack of a suitable model and training set in the initial stages of the experiment, this approach was provisionally shelved.

In this research, the semantic segmentation model DeepLabV3+ ResNet50 is compared with a new instance segmentation model, Segment Anything (SAM), to identify the optimal segmentation strategy. The output is projected to be in the form of mask files for each slice, capable of being applied directly to the original file, facilitating further manipulation by operators in Digital Content Creation (DCC) software such as Unreal Engine.

Finally, areas left blank, such as the sky, will be filled using image fill technology to ensure seamless organization. The ultimate goal is to achieve splitting in relatively complex environments, enabling natural pushing, pulling, and shifting movements. This will consequently reduce the cost of creation, aiding creators in producing high-quality, cost-effective content.

2.1 Deeplab3+

DeepLabV3+ is a deep learning model developed by the Google Research team, designed for semantic image segmentation tasks. This model aims to assign each pixel in an image to a specific class, thus identifying the object or region to which it belongs. It integrates the concepts of deep convolutional neural networks and atrous (dilated) convolutions and features an encoder-decoder structure supplemented with residual connections. This architecture allows DeepLabV3+ to capture rich contextual information within images and delineate object boundaries with high precision (Chen et al.2018).

In the present study, we utilized the DeepLabV3+ model, training it with the comprehensive ADE20K dataset. This dataset is extensive, encompassing 211 categories, including diverse themes such as urban, indoor, and natural environments. Theoretically, the broad scope of this dataset should meet the segmentation needs for film and video production.

We hypothesized that the combination of the DeepLabV3+ model and the ADE20K dataset could provide a robust solution for semantic segmentation in film and video production. The diverse categories covered by the ADE20K dataset should enable the model to accurately identify and segment a wide range of scenes and objects, potentially leading to more efficient and effective production processes.

The training process was a resource-intensive endeavor, consuming up to eight days to reach completion. Despite the extended duration, the model managed to successfully demarcate distinct blocks in the given images. For the purpose of this task, we simulated real-world scenarios that may necessitate filler shots, focusing primarily on medium shot urban environments.

While the partitioning results revealed competent performance from the model, several challenges were also apparent. Semantic segmentation tends to group together objects in proximity or objects sharing the same label, specifically when confronted with clusters of urban buildings. This behavior complicates achieving desired segmentation outcomes (Refer to Figure 1). Moreover, the model encountered difficulties when dealing with objects with obscure or unconventional features. For instance, when presented with a structure resembling both a bridge and a building, the model did not make a conforming judgement, and instead segmented the bridge into multiple segments (Refer to Figure 2).

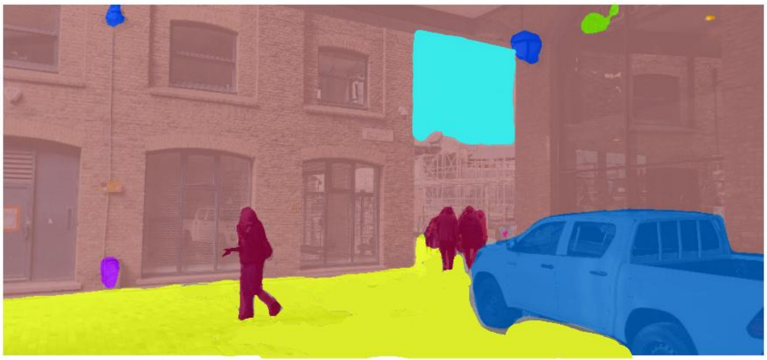


Figure 1. The buildings are all connected together.

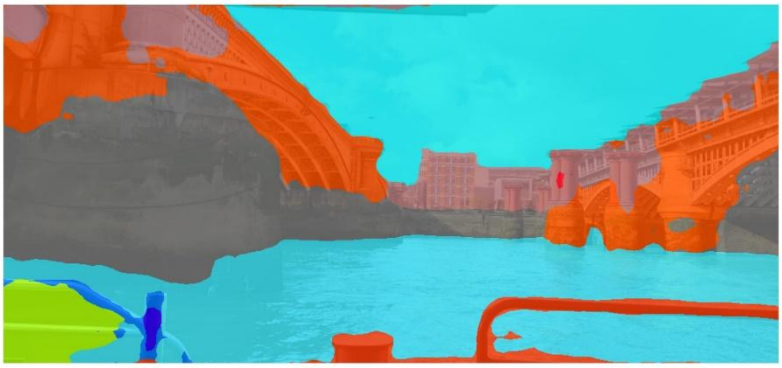


Figure 2. The bridge is divided into several pieces.

These findings suggest that semantic segmentation, despite its merits, may not be the optimal choice for this specific task due to its inherent limitations in handling complex objects and scenes. Further research should explore other techniques, such as instance segmentation, that may prove more effective in such contexts. Although the plans for deeplab3+ ended in failure, the experimental code is still stored in the deeplab3+ folder.

2.2 Segment Anything Model (SAM)

The Segment Anything Model (SAM) is an instance segmentation model, integrating the "Prompt" mechanism. This model, designed for optimal segmentation results, utilizes text, coordinate points, bounding boxes, among other auxiliary information (Kirillov et al.,2023). Due to the complexity of SAM, training the model using a standard personal computer poses challenges.

To overcome the limitations identified, this study turned to available open-source resources. Notably, the code adapted for this project was based on work done by Yu et al. (2023), who developed a project called " Inpaint Anything " that aimed at segmenting and replacing the nearest object based on its coordinates. This strategy involved identifying objects within the image and subsequently redrawing them in the blank areas. Given the alignment of this process with the requirements of the current study, this approach was deemed suitable for our objectives (Yu et al., 2023). Several significant modifications were made to the original project:

1. The input method was adjusted from a predetermined coordinate point to an interactive point selection coordinate point. An interactive window was integrated, allowing operators to manually select the part of the image they wished to segment. Furthermore, a threshold was established, allowing the operator to decide the number of coordinates for object determination.
2. The input was revised to accept a mask instead of a split block, making it more compatible with Digital Content Creation (DCC) software, such as Unreal Engine. This adjustment to output as a mask simplifies material creation for the operator.
3. Analogous modifications were made to the image fill section to facilitate a more accurate fill of the model.

The purpose of these modifications is to improve the versatility of the model and assist creators in acquiring the necessary resources. These modifications primarily focus on the changes implemented in 'inpaint_anything.py' and 'sam_segment.py'. Under appropriate circumstances, it is anticipated that these modifications will significantly facilitate the film and video production process. Furthermore, a code for downloading weights ('download_weight.py') has been prepared to simplify the deployment of custom programs for operators.

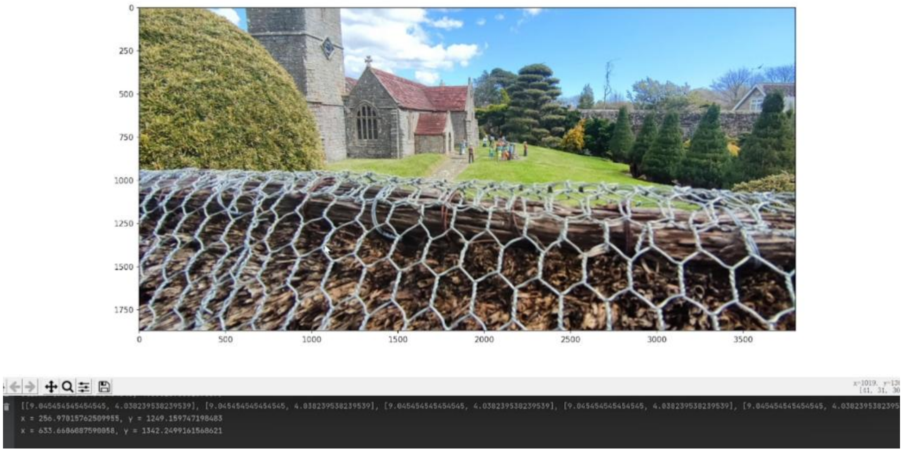
2.3 Layout to 2.5D

The primary focus of this research is the application of 2.5D technology, which simulates 3D effects on a 2D plane. Acting as an intermediary between 2D and 3D, this method presents the visual depth of 3D while maintaining the practicality and rendering complexity akin to 2D. This transitional approach proves valuable in the field of image segmentation, facilitating the generation of multi-layered depth-of-field effects.

Image segmentation allows for the division of an image into multiple distinct objects or regions, with each considered a separate "layer". Thus, the outcome of the image segmentation can be regarded as a 2D image imbued with depth information. This enhancement provides an increase in the perceived depth and dimensionality of the image. Compared to full 3D rendering, it offers a more cost-effective method that still delivers visually captivating effects exceeding those of standard 2D.

For this research, SAM was selected as the preferred image segmentation method and integrated into Unreal Engine for subsequent 2.5D effect processing. Configuring SAM requires the creation of an appropriate environment following the guidance provided in the "lama_requirements_windows.txt" file. Subsequently, running download_weight.py allows the retrieval of model weights. With this setup, the model is in an operational state, ready for use. This procedure necessitates the division of tasks into two distinct steps: segmentation and padding.

Initially, the image is loaded from the input folder. The user can execute ‘sam_segment.py’, which initiates a window prompting the user for point selections. The program then identifies the objects surrounding the selected points and groups them into a segmentation (Figure 3). The segmented portions are saved to the output (Figure 4). Subsequently, ‘inpaint_anything.py’ is executed, and a similar window appears, allowing the user to select the area that requires filling. If the masked image remains incomplete, additional points may be necessary to aid the model in identifying it. The final output is directed to the results folder.



```
[[9.045454545454545, 4.0382395382395539], [9.045454545454545, 4.0382395382395539], [9.045454545454545, 4.0382395382395539], [9.045454545454545, 4.0382395382395539], [9.045454545454545, 4.0382395382395
x = 256.97015762509955, y = 1249.159747198483
x = 633.6686087590858, y = 1342.2499161568621
```

Figure 3. The program brings up a pop-up window so that the operator can click on the sections that need to be grouped together.

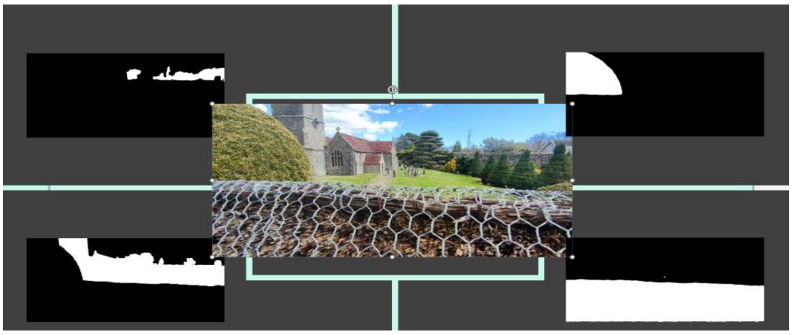


Figure 4. Getting the results of different masks and the original image

Within the context of PyCharm, the operator is empowered to manipulate the segmented areas visually and output different levels of masks. While there were attempts to port these functionalities to Jupyter Notebook, the lack of a compatible toolkit within the Jupyter environment rendered these efforts unsuccessful.

The methodology so far delivers a complete sky and mask file for each segment, thereby producing manipulable backdrop material. The material sphere of the Unreal Engine features an opaque mask mode, permitting the inclusion of a mask channel. This results in the black areas of the mask becoming transparent, with the white areas remaining visible (refer to Figure 5). These materials can be manually attached to the facets and adjusted according to the interrelationship between different layers in real-time as necessitated by the perspective.

Ultimately, this technique produces multiple facets that can be seamlessly integrated into any scene as required to support the actor's performance. Initial tests have shown that such a background can be manipulated in four ways (e.g., ±15 degrees) to maintain a relative perspective and avoid over-stretching the image (Figure 6). Additionally, Figure 7 demonstrates that the depth effect is maintained even when using a face piece in the next-door view.

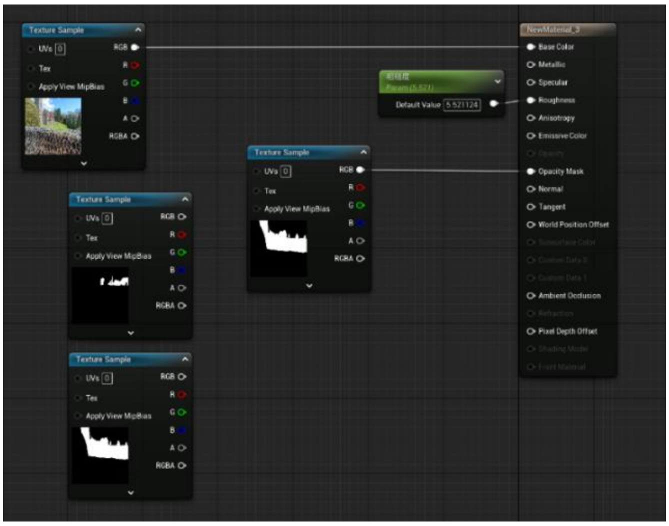


Figure 5. Using materials in Unreal Engine.

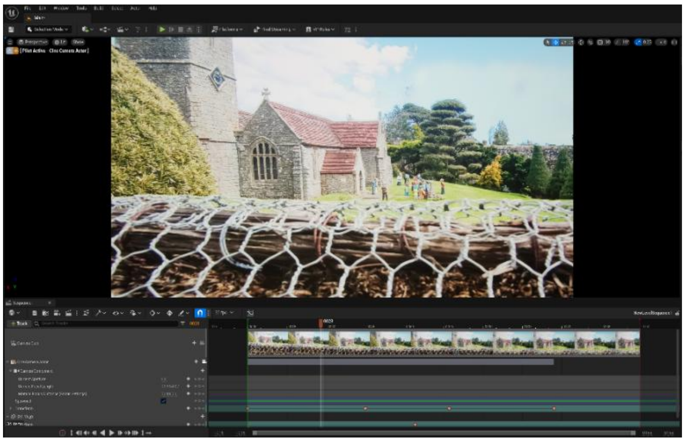


Figure 6. Implementing 2.5D effects in Unreal Engine

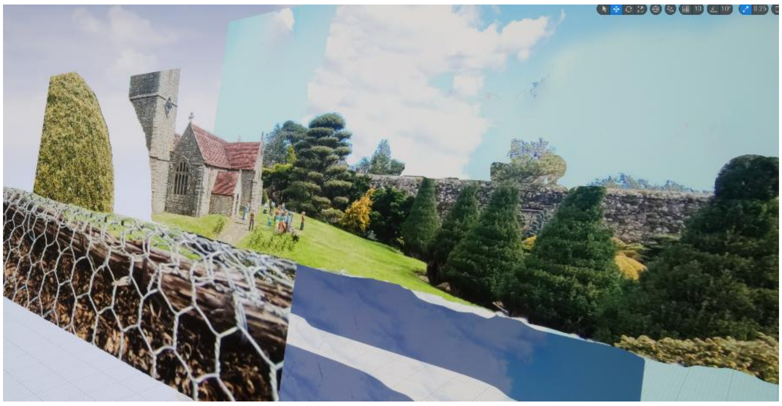


Figure 7. Observing the project in other views.

Conclusion

In practical shooting scenarios, this solution offers a cost-effective means of modifying existing material, thus reducing the requirement for costly re-shoots. By reworking the existing material, the proposed method offers significant savings in terms of setup and shooting costs.

The primary objective of this research has been to devise a pipeline, grounded in the principles of image segmentation, aimed at facilitating filmmakers and television producers to procure required supplementary footage material efficiently and through simplified operations. By contrasting semantic and instance segmentation, a more efficient solution has been identified.

Within the context of a practical experiment, the open-source SAM model underwent modifications to render its input and output interfaces more accessible and comprehensible to those in the film and television industries who lack training in artificial intelligence. As a result, the revised SAM model assists operators in obtaining masked materials for use within DCC software, yielding the desired results.

This enhanced model offers increased flexibility to filmmakers and television producers, providing them with easier access to necessary materials and thereby reducing creative costs. In practical filming scenarios, this solution promotes the reutilization of existing footage, circumventing the expense associated with re-shoots. In summary, this research has successfully engineered an image segmentation-based solution capable of assisting film and television producers in the effective utilization and modification of existing footage to satisfy specific production needs, all the while maintaining visual consistency.

Reference

1. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018) 'Encoder-decoder with atrous separable convolution for semantic image segmentation', in Proceedings of the European Conference on Computer Vision (ECCV), pp. 801-818.

2. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., ... & Girshick, R. (2023) 'Segment anything', arXiv preprint arXiv:2304.02643.

3. Yu, T., Feng, R., Feng, R., Liu, J., Jin, X., Zeng, W., & Chen, Z. (2023) "Inpaint Anything: Segment Anything Meets Image Inpainting." arXiv preprint arXiv:2304.06790.