UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Enhancing high-resolution segmentation using a low-resolution dataset

Author:
Bohdan SYDOR

Supervisor:
Ostap VINIAVSKYI

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science

in the

Department of Computer Sciences and Information Technologies
Faculty of Applied Sciences

Lviv 2023

Declaration of Authorship

I, Bohdan SYDOR, declare that this thesis titled, "Enhancing high-resolution segmentation using a low-resolution dataset" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"The World Tried to Catch Me but Could Not."

Skovoroda

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Enhancing high-resolution segmentation using a low-resolution dataset

by Bohdan SYDOR

Abstract

High-resolution images, now a common standard, presents challenges for existing segmentation approaches due to increased computational and memory requirements. This study aims to improve existing high-resolution image segmentation methods in terms of memory usage, processing time and accuracy. ...

Acknowledgements

I'd like to convey my heartfelt appreciation to all those who have been a part of my journey during these four years of academic pursuit. I'm particularly grateful to my advisor, Ostap VINIAVSKYI, whose consistent support and prompt responses to my queries have been invaluable...

Contents

List of Figures

List of Tables

List of Abbreviations

NN	**N**eural **N**etwork
SAM	**S**egment **A**nything **M**odel
CRM	**C**ontinuous **C**efinement **M**odel
SAM-SR	**S**AM-guided **S**egmentation **R**efinement

Dedicated to humanity

Chapter 1

Introduction

1.1 Problem description

Semantic segmentation - the process of image pixels classification based on their semantic information. In role of semantic information can be affiliation to a specific object, such as a cat, table, etc. Image segmentation can be separated into two categories; Semantic segmentation - pixels classification based on affiliation to a specific semantic class, and instance segmentation - pixels classification based on their semantics and association to the particular object.

In contrast to object detection using bounding boxes, segmentation techniques must generate accurate object boundaries. A detailed understanding of all object parts is essential. Predictions should be based not only on global but also local information to produce precise boundaries

Modern cameras now commonly capture 4K and even 8K images, making high-resolution images the new standard. Higher image resolutions enable the recording of greater details, which is crucial for various post-processing tasks, such as object removal, industrial defect detection, and more. However, existing segmentation approaches often struggle to directly handle these large-scale images while maintaining accuracy. Processing high-resolution images requires more computational resources and memory, which pose challenges for many segmentation methods. Furthermore, as most publicly available datasets consist of images with resolutions below 1K pixels, training segmentation models on high-resolution data using open datasets becomes difficult in many cases. Straightforward approaches to managing high-resolution images include downsampling and cropping. Nonetheless, downsampling can lead to the loss of details, while cropping may remove crucial context from the image.

1.2 Contribution

In recent years, several approaches have been proposed for high-quality segmentation of high-resolution images. The primary concept behind these methods is to first perform segmentation on a lower-resolution version of the image, followed by the application of a class-agnostic refinement step to enhance the mask.

This study examines high-resolution image segmentation methods and suggests improvements for memory, processing time and accuracy. Work includes:

- Evaluation of existing methods: Reviewing current techniques by comparing performance, memory, and processing speed to find ways to improve.
- Memory and processing optimizations: Suggesting changes based on the evaluation, using memory usage flattening and region of interest inference

- New refinement approach: Introducing a mask refinement technique based on near foundational vision model SAM
- Validation and benchmarking: Testing our improvements and proposed refinement method and comparing them with leading methods.

Chapter 2

Related Works

2.1 Segmentation methods

Classical image segmentation techniques typically provide rapid processing and good interpretability, making them suitable for use in controlled environments with low variability. They generally perform well when objects are clearly visible, exhibit simple shapes, possess distinct colors, and occupy fixed positions. However, in cases involving noise, varying object positions and poses, complex semantics, and other factors, classical approaches may not be applicable or effective

Fully connected neural networks (NNs) were initially employed for object detection tasks; however, their lack of spatial understanding of images led to their replacement by convolutional neural networks (CNNs). At present, CNNs are considered the standard for segmentation. They pose hierarchical feature learning, spatial awareness, parameter sharing, reduced complexity compared to MLP, translation invariance. Later work includes PSP-Net [Hengshuang Zhao and Jia, 2017], DeepLab [Liang-Chieh Chen and Adam, 2018] series methods, and other works

In all these methods, the output mask is predicted on resolution downsampled by 4 or 8 times and then bilinearly upscaled to match the final resolution. This approach serves to decrease memory usage and processing time, but it comes at the cost of reduced accuracy for the predicted masks. The downscaling and upscaling processes can lead to the loss of finer details and less precise object boundaries in the final segmentation output.

FIGURE 2.1: Training strategy of Patch-Based Segmentation

Also, patch based segmentation can be used for handling memory usage constraints, by reducing peak memory usage. During training, the input image is resized to the target scale and randomly cropped to fit the neural network's input size. During inference, images are divided into patches, processed separately, and then merged to create the final prediction mask. This approach provides a basis for handling high-resolution images, but the loss of context and potential increase in inference time highlight the need for alternative solutions.

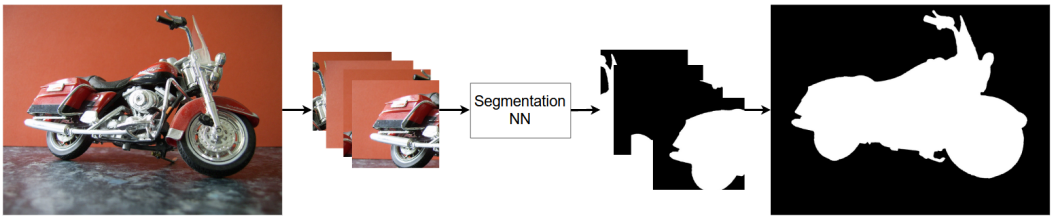

FIGURE 2.2: Inference strategy of Patch-Based Segmentation

The recently proposed LGNet (Wuyang Chen, 2021) combines both local and global context to improve segmentation results. The "L" in the name represents local, while the "G" stands for global context. This method integrates cropped images for local precision and resized images for global context. While the approach enhances outcomes, it still necessitates the use of high-resolution data.

2.2 Segmentation refinement

To deal with the issue of wrong boundary pixels of things in images, different ideas have been suggested. These ideas can be split into two main groups.

The first group can only make small changes to the existing boundaries based on the very close surrounding area. At first, this might seem like enough if you think we just need to fine-tune the boundaries. But, when an image that was split up on a small scale (like 512x512) is then upscaled up to 6k, making small changes to the existing boundaries doesn't really work. Tiny parts of things that you couldn't see at 512x512 become easy to see at 6k, and to include these parts in the original split-up image, looking at the very close surrounding area isn't enough.

Because of these issues, a second group of ideas were suggested. These ideas look at the larger surrounding area and can make changes to bigger parts of the image. By looking at more of the image, these ideas are better at dealing with problems when the boundary pixels of things in the image are wrong, especially when the image is upscaled dramatically

2.2.1 Local segmentation refinement

The SegFix (Yuhui Yuan, 2020) approach serves as an example of local mask refinement in image segmentation. The process it employs for mask refinement involves a couple of key steps: (1) identifying the boundary pixels of the segmented object and (2) determining the corresponding interior pixel for each identified boundary pixel.

Firstly, an image encoder is utilized to extract features for each pixel in the image. Simple backbones such as ResNet50 (Kaiming He, 2015) and EfficientNetB0 (Mingxing Tan, 2020) are often employed for this purpose, particularly for encoding images of nearly 4K resolution in a single pass. These encoders are selected to ensure the process fits within the time and memory constraints.

Subsequently, two branches are employed. The first branch is tasked with detecting pixels located on the object's edges. The second branch, on the other hand, is responsible for predicting offsets for the edge pixels. The authors of the SegFix method posit that interior pixels are generally more robust compared to boundary pixels. Therefore, for each boundary pixel, an interior pixel is assigned using the predicted offsets.

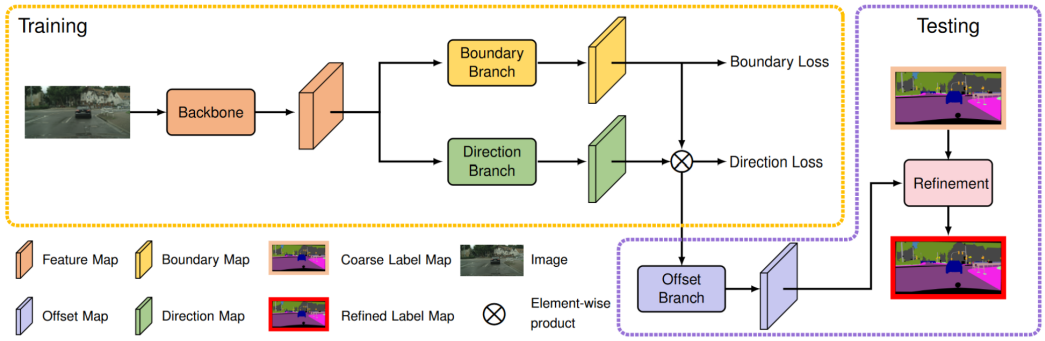

FIGURE 2.3: Illustrating the SegFix framework. Figure from the original paper

During the model training phase, the model is trained to segment edges and predict offsets. In the inference phase, the predicted offsets are utilized to reassign classes on the predicted masks, thereby refining them.

One important downside of this method is that it doesn't use information from the predicted masks. This makes it much harder to find the edges of the object that need to be improved. Figuring out the direction of the offset, or the amount the object needs to be shifted, also becomes more challenging.

Another key point is that this method needs high-resolution data to learn from. Usually, this method is trained on images that are nearly 1K in size, and then it's used on bigger images. However, this transfer from smaller to larger images isn't as efficient when compared to newer methods.

2.2.2 Global segmentation refinement

Global refinement methods, by taking into account the overall context of an image, can achieve more accurate refinements when the image has undergone significant resizing. This attribute is particularly crucial when there exists a considerable disparity between the segmentation scale and the original image scale, for instance, when the gap is over five times greater.

CascadePSP [Kei Cheng and Tang, 2020]

In 2020, the CascadePSP method was introduced as a solution for refining image masks using both global and local contexts. Notably, this method does not necessitate high-resolution data for training, owing to its efficient generalization capabilities to scales beyond the training distribution.

A crucial component of this technique is the refinement module, which comprises an encoder and a decoder. It bears some resemblance to the UNet architecture, but before the decoder, it fuses information from all encoder scales while preserving the original skip connections.

The refinement process consists of two steps. First, the global refinement step refines larger parts of masks based on the global context. Three refinement modules are employed for this purpose, working together to use skip connections from previous modules while refining features across different scales simultaneously.

Next, local mask refinement is applied following the global refinement step. Its objective is to refine masks using high-resolution local information. The image is divided into patches of a specific size, "L", which are processed individually. The

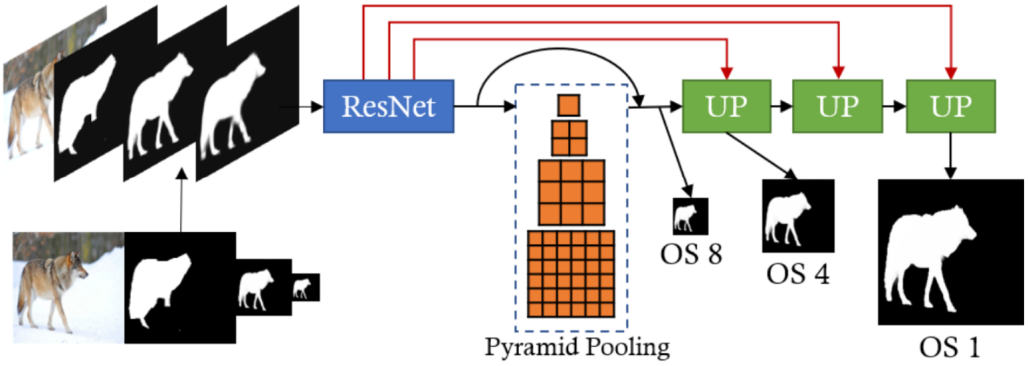

FIGURE 2.4: Illustrating the CascadePSP refinement module. Figure from the original paper

local refiner's architecture is similar to that of the global refiner, but it uses only two refinement modules instead of three.

During training, a single refinement module is utilized, with the learned weights shared among all other refinement modules during the inference stage. By separating the process into global and local steps, CascadePSP refines masks more efficiently than previous approaches.

CRM [Tiancheng Shen, 2021]

In 2021, an current SOTA approach inspired by LIIF (Yinbo Chen, 2021) was introduced. LIIF (Learning Implicit Image Functions) is utilized for continuous image representation, which enables flexible scaling of images. Initially, an image is encoded into a continuous representation using an encoder, which can then be decoded to any desired scale. The main advantage of LIIF lies in its capacity to upscale images to significantly larger scales than those encountered during the training phase. For example, in the LIIF paper, the model was trained to upscale images between 2-6 times their original size; however, during inference, it was capable of upscaling images by up to 30 times while still preserving acceptable image quality.

Assuming that high-resolution images are not available during training, LIIF's ability to work outside the training distribution becomes highly beneficial. In the paper "High-Quality Segmentation for Ultra High-Resolution Images," the authors utilize this property for the refinement task. They replace the original LIIF encoder with a more lightweight ResNet50 and incorporate other ideas from LIIF.

Each pixel is individually decoded based on the image and mask embeddings, while the original LIIF uses only image embeddings for super-resolution of the image. CRM utilizes not only the information encapsulated in the features, but also additional metadata concerning the pixel's position relative to these features.

The model is then trained to refine masks. During inference, a cascade strategy is employed to refine both global and local details of the mask, allowing for improved segmentation results even in the absence of high-resolution training data

2.2.3 CRM edge cases

In the analysis of the top-performing approaches for refining semantic segmentation masks, as proposed in the "High Quality Segmentation for Ultra High-resolution

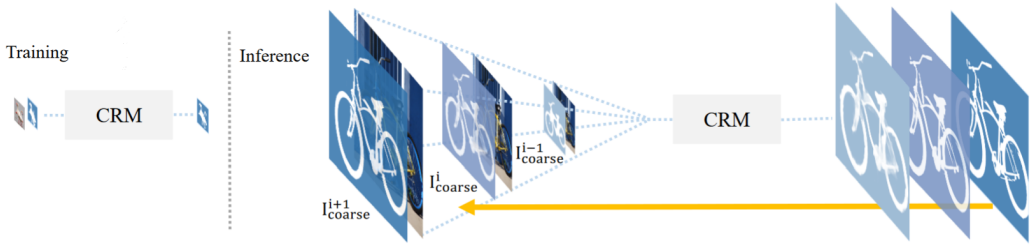

FIGURE 2.5: Illustrating the CRM refinement module training and inference. Figure from the original paper

Images" paper, it has been discovered that current models often struggle with understanding context. The leading model, CRM, utilizes ResNet50 as its only encoder to address memory limitations. However, this does not provide enough context understanding, as shown in Figure 2.4.

CRM primarily depends on color and pattern similarities. For instance, the front glass of a motorcycle is missed due to its resemblance to the background. Additionally, the inner part of a cow is incorrectly segmented due to the sudden color change from brown to white. When objects have initially poor edge quality, the model faces difficulties in accurately reconstructing elements like bottles.

The primary objective of this work is to enhance the deep contextual understanding of the mask refinement model. Through the experiments conducted in the subsequent section, it is demonstrated that the proposed approach effectively improves the behavior of the refiner across all tested scenarios.

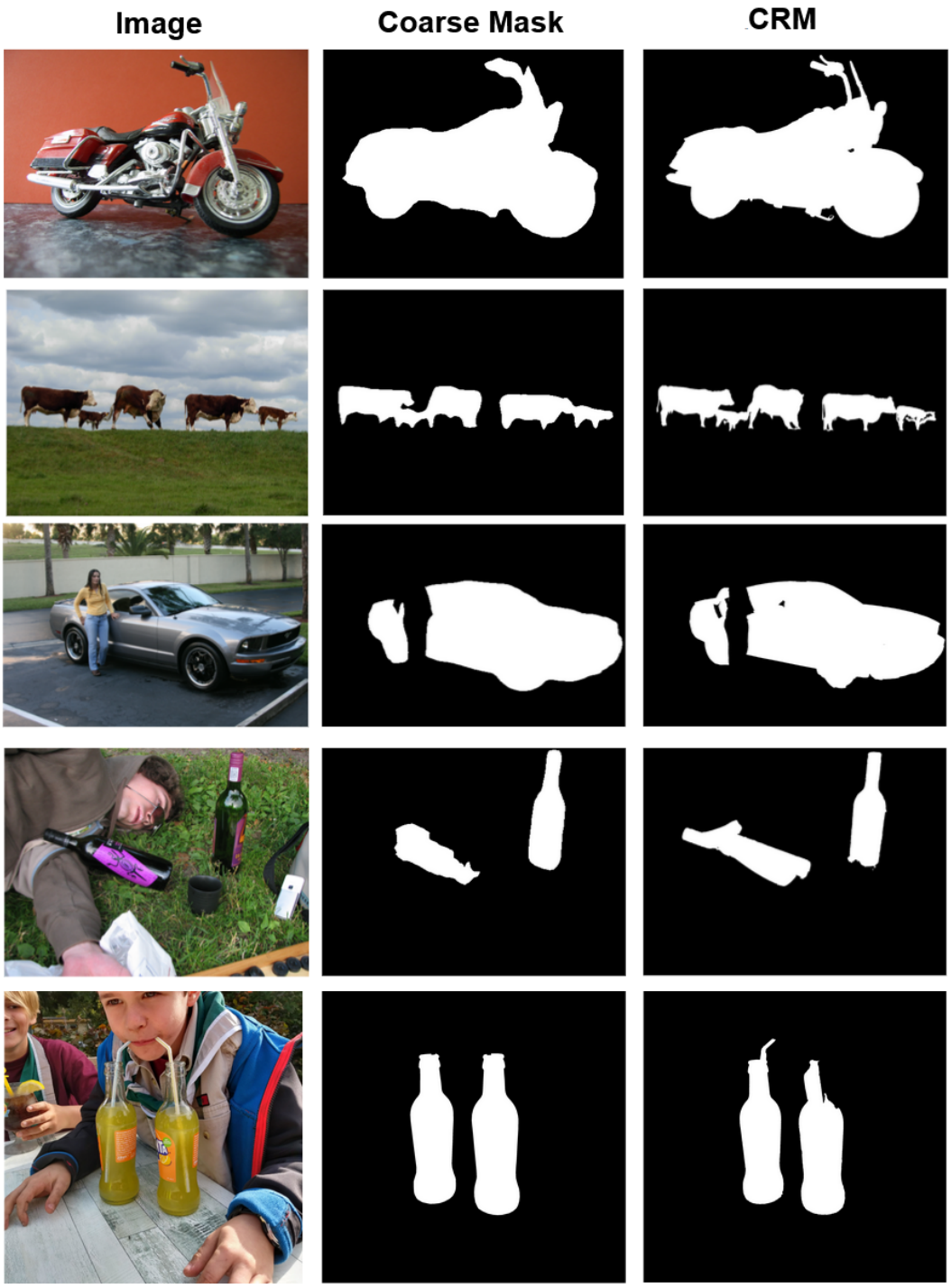

FIGURE 2.6: CRM edge cases visualization

Chapter 3

Method

3.1 CRM compression and acceleration

3.1.1 Memory

CRM is currently the most efficient approach regarding memory usage and inference speed, but it still uses many resources. In this section memory usage optimization approach is proposed. In most cases, NNs do not use memory uniformly, and there is always room for improvement. After profiling the CRM module, were noticed that there is a present memory usage spike which increases the memory requirement for the system by 1.5 times. During the features fusion from different scales, all features are interpolated to the largest scale spatial size and passed to convolution layers, which fuse them before the decoder. This operation requires a lot of memory if we try to fuse it all at once. But if split convolution into a few separate convolution operations on features patches and after predictions are merged into one features map, the peak memory usage can be reduced a lot.

3.1.2 Speed

Proposed Inference time optimizations use the fact that often most of the pixels on the mask are left the same after refinement, and there is no need to run a decoder for all of them. The straightforward optimization approach runs only on padded bounding boxes of objects on the mask. This optimization helps in cases when the object is small, but when the mask covers most of the image, then there will be no boost in speed.

The second optimization approach runs the decoder only on pixels near the rough mask's contours. The idea is that pixels far from contours will not be modified.

The last approach initially runs the decoder only on every nth pixel and decodes all other pixels. If the nearest four pixels are decoded with the same value, then this pixel preserves its value. If at least one of the nearest initially decoded pixels changes its value, it is also decoded by the refiner.

3.2 Segment Anything Model [Alexander Kirillov, 2023]

It's been observed that current mask refinement approaches have difficulty with contextual interpretation. They generally depend on basic patterns and similar colors, failing to deeply integrate context information into the image's description.

Recently, a novel model called the "Segment Anything Model" (SAM) was introduced, aimed at prompt-based segmentation. What makes SAM notable is its ambition to become a foundational model in the realm of image understanding. To

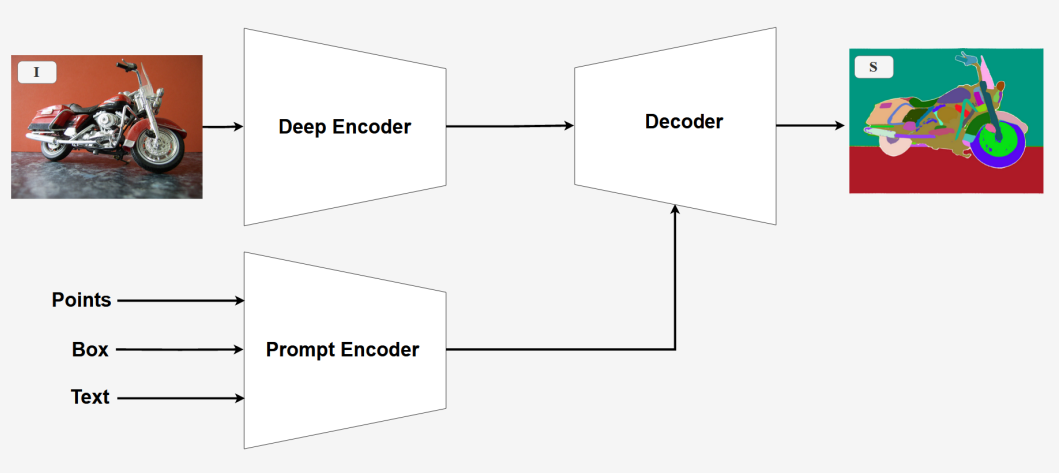

FIGURE 3.1: SAM pipeline

realize this, the model was trained on a dataset of 11 billion images. These images cover a wide range of areas, making the dataset diverse and comprehensive. It's important to note that the data represents one of the two critical components of a generic segmentation model. The team behind SAM has pushed the limits of existing dataset sizes in terms of image quantity and diversity.

SAM's architecture is built on two core parts Fig.3.1: the Vision Transformer (ViT Alexey Dosovitskiy, 2021) encoder and lightweight dencoder. Initially, images are resized to 1024x1024 pixels. Then, a 16x16 convolution is used to derive 1024 generic patterns from the entire image. The ViT encoder is then applied, which groups feature vectors with similar meanings to vectors with high similarity.

After the encoding process, masks can be pulled out using prompts with the assistance of the simple encoder. This encoder includes two transformer blocks that attends the features and prompts in both directions, a transposed convolution on the features, and a dot product operation with the prompts to generate the final mask.

SAM does not have specific domain restrictions, making it potentially valuable for improving the contextual understanding of current refinement methods. This study suggests various ways to incorporate SAM into the existing CRM pipeline.

3.3 SAM-guided Segmentation Refinement (SAM-SR)

This work proposes three conceptually different approaches to SAM integration with existing SOTA mask refiner CRM. They can be split by the level of integration between two pipelines. The first type of integration proposes IoU baed integration, where two pipelines runs sequentially, but without any complex integration between them. The second approach still runs SAM and CRM sequentially, but here CRM is trained to use SAM output as additional input. The last approach proposes four ways for SAM ViT encoder usage in CRM pipeline.

3.3.1 IoU based SAM-SR (SAM-SR-v1)

The IoU-based approach uses a uniform grid of points as prompts Fig3.2. Each point corresponds to a separate point and produces three masks. Three is a determined number of masks from SAM, corresponding to different semantic similarity thresholds. After NMS is applied to all masks to merge similar masks. In the result SAM

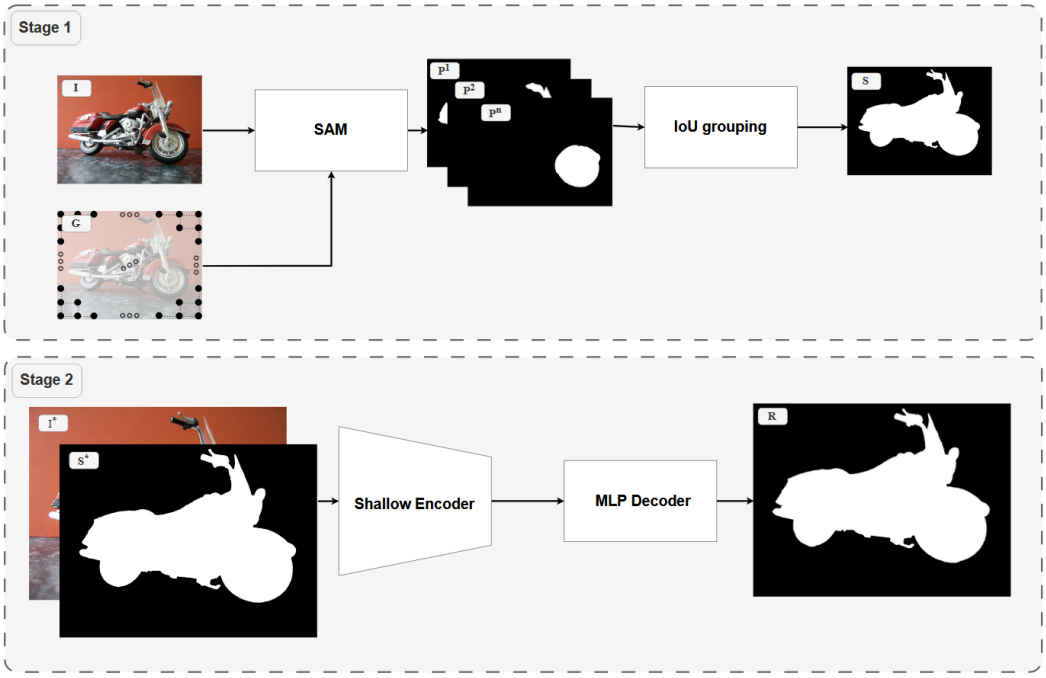

FIGURE 3.2: SAM-SR-v1 pipeline

itself produces set of masks $P^1, P^2, ...P^n$ in average n is equal to 900. After that, produced masks are overlapped with masks that should be refined. If the SAM mask is covered enough with the input mask, then it is preserved, else it's dropped. All preserved masks are merged into one and passed to CRM instead of the original input mask.

Experiments reveal that SAM struggles with segmenting sparse objects like bicycles and produces for them solid masks that ignore holes in objects, which dramatically degrades refinement quality. To handle this initial mask is analyzed on the sparsity coefficient, and if it's space enough, then the SAM step is skipped.

Also, sometimes SAM doesn't have suitable masks to cover the input mask perfectly; in such cases deformation coefficient after the SAM, step is analyzed. If it is too big, then the SAM step is also skipped.

This approach aims to validate SAM masks quality and determine weather they can be used for masks refinement.

3.3.2 Semi-integrated SAM-SR (SAM-SR-v2)

The second group of SAM integration approaches focuses on using SAM output as additional input to the CRM module. The main idea is to provide CRM preprocessed semantic information in a simple form to improve refinement quality.

SAM-SR-v2.1

SAM-SR-v2.1 Fig 3.3, in terms of prompts to SAM, works like SAM-SR-v1, but in the end, masks are sorted by area and merged to RGB with random colors assigned to each mask. Predicted RGB images with simplified semantics are used as additional input to CRM.

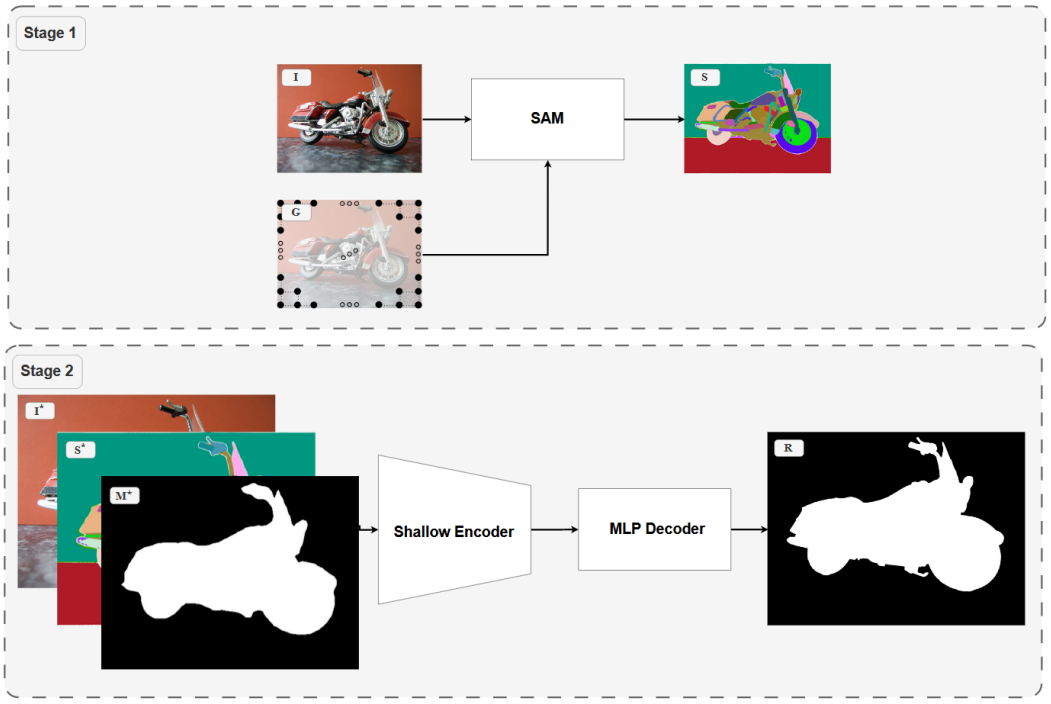

FIGURE 3.3: SAM-SR-v2.1 pipeline

Even though SAM works with images of size 1024x1024, it still can be utilized for ultra-high resolution mask refinement. After the visualization of the SAM output, gaps between masks can be noticed. This gaps highlights unconfident regions which, after, can be refined by CRM. In this case, SAM works like a context simplifier and edge detector at once.

SAM-SR-v2.2

SAM-SR-v2.2 (Fig 3.4) tries to employ SAM for potentially more precise mask prediction based on prompt points sampled from the initial mask. The idea is to validate SAM's ability to resegment rough masks with higher quality. For prompts, the points grid is sampled from a rough mask. After points are grouped in sets of three points, each set predicts the mask and confidence score for this mask. To prepare additional input for the CRM module, masks are merged based on confidence score. The IoU-based approach uses SAM masks instead of the initial rough mask, while SAM-SR-v2.2 provides the initial mask and mask after SAM refinement to the CRM module. Such an approach provides additional flexibility in terms of work with SAM.

3.3.3 SAM encoder (SAM-SR-v3)

As experiments showed, the previous approaches improved accuracy, leading to the decision to integrate the encoder into the CRM module. This integration aims to simplify the pipeline and incorporate mask information more flexibly. In SAM-SR-v1 and SAM-SR-v2, features from the encoder are transformed into simple images on output. During this process, a lot of deep semantic information is lost. SAM-SR-v3 (Fig 3.5) proposes using ViT features directly.

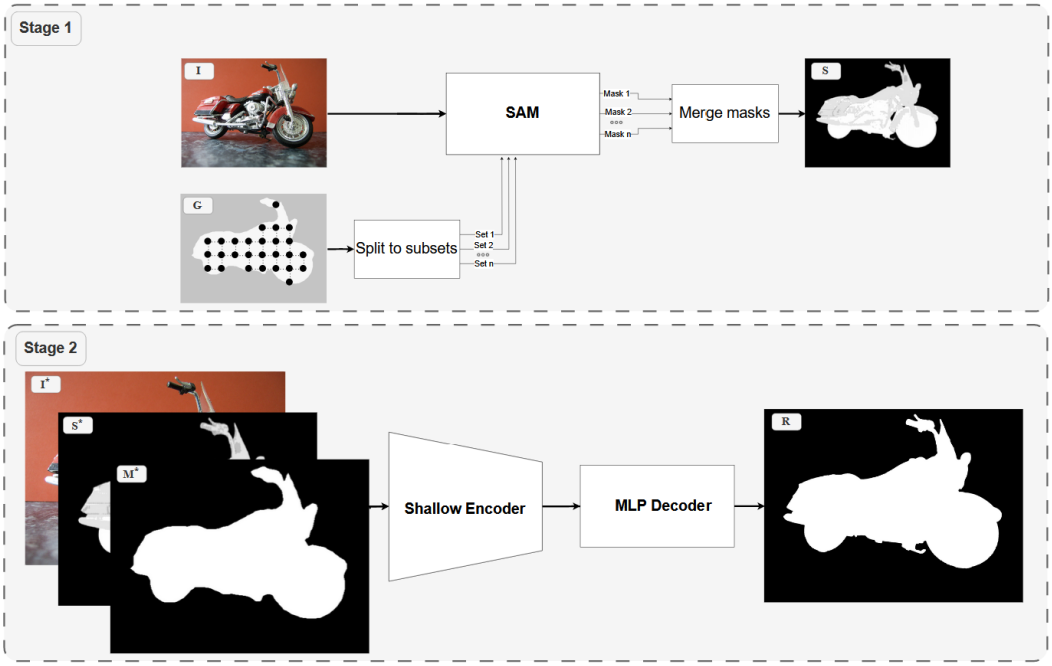

FIGURE 3.4: SAM-SR-v2.2 pipeline

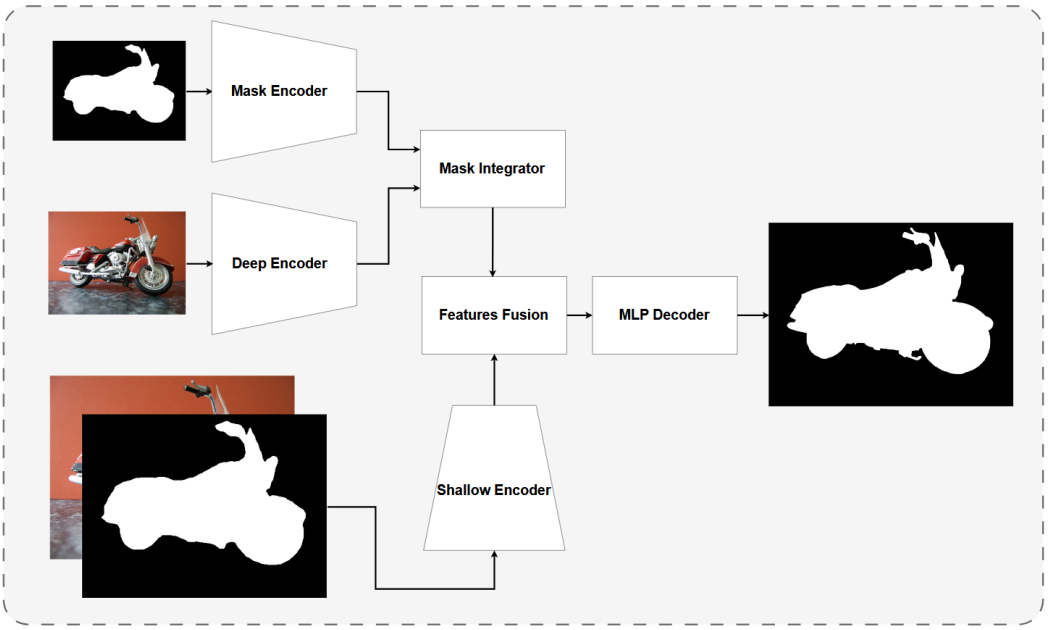

FIGURE 3.5: SAM-SR-v3 pipeline

SAM encoder utilizes 16x16 convolutions initially, resulting in the significant loss of local information crucial for precise boundary refinement. To handle the loss of high-level local semantic information shallow features encoder is used in pair with the SAM encoder. The shallow features encoder is designed to preserve local information, while the SAM encoder provides robust contextual information.

To incorporate mask information into ViT features, different approaches are proposed. CNN-based approach encodes mask to ViT domain features and directly adds mask information over ViT features. Transformer-based methods also use CNN encoder for the mask. After, the transformer block assigns a weighted sum

of mask feature vectors for each image feature vector. Weights are calculated based on the correlation between the current and other image-embedding vectors. The idea is to refine mask features using semantic similarity of image embeddings. After refinement, maks embeddings also are added over to image embeddings.

SAM-SR-V3.1 uses only ViT features without integration of any masks information. SAM-SR-V3.2 uses ViT features with integration of mask features by mask embedding addition. SAM-SR-V3.3 uses ViT features without with integration of mask features through multi-headed global attention block. SAM-SR-V3.4 uses ViT features without with integration of mask features through multi-head windowed attention block.

Before the decoder, features from deep and shallow encoders are processed through CNN features fuser. As in CRM, MLP pixel-wise decoder is used to decode refined mask.

Chapter 4

Experiments

4.1 Patch-based segmentation

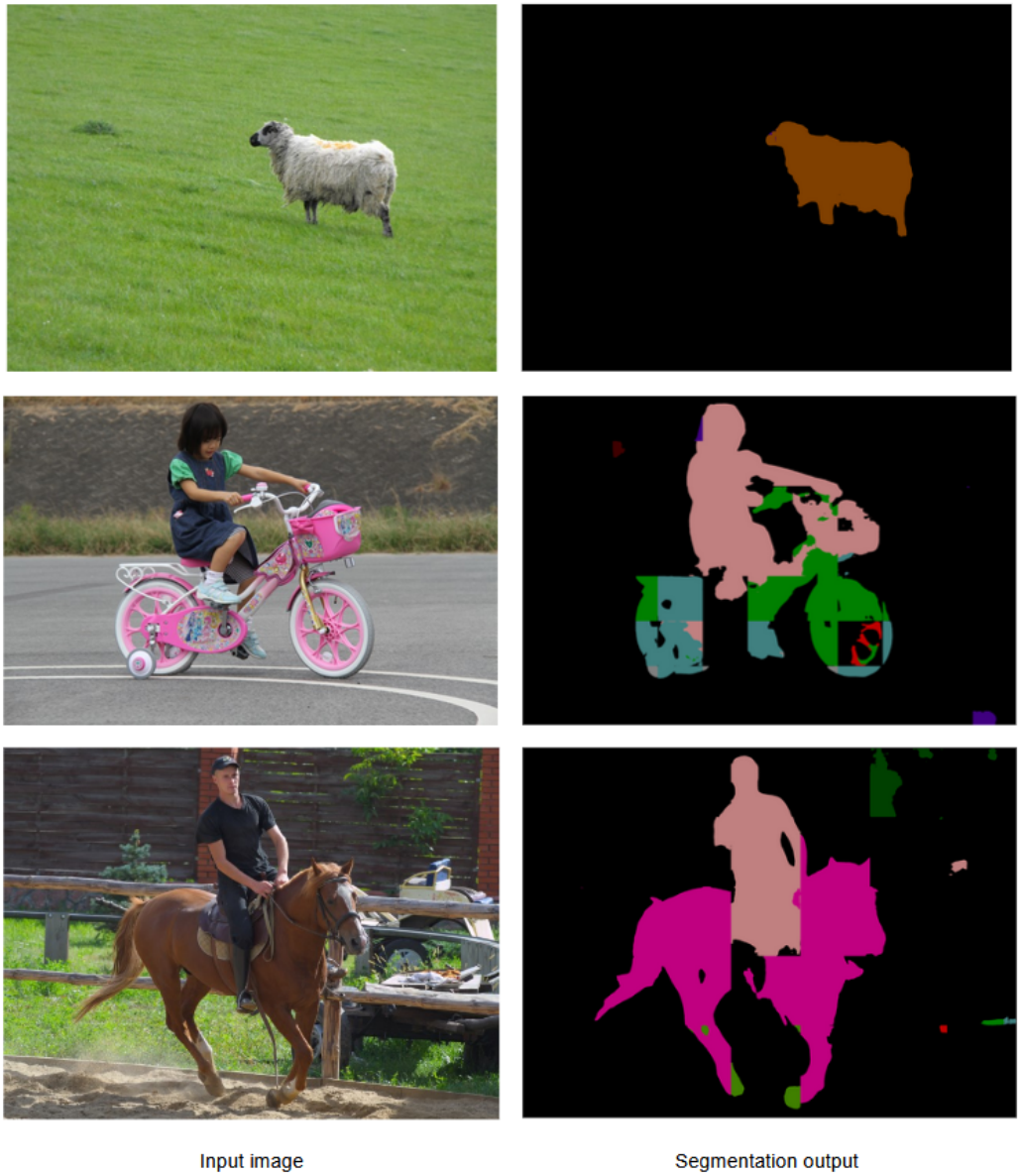

FIGURE 4.1: Illustrating patch-based approach results

Name	Peak memory usage (GiB)	Avg inference time (sec)	IoU/mBA
Original	23.8	8.97	94.18/76.09
Optimized	16.25	9.10	94.18/76.05

TABLE 4.1: Memory usage optimization. Avg inference time is calculated on test set of BIG dataset.

The primary objective of this experiment was to demonstrate the limitations of the straightforward patch-based approach in segmenting high-resolution images. The popular segmentation framework PSPNet was chosen for its proven effectiveness across a variety of segmentation domains. To enhance the model's data-fitting capabilities, ResNet101, the largest backbone in the ResNet family, was employed as the encoder.

PASCAL Visual Object Classes (PASCAL VOC) dataset was used for training, while the BIG dataset, selected for evaluation, due to its domain similarity with PASCAL VOC and its inclusion of very high-resolution images and masks. The original training methodology provided by the authors of PSPNet was followed. The only difference is the an increase in the number of epochs to account for the slower convergence rate of the model. Images from the training dataset were upscaled to sizes between 2k and 6k and subsequently cropped to 512x512 resolution. The experiment aimed to train the model to segment patches of 2k-6k images.

A notable observation was the increase in convergence time from the 8-hour estimate provided by the PSPNet authors for the original scale to 4 days for the crops. This can be attributed to the fact that patches often lacked sufficient information for accurate object class identification.

As the results indicate (Fig. 4.1), correct predictions were made when patches contained unique object details, but incorrect classifications occurred for inner parts of objects that lacked any useful information. This experiment reveals that the patch-based segmentation approach is not suitable for segmenting very high-resolution images.

4.2 CRM compression and acceleration

Memory usage. As experiments show (Tab. 4.1), patched convolution approach allows for a decrease in memory usage significantly while almost preserving inference time.

Inference time. Inference time is calculated on a test set of BIG dataset. The size of images is near uniformly distributed between 2k and 6k. Metrics are measured on PSPNet output refinement. Table 4.2 shows inference time optimization experiments results.

For box-based optimization precision lefts near the original but slightly lower. That is because the box limits mask refinement ranges, and sometimes this range is not sufficient to reproduce the original refinement effect.

Contour-based optimization speeds up inference better while adding more constraints on refinement flexibility and leading to a more significant drop in metrics, but they still are near the original.

Stride-based approach does not introduce any constraints on the refinement region and preserves metrics nearly the same while decreasing inference time more than two times. Different stride sizes were tested (Tab. 4.2).

Name	Avg inference time (sec)	IoU/mBA
Original	8.97	94.18/76.09
BBox	5.62	94.17/75.79
Contour	4.56	93.86/75.35
Strided 2	5.02	94.15/76.06
Strided 4	3.97	94.14/75.85
Strided 8	3.72	93.95/74.73

TABLE 4.2: Inference time optimization.

The best shows strided inference with stride size four. It decreases inference two times while practically preserving refinement quality.

4.3 SAM-guided Segmentation Refinement (SAM-SR)

4.3.1 Training details

In the training phase, the original procedure established by the CRM was used. The model was trained using various data sources to ensure a comprehensive training set. Specifically, we used four datasets - MSRA-10K (detection, 2014), DUT-OMRON, EC-SSD, and FSS-1000.

These individual datasets were brought together to create a large and diverse training set. The combined training set consists of 36,572 images with diverse semantic classes (>1,000 classes).

4.3.2 Evaluation details

Dataset. For the evaluation BIG dataset was used. Currently, it's standard for the evaluation of refinement methods. It contains images from 2k to 6k with very precise masks. Rough masks are produced by FCN-8s (Jonathan Long and Darre, 2015), DeepLabV3+ (Liang-Chieh Chen and Adam, 2018), RefineNet (Guosheng Lin and Reid, 2017), and PSPNet (Hengshuang Zhao and Jia, 2017) segmentation methods trained on PASCAL VOC.

Metrics. Two metrics for evaluation are used. The first one is intersection over union (IoU).

$$IoU = \frac{\text{area of overlap}}{\text{area of union}}$$

This metric focuses in general mask quality and calculates ration between intersection of predicted mask and gt mask to the union of this masks. The second metric is mean boundary accuracy (mBA) is used to evaluate the precision of object boundary predictions. This metric calculates IoU of fround truth and predicted masks on their boundaries with different dilation intensity of contours and calculating average between all intensities.

4.3.3 Results analysis

SAM-SR-v1 and SAM-SR-v2.1 show a huge improvement in metrics compared to the original CRM. Original CRM often incorrectly removes or adds significant regions

IoU/mBA	Coarse Mask	Original CRM	SAM-SR-v1
FCN-8s	72.39/53.63	79.62/69.47	**79.71/70.07**
DeepLabV3+	89.42/60.25	91.84/74.96	**93.00/76.69**
RefineNet	90.20/62.03	92.89/75.50	**93.55/76.93**
PSPNet	90.49/59.63	94.18/76.09	**94.45/77.17**
Avg. Improve.	0.00/0.00	4.01/15.12	**4.55/16.33**

TABLE 4.3: SAM-SR-v1 metrics

IoU/mBA	Coarse Mask	Original CRM	SAM-SR-v2.1	SAM-SR-v2.2
FCN-8s	72.39/53.63	79.62/69.47	**79.92/70.44**	76.19/70.13
DeepLabV3+	89.42/60.25	91.84/74.96	**93.04/76.30**	87.89/74.87
RefineNet	90.20/62.03	92.89/75.50	**93.71/76.35**	88.15/74.93
PSPNet	90.49/59.63	94.18/76.09	**95.09/76.98**	89.55/75.65
Avg. Improve.	0.00/0.00	4.01/15.12	**4.82/16.13**	-0.18/15.01

TABLE 4.4: SAM-SR-v2 metrics.

IoU/mBA	Coarse Mask	Original CRM	SAM-SR-v3.1	SAM-SR-v3.2	SAM-SR-v3.3	SAM-SR-v3.4
FCN-8s	72.39/53.63	79.62/69.47	79.70/69.61	**80.64**/**69.78**	79.74/69.21	79.79/69.12
DeepLabV3+	89.42/60.25	91.84/74.96	91.88/**75.05**	**92.33**/75.00	91.95/74.67	91.57/74.73
RefineNet	90.20/62.03	92.89/75.50	92.93/**75.61**	92.81/75.23	**92.96**/75.39	92.77/75.32
PSPNet	90.49/59.63	94.18/76.09	94.18/**76.25**	93.73/75.82	**94.23**/75.92	93.90/75.82
Avg. Improve.	0.00/0.00	4.01/15.12	4.05/**15.25**	**4.25**/15.14	4.1/14.91	3.88/14.87

TABLE 4.5: SAM-SR-v3 metrics.

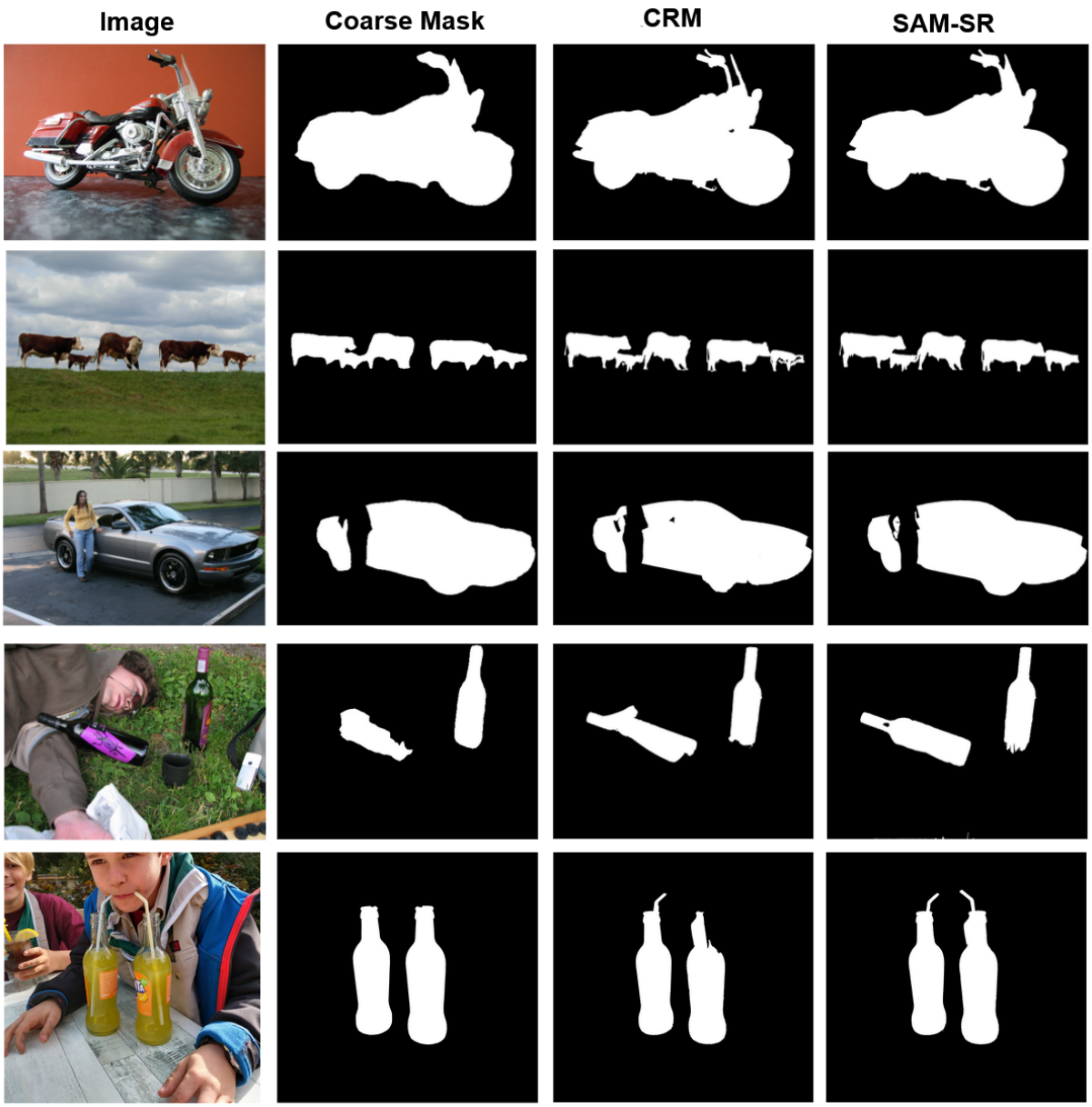

FIGURE 4.2: Illustrating SAM-SR-v2.1 improvement

and harms precision. These methods help to stabilize refiner behavior due to better context understanding provided by SAM. Figures 4.2 and 4.3 show examples of when SAM-SR-v2.1 stabilizes and produces more accurate masks after refinement.

SAM-SR-v2.2 showed that by itself, SAM is not enough to refine masks from points sampled from rough masks. The reason is that sometimes SAM confuses to segment required objects due to the insufficient information introduced by prompts, which often causes ambiguous situations for SAM. This causes even drop in metrics compare to rough masks.

SAM-SR-v3, in general, slightly changes the behavior of the original CRM model, but refinement quality lefts near the same. SAM decoder struggles to handle complex features provided by the SAM ViT encoder. SAM-SR-v2.1 receives simplified semantics and performs much better.

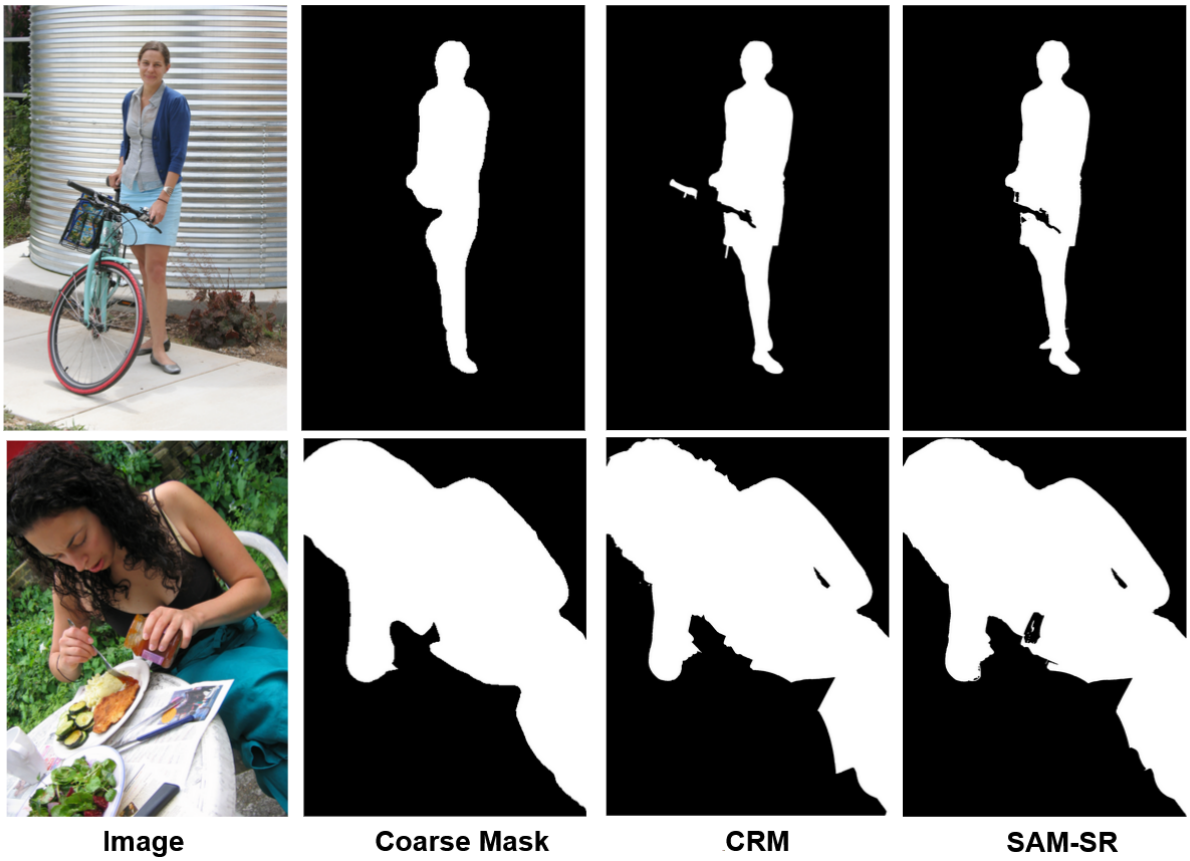

FIGURE 4.3: Illustrating SAM-SR-v2.1 improvement

Name	Avg inference time (sec)	Peak memory usage (GiB)	Avg improvement (IoU/mBA)
Original	8.97	23.8	4.01/15.12
SAM-SR-v2.1	5.7	22.5	4.82/16.13

TABLE 4.6: Summary comparison of previous SOTA refinement method and proposed SAM-SR-v2.1 with inference optimizations

Chapter 5

Conclusion

5.1 Conclusion

This work introduced novel optimization approaches for mask refinement inference, which decreases memory usage by 1.5 times and inference speed more than two times while preserving the same refinement quality.

Also, the main weaknesses of the SOTA refiner were addressed by introducing SAM for better context understanding. Different approaches to SAM integration were tested and achieved +0.8 IoU and +1 mBA.

Finally, after combining all these improvements proposed method outperforms the previous SOTA Tab.4.6 in terms of speed, memory, and accuracy

Bibliography

Alexander Kirillov Eric Mintun, Nikhila Ravi-Hanzi Mao Chloe Rolland Laura Gustafson Tete Xiao Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár Ross Girshick (2023). *Segment Anything*. arXiv: 2304.02643 [cs.CV].

Alexey Dosovitskiy Lucas Beyer, Alexander Kolesnikov-Dirk Weissenborn Xiaohua Zhai Thomas Unterthiner Mostafa Dehghani Matthias Minderer Georg Heigold Sylvain Gelly-Jakob Uszkoreit Neil Houlsby (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. arXiv: 2010.11929 [cs.CV].

detection, Global contrast based salient region (2014). *Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu*.

Guosheng Lin Anton Milan, Chunhua Shen and Ian Reid (2017). *Refinenet: Multi-path refinement networks for high- resolution semantic segmentation*. arXiv: 1611.06612 [cs.CV].

Hengshuang Zhao Jianping Shi, Xiaojuan Qi-Xiaogang Wang and Jiaya Jia (2017). *Pyramid scene parsing network*. arXiv: 1612.01105 [cs.CV].

Jonathan Long, Evan Shelhamer and Trevor Darre (2015). *Fully convolutional networks for semantic segmentation*. arXiv: 1411.4038 [cs.CV].

Kaiming He Xiangyu Zhang, Shaoqing Ren-Jian Sun (2015). *Deep Residual Learning for Image Recognition*. arXiv: 1512.03385 [cs.CV].

Kei Cheng Jihoon Chung, Yu-Wing Tai o and Chi-Keung Tang (2020). *Cascadepsp: toward class-agnostic and very high- resolution segmentation via global and local refinement*. arXiv: 2005.02551 [cs.CV].

Liang-Chieh Chen Yukun Zhu, George Papandreou Florian Schroff and Hartwig Adam (2018). *Encoder-decoder with atrous separable convolution for semantic image segmentation*. arXiv: 1802.02611 [cs.CV].

Mingxing Tan, Quoc V. Le (2020). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. arXiv: 1905.11946 [cs.CV].

Tiancheng Shen Yuechen Zhang, Lu Qi Jason Kuen Xingyu Xie Jianlong Wu Zhe Lin Jiaya Jia (2021). *High Quality Segmentation for Ultra High-resolution Images*. arXiv: 2111.14482 [cs.CV].

Wuyang Chen Ziyu Jiang, Zhangyang Wang Kexin Cui Xiaoning Qian (2021). *Collaborative Global-Local Networks for Memory-Efficient Segmentation of Ultra-High Resolution Images*. arXiv: 1905.06368 [cs.CV].

Yinbo Chen Sifei Liu, Xiaolong Wang (2021). *Learning Continuous Image Representation with Local Implicit Image Function*. arXiv: 2012.09161 [cs.CV].

Yuhui Yuan Jingyi Xie, Xilin Chen Jingdong Wang (2020). *SegFix: Model-Agnostic Boundary Refinement for Segmentation*. arXiv: 2007.04269 [cs.CV].