Dynamic Billboard Replacement in Videos

Shweta Agrawal
Stanford University
Google (SCPD)
ashweta@stanford.edu

Yawen Guo
Stanford University
Google (SCPD)
ywguo@stanford.edu

Abstract

Video monetization through product endorsements often relies on video creators physically integrating products in their videos during production, thus limiting both creative freedom and advertiser reach. This project explores building an end-to-end automated pipeline for product placement within existing video content post-production. We focus on the case of Billboard replacement in this report, and accomplish the task in 4 stages: Billboard object detection using YOLOv8 [18] or Grounding Dino [11], Mask refinement using YOLOv8 instance segmentation model or SAM[7], Video object tracking across frames using Cutie [5], and Video in-painting to render the final image using OpenCV library[4]. Additionally, we explore the potential of Stable Diffusion[21] with IP-adapter for in-painting, utilizing both text and image prompts. We present our findings, including observed limitations.

Figure 1. Pipeline output example: Automatic billboard replacement with an open-access painting (Trap) from National Gallery of Art.

1. Introduction

Video creators often monetize their videos by endorsing and promoting products directly in their videos during production. This project investigates the use of computer vision techniques for automatic post-production product placement, empowering creators with the ability to promote diverse products at any point. Conversely, these techniques could also be used to remove unwanted advertisements in a video or replacing them with something pleasant. In this report we focus on the use-case of automatic billboard replacement, and show some examples of replacing billboards with provided text prompt, product images or paintings.

We develop an automated pipeline to detect in-scene billboard advertisements in video content and seamlessly replace them by provided image, using object detection, segmentation, tracking and video in-painting techniques. The input to the automated pipeline is a raw video containing billboards and a replacement product image that needs to be seamlessly composited into the identified billboard. The output of the pipeline is a modified version of the original video where the new product or image has been realistically inserted in place of identified billboard.

Our approach yields good results for billboard detection, segmentation, and tracking in video. While in-painting using OpenCV[4]'s perspective transforms and mask replacement is generally reasonable, limitations are observed in flicker handling. We also try Stable diffusion models with IP-adapter, to achieve a more seamless in-painting with an image prompt. While the models perform well on individual frames, consistent generation across video frames proves challenging. This suggests that future work should explore and adapt existing video in-painting models such as STTN[23], FuseFormer[12], or investigate training generative models explicitly for video in-painting to improve adherence to image prompts and temporal stability.

2. Related Work

The first step in our pipeline involves using object detection models to detect potential areas within each video frame where new product can be inserted or existing products / ads can be replaced. YOLO (You Only Look Once) [18], which employs a convolutional neural network (CNN) architecture, is a fast and effective single-stage object detection algorithm for both images and videos that re-frames object detection as a regression problem, and directly predicts bounding boxes and class probabilities for objects in a single pass through the network. Grounding DINO [13] is an open-set object detector, created by marrying Transformer-based detector DINO [11] with grounded pre-training, which can detect arbitrary objects with human inputs such as category names or referring expressions. We fine-tune both of these for billboard detection and compare accuracy.

Once these regions of interest are detected, we can get precise masks by using image segmentation models. YOLO segmentation[9], an extension of the YOLO object detection framework, enables simultaneous object detection and segmentation, providing both bounding boxes and pixel-level masks for identified objects. Segment-Anything [10] is a new task, model, and dataset for image segmentation. The model is trained and available for use by prompting using bounded boxes, and can transfer zero-shot to new image distributions and tasks. Grounded SAM [19] uses Grounding DINO as an open-set object detector to combine with the segment anything model (SAM). This integration enables the detection and segmentation of any regions based on arbitrary text inputs and opens a door to connecting various vision models. We try various combination of these segmentation techniques with object detection.

After finding the precise mask of desired surface in one frame, we use Video Object Segmentation (VOS) methods to track them across frames, in a semi-supervised setting where a first-frame annotation is provided, and the method segments objects in all other frames as accurately as possible. XMem [6] has been leading architecture for VOS for long videos with unified feature memory stores inspired by the Atkinson-Shiffrin memory model. However, in 2024, the same authors came up with Cutie [5]. Cutie uses object-level memory reading instead of pixel level memory reading like XMem[6], and gives superior object tracking performance by putting the object representation from memory back into the video object segmentation result. In this report, we try Cutie with zero-shot learning.

With the desired product placement instance masks detected and propagated over time, video in-painting techniques can be employed to seamlessly blend new product or advertisement image into the tracked and segmented areas. In the simplest scenarios, such as when dealing with flat objects like billboards, perspective transformations provided by the OpenCV library[4] can be used with some smart handling of contours and masks across frames. For a more generic in-painting, Stable Diffusion models [22] would work more seamlessly. We get some mixed results with the pre-trained stable diffusion models with IP Adapter [27] which we share in subsequent sections.

Finally, we take inspiration from other works for end-to-end pipelines in such vision task. Track Anything paper [25] puts together a segmentation plus tracking pipeline using SAM and XMem. This project [3] from Amazon puts together a pipeline to insert products in a cooking show. Our pipeline however is different from Track-Anything as it combines automatic detection of billboard before, which Track anything does not, uses Cutie [5] instead of XMem [6], and adds in-painting step in the end. Also, as compared to [3], billboard replacement is not limited to specific environments in our approach. We also fine-tune our own models for billboard detection and utilize the best available tracking models for videos.

3. Data

We explored and studied a number of annotated video and image datasets available for object detection and segmentation. Most of these datasets including Youtube-VIS[26], Youtube-8M[1] etc. focus on foreground categories, like person, animals, vehicles or small objects like cups etc. Through our research and exploration, we ultimately determined that utilizing community-curated datasets from Roboflow was the most suitable approach for billboard detection and segmentation. We also curated a few of our own datasets for instance segmentation using the public accessible videos linked from Youtube-8M, since there weren't many good quality billboard segmentation datasets available.

Despite the limited size of the datasets, fine-tuning pre-trained models on a limited dataset comprising 3,000 to 4,000 well annotated images significantly improved their ability to detect billboards, even though they initially struggled with this task.

3.1. Object Detection Datasets

- Robolflow Open source billboard object detection dataset 1 [2]: 3399 images annotated with bounding boxes for billboards with 80:15:5 training:test:validation split;
- Robolflow Open source billboard object detection dataset 2 [24]: 2719 images annotated with bounding boxes for billboard with 70:20:10 training:test:validation split;
- Roboflow Open source billboard object detection dataset 3 [20]: 399 images with 80:10:10 training:test:validation split;

3.2. Instance Segmentation Datasets

- Roboflow Open source billboard segmentation dataset 1 [17]: 2327 images annotated with detailed masks for billboard segmentation with 80:20 training:validation split;
- Self-curated Roboflow billboard instance segmentation dataset 1 from a video[14]: 120 total video frames detailed annotated with precise billboard segmentation mask using roboflow annotation tools, with 70:20:10 training:validation:test split; now published for public use;
- Self-curated Roboflow billboard instance segmentation dataset 2 from a video[15]: 71 frames annotated with precise masks for billboards from a video clip, 100% validation.
- Self-curated Roboflow billboard instance segmentation dataset 3 from a Sports video[16]: 194 video frames from a public access baseball video from Youtube-8M, annotated with precise billboard masks, multiple per frame, with 70:20:10 training, validation and test split.

3.3. Hand selected video for e2e pipeline testing

Finally we demonstrate the end to end pipeline results on a hand-curated video clip that has one prominent and a couple of smaller billboards, and the camera moves enough to require handling of perspectives and object tracking. We aim to replace and track the prominent billboard in this clip. (See original frame in Figure 1).

4. Methods

We develop our pipeline in 4 stages combining object detection using YOLO [18] or Grounding Dino [13], instance segmentation using SAM [10], video object segmentation and tracking using Cutie [5] and video in-painting using OpenCV library [4] or Stable Diffusion [21] techniques.

4.1. First frame billboard detection

With the provided input video, our goal is to detect the first frame with one high quality billboard that we want to track and replace with new product / advertisement image in the whole video. We use YOLOv8 / GroundingDINO object detection models for this purpose. Since the YOLOv8[9] model doesn't have built-in billboard detection, we fine-tuned it using billboard datasets to create a custom model for our purposes. For comparison, we also try the GroundingDINO[13] model that is able to do this with zero-shot prompting.

We first apply the object dtection model on every frame of the video to detect billboards, then we choose the first frame in the video which has detected billboard(s) with high enough confidence threshold of 0.55. In cases where multiple billboards are detected with high confidence in the frame, we select the billboard with the largest area for further processing.

For this specific pipeline, we concentrate on tracking and replacing a single prominent billboard within the video clip. However, future iterations of this approach could potentially extend to replacing multiple billboards simultaneously.

4.2. Precise billboard mask segmentation

With the first frame in the video with bounding box of the detected billboard that we want to replace, we use the segmentation model to generate high quality segmentation mask for this bounding mask. If multiple masks are generated, we choose the one with highest score.

Additionally, we evaluate the performance of YOLO segmentation[9] and GroundedSAM[19], both of which integrate object detection and segmentation functionalities, and subsequently conduct a comparative analysis of their results.

4.3. Billboard tracking

Utilizing the first frame's mask as a reference, we perform semi-supervised VOS (Video Object Segmentation) to track the billboard mask on the following frames using Cutie [5], and generate mask for each following individual frame.

4.4. Billboard mask replacement / video in-painting

Lastly, each per-frame mask generated in Step 3 is replaced with the designated product image using the OpenCV libracy[4], and the resulting frames are compiled into a final video clip. We first detect the four-corner contour around the precise mask and use the four corners to perform a perspective transform on the replacement image, then we place the transformed image on the masked area in the original frame. This gives good results for image but we see considerable flicker in video.

Additionally we try in-painting using Stable Diffusion model from runwayml [22] with both text prompt and also image prompt using IP-adapter [27], and report findings. Despite strong performance on single frames, their effectiveness diminishes when applied to consecutive video frames, where consistent generation and in-painting across frames becomes problematic.

Given time, we would like to explore additional techniques to blend the image better into the background, as well as try to reduce the flicker across frames. Some ideas include using Gaussian blur on the edges, and using some penalty on too much contour change across frames, to keep perspective from changing a lot frame by frame. We also want to explore existing video in-painting models like

STTN[23] and FuseFormer[12], or investigate specialized generative models for enhanced temporal stability and adherence to prompts in video in-painting, which could potentially address the current challenges in achieving consistent and accurate results across video frames.

5. Experiments

5.1. Evaluation Method

To assess the accuracy of our pipeline, we employ a comprehensive evaluation strategy that combines both quantitative metrics and qualitative observations.

- Fine-tuned object detection and segmentation models were quantitatively evaluated using image datasets curated from both independent images and video frames. Details of the evaluation metrics are presented in the subsequent section.
- The assessment of video object tracking and in-painting effectiveness is conducted through qualitative analysis of test videos. Illustrative examples of image frames and accompanying links to demo results are provided for demonstration purposes.

5.2. Evaluation Metrics

Intersection over Union (IoU) is computed for each detected object and measures the ratio of overlap between the union of detected bounding box and the ground truth box. If the IoU exceeds a certain threshold (usually 0.5), the prediction is considered a true positive (TP). Otherwise, it's a false positive (FP). This IoU is then used to compute the following four metrics to evaluate model's performance in detecting bounding boxes of billboard objects (P, R, AP50, AP50-95).

- P (Precision): The accuracy of the detected objects, indicating how many detection are correct.

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

- R (Recall): The ability of the model to identify all instances of objects in the images.

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

- AP50: Mean average precision calculated at an IoU threshold of 0.50. It's a measure of the model's accuracy considering only the "easy" defections.
- AP50-95: The average of the mean average precision calculated at varying IoU thresholds, ranging from 0.50 to 0.95. It gives a comprehensive view of the model's performance across different levels of detection difficulty.

For instance segmentation, the same four metrics are used with a slightly adapted version of Intersection Over Union (IoU) which is computed for each segmented object as an overlap between the predicted segmentation mask and the ground truth mask.

5.3. Results for billboard detection

This task involves converting video into image frames and detecting bounding boxes for billboards with confidence scores. We evaluated two different models for this purpose:

- YoloV8 [8] is fine-tuned with two billboard detection datasets [2] and [24]. Validation accuracy is evaluated on the validation split of these datasets. Test accuracy is evaluated on dataset completely unseen during training and validation [20].
- GroundingDino [13] demonstrates sufficient performance in billboard detection via zero-shot text prompting.

Table1 reports the evaluation metrics (P, R, AP50 and AP50-95) comparing the performance of these two models. We observed that GroundingDINO [13] was able to perform as good as fine-tuned YoloV8, and many times even better, with zero-shot prompting. This demonstrates the strength of superior transformer based generalize learning architecture of this model. Some qualitative illustrations of object detection are shown in Figure 2 for various cases of single, multiple, flat and curved billboards.

5.4. Results for billboard mask segmentation

We fine-tuned and compared three different model combinations for precise mask segmentation for Billboards:

- Yolov8-instance-seg[9]: YOLO segmentation model is fine-tuned with one open-access [17] and one self-curated instance segmentation dataset [14], tested on a completely unseen self-curated dataset [15] on Roboflow.
- YoloV8+SAM: YOLO object detection model is fine-tuned on object detection datasets as described above, followed by zero-shot inference from Segment-Anything Model (SAM) [10].
- GroundedSAM: zero shot inference from Grounding DINO followed by Segment-Anything Model (SAM), also tested on same three datasets.

We report quantitative comparisons between these three models in table2. While the fine-tuned YOLO segmentation model performed better on its training validation set after fine-tuning, GroundedSAM's accuracy on the independent

Model	Stage	Images	Instances	P	R	AP50	AP50-95
YoloV8	Validation[2]	510	1404	0.716	0.622	**0.678**	0.403
GroundingDINO(Zero-shot)	Validation[2]	510	1404		0.685	0.622	**0.412**
YoloV8	Validation[24]	544	1536	0.788	0.735	**0.821**	**0.531**
GroundingDINO(Zero-shot)	Validation[24]	544	1536		0.709	0.622	0.405
YoloV8	Test[20]	39	46	0.663	0.739	**0.901**	0.54
GroundingDINO(Zero-shot)	Test[20]	39	46		0.909	0.828	**0.685**

Table 1. Billboard object detection (bounding box) accuracy

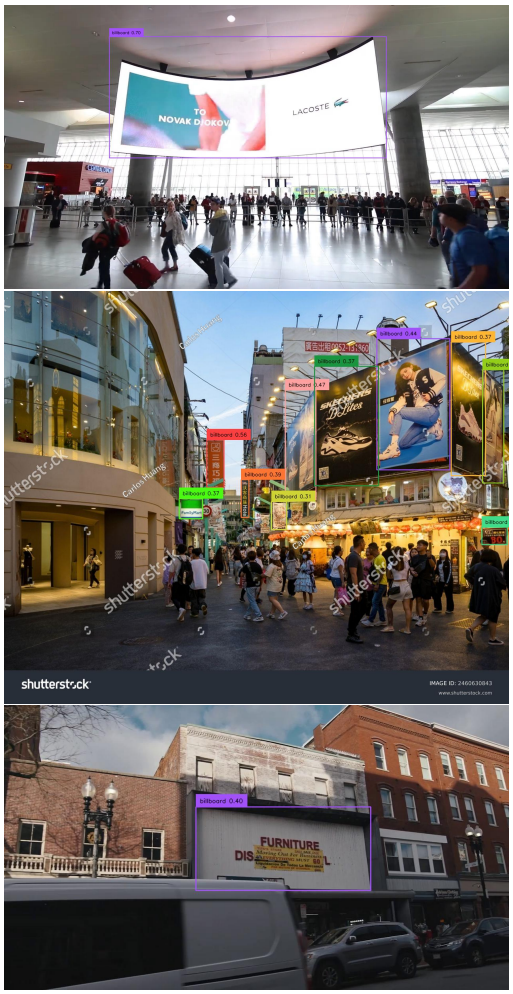

Figure 2. Billboard bounding box detection with GroundingDINO.

test dataset was notably higher, highlighting its stronger generalization capabilities.

We also found the qualitative results from GroundedSAM to be better, and illustrate them in Figure 4. YoloV8+SAM results on same frame are illustrated in figure 3.

Figure 3. YoloV8+SAM for billboard detection and precise mask segmentation.

Figure 4. GroundedSAM for billboard detection and precise mask segmentation

5.5. Results for video object segmentation and billboard tracking

With the billboard detected and segmentation mask generated, we applied Cutie [5] to track the target billboard in

Model	Stage	Images	Instances	P	R	AP50	AP50-95
YoloV8-seg	Validation [17]	100	154	0.705	0.714	**0.758**	**0.551**
YoloV8+SAM	Validation [17]	100	154		0.784	0.625	0.502
GroundedSAM(Zero-shot)	Validation [17]	100	154		0.703	0.652	0.504
Yolov8-seg	Validation [14]	24	30	0.889	0.857	**0.872**	**0.736**
Yolov8+SAM	Validation [14]	24	30		0.432	0.265	0.162
GroundedSAM(Zero-shot)	Validation [14]	24	30		0.636	0.417	0.278
Yolov8-seg	Test [15]	71	107	0.797	0.463	0.535	0.338
Yolov8+SAM	Test [15]	71	107		0.685	0.510	0.434
GroundedSAM(Zero-shot)	Test [15]	71	107		0.795	**0.697**	**0.593**

Table 2. Billboard segmentation (precise mask) accuracy

the rest of the frames. We used the precise mask obtained from the output of either YOLO+SAM model or GroundedSAM as the input for this subsequent stage. The Cutie model was able to track it through the rest of the video very effectively. A demo video of YoloV8+SAM+Cutie can be found here, showing precise full automated segmentation of billboard object through the same video clip using GroundedSAM+Cutie. While both models produce impressive results, GroundedSAM+Cutie exhibits a subtle yet significant advantage in terms of semantically understanding the billboard's composition and identifying the most relevant regions.

5.6. Results for video in-painting

At this point, we have a fully automated pipeline to detect a billboard object with precise mask throughout a video. The last step is to replace this object with another product painting or advertisement image. We explored two different methods for this task, yielding a range of results, some promising and others less successful.

- OpenCV library[4]: Using OpenCV library to do perspective transform after interpreting the 4 corners of detected mask was reasonably effective in placing the image with it looking natural. An example of inpainting with a painting is shown in figure1. And figure 5 shows an example of replacing with another product.

- Stable Diffusion[21]: Using Stable Diffusion with text prompting was fun and worked very well on single frames as shown in figure 6. We also tried using Stable Diffusion with IP adapter, however that also generated images with variations as shown in figure 7. However, the frame-by-frame variation in the generated image made this approach unsuitable for replacing objects in a whole video, particularly when our goal was to insert a specific and consistent product image rather than randomly generated ones frame by frame.

Figure 5. OpenCV inpainting with another product on GroundedSAM result

Figure 6. Stable Diffusion inpainting with text prompting on GroundedSAM result

5.7. End-to-End Pipeline Results

Demo videos for all the steps of the pipeline can be accessed here:

- Original example Video clip:
  https://youtu.be/7lBdDs8xRss
- GroundingDINO object detection:
  https://youtu.be/oDXU9mGAZuM
- GroundedSAM + Cutie segmentation and tracking:
  https://youtu.be/FaD-Y0lxCbQ
- GroundedSAM + Cutie + OpenCV in-painting:

Figure 7. OpenCV perspective transform (left) vs Stable Diffusion IP-adapter (right)

https://youtu.be/kIk4jNpPlH4

- Yolo+SAM+Cutie+OpenCV in-painting:

  https://youtu.be/45ECkRSkdbU?si=6VFB5bmIrYjVpcJn

6. Conclusion & Future Work

Overall the results from the pipeline are quite encouraging. Some conclusions:

- Object detection, segmentation and tracking across the video can be done pretty accurately.
- Transformer based Zero-shot prompting models like GroundingDINO and Segment-Anything Model seem to work better than YOLO models fine-tuned with relevant datasets.
- Cutie performed very well in tracking an object across video if the first mask given is precise and good. Also does not need to be fine-tuned.
- Achieving seamless video in-painting with a replacement image proved challenging using generative AI like Stable Diffusion, particularly given the model scale we employed. OpenCV transforms showed promise but presented difficulties in blending the replacement image with the scene and maintaining stability across frames. This could be a good topic of further research.

7. Contributions & Acknowledgements

7.1. Contributions

Equal contribution by Yawen and Shweta.

- Yawen wrote the code for YoloV8 object detection fine-tuning, GroundingDINO object detection and GroundedSAM segmentation, also performed the accuracy evaluation for object detection and instance segmentation tasks between models.
- Shweta wrote the code for instance segmentation using YOLO-instance-seg, SAM, tracking using Cutie and integration with OpenCV library. And also prepared instance segmentation datasets for testing videos.
- Both did further improvements and exploration of in-painting using Stable Diffusion and OpenCV, and helped with generating final outputs.

7.2. Acknowledgments

We would like to express our gratitude to the open-source community for providing valuable resources that facilitated our research. Specifically, we leveraged data from Roboflow and our implementations from the publicly available GitHub code demos:

- YOLO project:

  https://github.com/ultralytics/ultralytics

- Segment-Anything:

  https://github.com/facebookresearch/segment-anything

- GroundingDINO:

  https://github.com/IDEA-Research/GroundingDINO

- Stable Diffusion:

  https://github.com/runwayml/stable-diffusion

- GroundingSAM:

  https://github.com/IDEA-Research/Grounded-Segment-Anything

- Roboflow:

  https://universe.roboflow.com/

We acknowledge the invaluable contributions of the respective authors and developers who have made these resources accessible to the research community.

We would also like to thank our mentor **Nikil Ravi** and TA **Wenlong Huang** for valuable direction and suggestions in office hours and over emails.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Arslan. Billboard dataset. https://universe.roboflow.com/arslan-ongr8/billboard-xlvz1, sep 2022. visited on 2024-06-02.

[3] D. Bhargavi, K. Sindwani, and S. Gholami. Zero-shot virtual product placement in videos. In *ACM IMX 2023*, 2023.

[4] G. Bradski. The opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.

[5] H. K. Cheng, S. W. Oh, B. Price, J.-Y. Lee, and A. Schwing. Putting the object back into video object segmentation, 2024.

[6] H. K. Cheng and A. G. Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model, 2022.

[7] Y. Cheng, L. Li, Y. Xu, X. Li, Z. Yang, W. Wang, and Y. Yang. Segment and track anything, 2023.

[8] G. Jocher. Yolov8. https://github.com/ultralytics/ultralytics, 2022. Accessed: 2023-05-16.

[9] G. Jocher, A. Chaurasia, and J. Qiu. Ultralytics YOLO, Jan. 2023.

[10] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything, 2023.

[11] F. Li, H. Zhang, H. xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation, 2022.

[12] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, J. Dai, and H. Li. Fuseformer: Fusing fine-grained information in transformers for video inpainting. In *International Conference on Computer Vision (ICCV)*, 2021.

[13] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection, 2023.

[14] C. PP. Dataset - billboard video 2 dataset. https://universe.roboflow.com/cs231n-pp/dataset-billboard-video-2, may 2024. visited on 2024-06-02.

[15] C. PP. Dataset - billboard video 3 dataset. https://universe.roboflow.com/cs231n-pp/dataset-billboard-video-3, may 2024. visited on 2024-06-02.

[16] C. PP. Sports videos dataset. https://universe.roboflow.com/cs231n-pp/sports-videos, may 2024. visited on 2024-06-02.

[17] I. Processing. Billboards dataset. https://universe.roboflow.com/image-processing-awivd/billboards-4zz9y, apr 2024. visited on 2024-06-02.

[18] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[19] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[20] RoadEye. Roadeye dataset. https://universe.roboflow.com/roadeye/roadeye, nov 2022. visited on 2024-06-02.

[21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

[22] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[23] A. Siarohin, E. Sangineto, S. Lathuiliere, and S. Tulyakov. STTN: Space-time transformer networks for video restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3051–3060, 2020.

[24] test. billboard detection dataset. https://universe.roboflow.com/test-c8wix/billboard-detection-uo2ld, mar 2023. visited on 2024-06-02.

[25] J. Yang, M. Gao, Z. Li, S. Gao, F. Wang, and F. Zheng. Track anything: Segment anything meets videos, 2023.

[26] L. Yang, Y. Fan, and N. Xu. Video instance segmentation. In *ICCV*.

[27] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023.