Renovate Yourself: Calibrating Feature Representation of Misclassified Pixels for Semantic Segmentation

Hualiang Wang[1*], Huanpeng Chu[1*], Siming Fu[1], Zuozhu Liu[2], Haoji Hu[1†]

[1] College of Information Science and Electronic Engineering, Zhejiang University, China
[2] Zhejiang University-Angelalign Inc. R&D Center for Intelligent Healthcare, ZJU-UIUC Institute, Zhejiang University, China.
{hualiang_wang,chuhp,fusiming,haoji_hu}@zju.edu.cn, zuozhuliu@intl.zju.edu.cn

Abstract

Existing image semantic segmentation methods favor learning consistent representations by extracting long-range contextual features with the attention, multi-scale, or graph aggregation strategies. These methods usually treat the misclassified and correctly classified pixels equally, hence misleading the optimization process and causing inconsistent intra-class pixel feature representations in the embedding space during learning. In this paper, we propose the auxiliary representation calibration head (RCH), which consists of the image decoupling, prototype clustering, error calibration modules and a metric loss function, to calibrate these error-prone feature representations for better intra-class consistency and segmentation performance. RCH could be incorporated into the hidden layers, trained together with the segmentation networks, and decoupled in the inference stage without additional parameters. Experimental results show that our method could significantly boost the performance of current segmentation methods on multiple datasets (e.g., we outperform the original HRNet and OCRNet by 1.1% and 0.9% mIoU on the Cityscapes test set). Codes are available at https://github.com/VipaiLab/RCH.

Introduction

Semantic segmentation is one of the most fundamental and challenging tasks in computer vision, which aims to assign the correct category to each pixel in the image. An expressive representation for semantic segmentation should exhibit both consistency and discriminability, i.e., the representations ought to be as similar as possible for intra-class pixels while remaining distinctly different for inter-class pixels.

Recently, Fully Convolutional Network (FCN) (Long, Shelhamer, and Darrell 2015) adopts the softmax loss to learn discriminative features and achieves promising performance. Based on FCN, multi-scale methods (Chen et al. 2017; Zhao et al. 2017) explore multi-level feature fusion strategies to capture global features. Another mainstream methods employ attention (Vaswani et al. 2017) to enhance intra-class consistency and obtain non-local context (Wang et al. 2018) using pairwise affinities, including pixel-to-pixel (Huang et al. 2019), pixel-to-object (Yuan, Chen, and Wang 2020), and channel-to-channel (Fu et al. 2019) affinities. Despite the great performance, there are still a considerable number of indistinguishable pixels in images which are usually misclassified by existing methods due to complicated morphological or optical characteristics such as occlusion, sunlight, etc.

A non-trivial observation of the current segmentation methods is that the cosine similarities between the feature representations of the misclassified pixels produced by networks, such as DeepLabV3 (Chen et al. 2017) and OCRNet (Yuan, Chen, and Wang 2020), and their surrounding neighbor pixels are usually quite low, as illustrated in Figure 1(c). When such representations are introduced into the global contextual information, the representations of certain objects would be significantly affected and intertwined with representations from other objects in the image, resulting in wrong segmentations and inferior performance. In contrast, the networks trained with our proposed method can correctly recognize these indistinguishable pixels with a much higher cosine similarity. We further show statistically that many of these wrong segmentation predictions, either false negative (pixels belong to the current category which are misclassified as other categories, FN) or false positive (pixels belonging to other categories which are misclassified as the current category, FP), could be corrected by our method. As illustrated in Figure 1(a-b), the distribution of the logit scores of FNs would shift to the right when our method is applied, leading to more true positive (pixels which are correctly classified, TP) predictions. Similarly, there is a left shift of the distribution of the logit scores of FPs, which helps produce more true negative predictions.

Motivated by the aforementioned findings, we propose the representation calibration head (RCH) to calibrate the error-prone feature representations of misclassified pixels for better intra-class consistency and segmentation performance. RCH could be plugged into the latent layers, jointly trained with the segmentation networks, and discarded in the inference stage without additional trainable parameters. RCH consists of three modules and a metric loss function. The Image Decoupling module (ID) is responsible for categorizing pixels to TP, FN and FP for each object. The Prototype Clustering module (PC) dynamically uses TP pixels to calculate the object prototypes during training, inspired by (Wu

[*]These authors contributed equally.
[†]Corresponding author

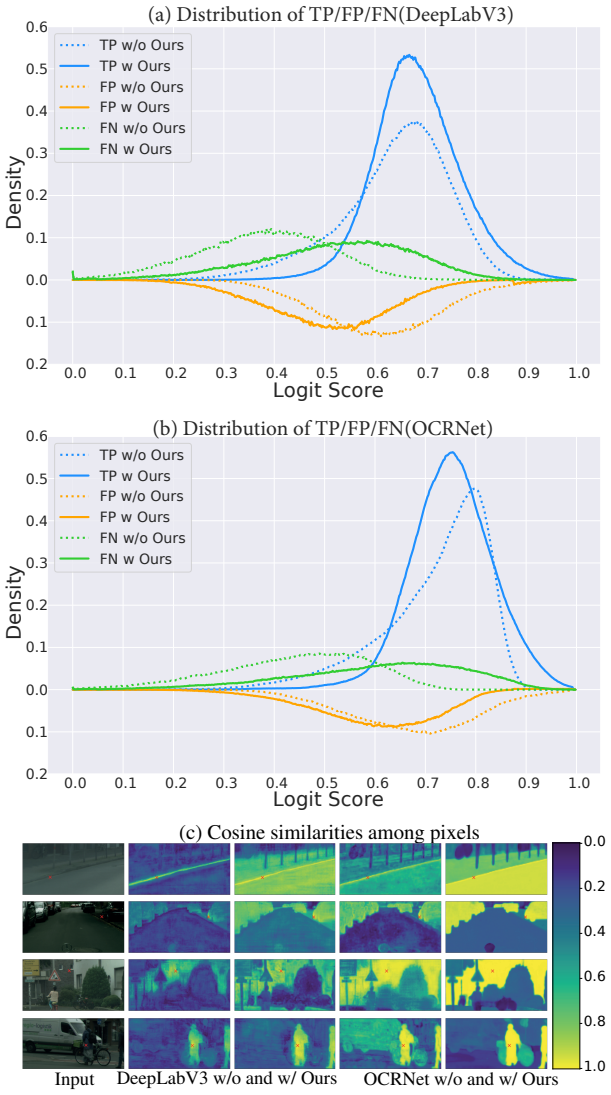


Figure 1: (a) and (b) are the score distributions (output before softmax layer) of all TP, FP and FN pixels in the validation set. In (c), we randomly sample 4 images and set misclassified pixels producted by DeepLabV3 and OCRNet as anchors (red cross). We display the cosine similarities between anchors and other pixels in input images. The corresponding colorbar is on the right.

et al. 2018; Ye et al. 2019). Specifically, we enhance the intra-class consistency by clustering the same class pixels together while preventing the misclassified pixels from participating in the generation of cluster centers to alleviate disturbances from irrelevant objects, as shown in Figure 2. The Error Calibration module (EC) is designated to minimize the distance between the TP and FN pixels while pushing away the FP pixels by virtue of a carefully designed metric loss (Deng et al. 2019; Xu et al. 2021). Extensive experimental results demonstrate that our method achieves consistent and impressive improvements on various networks (e.g., we outperform the original HRNet and OCRNet by 1.1% and 0.9%

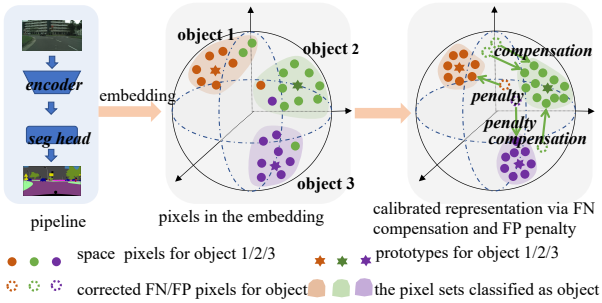


Figure 2: The illustration of our main idea. We show it by the toy example in 3D embedding space and take the pixels of object 2 as examples to show how to calibrate the representation. Specifically, we first use TP pixels of object 2 to learn the object prototype as clustering center. Then, we compensate the FN pixels which are far from prototype of object 2 and penalize FP pixels which are close to it.

mIoU on the Cityscapes dataset).

Our contributions could be summarized as follow:

- We propose a pluggable representation calibration head (RCH) to calibrate the features of misclassified pixels by minimizing the distance between TP and FN pixels while pushing away FP pixels. RCH is jointly trained with the segmentation network but discarded in the inference stage.
- We explicitly define the optimization objective for pixel representations in the semantic segmentation by regarding the TP pixels as representative examples to generalize prototypes and regarding the FN/FP pixels as calibrated items to enhance intra-class consistency.
- Extensive experiments on three challenging semantic segmentation datasets, including Cityscapes, Pascal Context and ADE20K demonstrate the superiority and effectiveness of our method over the SOTA segmentation methods.

Related Work

Networks for Semantic Segmentation. By leveraging on the powerful representations of deep neural networks (He et al. 2016; Sun et al. 2019; Dosovitskiy et al. 2020), semantic segmentation networks (Long, Shelhamer, and Darrell 2015) have made great progress. Recently, diverse novel networks are proposed, including multi-scale (Chen et al. 2017; Zhao et al. 2017, 2018), attention (Yuan, Chen, and Wang 2020; Huang et al. 2019; Zhang et al. 2019) and transformer-based (Zheng et al. 2021; Xie et al. 2021) methods. For example, DeepLabV3 (Chen et al. 2017) exploits atrous spatial pyramid pooling to learn global contextual representations. DANet (Fu et al. 2019) employs self-attention to capture dependencies among pixels and channels. (Xie et al. 2021) extracts non-local contextual features via transformer-based encoder.

Loss Function for Semantic Segmentation. There are extensive methods on designing novel loss functions to improve semantic segmentation performance. (Berman, Triki,

and Blaschko 2018) directly optimizes the network with an auxiliary intersection-over-union (IoU) loss. (Zhao et al. 2019) maximises the mutual information between predictions and labels for better performance. Resembling our method are works on consistency enhancement. (Yu et al. 2020) minimizes discrepancy between affinity matrices separately computed by feature maps and one-hot label maps. (He et al. 2021) learns the consistent features via minimizing distance among intra-class pixels while maximizing distance among inter-class pixels. (Wang et al. 2021) optimizes the feature distance via densely contrasting cross-image pixels and regions. However, it has to store a large number of samples into the memory bank, which increases storage cost. In stark contrast, we calculate only one prototype for each object and find the explicit contrastive centers, which reduces storage requirements by several orders of magnitude. Moreover, we dynamically calibrate the features according to the current misclassified pixels to ensure consistent intra-class feature representations.

Prototype Clustering. As one of applications of contrastive learning (Dosovitskiy et al. 2014; Chen et al. 2020; He et al. 2020), prototype clustering (Wu et al. 2018) aims to learn prototypes as clustering centers from variants of the same input. (Ye et al. 2019; Li et al. 2020a) apply it in unsupervised tasks to learn prototypes as non-parametric classifiers for all input instances. (Qiao et al. 2021; Yang et al. 2021; Ko, Gu, and Kim 2021) employ it in the few-shot tasks to maintain prototypes as centers for tail categories, preventing them from being overwhelmed by head categories. (Joseph et al. 2021) uses prototypes to acquire discriminative features to distinguish unknown objects in the open world. Different from the above methods, we emphasize on learning more accurate prototypes by ignoring misclassified pixels, which gives a disregarded but significant insight on improving the performance of prototype clustering.

Metric Learning. Deep metric learning methods (Schroff, Kalenichenko, and Philbin 2015; Wang et al. 2017; Sohn 2016; Cakir et al. 2019) aim to map the input to the embedding space so that we can effectively measure the similarities between two samples. The margin-based loss requires the distance between negative pairs to be larger than that between positive pairs with a fixed margin (Cheng et al. 2016; Yu and Tao 2019; Deng et al. 2019; Liu et al. 2017). (Wang and Deng 2020) adopts the reinforcement learning to adaptively obtain a large margin for the biased class. (Wang et al. 2020) only ensures a margin for hard pairs in the loss function. (Huang et al. 2020) introduces the curricular learning to set a small margin to focus on easy pairs on the early training while larger margin to hard pairs when the training goes on. Recently, (Xu et al. 2021) considers margin as FP rate. When the errors are eliminated, the margin also goes to zero.

Similar to (Xu et al. 2021), we also treat the margin as a calibration term for eliminating misclassified pixels. Rather than only considering FP rates, we categorize the calibration term into two types, where one is designated to pull the FN and TP pixels together while the other pushes the FP pixels to the opposite pole in the hyperspherical embedding space.

The Proposed Approach

In this section, we first present how our proposed representation calibration head (RCH) works in synergy with segmentation networks. Then, we introduce three modules, including Image Decoupling (ID), Prototype Clustering (PC), Error Calibration (EC) modules and a metric loss function. Finally, we analyze the effectiveness of the proposed method in terms of gradient and hard example mining.

Overview

The key component of our method is the representation calibration head (RCH) to improve the intra-class consistency via feature calibration during training, which could be abandoned during inference. As illustrated in Figure 3, we first utilize typical models as the encoder to extract intermediate features $\boldsymbol{F} \in \mathbb{R}^{N\times D}$ from the input images $\boldsymbol{X} \in \mathbb{R}^{N\times C}$, where N is the number of pixels and C is the number of input channels.

The extracted feature $\boldsymbol{F}$ is then fed into the segmentation head to output the probability maps $\boldsymbol{P} \in \mathbb{R}^{N\times K}$, where K is the number of categories. Suppose that the i-th pixel belongs to the k-th class, the segmentation loss is cross entropy between the ground truth and output probability.

$$L_i^{seg} = -y_{ik} \log p_{ik}, \tag{1}$$

where $\boldsymbol{Y} \in \mathbb{R}^{N\times K}$ is the one-hot ground truth.

The one-hot prediction map $\boldsymbol{Y}^{seg} \in \mathbb{R}^{N\times K}$ can be obtained via the index function of maximum values

$$y_{ik}^{seg} = \begin{cases} 1 & \text{if } k = \arg\max\{p_{ik}\}_{k=1\sim K}; \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

For RCH, we first use a 1×1 convolution layer to transform $\boldsymbol{F}$ to $\boldsymbol{E} \in \mathbb{R}^{N\times D}$. Given the ground truth y and one-hot prediction y^{seg} of the segmentation head, the image decoupling module categorizes the pixels into TP, FN and FP pixels. Then, the prototype clustering module updates the object prototypes by the mean embedding of TPs, and calculates the cosine similarities between pixels and prototypes. The error calibration module parallelly calculates the FN compensation and FP penalty terms for all pixels. RCH employs a metric loss function L_i^{metric} for each pixel. Finally, the metric loss is combined with the segmentation loss to obtain the total loss.

Calculating the Metric Loss

Image Decoupling Module Specifically, the image decoupling module groups the pixels of the input image into three sets with respect to their categories:

$$\begin{aligned} s_k^{tp} &= set\{y_{\cdot k} \odot y_{\cdot k}^{seg}\} \\ s_k^{fn} &= set\{y_{\cdot k} \odot (1 - y_{\cdot k}^{seg})\} \\ s_k^{fp} &= set\{(1 - y_{\cdot k}) \odot y_{\cdot k}^{seg}\}, \end{aligned} \tag{3}$$

where $s_k^{tp}, s_k^{fn}, s_k^{fp}$ are sets of TP, FN and FP pixels for category k. $y_{\cdot k}$ and $y_{\cdot k}^{seg}$ are one-hot ground truth and prediction map of k, and $\odot$ represents the element-wise product.

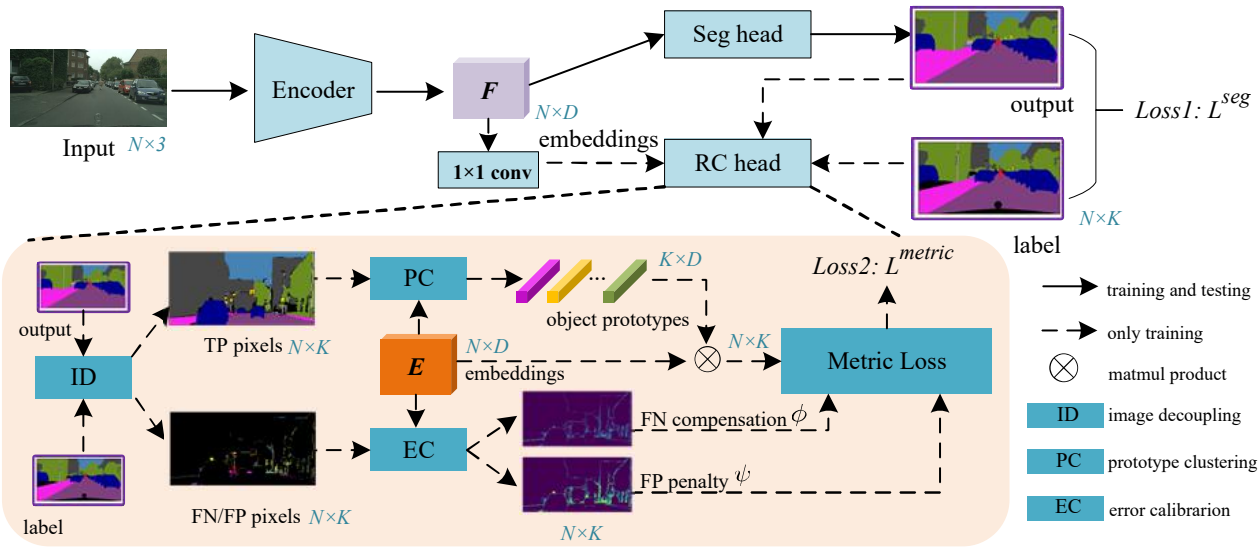


Figure 3: The pipeline of the proposed method. The segmentation head and representation calibration head (RCH) are jointly trained in the training phase, while RCH is discarded in the inference stage.

Prototype Clustering The PC module adopts the current and historical TP pixels to calculate the category prototype $\boldsymbol{\mu}_k \in \mathbb{R}^{1\times D}$. The prototype is updated via exponential moving average (EMA) (Ye et al. 2019; Wu et al. 2018):

$$\boldsymbol{\mu}_k = \rho\boldsymbol{\mu}_k + (1-\rho)\frac{1}{n_k^{tp}}\sum_{i\in s_k^{tp}} \boldsymbol{e}_i, \tag{4}$$

where n_k^{tp} is the number of current TP pixels for category k. $\boldsymbol{e}_i \in \mathbb{R}^{1\times D}$ is the embedding for pixel i, and ρ is the momentum value to adjust the retained proportion of historical prototypes. Each category maintains only one prototype, so the size of all category prototypes is $K \times D$. Compared with methods which restore all pixels into memory banks as prototypes (Wang et al. 2021), the storage capacity of our method is intensively reduced.

Then, the cosine similarity between pixel i and prototype $\boldsymbol{\mu}_k$ is calculated in the embedding space:

$$\cos\theta_{ik} = \tilde{\boldsymbol{\mu}}_k\tilde{\boldsymbol{e}}_i^\top = \frac{\boldsymbol{\mu}_k\boldsymbol{e}_i^\top}{||\boldsymbol{\mu}_k||_2||\boldsymbol{e}_i||_2}, \tag{5}$$

where $||\boldsymbol{\mu}_k||_2$ is the L_2 distance and $\tilde{\boldsymbol{\mu}}_k$ is the normalized vector with magnitude 1. Our method aims at maximizing $\cos\theta_{ik}$ to make $\boldsymbol{e}_i$ close to prototype $\boldsymbol{\mu}_k$, thus clustering pixels of the same category to improve intra-class consistency.

Error Calibration The EC module treats FP and FN errors by different strategies. The FP errors occur when pixels of other categories are excessively similar to category k in the feature space. To tackle this, we first take pixel i of category k as an anchor. Then we calculate the average cosine similarity between all FP pixels and the anchor to determine the penalty term:

$$\psi_i = \begin{cases} 1 + \frac{1}{n_k^{fp}}\sum_{j\in s_k^{fp}} \tilde{\boldsymbol{e}}_j\tilde{\boldsymbol{e}}_i^\top & \text{, if } n_k^{fp}>0; \\ 0 & \text{, otherwise.} \end{cases} \tag{6}$$

By minimizing $\psi_i \in [0,2]$, FP pixels are pushed to the opposite pole against anchor i in the hyperspherical embedding space. Ideally, ψ_i is 0 when there is no FP pixels or the average similarity converges to -1, indicating all FP pixels are at the opposite direction from anchor i.

On the other hand, the FN errors occur when pixels belonging to category k are misclassified as other categories. We reduce the FN errors by introducing a compensation term to pull the FN pixels back to its true category. Thus, we determine the compensation term ϕ_i for pixel i of object k as:

$$\phi_i = \begin{cases} 1 - \frac{1}{n_k^{fn}}\sum_{j\in s_k^{fn}} \tilde{\boldsymbol{e}}_j\tilde{\boldsymbol{e}}_i^\top & \text{, if } n_k^{fn}>0; \\ 0 & \text{, otherwise.} \end{cases} \tag{7}$$

By minimizing $\phi_i \in [0,2]$, we gradually pull the FN pixels towards the anchor i. Ideally, ϕ_i is reduced to 0 when there is no FN pixels or all FN and TP pixels are placed at the same direction in the feature space.

Metric Loss Function The proposed metric loss function integrates $\cos\theta_{ik}$, ψ_i and ϕ_i with the following form:

$$L_i^{metric} = -\log\frac{e^{\cos\theta_{ik}/\tau-(1-p_{ik})\phi_i}}{e^{\cos\theta_{ik}/\tau-(1-p_{ik})\phi_i}+\sum_{l\neq k}e^{\cos\theta_{il}/\tau}} - \log\frac{e^{\cos\theta_{ik}/\tau-(1-p_{ik})\psi_i}}{e^{\cos\theta_{ik}/\tau-(1-p_{ik})\psi_i}+\sum_{l\neq k}e^{\cos\theta_{il}/\tau}}, \tag{8}$$

where τ is the temperature hyper-parameter. The compensation term $\phi_i \geq 0$ and penalty term $\psi_i \geq 0$ are applied to

the numerator, resulting in a higher loss value, which means the existing prototype clustering loss (Wu et al. 2018; Ye et al. 2019) is a lower bound of the proposed metric loss function. By minimizing L_i^{metric}, all FP pixels are pushed to the opposite direction against anchor i, and all FN pixels are pulled to the same direction towards anchor i. Overall, The total loss value for pixel i can be written as follows:

$$L_i = L_i^{seg} + \lambda L_i^{metric}, \tag{9}$$

where λ is the factor to adjust the strength of the two losses.

Theoretical Analysis We elaborate the effectiveness of L_i^{metric} from the perspective of gradients. For simplicity, we denote the two parts of L_i^{metric} as $L_i^{'}$ and $L_i^{''}$. The values in the log function are denoted as $S^{'}$ and $S^{''}$. The gradients of pixel i is given as follows:

$$\frac{\partial L_i^{'}}{\partial \tilde{\boldsymbol{e}}_i} = -(1-S_k^{'})(\frac{\tilde{\boldsymbol{\mu}}_k}{\tau} + \frac{1-p_{ik}}{n_k^{fn}} \sum_{j \in s_k^{fn}} \tilde{\boldsymbol{e}}_j) + \sum_{l \neq k} S_l^{'} \frac{\tilde{\boldsymbol{\mu}}_l}{\tau}$$

$$\frac{\partial L_i^{''}}{\partial \tilde{\boldsymbol{e}}_i} = -(1-S_k^{''})(\frac{\tilde{\boldsymbol{\mu}}_k}{\tau} - \frac{1-p_{ik}}{n_k^{fp}} \sum_{j \in s_k^{fp}} \tilde{\boldsymbol{e}}_j) + \sum_{l \neq k} S_l^{''} \frac{\tilde{\boldsymbol{\mu}}_l}{\tau}. \tag{10}$$

Equation (10) shows that by minimizing L_i^{metric}, the anchor pixel i is pulled close to its label prototype $\boldsymbol{\mu}_k$ and far from the other $\boldsymbol{\mu}_l$. Besides, the two error calibration terms also produce extra directions of convergence. The compensation term drives pixel i close to both TP and FN pixels to shrink the intra-class variances, while the penalty term pushes pixel i away from FPs to enrich inter-class differences. p_{ik} is the parameter for hard example mining. It gives more weights to hard-to-identify pixels, i.e., pixels with lower p_{ik} values in the loss function. Furthermore, when there are no FP and FN pixels ($n_k^{fn} = n_k^{fp} = 0$), the loss function degrades to the standard negative log likelihood loss.

From the viewpoint of hard example mining, our calibration terms give stronger supervision to three types of hard pixels: (1) pixels with lower confidence p_{ik}. (2) pixels with higher penalty term ψ_i. (3) pixels with higher compensation term ϕ_i. The calibration terms emphasize on calibrating the above pixels to generate consistent features.

Experiment

In this section, we first introduce the implementation details. Next, we devise series of ablation experiments to analyze the effects of our RCH. Finally, we compare our method with SOTA networks on three popular datasets.

Implementation Details

Dataset We conduct experiments on Cityscapes (Cordts et al. 2016), ADE20K (Zhou et al. 2017) and Pascal Context (Mottaghi et al. 2014). The Cityscapes contains 19 categories from 5000 images of high resolution (2048 × 1024), of which 2975 images for training, 500 images for validation and 1525 for testing. The ADE20K is a scene parsing dataset covering 150 classes from 20210 images. The dataset is divided into 20K/2K/3K images for training, validation and testing, respectively. The Pascal Context dataset contains 59 semantic classes and 1 background class. The training set and test set consist of 4998 and 5105 images respectively.

Network	prototype	FN	FP	mIoU	$\triangle_{mIoU}$
OCRNet	-	-	-	81.11	-
OCRNet	Learned	-	-	81.34	0.23
OCRNet	GTs	-	-	81.55	0.44
OCRNet	TPs	-	-	81.75	0.64
OCRNet	TPs	✓	-	81.93	0.82
OCRNet	TPs	-	✓	81.88	0.77
OCRNet	TPs	✓	✓	82.24	1.13

Table 1: Ablation study on each component. The "prototype" in the table header means different ways to obtain the object prototypes, including learnable parameters, calculated by GT or TP pixels. The "FN" and "FP" denote the FN compensation and FP penalty terms, respectively. $\triangle_{mIoU}$ indicates the improvement in terms of mIoU.

Network We conduct experiments with four SOTA networks (including encoder and segmentation head): ResNet101+DeepLabV3 (Chen et al. 2017), HRNetW48+FCN (Sun et al. 2019), HRNetW48+OCRNet (Yuan, Chen, and Wang 2020) and MiT+SegFormer (Xie et al. 2021). For simplicity, we mark the network trained with RCH as $\star$. Note that, we choose the output of the encoder as the intermediate features for RCH. For reproducibility, we use $mmsegmentation$ (Contributors 2020) as our codebase and the networks are trained with 8 Nvidia Titan XP. The encoders are pre-trained on ImageNet-1k (Krizhevsky, Sutskever, and Hinton 2012). The segmentation head and RCH are randomly initialized.

Data Augmentation In the training phase, we first apply the random horizontal flip and random scale of $\{0.5, 0.75, 1.0, 1.5, 1.75, 2.0\}$ to augment the input images. Then, we randomly crop the large images or pad the small images into a fixed size for training (512 × 512 for ADE20K and Pascal Context, 512 × 1024 for Cityscapes).

Training and Inference We train the models using Adam optimizer with the initial learning rate 0.01, weight decay 0.0005 and momentum 0.9. The learning rate dynamically decays exponentially according to the 'ploy' strategy. To provide a fair comparison, we adopt the widely-used tricks: OHEM (Shrivastava, Gupta, and Girshick 2016) and auxiliary loss (Zhao et al. 2017) to all networks.

For the ablation study, we train networks for 40K iterations with a batch size of 8 on Cityscapes train set. The results are obtained by the whole test strategy on the validation set. For comparison with SOTA, we train networks with iterations of batch size of 160K and 8 on Cityscapes, 160K and 16 on ADE20K, 30K and 16 on Pascal Context, respectively. We do inference using sliding windows on input images with multiple scales: $\{0.5, 0.75, 1.0, 1.5, 1.75, 2.0\}$. All experimental configurations are the same no matter whether RCH is used or not.

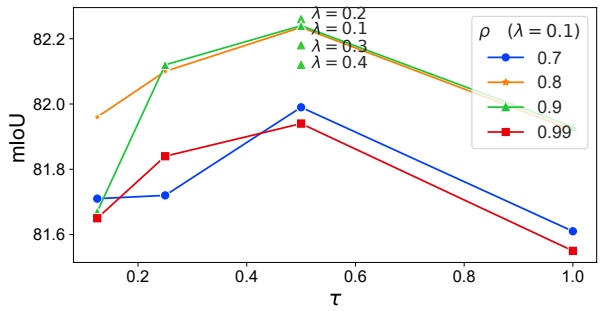


Figure 4: Ablation study on the temperature τ, momentum ρ of EMA and loss factor λ for our RCH.

Type	Network	Encoder	mIoU
CNN	DeepLabV3	ResNet101	79.78
	DeepLabV3⋆	ResNet101	80.72 (↑ 0.94)
	HRNet	HRNet-W48	80.69
	HRNet⋆	HRNet-W48	81.83 (↑ 1.14)
	OCRNet	HRNet-W48	81.11
	OCRNet⋆	HRNet-W48	82.24 (↑ 1.13)
Trans	SegFormer	MiT-B0	76.2
	SegFormer⋆	MiT-B0	76.8 (↑ 0.6)
	SegFormer	MiT-B1	78.5
	SegFormer⋆	MiT-B1	79.0 (↑ 0.5)
	SegFormer	MiT-B2	81.0
	SegFormer⋆	MiT-B2	81.6 (↑ 0.6)

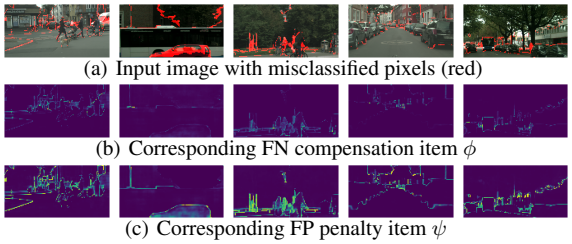
Table 2: Experiments on different networks.

Ablation Study

Effectiveness of the Components We first evaluate the effectiveness of the prototype clustering, FN and FP terms in our proposed RCH. Experimental results are reported in Table 1. We first train the OCRNet with different prototype clustering strategies, i.e., without prototype, or setting prototypes as learnable parameters or calculating by GT or TP pixels, respectively. Conclusions from results in Table 1 Row 1-4 are (1) Giving additional surveillance to the intermediate features is beneficial to the performance. (2) Benefitting from prototype clustering, collecting pixels as prototypes are more helpful than learnable parameters. (3) Generating prototypes without the misclassified pixels can further gain the performance improvements. The on-off experiments in Row 5-6 demonstrate the superior performance of the FN compensation and FP penalty terms, which bring the improvement of 0.82 and 0.77 in terms of mIoU, respectively. Notably, simultaneously adopting the above two calibration terms further improves the mIoU by 1.13.

Ablation on Hyper-parameters In our proposed method, there are three hyper-parameters including the temperature τ, momentum ρ of EMA and loss factor λ for the RCH. We first fix $\lambda = 0.1$ and conduct the grid search for τ and ρ as shown in Figure 4. For the ρ, a higher value means that the object prototypes favor retaining historical embeddings. A better choice would be $\rho = 0.8$ or $\rho = 0.9$. Bigger values of τ make the two calibration terms play more prominent roles, but the network is more difficult to converge if τ is too large. We can observe that setting τ in the range of $[0.25, 0.5]$ is a reasonable choice. For the loss factor λ in the RCH, referring to auxiliary loss (Zhao et al. 2017) as a priori, we set the upper value of λ as 0.4 and raise λ gradually from 0.1 to 0.4 in a step of 0.1, while $\tau = 0.5$ and $\rho = 0.9$. As shown in Figure 4, sampling λ from a low value range makes almost negligible impact on the performance.

In the following experiments, we set τ, ρ and λ as 0.5, 0.9 and 0.1, respectively.



(a) Input image with misclassified pixels (red)

(b) Corresponding FN compensation item ϕ

(c) Corresponding FP penalty item ψ

Figure 5: Visualization on two calibration terms. We feed input images into OCRNet⋆ to output the FN compensation ϕ and FP penalty ψ terms. For better demonstration, we mask the misclassified pixels with red in Row 1 and show the two calibration terms in Row 2-3. The brighter color means the higher value.

Performance on More Networks To demonstrate that our RCH can consistently improve performance on various networks, we conduct experiments on three CNN-based networks: DeepLabV3, HRNet, OCRNet and one transformer-based network SegFormer with three different model sizes. Note that SegFormer has to perform a warm-up training first before regular training. We are in line with that setting and do not use the RCH during the warm-up time. As shown in Table 2, for the DeepLabV3, HRNet and OCRNet, it is remarkable that our proposed method succeeds in boosting the mIoU by 0.94, 1.14 and 1.13, benefiting from improving the representation consistency via calibrating the misclassified features. More satisfactorily, our method is still effective for the transformer-based networks. As shown in Table 2, we achieve a performance improvement of 0.5-0.6 mIoU in the face of encoders with different sizes.

Visualization Analysis

Visualization on Two Calibration Terms As shown in Figure 5, in general, both calibration terms are not only distributed over the misclassified pixels, but also expanded to neighboring pixels. The above observations illustrate that our proposed calibration terms are capable of discovering the pixels similar to the misclassified ones. With the supervision of RCH, networks pull intra-class pixels together and push away inter-class pixels, leading to significant performance improvement.

Visualization on Feature Similarities To exhibit the improvement of intra-class consistency and inter-class discrim-

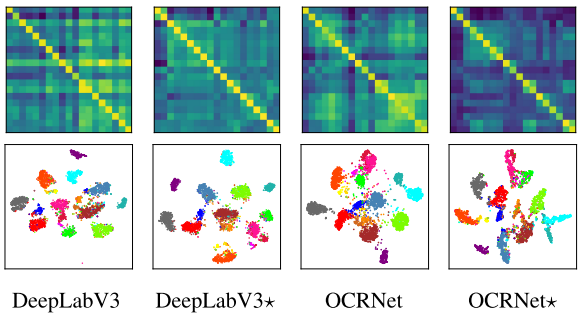


Figure 6: Visualization on feature similarities. Row 1 is the average cosine similarity matrices among all categories for four networks, where the lighter color indicates the higher similarity. Row 2 is the T-SNE (Van der Maaten and Hinton 2008) point map of the intermediate features obtained from four networks, where clusters with different colors represent different categories. We conduct experiments on Cityscapes validation set.

ination of feature representations by RCH, we visualize the cosine similarity matrix among categories, which is shown in Row 1 of Figure 6. Concretely, we feed all the images in the Cityscapes validation set into four well-trained networks to extract intermediate features for all pixels. Then, we separate the pixels into 19 sets based on categories. The value in an index ij of the matrix is the average cosine similarity between all pixels in the set i and j. The values on the diagonals reflect the intra-class similarities and the off-diagonals indicate the inter-class similarities. An obvious conclusion can be summarized that the values on the off-diagonal in the similarity matrix tend closer to -1 and the values on the diagonal lean more towards +1 when our RCH is applied. Meanwhile we adopt a common-used dimensionality reduction toolkit T-SNE (Van der Maaten and Hinton 2008) to reduce the dimensionality of the intermediate features to 2 and display pixels from different categories with different colors. The results show that our proposed method is adept in shrinking the distance among pixels from the same category and increasing the margin among pixels from different categories.

Qualitative Analysis on Predictions We qualitatively show that the calibrated features from our method could help reduce misclassified pixels, as illustrated in Figure 7. For example, in Row 3, the original OCRNet misclassified many pixels for the motorcycle and rider, while our method can clearly distinguish them.

Comparision with SOTA

To further demonstrate the superiority of our RCH, we conduct experiments on the SOTA networks across Cityscapes, ADE20K and Pascal Context datasets. As shown in Table 3, on Cityscapes, HRNet⋆ yields 82.7 mIoU using our RCH on Cityscapes test set, outperforming the original one by 1.1 mIoU. For OCRNet, we obtain an improvement of 0.9 mIoU. The existing SOTA networks usually require a trade-off between performance and computational complexity. For instance, HRNet adds a large amount of parameters and transformer-based networks have to bound extra datasets (Krizhevsky, Sutskever, and Hinton 2012) to guarantee results while suffering from high computational overheads. Our method improves performance in a cost-effective way, i.e., we keep the inference structure exactly the same as original networks. Compared to methods (He et al. 2021; Wang et al. 2021) whcih also design extra loss functions, we still outperform them in terms of mIoU. Finally, we evaluate our method on the ADE20K validation set and Pascal Context test set. Compared to Cityscapes, these two datasets have lower resolutions and more categories. The remarkable results are illustrated in Figure 8. Network segmentation performance usually suffers from more FN/FP pixels in complicated scene images which exacerbates the issue of disturbances among categories. Our RCH effectively alleviates the above issue and outperforms the original OCRNet

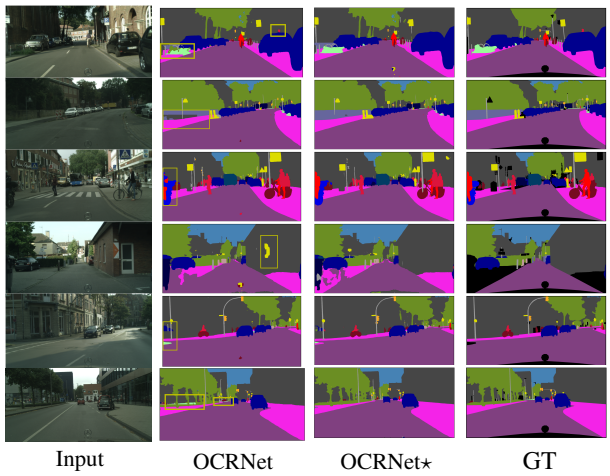


Figure 7: Quality analysis on predictions. From left to right, we list the input images, the predictions of OCRNet, OCRNet⋆ and the ground truth. From top to bottom, there are six demos and a colorbar for the corresponding categories. Black regions in GT indicate they are ignored.

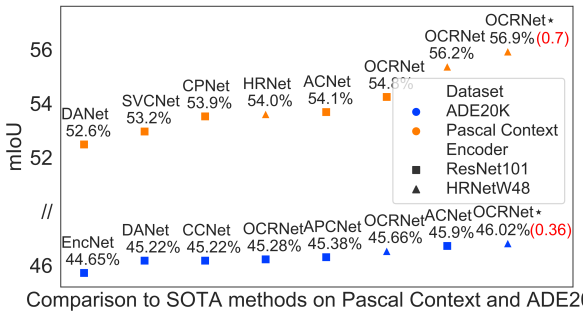


Figure 8: Comparison to SOTA networks on ADE20K validation set (marked as blue color) and Pascal Context test set (marked as orange color). We use the square and triangle to mark the networks using ResNet101 and HRNetW48 as the encoder, respectively.

Network	Encoder	mIoU
PSPNet (Zhao et al. 2017)	ResNet101	78.4
PSANet (Zhao et al. 2018)	ResNet101	80.1
SVCNet (Ding et al. 2019)	ResNet101	81.0
CPNet (Yu et al. 2020)	ResNet101	81.3
CCNet (Huang et al. 2019)	ResNet101	81.4
DANet (Fu et al. 2019)	ResNet101	81.5
OCRNet (Yuan, Chen, and Wang 2020)	ResNet101	81.8
ACFNet (Zhang et al. 2019)	ResNet101	81.9
GFFNet (Li et al. 2020b)	ResNet101	82.3
CSFRN† (He et al. 2021)	ResNet101	82.6
ContrastSeg† (Wang et al. 2021)	HRNet-W48	82.5
ContrastSeg+OCRNet†	HRNet-W48	83.2
SETR (Zheng et al. 2021)	ViT	81.6
SegFormer (Xie et al. 2021)	MiTb5	82.2
SegFormer‡	MiTb5	83.1
HRNet (Sun et al. 2019)	HRNet-W48	81.6
HRNet⋆	HRNet-W48	**82.7**
OCRNet	HRNet-W48	82.4
OCRNet⋆	HRNet-W48	**83.3**

Table 3: Comparison to SOTA networks on Cityscapes test set. We separately use ⋆, † and ‡to mark networks trained with our RCH, extra loss functions and using extra datasets, e.g., Mapillary, ImageNet-22K.

on two datasets by 0.36 and 0.7 mIoU, respectively.

Conclusion

In this paper, we present a representation calibration head (RCH) to rectify features of misclassified pixels and enhance intra-class consistency. RCH consists of the ID, PC, EC modules and a metric loss function. Moreover, RCH is abandoned in the inference stage without any additional computational cost. Extensive experimental results show that RCH as a plug-in could significantly improve the segmentation performance of SOTA networks, i.e., OCRNet, HRNet, SegFormer, on three challenging datasets, including Cityscapes, ADE20K, Pascal Context. Future work includes theoretical understanding of the metric loss functions and extension of RCH to more computer vision tasks.

Acknowledgments

This work is supported by the Zhejiang Provincial key RD Program of China (2021C01119) , and the Zhejiang University-Angelalign Inc. R&D Center for Intelligent Healthcare.

References

Berman, M.; Triki, A. R.; and Blaschko, M. B. 2018. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4413–4421.

Cakir, F.; He, K.; Xia, X.; Kulis, B.; and Sclaroff, S. 2019. Deep metric learning to rank. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1861–1870.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Cheng, D.; Gong, Y.; Zhou, S.; Wang, J.; and Zheng, N. 2016. Person re-identification by multi-channel parts-based cnn with improved triplet loss function. In *Proceedings of the iEEE conference on computer vision and pattern recognition*, 1335–1344.

Contributors, M. 2020. MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark. https://github.com/open-mmlab/mmsegmentation.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4690–4699.

Ding, H.; Jiang, X.; Shuai, B.; Liu, A. Q.; and Wang, G. 2019. Semantic correlation promoted shape-variant context for segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8885–8894.

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

Dosovitskiy, A.; Springenberg, J. T.; Riedmiller, M.; and Brox, T. 2014. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 27: 766–774.

Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; and Lu, H. 2019. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3146–3154.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, X.; Liu, J.; Fu, J.; Zhu, X.; Wang, J.; and Lu, H. 2021. Consistent-Separable Feature Representation for Semantic Segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1531–1539.

Huang, Y.; Wang, Y.; Tai, Y.; Liu, X.; Shen, P.; Li, S.; Li, J.; and Huang, F. 2020. Curricularface: adaptive curriculum learning loss for deep face recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5901–5910.

Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; and Liu, W. 2019. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 603–612.

Joseph, K.; Khan, S.; Khan, F. S.; and Balasubramanian, V. N. 2021. Towards open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5830–5840.

Ko, B.; Gu, G.; and Kim, H.-G. 2021. Learning with Memory-based Virtual Classes for Deep Metric Learning. *arXiv preprint arXiv:2103.16940*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105.

Li, J.; Zhou, P.; Xiong, C.; and Hoi, S. C. 2020a. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*.

Li, X.; Zhao, H.; Han, L.; Tong, Y.; Tan, S.; and Yang, K. 2020b. Gated fully fusion for semantic segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11418–11425.

Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; and Song, L. 2017. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 212–220.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3431–3440.

Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.-G.; Lee, S.-W.; Fidler, S.; Urtasun, R.; and Yuille, A. 2014. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 891–898.

Qiao, L.; Zhao, Y.; Li, Z.; Qiu, X.; Wu, J.; and Zhang, C. 2021. DeFRCN: Decoupled Faster R-CNN for Few-Shot Object Detection. *arXiv preprint arXiv:2108.09017*.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815–823.

Shrivastava, A.; Gupta, A.; and Girshick, R. 2016. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 761–769.

Sohn, K. 2016. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, 1857–1865.

Sun, K.; Xiao, B.; Liu, D.; and Wang, J. 2019. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5693–5703.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, J.; Zhou, F.; Wen, S.; Liu, X.; and Lin, Y. 2017. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*, 2593–2601.

Wang, M.; and Deng, W. 2020. Mitigating bias in face recognition using skewness-aware reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9322–9331.

Wang, W.; Zhou, T.; Yu, F.; Dai, J.; Konukoglu, E.; and Van Gool, L. 2021. Exploring cross-image pixel contrast for semantic segmentation. *arXiv preprint arXiv:2101.11939*.

Wang, X.; Girshick, R.; Gupta, A.; and He, K. 2018. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7794–7803.

Wang, X.; Zhang, S.; Wang, S.; Fu, T.; Shi, H.; and Mei, T. 2020. Mis-classified vector guided softmax loss for face recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12241–12248.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3733–3742.

Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. *arXiv preprint arXiv:2105.15203*.

Xu, X.; Huang, Y.; Shen, P.; Li, S.; Li, J.; Huang, F.; Li, Y.; and Cui, Z. 2021. Consistent Instance False Positive Improves Fairness in Face Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 578–586.

Yang, L.; Zhuo, W.; Qi, L.; Shi, Y.; and Gao, Y. 2021. Mining Latent Classes for Few-shot Segmentation. *arXiv preprint arXiv:2103.15402*.

Ye, M.; Zhang, X.; Yuen, P. C.; and Chang, S.-F. 2019. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6210–6219.

Yu, B.; and Tao, D. 2019. Deep metric learning with tuplet margin loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6490–6499.

Yu, C.; Wang, J.; Gao, C.; Yu, G.; Shen, C.; and Sang, N. 2020. Context prior for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12416–12425.

Yuan, Y.; Chen, X.; and Wang, J. 2020. Object-contextual representations for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, 173–190. Springer.

Zhang, F.; Chen, Y.; Li, Z.; Hong, Z.; Liu, J.; Ma, F.; Han, J.; and Ding, E. 2019. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6798–6807.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2881–2890.

Zhao, H.; Zhang, Y.; Liu, S.; Shi, J.; Loy, C. C.; Lin, D.; and Jia, J. 2018. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 267–283.

Zhao, S.; Wang, Y.; Yang, Z.; and Cai, D. 2019. Region mutual information loss for semantic segmentation. *arXiv preprint arXiv:1910.12037*.

Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H.; et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6881–6890.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 633–641.