Image Inpainting By Image Segmentation Using Mask Transformer

Wei Zhenyu[1]

[1] Digital Image Processing Report, Faculty of Innovation Engineering,
Macau University of Science and Technology

Abstract Modern image inpainting systems have improved considerably, however most methods still require a mask to be input, leading to the rapid drawing of a suitable mask becoming a major limitation in the use of these methods. Basically, current research on image inpainting systems rests on optimizing the processing effect of the image, which in practice requires masking the object that is expected to be inpainted. The mask is always irregular due to the diversity of the edges of the target object, making it difficult to obtain the exact mask area. When the desired object is very complex, the traditional drawing process becomes very difficult and slow. Meanwhile, the current image segmentation method is more as a tool to get the segmentation result, and the segmented instance region can then be used as a mask for inpainting. In order to solve the problem of drawing difficulty caused by irregular mask, it is necessary to automatically generate suitable masks for input and then perform image inpainting operation to achieve the goal. Based on Mask2Former, this project proposes a method to obtain different instance masks and improve the performance by deeply adjusting the model parameters and details, using the output of segmentation as its mask input, and eventually using multimodal fusion techniques in image segmentation and image generation to achieve efficient image inpainting.

1. Introduction

1.1. Background

Solving the image inpainting problem requires "understanding" the structure of the natural image before image composition. Image inpainting aims to complete the image with realistic content specifying the region to be restored, which can be either a missing or damaged region or a selected region[1]. The creation and improvement of image inpainting system has become a hotspot of current research, and its application has been used in various aspects of real life.

Recent developments in deep generative models provide an effective end-to-end framework for image synthesis and painting tasks[2]. Basically, the current image inpainting system is divided into the following three steps.

1. Input the necessary mask region for inpainting;
2. Overlay this mask region with the source image region;
3. Paint the selected region using the image inpainting method[3].

However, most of the existing generative model-based methods use image-level features to compute and do not use segmentation information to constrain the shape of the drawn region[4]. The mask is always irregular due to the diversity of the target object edges, making it difficult to obtain the exact mask region, and the mask region masks used for input still rely on manual

drawing [5]. This usually leads to blurred results on the boundaries and both the quality of the results and the speed of the repair process depend on the quality and speed of the selection of the input region, making the traditional drawing process very difficult and slow when the edges of the object to be drawn are very complex. This problem of slow and inaccurate selection of masked regions has greatly affected the final performance of the program[6].

Therefore, we choose the Masked-attention Mask Transformer for Universal Image Segmentation (Mask2Former)[7] paper for this study. By using Mask Transformer for Universal Image Segmentation with Masked-attention, the paper achieves high segmentation accuracy and precision while ensuring operational efficiency.

1.2. Solutions According to Mask2former

Image segmentation studies the problem of grouping pixels to the structure of a natural image. Different instances and semantics of grouped pixels, such as different classes or instance members, lead to different segmentation results. This approach that brings together semantic segmentation and instance segmentation is known academically as panoramic segmentation[8]. Current research has developed specialized architectures for such tasks[9].

To solve the problem of difficult drawing due to mask irregularity, we propose to introduce panoramic segmentation information so that it automatically generates a suitable mask for the input and then performs the (2)(3) steps of image inpainting. Using the segmented mask as an intermediate bridge before and after image inpainting, this information separates the differences and variations between instances in image inpainting, which delineates clearer recovery boundaries between regions with different instances and better textures within semantically consistent zones.

We decouple the process of obtaining instance masks into two steps: panoramic segmentation and guided filtering. We first use a state-of-the-art image panorama segmentation method[7] to generate segmentation labels of the input image with instance regions. Then, we extract the labels and masks of the segmentation results as input for the next step and provide an interface to allow the user to select the instance labels to be inpainted. Finally, we combine the segmentation masks corresponding to the labels with the input images and pass them to a state-of-the-art image inpainting system for prediction.

This leads to a segmentation-guided semantic segmentation process, and the whole system is able to combine deep learning for panoramic segmentation with image generation models. This allows for more automated and more realistic predictions, especially for the boundaries between different instances, greatly reducing the need to draw masks manually. Thus, the speed of obtaining masks is accelerated and mask drawing errors are reduced[10]. On the other hand, our method offers the possibility of interactive and multi-modal prediction compared to other methods that can only perform a single prediction on the input image. Examples are the selection of instance regions by input text or the generative modification of selected regions by text [11, 12].

1.3. Statement of Work Contribution

Our work has the following five parts:

1. Research and study mask2former paper;
2. Based on this paper, we propose an inpainting method based on panoramic segmentation, and obtain the labels and masks of each instance of the image through segmentation;
3. Optimize the algorithm, improve segmentation performance, and adjust model parameters and details in depth to improve performance;
4. Build a segmentation model and adjust its output, output the segmented label as an instance list for user selection, and output the segmented mask as the inpainting mask Input;
5. Select the most advanced inpainting model to run image inpainting with high efficiency;

Future work: Apply this project to a wider field, such as selecting the corresponding instance through text input, or modifying the generated content of the instance area according to the meaning of the text [13].

2. Relevant Research and Paper Selection Reasons

Object detection or localization is a gradual process from coarse to fine in digital images. It provides not only the class of the image object, but also the location of the object in the classified image. The position is given as bounding box or center.Semantic segmentation gives better inference by predicting the label of each pixel in the input image. Each pixel is labeled according to the object class it is in.To go a step further, instance segmentation provides distinct labels for separate instances of objects belonging to the same class. Therefore, instance segmentation can be defined as a technique that simultaneously solves the problem of object detection and semantic segmentation [14].

Instance segmentation has become one of the more important, complex and challenging areas in machine vision research. To predict object class labels and pixel-specific object instance masks, it localizes the different classes of object instances appearing in various images. The purpose of instance segmentation is mainly to help the computer refine the recognition of objects.

2.1. Introduction to Instance Segmentation Techniques

Previous techniques rely on bottom-up generation of mask proposals. Subsequently, it was replaced by new techniques with more efficient structures, such as RCNN. Although RCNN has a certain improvement in segmentation accuracy, the training is based on a multi-stage pipeline, which is slow and hard to optimize because each stage needs to be trained separately. In each image of CNN, each scheme needs to extract features, which lead to problems of storage, time and detection scale respectively. Testing is also slow because of the need to extract features from the CNN.

Subsequently, Fast RCNN and Faster RCNN appeared to solve its problem. Mask RCNN can be seen as a general instance segmentation architecture[15]. Mask RCNN is based on the prototype of Faster RCNN and adds a branch for segmentation tasks. For each Proposal Box of Faster RCNN, FCN is used for semantic segmentation[16]. The segmentation task is performed simultaneously with the positioning and classification tasks. Mask2Former uses the Detectron2 detection architecture[17] and follows the updated Mask R-CNN baseline settings to process the COCO dataset, and improves the training and inference details, and finally improves the accuracy and operation speed compared with Mask R-CNN (Figure 1).

	method	backbone	search space	epochs	AP
CNN backbones	Mask R-CNN [24]	R50	dense anchors	36	37.2
		R50	dense anchors	400	42.5
		R101	dense anchors	36	38.6
		R101	dense anchors	400	43.7
	Mask2Former (ours)	R50	100 queries	50	43.7
		R101	100 queries	50	**44.2**

Figure 1. Comparison of Mask2Former and Mask R-CNN in terms of accuracy

2.2. Mask2Former's Predecessor: MaskFormer

Current methods usually formulate semantic segmentation as a per-pixel classification task[18], while instance segmentation is handled using mask classification. The author's point of view is that mask classification is completely universal, that is, the exact same model, loss, and training procedure can be used to solve semantic and instance-level segmentation tasks in a unified manner. Accordingly, this paper[19] proposes a simple mask classification model - MaskFormer, which predicts a set of binary masks, each mask is associated with a single global class label prediction, and can seamlessly integrate any existing per-pixel classification model Convert to mask classification.

MaskFormer outperforms state-of-the-art semantic segmentation models (55.6 mIoU on ADE20K) and panoptic segmentation models (52.7 PQ on COCO), especially when the number of categories is large [20]. MaskFormer consists of three modules:

1. Pixel-level module: a backbone for extracting image features and a pixel-level decoder for generating per-pixel embeddings;
2. Transformer module: use stacked Transformer decoder layers to calculate N per-segment embeddings;
3. Segmentation module: Generate probability-mask pairs of prediction results from the above two embeddings.

2.2.1. Pixel-level modules

Any segmentation model based on per-pixel classification is suitable for pixel-level module design, MaskFormer can seamlessly convert such models to Mask classification, the backbones used in this paper are ResNet backbones and Swin-Transformer backbones [21].

The pixel decoder is based on the lightweight pixel decoder of the popular FPN architecture. After the FPN, the low-resolution feature maps in the decoder are 2× upsampled and summed with the projected feature maps of the corresponding resolution from the backbone. Next, the tandem features are fused by an additional 3×3 convolutional layer+ GN+ ReLU. This process is repeated until the final feature map is obtained. Finally, a single 1×1 convolutional layer is applied to obtain the peri-pixel embedding.

2.2.2. Transformer module

The standard Transformer decoder (same as DETR) is used to compute the output N per-segments embeddings Q from the image features F and N learnable positional embeddings (i.e., queries), which encode the global information of each segment predicted by MaskFormer[22]. Similar to DETR, the decoder generates all predictions in parallel. The N query embeddings are initialized as zero vectors and are each associated with a learnable position encoding.

In this paper, six Transformer decoder layers and 100 queries are used, and the same loss of DETR is applied after each decoder.

In their experiments, the authors observe that MaskFormer is also quite competitive in semantic segmentation using a single decoder layer, but multiple layers in instance segmentation are necessary to remove duplicate items from the final prediction.

2.2.3. Segmentation module

A linear classifier with softmax activation is used to generate class probability predictions on the per-segments embedding Q and predicts an additional "no-object" class to prevent embeddings from not corresponding to any region.

For Mask prediction, the per-segments embedding Q is converted into N Mask embeddings Emask using a multi-layer perceptron (MLP) with 2 hidden layers.

Finally, the corresponding binary mask prediction mi is obtained by the dot product between the computed i-th Mask embedding and the per-pixel embedding.

2.2.4. Result output

- For semantic segmentation, segments sharing the same category label are merged.
- For instance segmentation, the indexing of probability-mask pairs helps to distinguish different instances of the same category.

- To reduce the false alarm rate in panoramic segmentation, low confidence predictions are filtered out before inference and most of the prediction segments with binary masks (mi>0.5) obscured by other predictions are removed.

2.3. Overall Model Architecture

The overall architecture of the model in this paper is derived from the simple meta-architecture of MaskFormer and consists of three components [7].

1. A backbone feature extractor: that extracts low-resolution features from the image.
2. Pixel decoder: progressively upsampling low-resolution features from the output of the backbone to generate high-resolution per-pixel embeddings.
3. Transformer decoder: operates on image features to process object queries. The final binary mask prediction is decoded from the per-pixel embeddings with object queries.

The main improvement in this paper is in the Transformer decoder.

2.4. Transformer Decoder Optimization Improvements

To optimize the Transformer decoder design, the following three improvements are made.

1. Switch the order of self-attention and cross-attention (the new "masked attention") to make the computation more efficient: the query features for the first self-attention layer do not depend on the image features yet, so applying self-attention does not make any sense.
2. Make the query features also learnable (still retaining the learnable query location embedding) and the learnable query features are directly supervised before the prediction mask (M0) used in the Transformer decoder. The authors found that these learnable query features function similarly to the region proposal network[23] and are able to generate mask proposals.
3. Dropout is not required and usually degrades performance. The authors therefore removed dropout completely from the decoder.

2.5. Key Improvements

1. Use masked attention in the Transformer decoder to restrict attention to local features (either objects or regions depending on the specific semantics of the grouping) centered on the predicted segment. The masked attention gives faster convergence and better performance than the cross-attention used in the standard Transformer decoder that focuses on all locations.
2. Use multi-scale high-resolution features to help the model segment small objects/regions.
3. Optimization improvements are proposed, such as switching the order of self-attention and cross-attention, making query features learnable, and removing dropout; all of which improve performance without additional computation.
4. By computing mask loss over K random sampling points, 3 times more training memory is saved without affecting performance.

2.6. High-resolution features

High-resolution features can improve model performance, especially for small targets. However, it increases the computational requirements.

In this paper, authors propose an effective multi-scale strategy to introduce high-resolution features while controlling the increase of computation.A feature pyramid consisting of low-resolution and high-resolution features is utilized, and the different scale features of the multi-scale features are fed to different Transformer decoder layers separately.For each resolution, a sinusoidal position embedding is added as well as a learnable scale-level embedding [24]. he 3-layer Transformer decoder is repeated L times, and the final Transformer decoder thus has 3L layers.

2.7. Improving Training Efficiency

One limitation of training general-purpose architectures is the large memory consumption due to high-resolution mask prediction, which makes them more difficult to receive than dedicated architectures that are more memory-friendly. For example, MaskFormer can only accommodate a single image in a GPU with 32G of memory [7].

Inspired by PointRend and Implicit PointRend, a segmentation model can be trained by computing the mask loss over K random sampling points instead of the entire mask. In this paper, we set K = 12544, which is 112×112 points.

In this paper, we use the sampling points to calculate the mask loss in the matching loss and final loss calculation.

- In constructing the matching loss of the dichotomous matching cost matrix, the same set of K points of all predicted and true masks are sampled uniformly.
- In the final loss between predictions and their matched ground truth, use importance sampling to sample different K-point sets for different predictions and ground truth. This training strategy effectively reduces the training memory by a factor of 3, from 18GB to 6GB per image.

3. Method

Mask2Former uses the settings of the MaskFormer generation, but with the following differences[7].

3.1. Pixel Decoder

Mask2Former is compatible with any existing pixel decoder module. In MaskFormer, FPN is selected as the default due to its simplicity. Since our goal is to demonstrate robust performance in different segmentation tasks, they use the more advanced Multi-Scale Deformable Attention Transformer (MSDeformAttn)[25] as our default pixel decoder. Specifically, they use six MSDeformAttn layers applied to feature maps with resolutions of 1/8, 1/16, and 1/32, and use a simple upsampling layer with lateral connections on the final 1/8 feature map to generate a feature map with resolution of 1/4 as a per-pixel embedding [26]. In their ablation study, they show that this pixel decoder provides the best results in different segmentation tasks (Figure 3).

3.2. Transformer Decoder

The authors use the Transformer decoder proposed in Section 3.2 with L = 3 (i.e., 9 layers in total) and a default of 100 queries. An auxiliary loss is added to each intermediate Transformer decoder layer and learnable query feature before the Transformer decoder. Loss weights. They use the binary cross-entropy loss and the dice loss[27] as their mask loss. $\ell mask = \lambda ce \ell ce + \lambda dice \ell$dice. They set $\lambda ce = 5.0$ and $\lambda dice = 5.0$. The final loss is a combination of the mask loss and the classification loss. $\ell mask + \lambda cls \ell$cls, they set λcls = 2.0 for predictions that match the ground truth and 0.1 for "no object", i.e., predictions that do not match any ground truth (Figure 4).

3.3. Post Processing

The authors use exactly the same post-processing method as in[18] to obtain the expected output format of the generic and semantic segmentation from the binary mask and category prediction pairs. Instance segmentation requires an additional confidence score for each prediction [28]. They multiply the category confidence and the mask confidence (i.e., the average per-pixel binary mask foreground probability) to obtain the final confidence.

4. Result and Discussion

In this research, we build code that bridges image segmentation and image inpainting. Among them, image segmentation uses Mask2Former's method, which is also the core of this study, and

```
from torch import nn

@META_ARCH_REGISTRY.register()
# Main class for mask classification semantic segmentation architectures.
class MaskFormer(nn.Module):
    @classmethod
    def from_config(cls, cfg):
        # loss weights
        class_weight = cfg.MODEL.MASK_FORMER.CLASS_WEIGHT
        dice_weight = cfg.MODEL.MASK_FORMER.DICE_WEIGHT
        mask_weight = cfg.MODEL.MASK_FORMER.MASK_WEIGHT

        # building criterion
        matcher = HungarianMatcher(
            cost_class=class_weight,
            cost_mask=mask_weight,
            cost_dice=dice_weight,
            num_points=cfg.MODEL.MASK_FORMER.TRAIN_NUM_POINTS,
        )

        weight_dict = {"loss_ce": class_weight, "loss_mask": mask_weight, "loss_dice": dice_weight}

        return {
            "backbone": backbone,
            "sem_seg_head": sem_seg_head,
            "criterion": criterion,
            "num_queries": cfg.MODEL.MASK_FORMER.NUM_OBJECT_QUERIES
        }
```

Figure 2. Part of the codes implemented according to Mask2Former, using the method from the paper

image inpainting uses LaMa's method [29]. In Mask2Former, the authors use the Mask Transformer with Masked-attention for Universal Image Segmentation, which achieves very high segmentation accuracy and precision while ensuring operational efficiency, and we implement this method in accordance with the paper (Figure 2).

In implementing the Mask2Former method, we constructed the model of the method and entered the basic parameters including the pixel decoder (Figure 3) and Transformer decoder (Figure 4) as described in the paper.

Following the method mentioned in this paper, the runtime only requires the input of the original image to be processed and the target label that you want to repair (Figure 5). By referring to Mask2Former's method for instance segmentation, the final result with label and mask area can be obtained, and Mask2Former provides 3 times better performance compared to other segmentation methods [30].

The result of instance segmentation consists of the instance labels and the corresponding instance masks contained in the original image. The instance label is the name of each instance, which will be used as the basis for the user to select different instances for inpainting. The instance mask is the area in the image where the instance is located, which will be used as input for inpainting [29] the image (Figure 6).

The result of the run is the result of the user's image inpainting for the instance corresponding to the selected label. The result is that after image restoration, the area in the instance no longer contains the instance[31], but is filled in full with inpainting (Figure 7).

5. Conclusion

In this paper, we solve the problem of difficult drawing caused by irregular mask, and it is achievable to automatically generate suitable masks for input by Mask2Former based on instance segmentation, and then perform the image mapping operation to achieve the goal. Then we deeply adjust the model parameters and details to improve the performance, use the output of

```
MODEL:
  META_ARCHITECTURE: "MaskFormer"
  SEM_SEG_HEAD:
  NAME: "MaskFormerHead"
  IGNORE_VALUE: 65
  NUM_CLASSES: 65
  LOSS_WEIGHT: 1.0
  CONVS_DIM: 256
  MASK_DIM: 256
  PIXEL_DECODER_NAME: "MSDeformAttnPixelDecoder"
  COMMON_STRIDE: 4
  TRANSFORMER_ENC_LAYERS: 6
```

[t]

Figure 3. Comparison of Mask2Former and Mask R-CNN in terms of accuracy

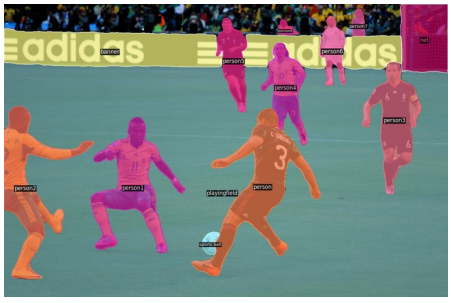
```
MASK_FORMER:
  TRANSFORMER_DECODER_NAME: "MultiScaleMaskedTransformerDecoder"
  TRANSFORMER_IN_FEATURE: "multi_scale_pixel_decoder"
  CLASS_WEIGHT: 2.0
  MASK_WEIGHT: 5.0
  DICE_WEIGHT: 5.0
  HIDDEN_DIM: 256
  NUM_OBJECT_QUERIES: 100
  NHEADS: 8
  DROPOUT: 0.0
  DIM_FEEDFORWARD: 2048
  ENC_LAYERS: 0
```

[t]

Figure 4. Comparison of Mask2Former and Mask R-CNN in terms of accuracy

[t]

Figure 5. The original image used to be processed



[t]

Figure 6. Comparison of Mask2Former and Mask R-CNN in terms of accuracy

[t]

Figure 7. Comparison of Mask2Former and Mask R-CNN in terms of accuracy

segmentation as its mask input, and finally achieve efficient image mapping using multimodal fusion techniques in the field of image segmentation and image generation.

However, there are still problems in this study, for example, the target object will have a shadow in the real world, and none of the current example segmentation studies can segment the shadow well, which will lead to missing regions in image inpainting and eventually affect the restoration results, and these problems are left for future workers to study[32].

Nonetheless, this study has greatly improved the efficiency of image inpainting and has had a profound impact on multimodal fusion technology because of its relevance to the fields of image segmentation and image generation.

References

[1] E. T. Hassan, H. M. Abbas, and H. K. Mohamed, "Image inpainting based on image segmentation and segment classification," in *2013 IEEE International Conference on Control System, Computing and Engineering*, 2013, pp. 28–33.

[2] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 417–424.

[4] C. Guillemot and O. Le Meur, "Image inpainting : Overview and recent advances," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 127–144, 2014.

[5] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother, "Instancecut: from edges to instances with multicut," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5008–5017.

[6] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4076–4084.

[7] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," 2022.

[8] M. Shen and B. Li, "Structure and texture image inpainting based on region segmentation," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 1, 2007, pp. I–701–I–704.

[9] L. Yang, T. Xiao-jian, W. Qing, S. Shang-xin, and S. Xiao-lin, "Image inpainting algorithm based on regional segmentation and adaptive window exemplar," in *2010 2nd International Conference on Advanced Computer Control*, vol. 5, 2010, pp. 656–659.

[10] X. Niu, B. Yan, W. Tan, and J. Wang, "Effective image restoration for semantic segmentation," *Neurocomputing*, vol. 374, pp. 100–108, 2020.

[11] Y. Song, C. Yang, Y. Shen, P. Wang, Q. Huang, and C.-C. J. Kuo, "Spg-net: Segmentation prediction and guidance network for image inpainting," *arXiv preprint arXiv:1805.03356*, 2018.

[12] A. Kirillov, Y. Wu, K. He, and R. Girshick, "Pointrend: Image segmentation as rendering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9799–9808.

[13] P. Wang, A. Yang, R. Men, J. Lin, S. Bai, Z. Li, J. Ma, C. Zhou, J. Zhou, and H. Yang, "Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework," *CoRR*, vol. abs/2202.03052, 2022.

[14] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and D. D. Dai J F, "Deformable transformers for end-to-end object detection," in *Proceedings of the 9th International Conference on Learning Representations. Virtual Event, Austria: OpenReview. net*, 2021.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[17] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[18] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 864–17 875, 2021.

[19] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," 2021.

[20] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen, "Max-deeplab: End-to-end panoptic segmentation with mask transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 5463–5474.

[21] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[22] D. Meng, X. Chen, Z. Fan, G. Zeng, H. Li, Y. Yuan, L. Sun, and J. Wang, "Conditional detr for fast training convergence," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3651–3660.

[23] R. Faster, "Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 9199, no. 10.5555, pp. 2 969 239–2 969 250, 2015.

[24] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," *arXiv preprint arXiv:2005.10821*, 2020.

[25] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[27] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 565–571.

[28] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *European conference on computer vision*. Springer, 2020, pp. 282–298.

[29] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," *arXiv preprint arXiv:2109.07161*, 2021.

[30] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 475–12 485.

[31] J. Jam, C. Kendrick, K. Walker, V. Drouard, J. G.-S. Hsu, and M. H. Yap, "A comprehensive review of past and present image inpainting methods," *Computer vision and image understanding*, vol. 203, p. 103147, 2021.

[32] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and Y. Akbari, "Image inpainting: A review," *Neural Processing Letters*, vol. 51, no. 2, pp. 2007–2028, 2020.