This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

PEM: Prototype-based Efficient MaskFormer for Image Segmentation

Niccolò Cavagnero[*,1], Gabriele Rosi[*,1,2], Claudia Cuttano[1], Francesca Pistilli[1],
Marco Ciccone[1], Giuseppe Averta[1,2], Fabio Cermelli[2]
[1] Politecnico di Torino, [2] Focoos AI
[1] name.surname@polito.it, [2] name.surname@focoos.ai

Abstract

Recent transformer-based architectures have shown impressive results in the field of image segmentation. Thanks to their flexibility, they obtain outstanding performance in multiple segmentation tasks, such as semantic and panoptic, under a single unified framework. To achieve such impressive performance, these architectures employ intensive operations and require substantial computational resources, which are often not available, especially on edge devices. To fill this gap, we propose Prototype-based Efficient MaskFormer (PEM), an efficient transformer-based architecture that can operate in multiple segmentation tasks. PEM proposes a novel prototype-based cross-attention which leverages the redundancy of visual features to restrict the computation and improve the efficiency without harming the performance. In addition, PEM introduces an efficient multi-scale feature pyramid network, capable of extracting features that have high semantic content in an efficient way, thanks to the combination of deformable convolutions and context-based self-modulation. We benchmark the proposed PEM architecture on two tasks, semantic and panoptic segmentation, evaluated on two different datasets, Cityscapes and ADE20K. PEM demonstrates outstanding performance on every task and dataset, outperforming task-specific architectures while being comparable and even better than computationally expensive baselines. Code is available at https://github.com/NiccoloCavagnero/PEM.

1. Introduction

Image segmentation stands as a cornerstone within the realm of computer vision and image processing, playing a pivotal role in the extraction of meaningful information from digital images. At its core, it involves partitioning an image into distinct regions, or segments, each representing a significant object or component within the visual scene.

One of the recent and noteworthy developments in this domain is the emergence of transformer-based approaches [4, 5, 20, 37], which offer a unified framework for the various flavors of image segmentation, such as semantic and panoptic segmentation. These architectures rely on an end-to-end set prediction objective, inspired by DETR [2], and they employ an encoder-decoder architecture to generate high-resolution visual features and a transformer decoder to yield a representation for each object. By leveraging the same architecture, loss function and training pipeline for different segmentation tasks, these methods obtain outstanding results, outperforming task-specific architectures while streamlining the segmentation process as a whole.

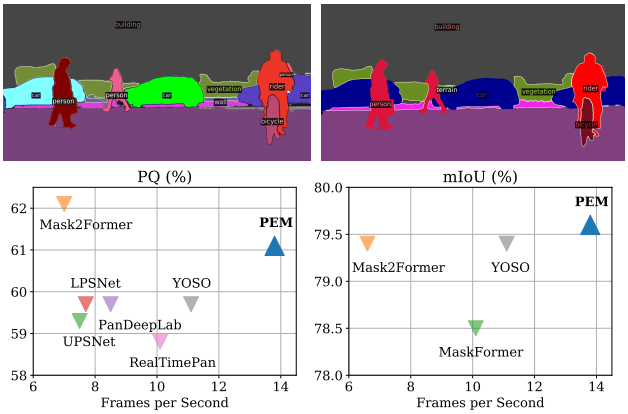

Figure 1. PEM delivers comparable or superior performance in comparison to existing methods while being the fastest multi-task architecture for image segmentation.

To achieve their remarkable performance and appealing properties, however, these models require expensive architectural components, resulting in slow and cumbersome architectures. The overly inefficient inference of existing methods leads to two important consequences: i) the deployment and inference cost, as well as the carbon footprint [12, 27], is not negligible, especially when offering cloud services to millions of users; ii) the computational demand makes the deployment on edge devices unfeasible, preventing the use on resource-constrained downstream tasks.

In this paper, we rethink their approach with the goal of doing more with less. We propose a novel image seg-

[*]Equal contribution

mentation architecture, named **P**rototype-based **E**fficient **M**askFormer (PEM), that properly considers the complexity of each component and it substitutes standard computationally demanding choices with novel and lighter counterparts. We first focus on the transformer decoder, originally constituted by a sequence of expensive attention operations between object descriptors and high-resolution image features. To reduce the computation and enhance scalability with increasing image resolutions, we incorporate a prototype selection mechanism which leverages a single visual token for each object descriptor. Furthermore, the introduction of prototypes allows us to design a novel efficient attention mechanism. Second, we revisit the visual decoder which is crucial for the extraction of high-resolution features. While previous works enhanced a feature pyramid network (FPN) with transformer-based attention modules [4, 5, 37], we employ a more efficient fully convolutional FPN. We supplement it with a context-based self-modulation module to recover the global context and deformable convolutions [7] to allow each kernel to dynamically modify its receptive field to focus on relevant regions.

We benchmark the proposed PEM architecture on two distinct tasks, semantic and panoptic segmentation, on two datasets: Cityscapes [6] and ADE20K [38] (see Fig. 1). PEM exhibits outstanding performance, showcasing similar or superior results compared to the computationally expensive baselines. Remarkably, PEM is able to outperform task-specific architectures on the challenging ADE20K.

By addressing both the major bottlenecks of modern segmentation models, PEM represents a significant step forward in the ongoing pursuit of efficient image segmentation methodologies, making them more sustainable and amenable to real-world applications. In summary, this paper provides the following contributions:

- A novel efficient cross-attention mechanism endowed with a prototype selection strategy that lowers the computational burden without affecting the results;
- A multi-scale FPN that presents the benefits of heavy transformer-based decoders in a convolutional fashion;
- Through comprehensive quantitative and qualitative analysis, we showcase the generality, efficiency, and state-of-the-art performance of PEM on both semantic and panoptic segmentation on two challenging benchmarks.

2. Related Works

Image Segmentation. A growing field of research aspires to design architecture able to operate in multiple image segmentation settings, without any change in loss function or architectural component. The seminal work of DETR [2] showed that it is possible to achieve competitive object detection and panoptic segmentation results using an end-to-end set prediction network based on mask classification. Inspired by this work, MaskFormer [4] proposed an architecture for image segmentation based on the mask classification approach, achieving state-of-the-art performance both in semantic and in panoptic segmentation. Mask2Former [5] further improved it, proposing various architectural enhancements that led to faster convergence and results that outperformed not only general-purpose approaches but also specialized architectures. More recently, kMaX-DeepLab [37] attempted to improve the cross-attention mechanism by replacing the classic attention with k-Means clustering operation. While significant progress has been made in improving overall performance across all tasks, the high resource requirements and slow inference time still hinder the deployment of these models on edge devices.

Efficient Image Segmentation. To reduce the computational complexity of image segmentation models, multiple works [3, 11, 16, 18, 31, 34, 35] proposed efficient network architectures which can be effectively deployed on devices and run in real-time. However, previous works proposed architectures specific for a single segmentation task only. Efficient semantic segmentation works were based on a two-branch architecture [11, 15, 35, 36] in which high-resolution and highly semantic features are separately processed and then merged. Recently, PIDNet [34] proposed a three-branch semantic segmentation architecture to enhance the object boundaries and improve performance on small objects. In panoptic segmentation, UPSNet [33] proposed a network that incorporates a parameter-free panoptic head to efficiently merge the predictions of segmentation and instance heads. Meanwhile, FPSNet [8] eliminated the need for additional segmentation heads and it introduced an architecture based on attention mask merging. Other approaches [3, 14, 16] localized objects using points or boxes. Recently, YOSO [18] proposed an efficient method that predicts masks for both things and stuff using a transformer architecture. Despite significant progress achieved in various settings, these architectures only operate on a single task, duplicating the research efforts and jeopardizing the research landscape. Our work aims to address this gap by presenting an efficient architecture that can be seamlessly employed in multiple segmentation tasks.

Efficient Attention Mechanism. Since its introduction, Attention [30] has exhibited remarkable capabilities in several tasks thanks to its generality and its ability to model global relationships among all input elements. MaskFormer and its variants [4, 5, 37] established both self-attention and cross-attention as two fundamental components in their architectures. Nevertheless, by considering all pairwise relationships among input tokens, the attention mechanism does not scale effectively for large input dimensions. This fundamental issue has prompted the scientific community to explore methods for mitigating the computational complex-

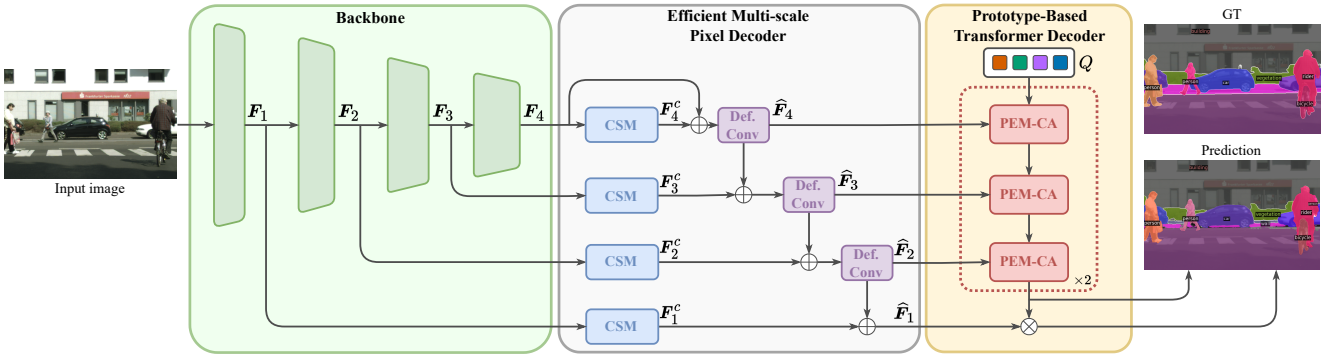

Figure 2. **Architecture of PEM** with the three main components highlighted: backbone, pixel decoder and transformer decoder. The backbone extracts features from the input image; the pixel decoder provides features upsampling to extract high-resolution features; the transformer decoder, which takes as input a set of learnable queries and the high-resolution features to produce refined queries for inference.

ity of this module. Some approaches aimed to reduce the computational cost of self-attention by integrating downsampling operators, typically pooling layers or strided convolutions, within the projections [21]. Consequently, the self-attention operation is performed at a lower resolution, improving efficiency. MobileViT V2 [25] and SwiftFormer [29] took a step forward, by replacing the expensive dot products between the input tokens with cheap element-wise multiplications of the features with a small context vector. These methods demonstrated how pairwise interactions can be redundant and global information can be condensed into lightweight context vectors, which are computationally inexpensive. While these works focus on reducing the complexity of self-attention, we propose a method for efficiently computing cross-attention, which constitutes one of the major bottlenecks in MaskFormer architectures. Our custom cross-attention effectively combines visual tokens and object queries to yield fast and precise image segmentation.

3. Method

3.1. Mask-based Segmentation Framework

In the field of image segmentation, the objective is to partition images into regions that exhibit shared characteristics. This includes tasks as semantic segmentation, where pixels with similar semantic attributes are grouped together, and panoptic segmentation, which involves differentiating instances within the same class. The goal is to develop a model that segments the image into separate regions with unique masks with assigned class probabilities. Formally, given an image, $\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3}$, we aim to predict a set of N binary masks $\boldsymbol{M} \in \{0,1\}^{N \times H \times W}$ each associated with a probability distribution $\boldsymbol{p}_i \in \Delta^{K+1}$, where (H, W) is the height and width of the image, and $K+1$ is the number of classes plus an additional "*no object*" class.

To achieve this goal, we follow the framework provided by the MaskFormer architecture [4, 5, 37] that consists of three main components, depicted in Fig. 2: (i) a *backbone* extracting feature maps $\boldsymbol{F}_i \in \mathbb{R}^{H_i \times W_i \times B_i}$ from the image $\boldsymbol{I}$, (ii) *a pixel decoder* that processes $\boldsymbol{F}_i$ to produce high-resolution multi-scale features $\widehat{\boldsymbol{F}}_i \in \mathbb{R}^{H_i \times W_i \times C}$, with $i \in \{1,2,3,4\}$ and H_i, W_i equal to the image resolution divided by, respectively, 4, 8, 16, and 32, and (iii) a *transformer decoder* which accepts three multi-scale features $\widehat{\boldsymbol{F}}_i$, $i \in \{2,3,4\}$ as input together with N learnable queries $\boldsymbol{Q} \in \mathbb{R}^{N \times C}$ and it generates N refined queries $\widehat{\boldsymbol{Q}} \in \mathbb{R}^{N \times C}$. To generate the N binary masks, the refined queries $\widehat{\boldsymbol{Q}}$ are then multiplied by the highest resolution features of the pixel decoder $\widehat{\boldsymbol{F}}_1$. Finally, the class probabilities are obtained through a linear classifier applied on $\widehat{\boldsymbol{Q}}$.

3.2. Prototype-based Masked Cross-Attention

A core component of MaskFormer architecture family [4, 5, 37] is the transformer decoder which, taking as input N learnable queries and high-resolution image features, has the objective of refining the queries which are later used to obtain the predictions. This is achieved through several transformer blocks that attend to the feature representations and model relations between different objects. Each block computes a cross-attention between visual features and the object queries, acting as anchors, relying on expensive dot products. Despite its remarkable performance, it is inevitably inefficient when applied to large input features, which are typical in segmentation tasks.

In this work, we improve the efficiency of this module by proposing an architectural enhancement, denoted as Prototype-based Masked Cross-Attention (PEM-CA) and illustrated in Fig. 3. First, PEM-CA capitalizes on the intrinsic redundancy of visual features in segmentation to significantly reduce the number of input tokens in attention layers through a prototype selection mechanism. Indeed, during training, features related to the same segment naturally align and we can therefore exploit this redundancy to process only a subset of the visual tokens. Second, inspired by recent advancements in the efficiency of attention modules [24, 29], PEM-CA redesigns the cross-attention operation, modeling interactions by means of computationally cheap element-wise operations.

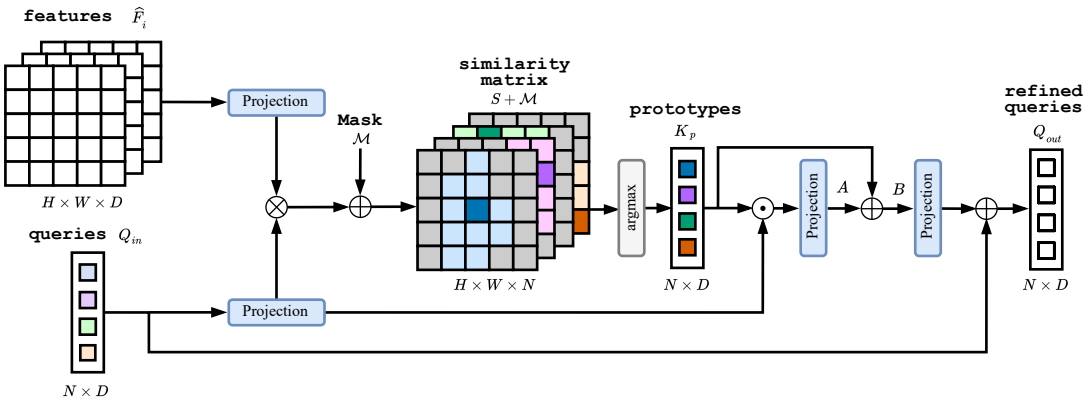

Figure 3. **Scheme of the proposed Prototype-based Masked Cross-Attention.** The prototype selection mechanism reduces the token dimension from HW to N, the number of queries, significantly reducing the computational burden.

Prototype Selection. The goal of the cross-attention is to refine each input query based on the visual features of the object it represents. However, we argue that using all the pixels belonging to an object for the refinement is redundant, since pixels associated with a specific object query will naturally become close to each other as training progresses. We can leverage this inherent redundancy to focus solely on the most relevant feature for each object, *i.e. the prototype*, while discarding the others for subsequent operations and reducing the computation.

In practice, to compute the prototypes, we first project the high-resolution features $\widehat{\boldsymbol{F}}_i \in \mathbb{R}^{H_i \times W_i \times C}$ and the object queries $\boldsymbol{Q}_{in} \in \mathbb{R}^{N \times C}$ in the same dimensional space and we obtain, respectively, $\boldsymbol{K} \in \mathbb{R}^{H_i W_i \times D}$ and $\boldsymbol{Q} \in \mathbb{R}^{N \times D}$. Then, the similarity matrix $\boldsymbol{S} \in \mathbb{R}^{H_i W_i \times N}$ is computed as:

$$\boldsymbol{S} = \boldsymbol{K}\boldsymbol{Q}^T, \tag{1}$$

which represents the relationship between each pixel of the image feature with the object queries. Once $\boldsymbol{S}$ is obtained, for each query, we select its most similar pixel according to $\mathbf{S}$. The subset of selected pixels is termed *prototypes*, as they represent the most representative token of their respective object. Formally, the prototypes $\boldsymbol{K}_p$ are computed as:

$$\boldsymbol{G} = \underset{H_i W_i}{\arg\max}(\boldsymbol{S} + \boldsymbol{\mathcal{M}}), \tag{2}$$

$$\boldsymbol{K}_p = \boldsymbol{K}\left[\boldsymbol{G}\right], \tag{3}$$

where $\boldsymbol{\mathcal{M}}$ is a binary mask applied to the similarity matrix, $\boldsymbol{G}$ the selected token indices and $\boldsymbol{K}\left[\boldsymbol{G}\right]$ denotes the selection of the indices in $\boldsymbol{G}$ on the first dimension of $\boldsymbol{K}$. The binary mask $\boldsymbol{\mathcal{M}}$ introduces a masking mechanism as in [5] that forces the selection to only consider foreground pixels, leading to more consistent assignments and improved training performance. We compute $\boldsymbol{\mathcal{M}}$ as:

$$\boldsymbol{\mathcal{M}}(x, y) = \begin{cases} 0 & \text{if } \boldsymbol{M}_{l-1}(x, y) = 1 \\ -\infty & \text{otherwise,} \end{cases} \tag{4}$$

where $\boldsymbol{M}$ is the binarized output of the previous transformer decoder layer resized at the same resolution of $\widehat{\boldsymbol{F}}_i$.

We note that the whole selection process is performed in a multi-head fashion, with features and queries divided in heads across the channel dimension. Hence, each group of channels of a token can be assigned to the corresponding group of a given query, thereby improving the modeling capability of the selection process.

Prototype-based Cross-Attention. The prototype selection mechanism reduces the input from $\boldsymbol{K} \in \mathbb{R}^{HW \times D}$ to $\boldsymbol{K}_p \in \mathbb{R}^{N \times D}$, significantly decreasing the complexity of any subsequent operation. In addition, it establishes a matching between the queries and their corresponding prototype. Consequently, the $\boldsymbol{Q}$-$\boldsymbol{K}_p$ interaction can be modeled through a cheap element-wise product and a projection $\boldsymbol{W}_A \in \mathbb{R}^{D \times D}$, avoiding the need of leveraging all pairwise relationships. Formally,

$$\boldsymbol{A} = (\boldsymbol{Q} \odot \boldsymbol{K}_p)\boldsymbol{W}_A. \tag{5}$$

The matrix $\boldsymbol{A}$ is then normalized across the channel dimension and scaled by a learnable parameter $\boldsymbol{\alpha} \in \mathbb{R}^D$. The scaled attention matrix indicates the strength of interaction between prototypes and queries, and we use it to dynamically reweight the prototypes $\boldsymbol{K}_p$ in an additive manner:

$$\boldsymbol{B} = \boldsymbol{\alpha} \odot \frac{\boldsymbol{A}}{||\boldsymbol{A}||_2} + \boldsymbol{K}_p. \tag{6}$$

To obtain the output $\boldsymbol{Q}_{out}$, a final linear projection $\boldsymbol{W}_{out} \in \mathbb{R}^{D \times C}$ brings the hidden states of the module back to the queries space before applying a residual connection:

$$\boldsymbol{Q}_{out} = \boldsymbol{B}\boldsymbol{W}_{out} + \boldsymbol{Q}. \tag{7}$$

The incorporation of PEM-CA allows a more efficient interaction between visual features and object queries compared to traditional cross-attention mechanisms. As depicted in Figure 4, it is evident that as the resolution of the input features increases, the gap in terms of latency also widens. PEM-CA demonstrates a notable advantage, being $2\times$ faster than the masked cross-attention counterpart. This observation underscores the efficiency gains achieved through the adoption of PEM-CA in managing the computational demands of attention mechanisms.

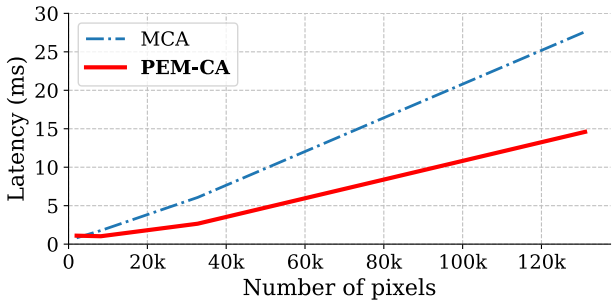

Figure 4. **Latency comparison between PEM-CA and Masked Cross-Attention.** PEM-CA scales better w.r.t. Masked Cross-Attention [5] when the input dimension increases. Note that, for Cityscapes images (1024×2048 pixels), the features have dimensions 2048 (F_4), 8192 (F_3), 32768 (F_2), 131072 (F_1) pixels.

3.3. Efficient Multi-scale Pixel Decoder

The pixel decoder covers a fundamental role in extracting multi-scale features which allow a precise segmentation of the objects. Mask2Former [5] implements it as a feature pyramid network (FPN) enhanced with deformable attention. Deformable attention is characterized by three fundamental properties: (i) it attends global context for feature refinement, (ii) it computes dynamic weights based on the input, and (iii) it leverages deformability to make the receptive field input-dependent, favoring the focus on relevant regions for the input. Despite its performance gains, using deformable attention upon an FPN introduces a computation overhead that makes the pixel decoder inefficient and unsuitable for real-world applications. To maintain the performance while being computationally efficient, we use a fully convolutional FPN where we restore the benefits of deformable attention by leveraging two key techniques. First, to reintroduce the global context (i) and the dynamic weights (ii), we implement context-based self-modulation (CSM) modules that adjust the input channels using a global scene representation [17]. Moreover, to enable deformability (iii), we adopt deformable convolutions that focus on relevant regions of the image by dynamically adapt the receptive field. This dual approach yields competitive performance while preserving the computational efficiency.

Context-based Self-Modulation. Features coming from the backbone are highly localized and contain rich spatial details but they lack a general understanding of the scene context. To restore and efficiently inject this information at all scales, we take inspiration from previous works [11, 17, 28, 35] and employ a context-based self-modulation (CSM) mechanism to reweight the importance of each channel based on a global scene representation, which suppresses less informative channels and enhances the most informative ones. Specifically, given the input features from the i-th stage $\boldsymbol{F}_i \in \mathbb{R}^{H_i \times W_i \times B_i}$, with $i \in \{1, 2, 3, 4\}$, we first project them into a low-dimensional space C, obtaining $\boldsymbol{F}'_i \in \mathbb{R}^{H_i \times W_i \times C}$. Then, we compute the context representation $\boldsymbol{\Omega}_i \in \mathbb{R}^{1 \times C}$ as the projection of the globally pooled visual features:

$$\boldsymbol{\Omega}_i = \texttt{MLP}(\texttt{GAP}(\boldsymbol{F}'_i)), \tag{8}$$

where MLP denotes a two-layer network made of 1×1 convolutions and GAP a global average pooling operation. Finally, we obtain the relevance of each channel by passing $\boldsymbol{\Omega}_i$ through a sigmoid function σ and compute the contextualized features $\boldsymbol{F}^c_i \in \mathbb{R}^{H_i \times W_i \times C}$ by:

$$\boldsymbol{F}^c_i = \boldsymbol{F}'_i \odot \sigma(\boldsymbol{\Omega}_i) + \boldsymbol{F}'_i, \tag{9}$$

where $\odot$ represents the Hadamard (element-wise) product between $\boldsymbol{\Omega}_i$ and all the pixels in $\boldsymbol{F}_i$. Note that all these operations, employed to restore deformable attention characteristics, are highly efficient, being composed only by 1×1 convolutions, normalizations, or element-wise products.

Feature Aggregation. Having obtained the contextualized features, we now aggregate them to construct the features pyramid network. To do so, we follow previous efficient segmentation works [1, 18] relying on deformable convolutions [7] to fuse features coming from different scales.

In practice, given the features $\boldsymbol{F}^c_i$, with $i \in \{1, 2, 3, 4\}$, where $i = 4$ corresponds to the lowest resolution, we compute the intermediate FPN features $\widehat{\boldsymbol{F}}_i$ as follows:

$$\widehat{\boldsymbol{F}}_i = \begin{cases} \texttt{DefConv}(\boldsymbol{F}^c_i + \texttt{Proj}(\texttt{GAP}(\boldsymbol{F}_i))), & i = 4 \\ \texttt{DefConv}(\boldsymbol{F}^c_i + \texttt{Up}(\widehat{\boldsymbol{F}}_{i+1})), & i = 2, 3 \\ \boldsymbol{F}^c_i + \texttt{Up}(\widehat{\boldsymbol{F}}_{i+1}), & i = 1 \end{cases} \tag{10}$$

where Proj indicates a linear projection, DefConv is the deformable convolution and Up is an upsampling operation. Note that the $\widehat{\boldsymbol{F}}_4$ is obtained starting using a scene representation that further injects the global context into the FPN, while for the others we mix the upsampled intermediate features of the FPN with the higher-resolution features.

As illustrated in Fig. 2, the three lowest resolution features $\widehat{\boldsymbol{F}}_i$ ($i \in \{2, 3, 4\}$) are employed as visual features in the PEM-CA (see Sec. 3.2), while the highest resolution feature $\widehat{\boldsymbol{F}}_1$ is used for computing the final predictions.

4. Experiments

4.1. Experimental Protocol

Datasets. We evaluate our method on two segmentation datasets: Cityscapes [6] and ADE20K [38]. The Cityscapes dataset features 19 distinct classes situated in an urban environment, further classified into 8 *things* and 11 *stuff* categories. On the other hand, ADE20K is a comprehensive dataset consisting of 150 diverse classes, encompassing 100 *things* and 50 *stuff* categories.

Baselines. We conduct a comprehensive comparison of PEM with both state-of-the-art task-specific and multi-task architectures for panoptic and semantic segmentation.

Metrics. For semantic segmentation, we employ mean Intersection over Union (mIoU) [10] to measure the performance. For panoptic segmentation, we rely on the Panoptic Quality metric (PQ) [19], which encapsulates the overall performance of the models. PQ is defined as the product of two components: Segmentation Quality, which takes into account the Intersection over Union (IoU) between correctly classified segments, and Recognition Quality, which scores the classification accuracy. Furthermore, we report the PQ averaged only on *thing* classes (PQ_{th}) and on *stuff* classes (PQ_{st}). If not stated otherwise, latency and FPS are measured using PyTorch 1.12 in FP32 on a V100 GPU with a batch size of 1, using a standard resolution 2048×1024 on Cityscapes and by taking the average runtime on the entire validation set on ADE20K.

4.2. Implementation Details

We train our models with AdamW [23] optimizer paired with a Cosine learning rate schedule [22]. Specifically, we set a learning rate of 0.0007 for Cityscapes and 0.0004 for ADE20K, with a 0.1 multiplier on the backbone. We adopt a batch size of 32 and the weight decay is set to 0.05 for both datasets. The models are trained for 90k iterations on Cityscapes and 160k iterations on ADE20K.

Losses. In terms of loss functions, we follow the configuration established in Mask2Former [5]. The classification head is supervised using Binary Cross-entropy (BCE), while for the masks, a combination of BCE and Dice loss [26] is employed. To balance the influence of these distinct loss components, weight factors of 2.0, 5.0, and 5.0 are assigned to BCE for classification, BCE for masks, and Dice loss, respectively. Furthermore, deep supervision is enabled at each transformer decoder block by default.

Architecture. Unless explicitly specified, our models employ a ResNet50 [13] pretrained on ImageNet-1k [9] as backbone. The transformer decoder utilized in our architecture is derived from Mask2Former [5], where the masked cross-attention layers are replaced by our proposed PEM-CA. The architectural configuration consists of two transformer decoder stages with a hidden dimension of 256 and 8 heads in the attention layers. An expansion factor of 8 is applied in the feed-forward networks and a fixed number of 100 object queries is employed. Furthermore, the hidden dimension of our pixel decoder is set to 128.

4.3. Results

Panoptic Segmentation on Cityscapes. Tab. 1 presents the results in the panoptic setting for Cityscapes. The key challenge of this dataset is to reach reasonable FPS due to the high-resolution of input images, stressing the need for efficient approaches. Notably, PEM demonstrates remarkable performance, achieving 61.1 PQ, while being the fastest architecture, with 13.8 FPS. PEM outperforms all competitor models except for the heavyweight and slower Mask2Former. Indeed, for a loss of 1 PQ, PEM is twice as fast as Mask2Former. Furthermore, our architecture exhibits a substantial improvement, with a 1.4 PQ gain and 2.7 FPS advantage over the second-fastest approach, YOSO. Remarkably, while showing similar PQ_{st} compared to YOSO, our architecture outperforms it on PQ_{th} (+3.3). Overall, PEM exhibits the most favorable performance-speed trade-off among all the models.

Method	PQ	PQ_{th}	PQ_{st}	FPS	FLOPs	Params
Mask2Former [5]	62.1	-	-	4.1	519G	44.0M
UPSNet [33]	59.3	54.6	62.7	7.5	-	-
LPSNet [14]	59.7	54.0	63.9	7.7	-	-
PanDeepLab [3]	59.7	-	-	8.5	-	-
FPSNet [8]	55.1	-	-	8.8†	-	-
RealTimePan [16]	58.8	52.1	63.7	10.1	-	-
YOSO [18]	59.7	51.0	66.1	11.1	265G	42.6M
PEM	61.1	54.3	66.1	13.8	237G	35.6M

Table 1. **Panoptic segmentation on Cityscapes with 19 categories.** †: measured on a Titan GPU. ResNet50 is employed as backbone for all the architectures.

Method	PQ	PQ_{th}	PQ_{st}	FPS	FLOPs	Params
BGRNet [32]	31.8	34.1	27.3	-	-	-
MaskFormer [4]	34.7	32.2	39.7	29.7	86G	45.0M
Mask2Former [5]	39.7	39.0	40.9	19.5	103G	44.0M
kMaxDeepLab [37]	42.3	-	-	-	-	-
YOSO [18]	38.0	37.3	39.4	35.4	52G	42.0M
PEM	38.5	37.0	41.1	35.7	47G	35.6M

Table 2. **Panoptic segmentation on ADE20k with 150 categories.** ResNet50 is employed as backbone for all the architectures. FLOPs are measured at resolution 640x640.

Panoptic Segmentation on ADE20K. The ADE20K panoptic results are outlined in Tab. 2. The large-scale dimension of the dataset together with the high number of classes make ADE20K one of the most challenging benchmarks in segmentation. Nonetheless, PEM attains a PQ of 38.5 at 35.7 FPS. Within this context, only heavy and slow architectures, such as Mask2Former and kMaxDeepLab, manage to surpass the PQ of PEM, while being slower. In particular, Mask2Former is 16.2 FPS slower than PEM. When compared to the fastest competitor, YOSO, our model demonstrates an improvement of 0.5 PQ. In essence, PEM demonstrates the best performance-speed trade-off,

Method	Backbone	mIoU	FPS	FLOPs	Params
Task-specific Architectures					
BiSeNetV1 [35]	R18	74.8	65.5†	55G	49.0M
BiSeNetV2-L[36]	-	75.8	47.3‡	119G	-
STDC1-Seg75 [11]	STDC1	74.5	74.8†	-	-
STDC2-Seg75 [11]	STDC2	77.0	58.2†	-	-
DDRNet-23-S [15]	-	77.8	108.1	36G	5.7M
DDRNet-23 [15]	-	79.5	51.4	143G	20.1M
PIDNet-S [34]	-	78.8	93.2	46G	7.6M
PIDNet-M [34]	-	80.1	39.8	197G	34.4M
PIDNet-L [34]	-	80.9	31.1	276G	36.9M
Multi-task Architectures					
MaskFormer [4]	R101	78.5	10.1	559G	60.2M
Mask2Former [5]	R50	79.4	6.6	523G	44.0M
YOSO [18]	R50	79.4	11.1	268G	42.6M
PEM	STDC1	78.3	24.3	92G	17.0M
PEM	STDC2	79.0	22.0	118G	21.0M
PEM	R50	79.9	13.8	240G	35.6M

Table 3. **Semantic segmentation on Cityscapes with 19 categories.** †: resolution of 1536x768. ‡: resolution of 1024x512.

Method	Backbone	mIoU	FPS	FLOPs	Params
Task-specific Architectures					
BiSeNetV1 [35]	R18	35.1	143.1	15G	13.3M
BiSeNetV2-L [36]	-	28.5	106.7	12G	3.5M
STDC1 [11]	STDC1	37.4	116.1	8G	8.3M
STDC2 [11]	STDC2	39.6	78.5	11G	12.3M
DDRNet-23-S [15]	-	36.3	96.2	4G	5.8M
DDRNet-23 [15]	-	39.6	94.6	18G	20.3M
PIDNet-S [34]	-	34.8	73.5	6G	7.8M
PIDNet-M [34]	-	38.8	73.3	22G	28.8M
PIDNet-L [34]	-	40.5	65.4	34G	37.4M
Multi-task Architectures					
MaskFormer [4]	R50	44.5	29.7	55.1G	41.3M
Mask2Former [5]	R50	47.2	21.5	70.1G	44.0M
YOSO [18]	R50	44.7	35.3	37.3G	42.0M
PEM	STDC1	39.6	43.6	16.0G	17.0M
PEM	STDC2	45.0	36.3	19.3G	21.0M
PEM	R50	45.5	35.7	46.9G	35.6M

Table 4. **Semantic segmentation on ADE20K with 150 categories.** FLOPs are measured at resolution 512x512.

confirming the findings observed on the Cityscapes dataset.

Semantic Segmentation on Cityscapes. Tab. 3 presents the results for Cityscapes in the semantic setting. PEM achieves a mIoU of 79.9 at 13.8 FPS. Although slower, PEM shows superior performance w.r.t. all competitors with the exception of the highly engineered and tailored PIDNet. Conversely, when compared to multi-task architectures, PEM obtains a gain of 0.5 mIoU over both Mask2Former and YOSO and 1.4 mIoU over MaskFormer, being faster than them by more than 2 FPS. In summary, PEM emerges as the general architecture with the best trade-off between performance and latency for semantic segmentation on Cityscapes. Additionally, when equipped with STDC[11] as backbone, PEM maintains its performance while demonstrating a consistent improvement in speed.

Semantic Segmentation on ADE20K. Tab. 4 outlines the results for semantic segmentation on the large-scale ADE20K dataset. Although it features over 150 distinct classes, PEM attains a noteworthy 45.5 mIoU at 35.7 FPS. When compared to general-purpose architectures, PEM shows an outstanding performance vs efficiency trade-off, with only Mask2Former able to achieve higher mIoU, while being almost 15 FPS slower. Contrary to the Cityscapes results, PEM outperforms all task-specific approaches by a large margin, showcasing a 5 mIoU gain over the highest-performing competitor, PIDNet-L. Furthermore, also in this scenario, with the integration of an efficient backbone, such as STDC1 or STDC2, PEM yields remarkable performance while significantly reducing inference time.

4.4. Ablation Study

We perform ablation studies on the various components of the proposed architecture, in order to assess the contribution of each module. All the ablations are performed on Cityscapes [6] in the panoptic setting.

Prototype-based Masked Cross-Attention. Tab. 5 serves as a comprehensive comparison for evaluating the efficacy of Prototype-based Masked Cross-Attention against traditional cross-attention mechanisms, shedding light on the impact of removing masking and prototype selection from our module. The absence of the masking mechanism, designed to enhance focus on foreground regions, results in a noticeable 3.3 panoptic quality (PQ) loss. This highlights the crucial role played by the masking mechanism in guiding the model attention to relevant regions, especially those representing foreground objects. Complete removal of the prototype selection strategy, with token aggregation by summation, akin to SwiftFormer [29], leads to a more consistent performance loss of 13.4 PQ and almost 20 PQ_{th}. The substantial performance drop underlines the critical point that tokens cannot be naively collapsed. This emphasizes the necessity of a careful selection strategy, demonstrating the importance of our proposed prototype selection mechanism for optimal token representation. Notably, all observed performance drops are mostly associated with the *things* category. This suggests that the prototype selection strategy is particularly critical for distinguishing between different instances of objects. Overall, Prototype-based Masked Cross-Attention demonstrates superior per-

Method	PQ	PQ_{th}	PQ_{st}	Latency	FLOPs
CA [4]	58.4	47.4	66.4	11.4 ms	228G
Masked CA [5]	60.4	52.0	66.5	17.4 ms	246G
PEM-CA w/o Prototypes	48.7	24.7	66.1	5.6 ms	218G
PEM-CA w/o Masking	57.8	47.0	65.6	6.8 ms	221G
PEM-CA	**61.1**	**54.3**	**66.1**	9.8 ms	237G

Table 5. **Ablation of PEM-CA on Cityscapes**. We report the cumulative latency of the cross-attention modules.

Method	PQ	PQ_{th}	PQ_{st}	Latency	FLOPs
PEM w/ MF decoder [4]	57.8	48.5	64.5	84.7ms	414G
PEM w/ M2F decoder [5]	61.4	54.6	66.3	131.6ms	497G
PEM w/o CSM	60.0	51.5	66.1	71.9ms	237G
PEM w/o deformable	57.1	47.6	64.0	69.9ms	250G
PEM	**61.1**	**54.3**	**66.1**	72.4ms	237G

Table 6. **Ablation of Pixel Decoders on Cityscapes.** We report the total latency for the whole model.

formance compared not only to standard cross-attention but also its masked counterpart [5], while being significantly faster than both methods. The achieved PQ_{th} improvement exceeds 2, showcasing the efficacy of the proposed Prototype-based Masked Cross-Attention in the realm of panoptic segmentation.

Lightweight Pixel Decoder. Tab. 6 illustrates the behavior of PEM when certain components are excluded from the lightweight pixel decoder and it presents a comparison with MaskFormer and Mask2Former pixel decoders. Specifically, the absence of CSM modules results in more than 1 PQ and almost 3 PQ_{th} loss, thereby demonstrating the need of global context modeling and dynamic weights. Furthermore, substituting Deformable convolutions with standard ones exacerbates the PQ loss, yielding a decrease of 4 PQ and nearly 7 PQ_{th}. This underscores the critical role of kernel deformability, especially for discerning multiple instances within the *things* category. When compared to the heavy convolutional decoder of MaskFormer, our decoder demonstrates superior performance with a higher PQ (+3.3) and higher FPS (+2.0). This attests to the effectiveness of our approach, which integrates the strengths of the Mask2Former decoder in a convolutional manner. Notably, while our decoder and the Mask2Former decoder achieve similar performance levels, our implementation stands out for its efficiency, running at twice the FPS.

Number of Transformer Decoder Layers. Figure 5 shows the variation in performance and latency when the number of transformer decoding layers varies from zero, where prediction is carried out solely on initialized queries, to six, representing the complete configuration. Notably, PEM exhibits robust performance even with a single stage of transformer decoder, *i.e*. three decoding layers, offering a flexible trade-off between performance and speed. Moreover, on the challenging ADE20K dataset, the model attains nearly zero PQ without cross-attention layers, emphasizing the need for an efficient query refinement process.

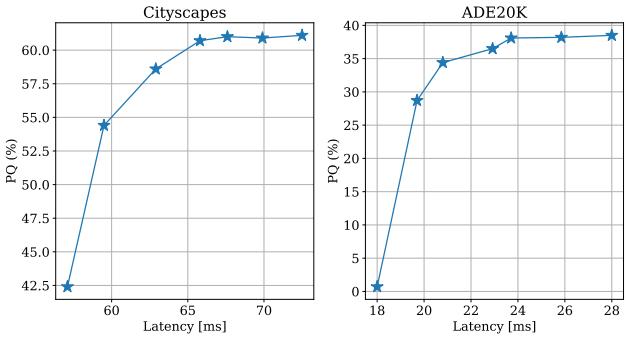

Figure 5. **PQ versus latency on Cityscapes and ADE20K.** We report performance and latency across different numbers of PEM transformer decoder blocks, ranging from zero to six.

5. Conclusions

Our work introduces PEM, a transformer-based architecture addressing the efficiency challenges posed by current models in image segmentation tasks. Leveraging a novel prototype-based cross-attention mechanism and an efficient multi-scale feature pyramid network, PEM achieves outstanding performance in both semantic and panoptic segmentation tasks on multiple datasets. PEM's efficiency surpasses state-of-the-art multi-task architectures and competes favorably with computationally expensive baselines, demonstrating its potential for deployment in resource-constrained environments. This work represents a significant step towards efficient and high-performance segmentation solutions using transformer-based architectures.

Acknowledgements This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. We acknowledge the CINECA award under the ISCRA initiative, for the availability of high performance computing resources and support.

References

[1] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019. 5

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2

[3] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020. 2, 6

[4] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 34:17864–17875, 2021. 1, 2, 3, 6, 7, 8

[5] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 5, 7

[7] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 2, 5

[8] Daan de Geus, Panagiotis Meletis, and Gijs Dubbelman. Fast panoptic segmentation network. *IEEE Robotics and Automation Letters*, 5(2):1742–1749, 2020. 2, 6

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[10] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111:98–136, 2015. 6

[11] Mingyuan Fan, Shenqi Lai, Junshi Huang, Xiaoming Wei, Zhenhua Chai, Junfeng Luo, and Xiaolin Wei. Rethinking bisenet for real-time semantic segmentation. In *CVPR*, 2021. 2, 5, 7

[12] Shreyank N Gowda, Xinyue Hao, Gen Li, Laura Sevilla-Lara, and Shashank Narayana Gowda. Watt for what: Rethinking deep learning's energy-performance relationship. *arXiv preprint arXiv:2310.06522*, 2023. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[14] Weixiang Hong, Qingpei Guo, Wei Zhang, Jingdong Chen, and Wei Chu. Lpsnet: A lightweight solution for fast panoptic segmentation. In *CVPR*, 2021. 2, 6

[15] Yuanduo Hong, Huihui Pan, Weichao Sun, and Yisong Jia. Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv preprint arXiv:2101.06085*, 2021. 2, 7

[16] Rui Hou, Jie Li, Arjun Bhargava, Allan Raventos, Vitor Guizilini, Chao Fang, Jerome Lynch, and Adrien Gaidon. Real-time panoptic segmentation from dense detections. In *CVPR*, 2020. 2, 6

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 5

[18] Jie Hu, Linyan Huang, Tianhe Ren, Shengchuan Zhang, Rongrong Ji, and Liujuan Cao. You only segment once: Towards real-time panoptic segmentation. In *CVPR*, 2023. 2, 5, 6, 7, 1, 3

[19] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, 2019. 6

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[21] Yanyu Li, Ju Hu, Yang Wen, Georgios Evangelidis, Kamyar Salahi, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Rethinking vision transformers for mobilenet size and speed. In *ICCV*, 2023. 3

[22] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6, 1

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. 6, 1

[24] Sachin Mehta and Mohammad Rastegari. Mobilevit: Lightweight, general-purpose, and mobile-friendly vision transformer. In *ICLR*, 2021. 3

[25] Sachin Mehta and Mohammad Rastegari. Separable self-attention for mobile vision transformers. *Transactions on Machine Learning Research*, 2022. 3

[26] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016. 6

[27] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021. 1

[28] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *AAAI*, 2018. 5

[29] Abdelrahman Shaker, Muhammad Maaz, Hanoona Rasheed, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Swiftformer: Efficient additive attention for transformer-based real-time mobile vision applications. In *ICCV*, 2023. 3, 7

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 2

[31] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *ECCV*, 2020. 2

[32] Yangxin Wu, Gengwei Zhang, Yiming Gao, Xiajun Deng, Ke Gong, Xiaodan Liang, and Liang Lin. Bidirectional graph reasoning network for panoptic segmentation. In *CVPR*, 2020. 6

[33] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, 2019. 2, 6

[34] Jiacong Xu, Zixiang Xiong, and Shankar P Bhattacharyya. Pidnet: A real-time semantic segmentation network inspired by pid controllers. In *CVPR*, 2023. 2, 7

[35] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *ECCV*, 2018. 2, 5, 7

[36] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129: 3051–3068, 2021. 2, 7

[37] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. kmax-deeplab: k-means mask transformer. In *ECCV*, 2022. 1, 2, 3, 6

[38] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2, 5