Fast-SAM
(Segment Anything Model)

CS-337 Course Project

Arhaan Ahamad *Roll No: 210050016*
Chaitnaya Agarwal *Roll No: 210050038*
Akshat Kumar Gupta *Roll No: 210050010*
Yash Virani *Roll No: 210050170*
Hitesh Kumar *Roll No: 210050066*

I. Introduction to SAM (Segment Anything Model)

The Segment Anything Model (SAM) is a groundbreaking development in artificial intelligence and computer vision. It represents a significant stride in machine learning models' ability to interact with and understand visual data. Developed with a focus on versatility and adaptability, SAM is designed to perform a wide range of tasks related to image segmentation. At its core, SAM is equipped to process and interpret images across various contexts, ranging from everyday objects to complex scenes in diverse environments such as underwater imagery and artistic renditions. This model showcases impressive zero-shot learning capabilities, enabling it to recognize and segment objects it has never encountered during its training phase. Furthermore, SAM's architecture is tailored to accommodate flexible prompts, allowing it to generate high-quality segmentation masks from minimal input and adapt to various use cases.

II. Why Fast-SAM

A. Enhanced Processing Speed

Fast-SAM significantly improves processing speed compared to the original SAM model. This acceleration is crucial for real-time applications where quick response times are essential, such as in autonomous vehicles, real-time surveillance, and interactive systems. The ability to process images swiftly without compromising accuracy makes Fast-SAM a valuable tool in scenarios where time efficiency is as critical as accuracy.

B. Reduced Computational Resources

Fast-SAM's architecture requires fewer computational resources, making it an energy-efficient and cost-effective solution. This aspect is particularly beneficial for applications with limited hardware capabilities or those that need to minimize energy consumption for environmental and economic reasons. The model's efficiency in resource usage broadens its accessibility and applicability, especially in resource-constrained environments.

C. Preserving Performance Quality

Despite its enhanced speed, Fast-SAM maintains a high level of performance quality. It is adept at handling various tasks, including anomaly detection and object segmentation, with comparable accuracy to SAM. This balance between speed and performance ensures that Fast-SAM can be reliably used in a wide range of applications without sacrificing the quality of results.

D. Training Efficiency

Fast-SAM is engineered for high training efficiency, requiring only about 2% of SAM's dataset to achieve comparable performance levels. This significant reduction in data requirement accelerates the training process substantially and diminishes the computational load. Using just a fraction — 2% — of the dataset makes the model more sustainable in terms of computational resources. It facilitates more accessible updates and retraining, adapting swiftly to new data or evolving requirements.

E. Architecture of Fast-SAM

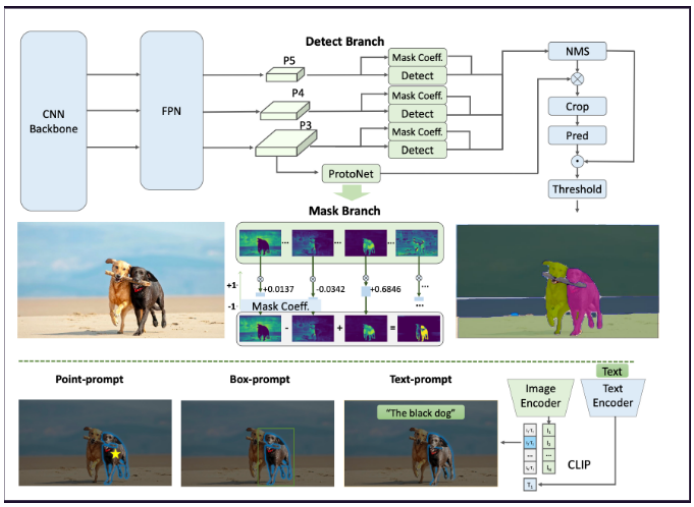

Fig. 1: Fast_SAM's Architecture

Refer to Figure 1 for a visual representation of the Fast-SAM architecture. Built upon a Convolutional Neural Network (CNN)-based architecture, Fast-SAM diverges from SAM's

Transformer-based structure, leading to increased speed and reduced computational demand. The two-stage process entails all-instance segmentation and prompt-guided selection. The first stage employs a CNN backbone and FPN for detecting and segmenting all image objects. The second stage uses prompts to select specific objects, a method that simplifies segmentation and enables real-time operation. This architecture capitalizes on the strengths of CNNs for image analysis and is optimized for speed, without significantly compromising segmentation accuracy. Consequently, Fast-SAM stands out as an efficient alternative to SAM, especially in scenarios where speed and resource optimization are essential.

III. About our Model

Our project focuses on image segmentation and analysis, employing a pipeline with two main segments. The first segment involves generating various possible segments given an input image. The second segment takes a text prompt and decides which segment to select, providing a method for intelligent and context-aware image segmentation.

IV. Model Description

A. Segmentation Models

We utilize a combination of state-of-the-art segmentation models to generate diverse segments for a given input image. The models include:

- **DETR (Detection Transformer):** A Facebook model based on the Transformer architecture, specifically designed for object detection and segmentation.
- **Mask2Former:** A Facebook model that utilizes the Swin Transformer architecture, fine-tuned on the COCO Panoptic dataset for segmentation tasks.
- **YOLO (You Only Look Once):** An Ultralytics model based on the YOLOv8x architecture, providing real-time object detection and segmentation capabilities.
- **SegFormer:** An NVIDIA model fine-tuned on the Cityscapes dataset for high-quality segmentation, particularly suited for urban scene analysis.

B. Text-Driven Selection

The second segment of our pipeline integrates OpenAI's CLIP (Contrastive Language-Image Pre-training) model. This model is capable of understanding both text and image representations and enables us to rank segmented images based on a provided text prompt. The CLIP model assists in selecting the most contextually relevant segment for a given text description.

V. Approach

1) **Image Segmentation:**
   - Users can choose from various segmentation models (DETR, Mask2Former, YOLO, SegFormer) to process input images.
   - The selected model generates multiple possible segments for a given image, providing a range of perspectives for analysis.
2) **Text-Driven Selection:**
   - Leveraging the CLIP model, users can provide a text prompt that describes the desired features or characteristics.
   - The CLIP model ranks the generated segments based on their alignment with the provided text, helping to identify the most suitable segment.
3) **Visualization:**
   - The pipeline includes visualization steps to showcase the original image, the generated segments, and the selected segment based on the text prompt.

VI. Segmentation Approach

- The DETR model, featuring an End-to-End Object Detection framework with a ResNet-50 backbone, has been effectively trained on the COCO dataset to deliver instance segmentation, showcasing impressive performance in identifying and delineating distinct object instances within images.
- Experiments with YOLOv8, mask2former, and segformer were conducted to evaluate their segmentation capabilities; however, segformer, having been pre-trained on the cityscapes dataset, exhibited limited generalizability when applied to diverse image sets beyond urban scenes.
- Comparative analysis revealed that segments produced by YOLOv8 and mask2former did not match the quality of those generated by DETR, indicating DETR's superior accuracy and effectiveness in creating distinct and precise segmentations of objects in various images.

A. Segmentation result using the DETR model

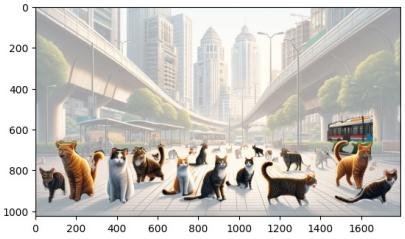

Fig. 2: Segmentation result for the cat prompt using DETR model

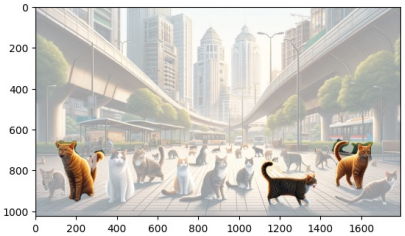

Fig. 3: Segmentation result for the ginger cat prompt using DETR model

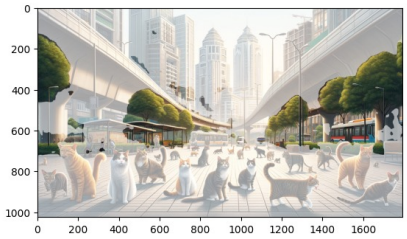

Fig. 4: Segmentation result for the trees prompt using DETR model

Now, here we can see that the DETR model consistently delivers precise results in object segmentation, as illustrated by three distinct images generated in response to prompts for "cat," "ginger_cat," and "trees." Employing panoptic segmentation, DETR excels in accurately distinguishing ginger cats from other feline counterparts within the images. This capability underscores the model's effectiveness in producing fine-grained segmentation results, showcasing its robustness in handling diverse object classes. These visual demonstrations reinforce DETR's superiority over other models such as SegFormer, Mask2Former, and YOLOv8, particularly in scenarios requiring nuanced object identification and segmentation.

B. Segmentation result using the Segformer model

Fig. 5: Original image used for segformer segmentation

Fig. 6: Segmentation result for the cat prompt using Segformer on the above image

Fig. 7: Original image used for segformer segmentation

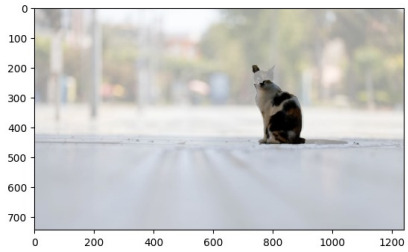

Fig. 8: Segmentation result for the cat prompt using Segformer on the above image

Now, here we can see that as the Segformer model is trained on the Cityscapes dataset, it is able to correctly segment the cars from the image when given the car prompt. However, when attempting the same for segmenting the cat from another image, we obtain a disrupted result. Segformer is not able to correctly segment the cat, as it struggles to accurately identify the feline.

C. Segmentation result using the mask2former model

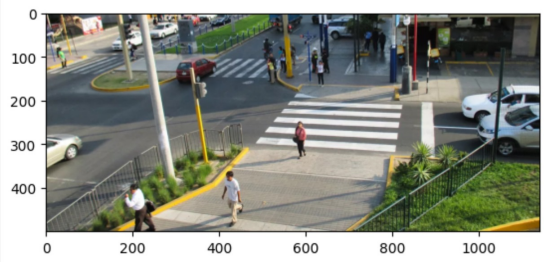

Fig. 9: Original image of a traffic signal processed by Mask2former

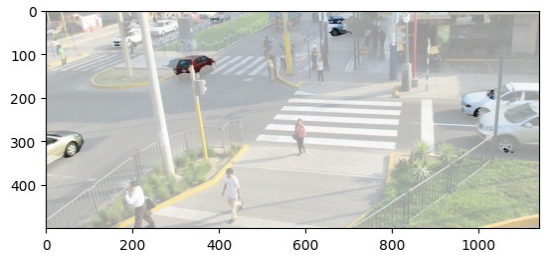

Fig. 10: Segmentation result for the car prompt using Mask2former

Here also, we can see that when given the prompt 'car,' Mask2former is not able to correctly classify every car, leading to incorrect segmentation results.

D. Comparison

The DETR (DEtection Transfomer) model has demonstrated superior object segmentation capabilities when compared to SegFormer, Mask2Former, and YOLOv8. Leveraging a transformer architecture, DETR excels in accurately identifying and delineating objects within images. Its attention mechanisms enable comprehensive context understanding, resulting in precise object segmentation. This stands in contrast to SegFormer and Mask2Former, which may face challenges in capturing nuanced details, and YOLOv8, which relies on grid-based predictions and may struggle with fine-grained segmentation. The DETR model's success in object segmentation highlights

the effectiveness of transformer-based approaches in computer vision tasks, underscoring their potential to outperform traditional architectures in complex visual recognition scenarios.

VII. CLIP

The CLIP (Contrastive Language-Image Pretraining) model is utilised by us to match a text prompt with a given segment of the image. The model embeds images and text in the same vector space and hence we can calculate similarity between a text prompt and an image.Developed by OpenAI, CLIP demonstrates remarkable capabilities in understanding and relating images and text, showcasing a versatile and generalized approach to tasks that traditionally required domain-specific models.

CLIP leverages a novel training paradigm based on contrastive learning, a technique where the model learns to associate similar instances and differentiate dissimilar ones. This approach enables CLIP to grasp intricate relationships between images and their corresponding textual descriptions without the need for paired datasets explicitly linking them. In essence, CLIP is trained to understand the relationships between images and text in a way that allows it to perform a wide range of tasks without task-specific fine-tuning.

At the core of CLIP's success is its ability to learn a shared representation space for images and text. The model is pretrained on a massive dataset that includes a diverse set of images and their associated textual descriptions from the internet. This pretraining process allows CLIP to learn a rich and generalized understanding of the content present in both modalities. Importantly, CLIP does not rely on manually annotated datasets for specific tasks, making it more versatile and applicable to a broader range of applications.

CLIP is the major source of the zero-shot capabilities in the model. The model can generalize its understanding to tasks it has never encountered during training. For instance, CLIP can recognize and describe objects in images, generate textual descriptions for images, and even perform more abstract tasks like answering questions about images or identifying semantically similar images and text pairs. This zero-shot capability showcases the model's versatility and highlights its potential for real-world applications.

VIII. Conclusion

In conclusion, the Segment Anything Model (SAM) and its accelerated counterpart, Fast-SAM, represent a groundbreaking leap in the realm of artificial intelligence and computer vision. Our project adopts Fast-SAM and integrates state-of-the-art segmentation models like DETR, Mask2Former, YOLO, and SegFormer. The CLIP model enables text-driven selection, ranking segments based on textual prompts. This two-segment pipeline facilitates intelligent and context-aware image segmentation. DETR demonstrates superior segmentation accuracy, especially in nuanced object identification. SegFormer, while effective in urban scenes, exhibits limitations in diverse contexts. Mask2Former struggles with precise classification.

CLIP's unique contrastive learning approach empowers it to understand complex relationships between images and text, achieving zero-shot capabilities and versatile applications. It excels in tasks like object recognition, textual description generation, and semantically similar image-text pairing without task-specific fine-tuning.

In conclusion, our project unveils a powerful image segmentation and analysis pipeline, marrying Fast-SAM's speed and efficiency with diverse segmentation models and CLIP's contextual understanding. The result is a versatile solution applicable across various domains, balancing speed, resource optimization, and high-quality segmentation.

References

[1] Ultralytics Roboflow. (n.d.). YOLOv8 Roboflow Segment Prediction. GitHub. Retrieved from https://github.com/roboflow/ultralytics-roboflow/blob/main/ultralytics/yolo/v8/segment/predict.py

[2] CASIA IVA Lab. (n.d.). FastSAM Model. GitHub. Retrieved from https://github.com/CASIA-IVA-Lab/FastSAM/blob/main/fastsam/model.py

[3] Deci AI. (n.d.). Image Segmentation Using YOLO, NAS, and Segment Anything. Deci AI Blog. Retrieved from https://deci.ai/blog/image-segmentation-using-yolo-nas-and-segment-anything/

[4] Alec Radford et al. (2021). Learning Transferable Visual Models From Natural Language Supervision. arXiv preprint arXiv:2103.00020. Retrieved from https://arxiv.org/abs/2103.00020

[5] Xu Zhao et al. (2023). Fast Segment Anything. arXiv preprint arXiv:2306.12156. Retrieved from https://arxiv.org/abs/2306.12156

[6] Alexander Kirillov et al. (2023). Segment Anything. arXiv preprint arXiv:2304.02643. Retrieved from https://arxiv.org/abs/2304.02643

[7] Nicolas Carion et al. (2020). End-to-End Object Detection with Transformers. arXiv preprint arXiv:2005.12872. Retrieved from https://arxiv.org/abs/2005.12872

[8] Enze Xie et al. (2021). SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. arXiv preprint arXiv:2105.15203. Retrieved from https://arxiv.org/abs/2105.15203

[9] Joseph Redmon et al. (2016). You Only Look Once: Unified, Real-Time Object Detection. arXiv preprint arXiv:1506.02640. Retrieved from https://arxiv.org/abs/1506.02640