MuGE: Multiple Granularity Edge Detection

Caixia Zhou[1], Yaping Huang[1]*, Mengyang Pu[2], Qingji Guan[1], Ruoxi Deng[3], Haibin Ling[4]
[1]Beijing Key Laboratory of Traffic Data Analysis and Mining, Beijing Jiaotong University
[2] North China Electric Power University [3] Wenzhou University [4]Stony Brook University
{cxzhou,yphuang,qjguan}@bjtu.edu.cn, mengyang.pu@ncepu.edu.cn,
ruoxii.deng@gmail.com, hling@cs.stonybrook.edu

Abstract

Edge segmentation is well-known to be subjective due to personalized annotation styles and preferred granularity. However, most existing deterministic edge detection methods produce only a single edge map for one input image. We argue that generating multiple edge maps is more reasonable than generating a single one considering the subjectivity and ambiguity of the edges. Thus motivated, in this paper we propose multiple granularity edge detection, called MuGE, which can produce a wide range of edge maps, from approximate object contours to fine texture edges. Specifically, we first propose to design an edge granularity network to estimate the edge granularity from an individual edge annotation. Subsequently, to guide the generation of diversified edge maps, we integrate such edge granularity into the multi-scale feature maps in the spatial domain. Meanwhile, we decompose the feature maps into low-frequency and high-frequency parts, where the encoded edge granularity is further fused into the high-frequency part to achieve more precise control over the details of the produced edge maps. Compared to previous methods, MuGE is able to not only generate multiple edge maps at different controllable granularities but also achieve a competitive performance on the BSDS500 and Multicue benchmark datasets.

1. Introduction

As a fundamental low-level vision task, edge detection can greatly benefit numerous downstream tasks, such as image inpainting [39], semantic segmentation [62], low-light image enhancement [57], salient object detection [43]. The success of deep learning techniques largely improves the performance of edge detection, where some advanced architectures [34, 42, 55], efficient loss functions [7, 8, 19], and lightweight networks [9, 13, 48, 49] have been proposed and successfully surpassed human performance.

*Corresponding author.

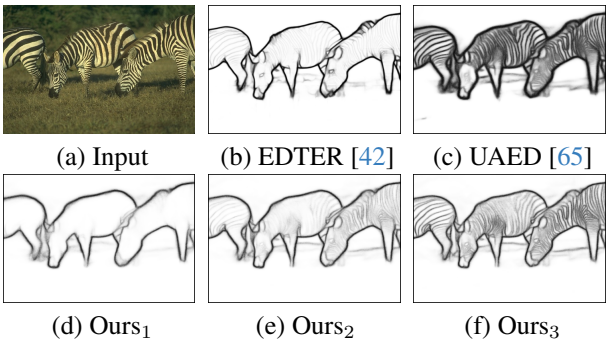

Figure 1. Comparisons between the state-of-the-art methods and our proposed MuGE. (a) shows an image from the BSDS500 test set. (b) and (c) show the predictions from the recent best methods, EDTER [42] and UAED [65]. (d)-(f) shows our generated plausible edge maps at different granularities that better reflect the ambiguity and subjectivity of edge detection.

It is common that people produce different edge maps of one image since they have different annotation styles or granularity preferences. Consequently, in view of inherent ambiguity and the diversity of human perception, we argue that it is more reasonable to design a model that can generate multiple predictions to satisfy such discrepancy for edge detection. In fact, most edge detection datasets, such as BSDS500 [1] and Multicue [37], invite multiple annotators to label one image, and thus provide multiple annotated edge maps. However, most existing methods [7, 8, 19, 34, 42, 55] simply treat the edge detection as a deterministic dense prediction task, where the supervisions are the fusion of multiple annotations by the majority voting strategy. Therefore, for a given image (Fig. 1(a)), the trained detectors can only generate a single edge map (Fig. 1(b)). Recent probabilistic methods [30, 65] consider the uncertainty of annotations by modeling the label distributions as Gaussian or Beta distributions to improve the performance. Sampling from the distribution, as a byproduct, can generate multiple results, but the diversity is severely limited and uncontrollable (Fig. 1(c)).

In this paper, we propose multiple granularity edge de-

tection, called MuGE, which captures the diversity in human perception of edges. Our proposed MuGE is the first to have the capability of producing diverse and plausible edge predictions with different granularities (Fig. 1(d)-(f)). To realize our goal, two key challenges need to be solved. One is to estimate the granularity of edge maps provided by different annotators, and the other is to embed the estimated edge granularity into the feature maps to control the detail levels of the edge predictions precisely.

Firstly, to address the first issue, we devise a binary classification network to encode the edge granularity, where edge maps with different annotation complexities are involved for training. Specifically, for each image of the training dataset, we calculate the number of edge pixels in each edge annotation, and then label the one with the fewest pixels as the sample with simple granularity, and the one with the most pixels as the sample with complex granularity. After training, we feed the remaining edge annotations into the network and take the output value ranging from 0 to 1 as the estimated edge granularity. During the inference stage, we simply set the edge granularity from 0 to 1 with a fixed interval to output various edge maps.

Secondly, to generate diverse edge maps flexibly, we embed the edge granularity into the multi-scale feature maps. We first modulate the edge granularity into the feature maps in the spatial domain. Considering that low-frequency components generally reflect the rough object contours and high-frequency components depict the detailed textures, we utilize the Discrete Fourier Transform (DFT) to decouple the feature maps into low-frequency and high-frequency parts. Subsequently, we multiply the high-frequency components with the obtained edge granularity and further concatenate the features of the spatial domain for the final prediction. Embedding the edge granularity into the feature maps in both spatial and frequency domains can help the model enhance the ability of producing distinct edge maps with varying levels of granularity.

Our contributions can be summarized as follows:

- We propose multiple granularity edge detection, called MuGE, which can produce differing edges, covering from simple object contours to complex edge maps that include richer details. To our best knowledge, MuGE is the first edge detector with the capability of producing plausible diverse predictions for edge detection.
- We design an edge granularity network to encode the granularity of the annotated edge maps, and further propose to embed the estimated edge granularity into both spatial and frequency domains to effectively generate multiple edge maps of different granularity.
- Comprehensive experiments on benchmarks demonstrate that our proposed MuGE can not only produce plausible diverse edge maps, but also achieve new state-of-the-art (SOTA) on the BSDS500 and Multicue datasets.

2. Related Work

Edge detection. Edge detection has long been a focal research task. Traditional methods [3, 26] rely on gradient computing of density, color, or texture. In the past 10 years, deep learning-based solutions with supervision have gained prominence, which mainly concentrate on exploring effective network structures and loss functions. HED [55], CEDN [60], RDS [33], RCF [34], CED [52], and BDCN [16] utilize VGG16 [47] to extract feature maps. EDTR [14] and EDTER [42] introduce Transformer for edge detection. PiDiNet [49] and LDC [9] aim to build lightweight networks for the requirement of high efficiency. RindNet [41] investigates fine-grained edge detection. To obtain crisp edges, new loss functions are developed for edge detection, such as LPCB [8], DSCD [7], and CATS [19]. The methods in [59] and [61] tackle the problem from the noisy label perspective. Recent probabilistic methods [30, 65] are proposed to explore the uncertainty underlying the multiple annotations. UAED [65] constructs Gaussian distributions to take full use of all available labels, and BetaNet [30] replaces Bernoulli distribution with Beta distribution in the head function and uses recurrent voting strategy to merge multiple labels.

Previous deterministic works mainly aim at generating a single edge map given one image. Although UAED [65] and BetaNet [30] can generate multiple results by sampling from the learned distributions, the predictions lack diversity and controllability. In contrast, we are the first to focus on producing multiple edge predictions with distinct granularities, which is more suitable for various downstream tasks.

Diverse image generation. Many tasks involve diverse image generation due to the subjectivity, complexity, or insufficient cues, such as medical image segmentation [27], image inpainting [31], translation [5], and colorization [54].

The generative model [10, 15, 18, 24] is commonly used for this purpose. For example, MSGAN [36] increases the distance between images generated from different latent codes, and Divco [32] simultaneously considers the positive and negative relationships between generated images by contrastive learning. StarGAN [5] and StarGAN v2 [6] address image-to-image translation among multiple domains by using the domain labels as the hints. Fang *et al.* [12] integrate a line control matrix into the generator to control the level of details. To generate multiple segmentation variants for a reliable diagnosis in medical image segmentation, Probabilistic UNet [27] and PhiSeg [2] construct ambiguous latent space using Variational Autoencoder, and Valiuddin *et al.* [51] boost the expressiveness of posterior distribution in latent space with normalizing flow. Wolleb *et al.* [53] and Rahman *et al.* [45] harness the powerful generation and rich diversity capabilities of diffusion models. A relevant work to ours is the exploration of label style in [64], which introduces the concept of defining a specific label

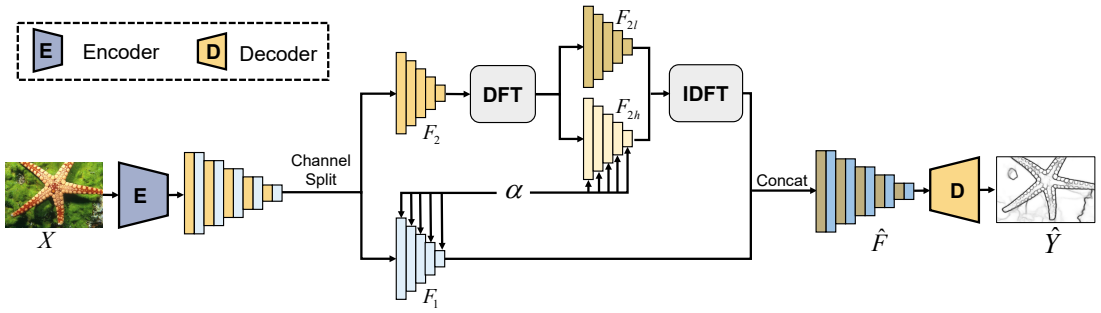

Figure 2. The overall framework of our proposed MuGE. Given an input X and its corresponding label set $\{Y^k\}_{k=1}^K$ and a estimated edge granularity α of one edge map Y^k, we first extract the multi-scale feature maps F from the encoder. Then, we split the features along the channel dimensions (F_1 and F_2) and embed α into the first part F_1. For the second part F_2, we decompose it into high-frequency and low-frequency components (F_{2h} and F_{2l}) by discrete Fourier transform (DFT), where the high-frequency features are multiplied by the edge granularity α because high-frequency components learn local texture details. The embedded features are then converted to the spatial domain using inverse DFT (IDFT). Finally, the enhanced features are recovered along the channel dimension $\hat{F}$ through a connection operator and are fed to the decoder, which generates the final edge map $\hat{Y}$.

style and instructing annotators to follow this style when labeling. The defined style is then fused into probabilistic UNet [27] and stochastic segmentation networks [38].

The above generative approaches, if applied to edge detection, could be challenging to train and potentially degrade the performance [4, 63]. In contrast to these methods, we propose to design an edge granularity network that serves to encode the granularity of edge maps and then integrate such edge granularity into both spatial and high-frequency domains, which not only yields diversified edge maps, but also improves the edge detection performance.

3. Proposed Method

Given an image $X \in \mathbb{R}^{H\times W\times 3}$ and its corresponding annotations $\{Y^k\}_{k=1}^K$, where $Y^k \in \{0,1\}^{H\times W}$ is the k-th annotation and K is the total number of annotations, our approach aims to produce a series of edge maps with diverse detail levels. The training framework of the proposed MuGE is shown in Fig. 2, which follows the encoder-decoder framework in [65]. For an input image X, we first extract the multi-scale feature maps F from the encoder, and split the feature maps into two parts (*i.e.*, F_1 and F_2), where F_1 is directly combined with the estimated edge granularity α and F_2 is decomposed into high-frequency and low-frequency components. Then α is also integrated into the high-frequency components to control the detailed patterns of the produced edge map more flexibly.

3.1. Edge Granularity Network

The first challenge is how to properly encode the granularity of edge maps. We expect that a scalar value between 0 and 1 can be obtained to represent the granularity of an edge map, where a small value indicates a simple edge map that sketches the object boundaries roughly, and a larger value represents a more complex edge map, capable of depicting rich details within objects and background areas.

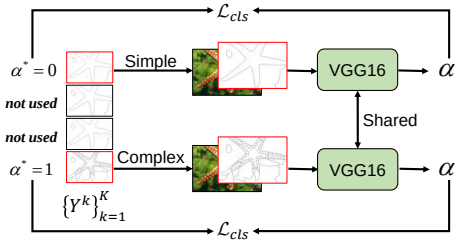

Figure 3. The training framework for the edge granularity network S. Red boxes indicate the simplest and most complex edge maps we selected for training. We connect the selected edge maps with the image as the input, and obtain the predicted edge granularity α by training a VGG16 with ground truth edge granularity α^*.

To achieve the goal, we train a binary classifier S to predict the edge granularity, where class 0 and class 1 refer to the simplest and the most complex edge map. For each image in the training dataset, we first count the number of edge pixels in each annotation Y^k, and then the edge maps with maximum and minimum edge pixels are chosen as the training samples for class 1 and 0, respectively.

Fig. 3 shows the process for training the classifier S. Specifically, we concatenate the image and the edge map with minimum or maximum edge pixels as the input, and the corresponding ground truth edge granularity α^* is set to 0 or 1. VGG16 [17] is chosen as the backbone due to its simplicity and effectiveness. As a typical binary classification task, the binary cross entropy (BCE) is used as the loss function. The whole training process can be written as:

$$\mathcal{L}_{\text{cls}} = -\sum_{i=1}^{N}\Big(\alpha^*\log(\alpha) + (1-\alpha^*)\log(1-\alpha)\Big), \quad (1)$$

where N denotes the number of the training samples, and α denotes the predicted edge granularity.

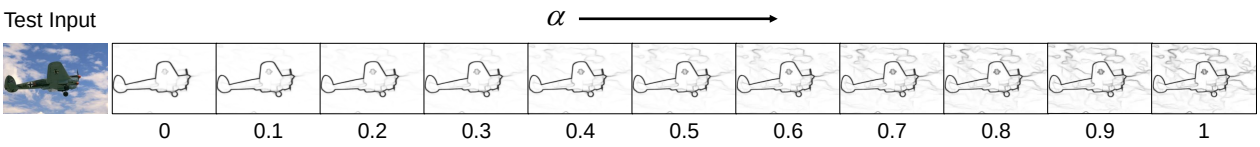

Figure 4. The generated diverse edge predictions for the given test image with different edge granularity α, where α ranges from 0 to 1 in intervals of 0.1 under the MS-VOC setting (multi-scale test setup trained with extra PASCAL VOC data).

After training the edge granularity network, we fix the parameters and take the predictions that range from 0 to 1 as the estimated granularity for the other edge maps, which will be further used for training the edge detection model. In the inference stage, given a test image, we vary the edge granularity α from 0 to 1 with a fixed interval t_α to control the granularity of generated edge maps. The interval t_α can be any arbitrary value, and Fig. 4 displays a test image and the corresponding M predicted edge maps ($M = 11$) with $t_\alpha = 0.1$. As the α value increases, the produced edge map becomes increasingly richer in details. More visualization results about α are given in supplementary materials.

3.2. Edge Detector with Granularity

Our method is built upon the backbone of UAED [65], which consists of an EfficientNet encoder [50] ($\mathcal{E}$) and two UNet++ decoders [66] ($\mathcal{D}_1$ and $\mathcal{D}_2$). The prediction head ($\mathcal{H}_1$ and $\mathcal{H}_2$) is constructed as a Gaussian distribution, whose standard deviation is supervised by the value computed from the label sets to capture ambiguity and fully exploit all annotations. We operate on the extracted feature maps F, which are down-sampled to $1/2$, $1/4$, $1/8$, $1/16$, and $1/32$ of the original images respectively. And the corresponding output channels are 64, 48, 80, 224, and 640.

To better control the granularity of the predictions, in addition to preserving spatial features, we also transform features into the frequency domain, where the frequency features are partitioned into high-frequency and low-frequency components. Elaborately, we split the feature maps F into two parts (F_1 and F_2) along the channel dimension, and directly embed the edge granularity into the first part. Then we decompose the second part into low-frequency F_{2l} and high-frequency F_{2h} using DFT:

$$F_{2l} = B \odot \mathrm{DFT}(F_2), F_{2h} = (\mathbb{1} - B) \odot \mathrm{DFT}(F_2), \quad (2)$$

where $\odot$ is element-wise multiplication and $\mathbb{1}$ is a matrix of ones. $B \in \mathbb{R}^{H \times W}$ is a binary mask with 1 in the center region and 0 elsewhere, where the region size is controlled by a ratio r to distinguish between high-frequency and low-frequency components. r is empirically set to 0.5.

Motivated by the fact that low-frequency components mainly reflect object contours and high-frequency components prefer to depict the details within objects [29], we embed the edge granularity into the high-frequency components. Specifically, if the edge granularity has a relatively small value, indicating a simpler edge map that roughly outlines the contours of objects, the response of the high-frequency components should be suppressed, whereas it should be retained otherwise. Therefore, we multiply high-frequency information F_{2h} by the edge granularity α to obtain new high-frequency $\hat{F}_{2h} = \alpha F_{2h}$ to control the detail levels of the generated edge maps.

Subsequently, we use the inverse DFT (IDFT) to convert the frequency domain back to the spatial domain, *i.e.*, $\hat{F}_2 = \mathrm{IDFT}(F_{2l}) + \mathrm{IDFT}(\hat{F}_{2h})$.

Finally, we concatenate the two features ($\hat{F}_1$ and $\hat{F}_2$) on the channel dimension, then forward them to the decoder ($\mathcal{D}_1$ and $\mathcal{D}_2$) and prediction head ($\mathcal{H}_1$ and $\mathcal{H}_2$) to obtain the final prediction $\hat{Y}$:

$$\begin{aligned} \hat{F} &= \mathrm{concat}(\hat{F}_1; \hat{F}_2), \qquad \hat{F}_1 = \alpha F_1 \\ \hat{\mu} &= \mathcal{H}_1(\mathcal{D}_1(\hat{F})), \;\; \hat{\sigma}^2 = \mathcal{H}_2(\mathcal{D}_2(\hat{F})), \\ \hat{Y} &= \mathrm{sigmoid}(\hat{\mu} + \epsilon\hat{\sigma}), \;\; \epsilon \sim \mathcal{N}(0, \mathrm{I}), \end{aligned} \quad (3)$$

where $\hat{\mu}$ is the mean, $\hat{\sigma}^2$ is the variance of the predicted edge maps, and the prediction $\hat{Y}$ is randomly sampled from the learned distribution by the reparameterization trick [25].

3.3. Network Training

The loss function used for training can be divided into three parts: the edge loss function, which supervises the training of edge predictions; the frequency loss function, which facilitates the recovery of frequency domain information; and the CLIP loss function which constrains the consistency between the granularity of prediction and ground truth.

Edge loss function. We use the loss function developed in UAED [65] to supervise the training of the predicted edge maps, which includes a balanced mean squared error (MSE) loss for uncertainty estimation ($\mathcal{L}_{\mathrm{bvar}}$) and an uncertainty-driven loss for edge detection ($\mathcal{L}_{\mathrm{ue}}$):

$$\begin{aligned} \mathcal{L}_{\mathrm{uaed}} &= \mathcal{L}_{\mathrm{bvar}} + \mathcal{L}_{\mathrm{ue}}, \\ \mathcal{L}_{\mathrm{bvar}} = \sum_{j=1}^{HW} M_j(\hat{\sigma}_j^2 - \sigma_j^2)^2, \;\; \mathcal{L}_{\mathrm{ue}} &= \sum_{j=1}^{HW} \exp(\beta_t \hat{\sigma}_j)\mathcal{L}_{\mathrm{e}}, \\ M_j = \gamma Y_j + (1-\gamma)(1-Y_j), \;\; \gamma &= [Y_-^k]/([Y_-^k] + [Y_+^k]), \\ \mathcal{L}_{\mathrm{e}} = -\sum_{j=1}^{HW} M_j \Big(Y_j^k \log(\hat{Y}_j) &+ (1 - Y_j^k)\log(1 - \hat{Y}_j) \Big), \end{aligned} \quad (4)$$

where j denotes the j-th pixel, $\beta_t = t/T$ is an adaptive factor, t is the current epoch, and T is the total epochs. The notation $[\cdot]$ denotes the number of pixels, and Y_-^k and Y_+^k denote the non-edge and edge pixels respectively in k-th annotated edge map.

Frequency loss function. Edge details can be perceived in the frequency domain. To generate crisp edge maps, we use focal frequency loss (FFL) [22] to close the frequency distance between the prediction and the ground truth at the frequency domain. Due to the prediction with large frequency distance being more likely regarded as a hard sample, the corresponding training weight should be strengthened. Therefore, a weighting strategy emphasizes the learning from these hard samples. Specifically, we transform the prediction and the ground truth into the frequency domain using DFT, and the frequency loss function is presented as:

$$\mathcal{L}_{\text{ffl}} = \sum_{j=1}^{HW} Z_j \left|\text{DFT}_j(Y^k) - \text{DFT}_j(\hat{Y})\right|^2, \tag{5}$$

where $Z_j = |\text{DFT}_j(Y^k) - \text{DFT}_j(\hat{Y})|^\eta$ is the weight put on each frequency. Here we set η 1.

CLIP loss function. To ensure that the granularity of the prediction is similar to that of ground truth, we feed prediction and ground truth to the CLIP visual encoder [44] to extract 512-D features, and use MSE loss to close the distance between the two feature maps. The CLIP loss function can be written as:

$$\mathcal{L}_{\text{clip}} = (\text{CLIP}(\hat{Y}) - \text{CLIP}(Y^k))^2. \tag{6}$$

Total loss function. The final optimization objection is the sum of the above three losses:

$$\mathcal{L} = \mathcal{L}_{\text{uaed}} + \mathcal{L}_{\text{ffl}} + \mathcal{L}_{\text{clip}}. \tag{7}$$

4. Experiments

4.1. Experimental Setting

Dataset. In this section, we conduct experiments on two datasets, BSDS500 [1] and Multicue [37], which contain multiple annotations for evaluation. **BSDS500** contains 500 high-resolution RGB natural scene images with a size of 321×481, divided into 200 for training, 100 for validation, and 200 for testing. Each image is manually annotated by 4-9 annotators. To augment and make full use of the dataset, we process the images following UAED [65], which rotates each image at 25 different angles and flips each image (horizontally, vertically, and both) at each angle. Moreover, we incorporate the PASCAL VOC Context Dataset [11], consisting of 10,103 images, as supplementary training data. **Multicue** contains 100 scenes designed for the study of boundary and edge detection in challenging natural scenes. Each scene includes both left-view and right-view short (10-frame) sequences. The last frame of each left-view sequence is annotated with edges by 6 annotators and boundaries by 5 annotators. Data augmentation involves rotation at 4 different angles (0, 90, 180, 270) and flipping. 80 images are randomly selected for training, leaving the remaining 20 for testing. This process is repeated three times and the average scores of three independent trials are regarded as the final results.

Implementation Details. Our training details are the same as UAED [65], where Pytorch [40] based image segmentation (SMP) neural network library [21] is utilized as the deep learning framework. To speed up the training process, we follow LPCB [8] and UAED [65] to make all training samples the same size, so that the model can be trained in a mini-batch way. For the BSDS500 dataset, we rotate the images to maintain the same size (321×481). For the Multicue dataset, each image with a size of 720×1280 is randomly cropped to 512×512 sub-images for training. The batch size is set to 4 for the edge detector and 16 for the edge granularity network. All parameters are updated by Adam optimizer [23] with a learning rate of $1e$-4. All experiments are conducted on a single RTX 3090.

Evaluation Protocols. We access the diversity of the generated multiple predictions and evaluate the performance using widely used metrics, including optimal dataset scale (ODS), optimal image scale (OIS), and Average Precision (AP). The predictions are processed by non-maximum suppression (NMS) before evaluation following previous works [34, 65]. The localization tolerance is set to 0.0075 to control the maximum allowed distance in matches between the predictions and the ground truth maps.

4.2. Comparison with State-of-the-arts

We first discuss and visualize the diversity of our predictions. Then we compare the performance of the proposed MuGE with existing excellent edge detectors, including traditional detectors such as Canny [3], CNN-based detectors such as RCF [34] and UAED [65], and transformer-based detector EDTR [14] and EDTER [42].

Diverse results on BSDS500. Fig. 5 clearly visualizes that our proposed MuGE can yield diversified plausible predictions with varying detail levels, which better align with human perception and can also benefit different kinds of downstream tasks. Despite that UAED [65] can also generate multiple edge maps by sampling from distributions, its results lack diversity. For further quantitative comparison of the diversity of generated edge predictions, we compute the LPIPS metric [67] in Table 1, where UAED generates $M = 3$ samples with μ and $\mu \pm \sigma$, as well as $M = 11$ samples with $\mu, \mu \pm \sigma, \mu \pm 1.5\sigma, \mu \pm 2\sigma, \mu \pm 2.5\sigma$, and $\mu \pm 3\sigma$, respectively. For MuGE, $M = 3$ and $M = 11$ mean we

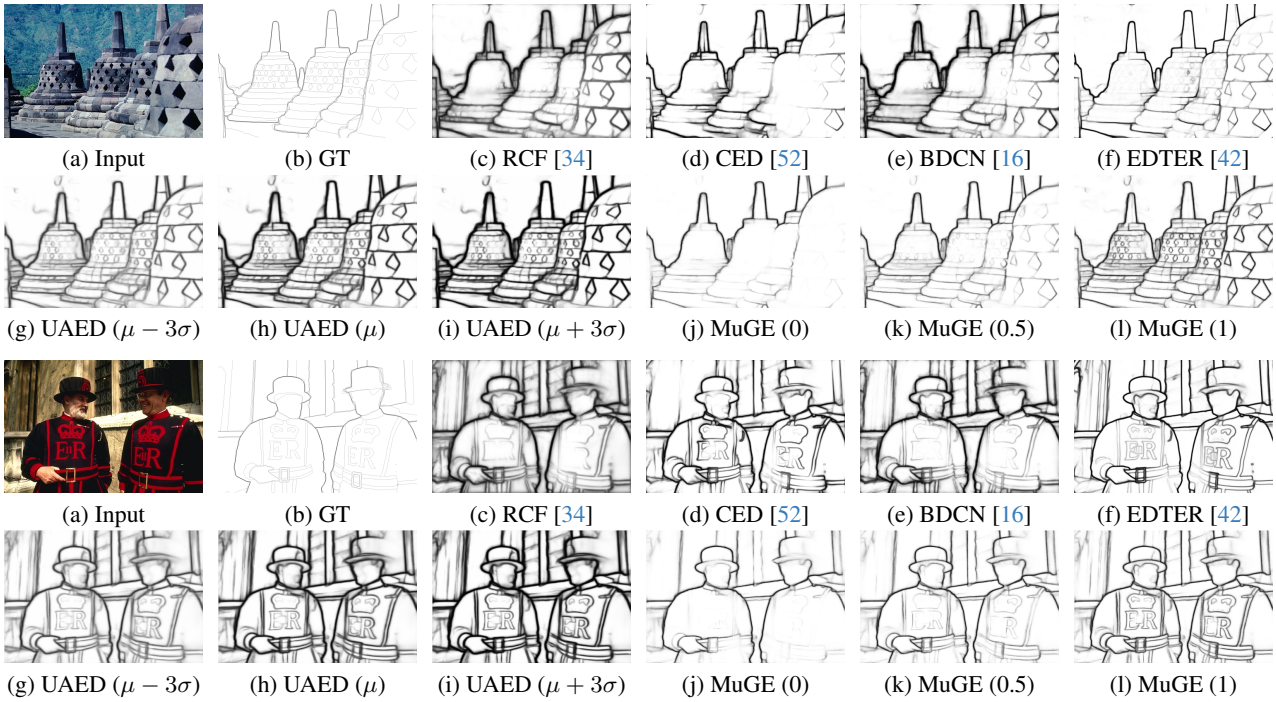

Figure 5. Qualitative comparisons on challenging samples in the BSDS500 test set under MS setting. Note that UAED samples from the learned distribution with μ and $\mu \pm 3\sigma$, and MuGE produces diverse results with edge granularity of 0, 0.5, and 1, respectively.

Table 1. The calculated LPIPS on **BSDS500** [1] (higher is better). M means the number of predictions.

Method	$M = 3$	$M = 11$
UAED	0.0380	0.0118
MuGE (Ours)	0.1663	0.1065

generate 3 and 11 maps with an interval of $t_\alpha = 0.5$ and $t_\alpha = 0.1$ respectively. Table 1 shows that MuGE has a big improvement in producing diverse predictions. More visualization results are given in supplementary materials.

Quantitative results on BSDS500. Since MuGE is capable of producing diverse results, we use the best-matching strategy to report the metrics for the comparison between the best edge prediction generated by our approach and the single final ground truth fused by all annotators' maps for each test image. The results are summarized in Table 2, where our results with different intervals ($t_\alpha = \{0.5, 0.1\}$ produces $M = \{3, 11\}$ results) are reported. For a fair comparison, UAED also generates the same number of predictions for evaluation with the best-matching strategy, which is denoted as UAED*. Obviously, a smaller interval typically generates more potential predictions, leading to better performance. It should be noted that our training model remains the same across various intervals, with the only difference being the generation of varying numbers of predictions in MuGE. The following discussions and experiments are conducted with $M = 11$.

We can see that our proposed MuGE achieves a new state-of-the-art, outperforming other CNN-based and Transformer-based methods. In the single-scale setting, compared with the second best method UAED [65], we obtain a performance gain of 0.9%, 0.9%, and 1.5% in terms of ODS, OIS, and AP. We also achieve ODS=0.861, OIS=0.867, and AP=0.909 under the MS-VOC setting, which also surpasses UAED [65]. Fig. 6 shows more intuitive results through the Precision-Recall curve.

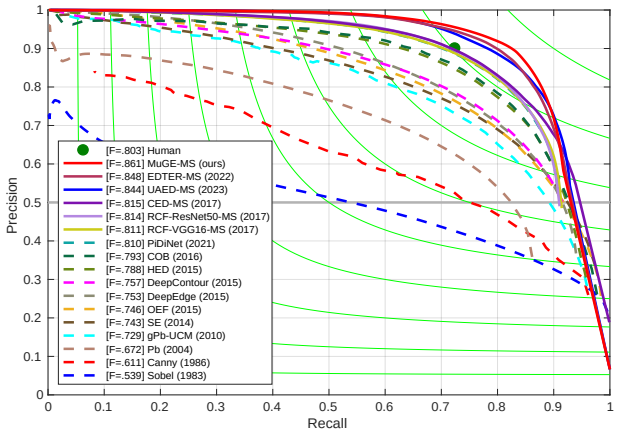

Figure 6. The precision-recall curves for the BSDS500 dataset.

In addition to evaluation with best-matching strategy, we also report the results with a single specific edge granularity α in Table 3. When the granularity of the prediction is the simplest ($\alpha = 0$) or most complex ($\alpha = 1$), the per-

Table 2. Comparisons on the **BSDS500** [1] testing set. VOC means training with extra PASCAL VOC data. The best results are denoted as **red**.

Setup	Method	Backbone	ODS	OIS	AP
Single-Scale (SS)	Canny TPAMI'86 [3]	-	0.611	0.676	0.520
	gPb-UCM TPAMI'10 [1]	-	0.729	0.755	0.745
	DeepContour CVPR'15 [46]	AlexNet	0.757	0.776	0.790
	HED ICCV'2015 [55]	VGG16	0.788	0.808	0.840
	Deep Boundary ICLR'15 [28]	VGG16	0.789	0.811	0.789
	RDS CVPR'16 [33]	VGG16	0.792	0.810	0.818
	COB ECCV'16 [35]	VGG16	0.793	0.820	0.859
	AMH-Net NIPS'17 [56]	ResNet50	0.798	0.829	0.869
	RCF CVPR'17 [34]	VGG16	0.798	0.815	-
	LPCB ECCV'18 [8]	VGG16	0.800	0.816	-
	BDCN CVPR'19 [16]	VGG16	0.806	0.826	0.847
	DSCD ACM'20 [7]	VGG16	0.802	0.817	-
	LDC ACM'21 [9]	MobileNet	0.799	0.816	0.837
	EDTR ICONIP'21 [14]	Transformer	0.820	0.839	0.861
	EDTER CVPR'22 [42]	Transformer	0.824	0.841	0.880
	FCL-Net NN'22 [58]	VGG16	0.807	0.822	-
	UAED CVPR'23 [65]	EfficientNet	0.829	0.847	0.892
	BetaNet KBS'23 [30]	VGG16	0.803	0.822	-
	PEdger ACM'23 [13]	Recurrent	0.823	0.841	-
	UAED* ($M=3$)	EfficientNet	0.838	0.847	0.882
	UAED* ($M=11$)	EfficientNet	0.841	0.847	0.881
	MuGE ($M=3$)	EfficientNet	0.845	0.854	0.895
	MuGE ($M=11$)	EfficientNet	**0.850**	**0.856**	**0.896**
Multi-Scale (MS)	Deep Boundary ICLR'15 [28]	VGG16	0.803	0.820	0.848
	EDTR ICONIP'21 [14]	Transformer	0.830	0.851	0.886
	EDTER CVPR'22 [42]	Transformer	0.840	0.858	0.896
	FCL-Net NN'22 [58]	VGG16	0.816	0.833	-
	UAED CVPR'23 [65]	EfficientNet	0.837	0.855	0.897
	UAED* ($M=3$)	EfficientNet	0.847	0.856	0.879
	UAED* ($M=11$)	EfficientNet	0.850	0.855	0.885
	MuGE ($M=3$)	EfficientNet	0.853	0.863	0.901
	MuGE ($M=11$)	EfficientNet	**0.858**	**0.864**	**0.902**
Single-Scale-VOC (SS-VOC)	Deep Boundary ICLR'15 [28]	VGG16	0.809	0.827	0.861
	RCF CVPR'17 [34]	VGG16	0.806	0.823	-
	CED CVPR'17 [52]	VGG16	0.815	0.833	0.889
	LPCB ECCV'18 [8]	VGG16	0.808	0.824	-
	BDCN CVPR'19 [16]	VGG16	0.820	0.838	0.888
	DSCD ACM'20 [7]	VGG16	0.813	0.836	-
	LDC ACM'21 [9]	MobileNet	0.812	0.826	0.857
	PiDiNet ICCV'21 [49]	PDC	0.807	0.823	-
	EDTER CVPR'22 [42]	Transformer	0.832	0.847	0.886
	FCL-Net NN'22 [58]	VGG16	0.815	0.834	-
	UAED CVPR'23 [65]	EfficientNet	0.838	0.855	0.902
	UAED* ($M=3$)	EfficientNet	0.849	0.856	0.891
	UAED* ($M=11$)	EfficientNet	0.851	0.857	0.892
	MuGE ($M=3$)	EfficientNet	0.852	0.859	0.904
	MuGE ($M=11$)	EfficientNet	**0.855**	**0.860**	**0.905**
Multi-Scale-VOC (MS-VOC)	Deep Boundary ICLR'15 [28]	VGG16	0.813	0.831	0.866
	RCF CVPR'17 [34]	VGG16	0.811	0.830	0.846
	LPCB ECCV'18 [8]	VGG16	0.815	0.834	-
	BDCN CVPR'19 [16]	VGG16	0.828	0.844	0.890
	DSCD ACM'20 [7]	VGG16	0.822	0.859	-
	LDC ACM'21 [9]	MobileNet	0.819	0.834	0.860
	EDTER CVPR'22 [42]	Transformer	0.848	0.865	0.903
	FCL-Net NN'22 [58]	VGG16	0.826	0.845	-
	UAED CVPR'23 [65]	EfficientNet	0.844	0.864	0.905
	UAED* ($M=3$)	EfficientNet	0.856	0.865	0.880
	UAED* ($M=11$)	EfficientNet	0.859	0.866	0.890
	MuGE ($M=3$)	EfficientNet	0.858	0.865	0.907
	MuGE ($M=11$)	EfficientNet	**0.861**	**0.867**	**0.909**

Table 3. Comparisons on **BSDS500** for a single edge granularity.

Method	SS			MS		
	ODS	OIS	AP	ODS	OIS	AP
UAED [65]	0.829	0.847	0.892	0.837	0.855	0.897
Ours ($\alpha=0$)	0.803	0.818	0.855	0.814	0.829	0.887
Ours ($\alpha=0.5$)	0.830	0.846	0.885	0.837	0.855	0.892
Ours ($\alpha=0.6$)	0.831	0.847	0.886	0.838	0.857	0.893
Ours ($\alpha=1$)	0.822	0.839	0.873	0.831	0.849	0.887

Table 4. Comparisons on **Multicue** [37]. All results are obtained by a single-scale input. The best two results are denoted as **red** and **blue** respectively.

	Method	ODS	OIS	AP
Edge	Human VR'16 [37]	0.750 (0.024)	-	-
	Multicue VR'16 [37]	0.830 (0.002)	-	-
	HED ICCV'15 [55]	0.851 (0.014)	0.864 (0.011)	-
	RCF CVPR'17 [34]	0.857 (0.004)	0.862 (0.004)	-
	BDCN CVPR'19 [16]	0.891 (0.001)	0.898 (0.002)	0.935(0.002)
	DSCD ACM'20 [7]	0.871 (0.007)	0.876 (0.002)	-
	LDC ACM'21 [9]	0.881 (0.012)	0.893 (0.011)	-
	PiDiNet ICCV'21 [49]	0.855 (0.007)	0.860 (0.005)	-
	FCL-Net NN'22 [58]	0.875 (0.005)	0.880 (0.005)	-
	EDTER CVPR'22 [42]	0.894 (0.005)	**0.900 (0.003)**	0.944 (0.002)
	UAED CVPR'23 [65]	**0.895 (0.002)**	**0.902 (0.001)**	**0.949 (0.002)**
	MuGE (Ours)	**0.898 (0.004)**	**0.900 (0.004)**	**0.950 (0.004)**
Boundary	Human VR'16 [37]	0.760 (0.017)	-	-
	Multicue VR'16 [37]	0.720 (0.014)	-	-
	HED ICCV'15 [55]	0.814 (0.011)	0.822 (0.008)	0.869 (0.015)
	RCF CVPR'17 [34]	0.817 (0.004)	0.825 (0.005)	-
	BDCN CVPR'19 [16]	0.836 (0.001)	0.846 (0.003)	0.893 (0.001)
	DSCD ACM'20 [7]	0.828 (0.003)	0.835 (0.004)	-
	LDC ACM'21 [9]	0.839 (0.012)	0.853 (0.006)	-
	PiDiNet ICCV'21 [49]	0.818 (0.003)	0.830 (0.005)	-
	FCL-Net NN'22 [58]	0.834 (0.016)	0.840 (0.016)	-
	EDTER CVPR'22 [42]	0.861 (0.003)	0.870 (0.004)	0.919 (0.003)
	UAED CVPR'23 [65]	**0.864 (0.004)**	**0.872 (0.006)**	**0.927 (0.006)**
	MuGE (Ours)	**0.875 (0.006)**	**0.879 (0.006)**	**0.932 (0.004)**

formance is lower than the middle granularity α. It is reasonable since the final ground truth map is merged with all edge maps. Not surprisingly, $\alpha=0.5$ and $\alpha=0.6$ achieve superior results, which also surpass the UAED in terms of ODS and OIS. Since multiple ground truth maps are also provided in BSDS500, we also evaluate the average performance with multiple ground truths, and the results are given in supplementary materials.

Quantitative results on Multicue. Experiments are also performed on the Multicue dataset. Table 4 shows the quantitative results. Our proposed MuGE also achieves a new state-of-the-art on the Multicue edge and boundary (ODS=0.898, OIS=0.900, AP=0.950 on the edge, and ODS=0.875, OIS=0.879, AP=0.932 on the boundary). We surpass the second best UAED [65] by 1.1%, 0.7%, and 0.5% in terms of ODS, OIS, and AP scores on the boundary. Some visual examples with different edge granularity are given in supplementary materials.

4.3. Ablation Study

The crucial designs of MuGE include embedding the edge granularity into both spatial and frequency domains, FFL

Table 5. The ablation study on the BSDS500 dataset for the role of every part plays. All results are obtained by a single-scale input.

Method	Embed (F)	Embed (S)	$\mathcal{L}_{\text{clip}}$	$\mathcal{L}_{\text{ffl}}$	ODS	OIS	AP
UAED					0.829	0.847	0.892
Ours	✓				0.846	0.851	0.899
	✓	✓			0.846	0.853	0.899
	✓	✓	✓		0.849	0.857	0.898
	✓	✓		✓	0.846	0.852	0.894
	✓	✓	✓	✓	0.850	0.856	0.896

loss, and CLIP loss. We conduct ablation experiments on these components to demonstrate the effectiveness of our method. The results are summarized in Table 5.

Effect of embedding the edge granularity. We embed the encoded edge granularity into spatial and frequency domains to control the generation of diverse edges. We can see that embedding the edge granularity into the frequency domain (1st row) can not only generate diverse maps, but also improve the performance by large margins (1.7%, 0.4%, 0.7% in ODS, OIS, and AP), while embedding the edge granularity into both the spatial and frequency domains (2nd row) contributes to the largest performance gain.

Effect of CLIP loss. We introduce the CLIP loss to ensure the detail level of predictions and ground truths remains consistent. The experiments show that CLIP loss can further improve performance by 0.3%, 0.4% (3rd row) and 0.4%, 0.4% (5th row) in ODS and OIS. In addition, we empirically find that introducing the CLIP loss can also help stabilize the training process.

Effect of FFL loss. FFL loss is responsible for aligning the details of edges by recovering the frequency information. Although FFL loss does not significantly improve performance (4th row), it can yield more crisp edges shown in Fig. 7, which is validated by the increased Average Crispness metric [61] (*e.g.*, from 0.228 to 0.291 when $\alpha = 0$).

4.4. Further Analysis

Comparison with fixed edge granularity encoding strategy. In MuGE, we train an edge granularity network to encode the granularity of each edge annotation. Another naive way is using a simple normalization operator, *i.e.*, for all edge maps of each image, we calculate the number of edge pixels in each map and then normalize each edge map according to the maximum and minimum values, resulting in a scalar between 0 and 1 served as the estimated granularity. As shown in Table 6, a simple normalization strategy can lead to a significant performance gain, but our learnable granularity encoding strategy achieves better performance in the single-scale setting.

Different edge granularity embedding strategies. To embed the edge granularity into feature maps for generating diverse predictions, we multiply the edge granularity into the feature maps in the spatial and frequency domains (αF). We also try other embedding strategies for comparison: (1) concat(X; α): connect the edge granularity with the input image along the channel dimension; (2) concat(F; α): connect the edge granularity with the multi-scale feature maps along the channel dimension; (3) AdaIN: use a fully connected layer to map the edge granularity to a scale and bias, and then perform the AdaIN [20] on the multi-scale feature maps. From Table 6, we can see that AdaIN can not bring performance gain. Besides, simply connecting the edge granularity with input image or feature maps can achieve better results, but still lower than our strategy, especially in the SS-VOC setting.

Table 6. The ablation study on the BSDS500 dataset for the different edge granularity encoding and embedding strategies.

Edge Granularity Controlling Strategy		SS			SS-VOC		
		ODS	OIS	AP	ODS	OIS	AP
Encoding	Normalization	0.847	0.852	0.891	0.857	0.862	0.903
	Classifier	0.850	0.856	0.896	0.855	0.860	0.902
Embedding	concat(X; α)	0.849	0.857	0.898	0.848	0.857	0.897
	concat(F; α)	0.848	0.854	0.896	0.848	0.855	0.897
	AdaIN	0.804	0.821	0.857	0.833	0.846	0.893
	Ours	0.850	0.856	0.896	0.855	0.860	0.902

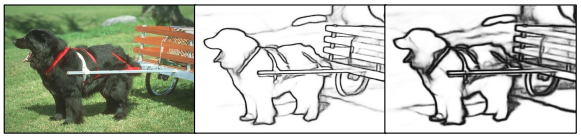

Figure 7. Qualitative visualization of the role of FFL loss on BSDS500 under SS setting. From left to right are the input, and the results with FFL loss and without FFL loss, respectively ($\alpha = 1$).

5. Conclusion

In this paper, we propose a novel edge detector called MuGE which, for the first time, explicitly considers edge granularity and integrates the edge granularity into the feature maps of both spatial and frequency domains, thus generating diverse edge predictions with varying levels of detail. The capability can benefit different kinds of downstream tasks with various demands. Comprehensive experiments are conducted on the BSDS500 and Multicue datasets to demonstrate the superiority of the proposed MuGE.

Limitation. Encoding the edge granularity in the annotation is not the only way to control the detail levels of edge maps, and we will also explore implementing a more flexible edge detector using other schemes (*e.g.*, text prompt).

Acknowledgements. This work was supported by National Natural Science Foundation of China (62271042, 62376021, 62302032, 62106017, 62201401), Beijing Natural Science Foundation (L211015, 4232032), and Hebei Natural Science Foundation (F2022105018). HL was not supported by any fund for this research.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5): 898–916, 2010. 1, 5, 6, 7

[2] Christian F Baumgartner, Kerem C Tezcan, Krishna Chaitanya, Andreas M Hötker, Urs J Muehlematter, Khoschy Schawkat, Anton S Becker, Olivio Donati, and Ender Konukoglu. Phiseg: Capturing uncertainty in medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 119–127. Springer, 2019. 2

[3] John Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986. 2, 5, 7

[4] Jiaqi Chen, Jiachen Lu, Xiatian Zhu, and Li Zhang. Generative semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7111–7120, 2023. 3

[5] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8789–8797, 2018. 2

[6] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8188–8197, 2020. 2

[7] Ruoxi Deng and Shengjun Liu. Deep structural contour detection. In *ACM Int. Conf. Multimedia*, pages 304–312, 2020. 1, 2, 7

[8] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. Learning to predict crisp boundaries. In *Eur. Conf. Comput. Vis.*, pages 562–578, 2018. 1, 2, 5, 7

[9] Ruoxi Deng, Shengjun Liu, Jinxin Wang, Huibing Wang, Hanli Zhao, and Xiaoqin Zhang. Learning to decode contextual information for efficient contour detection. In *ACM Int. Conf. Multimedia*, pages 4435–4443, 2021. 1, 2, 7

[10] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 2

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010. 5

[12] Chengyu Fang and Xianfeng Han. Flow-guided controllable line drawing generation. *arXiv preprint arXiv:2307.07540*, 2023. 2

[13] Yuanbin Fu and Xiaojie Guo. Practical edge detection via robust collaborative learning. In *ACM Int. Conf. Multimedia*, 2023. 1, 7

[14] Yi Gao, Chenwei Tang, Jiulin Lang, and Jiancheng Lv. End-to-end edge detection via improved transformer model. In *International Conference on Neural Information Processing*, pages 514–525. Springer, 2021. 2, 5, 7

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Adv. Neural Inform. Process. Syst.*, 27, 2014. 2

[16] Jianzhong He, Shiliang Zhang, Ming Yang, Yanhu Shan, and Tiejun Huang. Bi-directional cascade network for perceptual edge detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3828–3837, 2019. 2, 6, 7, 3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 3

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Adv. Neural Inform. Process. Syst.*, 33:6840–6851, 2020. 2

[19] Linxi Huan, Nan Xue, Xianwei Zheng, Wei He, Jianya Gong, and Gui-Song Xia. Unmixing convolutional features for crisp edge detection. *PAMI*, 44(10):6602–6609, 2021. 1, 2

[20] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Int. Conf. Comput. Vis.*, pages 1501–1510, 2017. 8

[21] Pavel Iakubovskii. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2019. 5

[22] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Focal frequency loss for image reconstruction and synthesis. In *Int. Conf. Comput. Vis.*, pages 13919–13929, 2021. 5

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[25] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *Adv. Neural Inform. Process. Syst.*, 28, 2015. 4

[26] Josef Kittler. On the accuracy of the sobel edge detector. *Image and Vision Computing*, 1(1):37–42, 1983. 2

[27] Simon Kohl, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R Ledsam, Klaus Maier-Hein, SM Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. A probabilistic u-net for segmentation of ambiguous images. In *Adv. Neural Inform. Process. Syst.*, 2018. 2, 3

[28] Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. *Int. Conf. Learn. Represent.*, 2016. 7

[29] Yunsung Lee, Jin-Young Kim, Hyojun Go, Myeongho Jeong, Shinhyeok Oh, and Seungtaek Choi. Multi-architecture multi-expert diffusion models. *arXiv preprint arXiv:2306.04990*, 2023. 4

[30] Mingchun Li, Dali Chen, and Shixin Liu. Beta network for boundary detection under nondeterministic labels. *Knowledge-Based Systems*, 266:110389, 2023. 1, 2, 7

[31] Hongyu Liu, Ziyu Wan, Wei Huang, Yibing Song, Xintong Han, and Jing Liao. Pd-gan: Probabilistic diverse gan for image inpainting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9371–9381, 2021. 2

[32] Rui Liu, Yixiao Ge, Ching Lam Choi, Xiaogang Wang, and Hongsheng Li. Divco: Diverse conditional image synthesis via contrastive generative adversarial network. In *IEEE*

Conf. Comput. Vis. Pattern Recog., pages 16377–16386, 2021. 2

[33] Yu Liu and Michael S Lew. Learning relaxed deep supervision for better edge detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 231–240, 2016. 2, 7

[34] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer convolutional features for edge detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3000–3009, 2017. 1, 2, 5, 6, 7, 3

[35] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. Convolutional oriented boundaries. In *Eur. Conf. Comput. Vis.*, pages 580–596. Springer, 2016. 7

[36] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1429–1437, 2019. 2

[37] David A Mély, Junkyung Kim, Mason McGill, Yuliang Guo, and Thomas Serre. A systematic comparison between visual cues for boundary detection. *Vision research*, 120:93–107, 2016. 1, 5, 7

[38] Miguel Monteiro, Loïc Le Folgoc, Daniel Coelho de Castro, Nick Pawlowski, Bernardo Marques, Konstantinos Kamnitsas, Mark van der Wilk, and Ben Glocker. Stochastic segmentation networks: Modelling spatially correlated aleatoric uncertainty. *Adv. Neural Inform. Process. Syst.*, 33:12756–12767, 2020. 3

[39] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 0–0, 2019. 1

[40] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[41] Mengyang Pu, Yaping Huang, Qingji Guan, and Haibin Ling. Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth. In *Int. Conf. Comput. Vis.*, pages 6879–6888, 2021. 2

[42] Mengyang Pu, Yaping Huang, Yuming Liu, Qingji Guan, and Haibin Ling. Edter: Edge detection with transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1402–1412, 2022. 1, 2, 5, 6, 7, 3

[43] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7479–7489, 2019. 1

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 5

[45] Aimon Rahman, Jeya Maria Jose Valanarasu, Ilker Hacihaliloglu, and Vishal M Patel. Ambiguous medical image segmentation using diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11536–11546, 2023. 2

[46] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3982–3991, 2015. 7

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[48] Xavier Soria, Yachuan Li, Mohammad Rouhani, and Angel D Sappa. Tiny and efficient model for the edge detection generalization. *arXiv preprint arXiv:2308.06468*, 2023. 1

[49] Zhuo Su, Wenzhe Liu, Zitong Yu, Dewen Hu, Qing Liao, Qi Tian, Matti Pietikäinen, and Li Liu. Pixel difference networks for efficient edge detection. In *Int. Conf. Comput. Vis.*, pages 5117–5127, 2021. 1, 2, 7

[50] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 4

[51] MM Valiuddin, Christiaan GA Viviers, Ruud JG van Sloun, Fons van der Sommen, et al. Improving aleatoric uncertainty quantification in multi-annotated medical image segmentation with normalizing flows. In *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Perinatal Imaging, Placental and Preterm Image Analysis*, pages 75–88. Springer, 2021. 2

[52] Yupei Wang, Xin Zhao, and Kaiqi Huang. Deep crisp boundaries. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3892–3900, 2017. 2, 6, 7, 3

[53] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C Cattin. Diffusion models for implicit image segmentation ensembles. In *International Conference on Medical Imaging with Deep Learning*, pages 1336–1348. PMLR, 2022. 2

[54] Yanze Wu, Xintao Wang, Yu Li, Honglun Zhang, Xun Zhao, and Ying Shan. Towards vivid and diverse image colorization with generative color prior. In *Int. Conf. Comput. Vis.*, pages 14377–14386, 2021. 2

[55] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Int. Conf. Comput. Vis.*, pages 1395–1403, 2015. 1, 2, 7

[56] Dan Xu, Wanli Ouyang, Xavier Alameda-Pineda, Elisa Ricci, Xiaogang Wang, and Nicu Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. In *Adv. Neural Inform. Process. Syst.*, pages 3961–3970, 2017. 7

[57] Xiaogang Xu, Ruixing Wang, and Jiangbo Lu. Low-light image enhancement via structure modeling and guidance. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9893–9903, 2023. 1

[58] Wenjie Xuan, Shaoli Huang, Juhua Liu, and Bo Du. Fcl-net: Towards accurate edge detection via fine-scale corrective learning. *Neural Networks*, 145:248–259, 2022. 7

[59] Wenjie Xuan, Shanshan Zhao, Yu Yao, Juhua Liu, Tongliang Liu, Yixin Chen, Bo Du, and Dacheng Tao. Pnt-edge: Towards robust edge detection with noisy labels by learning pixel-level noise transitions. In *ACM Int. Conf. Multimedia*, 2023. 2

[60] Jimei Yang, Brian Price, Scott Cohen, Honglak Lee, and Ming-Hsuan Yang. Object contour detection with a fully

convolutional encoder-decoder network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 193–202, 2016. 2

[61] Yunfan Ye, Renjiao Yi, Zhirui Gao, Zhiping Cai, and Kai Xu. Delving into crispness: Guided label refinement for crisp edge detection. *IEEE Trans. Image Process.*, 2023. 2, 8

[62] Zhiding Yu, Rui Huang, Wonmin Byeon, Sifei Liu, Guilin Liu, Thomas Breuel, Anima Anandkumar, and Jan Kautz. Coupled segmentation and edge learning via dynamic graph propagation. *Adv. Neural Inform. Process. Syst.*, 34:4919–4932, 2021. 1

[63] Zhiliang Zeng, Ying Kin Yu, and Kin Hong Wong. Adversarial network for edge detection. In *2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, pages 19–23. IEEE, 2018. 3

[64] Kilian Zepf, Eike Petersen, Jes Frellsen, and Aasa Feragen. That label's got style: Handling label style bias for uncertain image segmentation. In *Int. Conf. Learn. Represent.*, 2023. 2

[65] Caixia Zhou, Yaping Huang, Mengyang Pu, Qingji Guan, Li Huang, and Haibin Ling. The treasure beneath multiple annotations: An uncertainty-aware edge detector. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 15507–15517, 2023. 1, 2, 3, 4, 5, 6, 7

[66] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer, 2018. 4

[67] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *Adv. Neural Inform. Process. Syst.*, 30, 2017. 5

MuGE: Multiple Granularity Edge Detection

Supplementary Material

In this supplementary material, we display more results with different numbers of edge maps and qualitative comparisons of the predictions produced by UAED [65] and our proposed MuGE. We also provide additional visualization results on BSDS500 [1] and Multicue [37] datasets. Besides, since multiple annotations are available in BSDS500, we report the metrics for comparison with multiple ground truths for further evaluation.

A. Different intervals of the edge granularity

The choice of interval value is arbitrary, and we present the results with an interval of 0.1 in the main paper. In the supplementary, we further show the results with intervals of 0.05 in Fig. 8. Moreover, we also summarize the quantitative comparisons of different numbers of produced maps shown in Table 8. Obviously, a smaller interval typically generates more potential predictions, thus leading to better performance. It is worth noting that no matter how many predictions are generated, the model training is fixed, and we just need to feed different α to control the granularity of edge maps.

B. Qualitative comparisons of multiple Predictions.

Fig. 9 shows the qualitative comparisons of the multiple predictions between UAED and MuGE. From the visualized results, we can see that the diversity of UAED is quite limited, in contrast, our proposed MuGE presents a better diversity with different edge granularity, which can be beneficial for the downstream tasks by controlling the granularity of the edge maps.

C. More Visualization Results

In this section, we report more qualitative results on BSDS500 [1] and Multicue [37] datasets. More specifically, Fig. 10 shows the visual results compared with other approaches on BSDS500 [1] dataset, and more results of the proposed MuGE on BSDS500 [1] dataset are given in Fig. 11. Moreover, Fig. 12 depicts qualitative results for Multicue boundary [37].

Table 7. Comparisons on **BSDS500** [1] for multiple ground truth edge maps. All results are obtained under the MS-VOC setting.

Method	ODS	OIS	AP
EDTER [42]	0.727	0.769	0.775
UAED [65]	0.732	0.775	0.794
MuGE (Ours)	0.756	0.788	0.815

D. Results on multiple ground truth edge maps

In Table 2 of the main paper, we compare with previous works by evaluating the performance between the selected best-matching edge map and the single final fused ground truth for each test image. Since multiple ground truth maps are also provided in BSDS500 dataset, we further evaluate the performance on multiple ground truths, *i.e.*, we select the corresponding matched prediction from generated 11 predictions for each annotator's ground truth, and report the average performance. From Table 7, we can see that MuGE also outperforms the current SOTA UAED [65] by 2.4%, 1.3%, and 2.1%, and Transformer-based method EDTER [42] by 2.9%, 1.9%, and 4.0% in terms of the average ODS, OIS, and AP scores.

Table 8. Results on the **BSDS500** [1] testing set with different intervals of the edge granularity.

# Predictions	Edge Granularity	SS			MS			SS-VOC			MS-VOC		
		ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP	ODS	OIS	AP
$M=1$	$\alpha=0.6$	0.831	0.847	0.886	0.838	0.857	0.893	0.838	0.853	0.897	0.843	0.860	0.900
$M=2$	$\alpha\in\{0,1\}$	0.835	0.849	0.888	0.845	0.859	0.897	0.847	0.857	0.901	0.853	0.864	0.905
$M=3$	$\alpha\in\{0,0.5,1\}$	0.845	0.854	0.895	0.853	0.863	0.901	0.852	0.859	0.904	0.858	0.865	0.907
$M=6$	$\alpha\in\{0,0.2,\ldots,0.8,1\}$	0.849	0.855	0.896	0.856	0.864	0.902	0.854	0.860	0.905	0.860	0.867	0.909
$M=11$	$\alpha\in\{0,0.1,\ldots,0.9,1\}$	0.850	0.856	0.896	0.858	0.864	0.902	0.855	0.860	0.905	0.861	0.867	0.909
$M=21$	$\alpha\in\{0,0.05,\ldots,0.95,1\}$	0.851	0.857	0.897	0.858	0.864	0.903	0.856	0.861	0.905	0.862	0.867	0.909

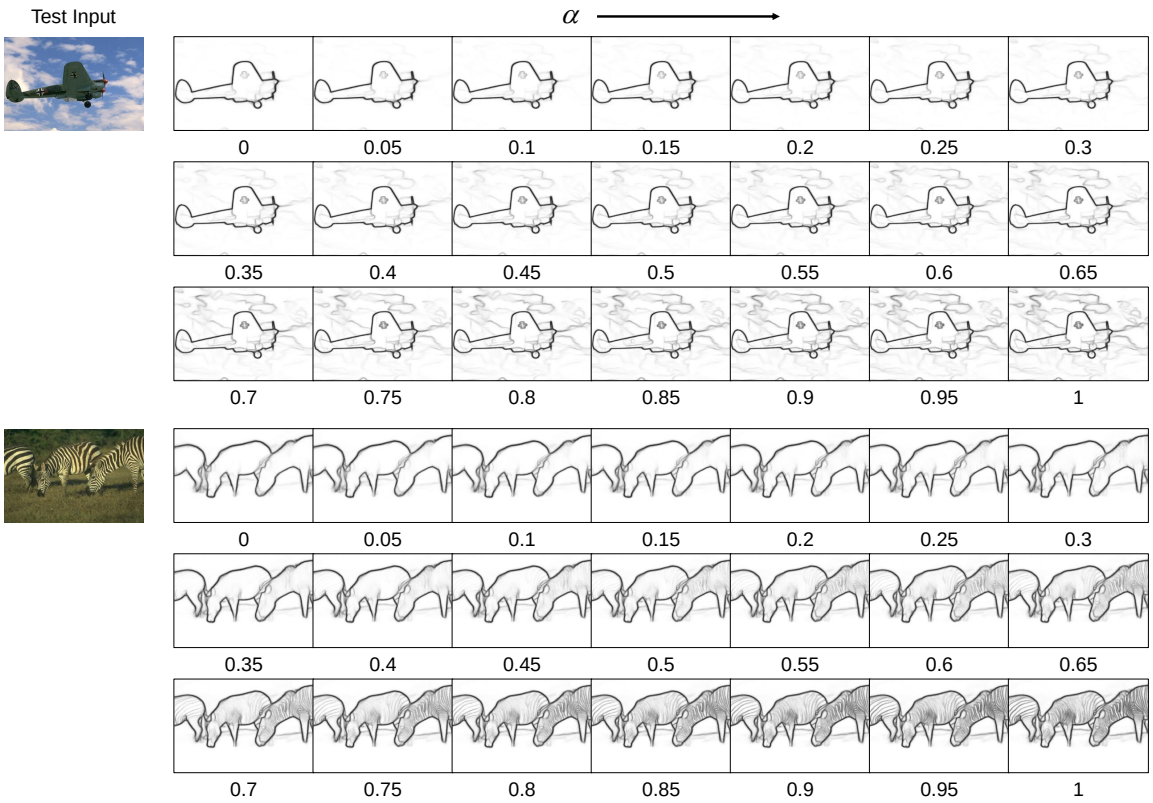

Figure 8. The generated diverse edge predictions for the given test image with different edge granularity α, where α ranges from 0 to 1 in intervals of 0.05 under the MS-VOC setting.

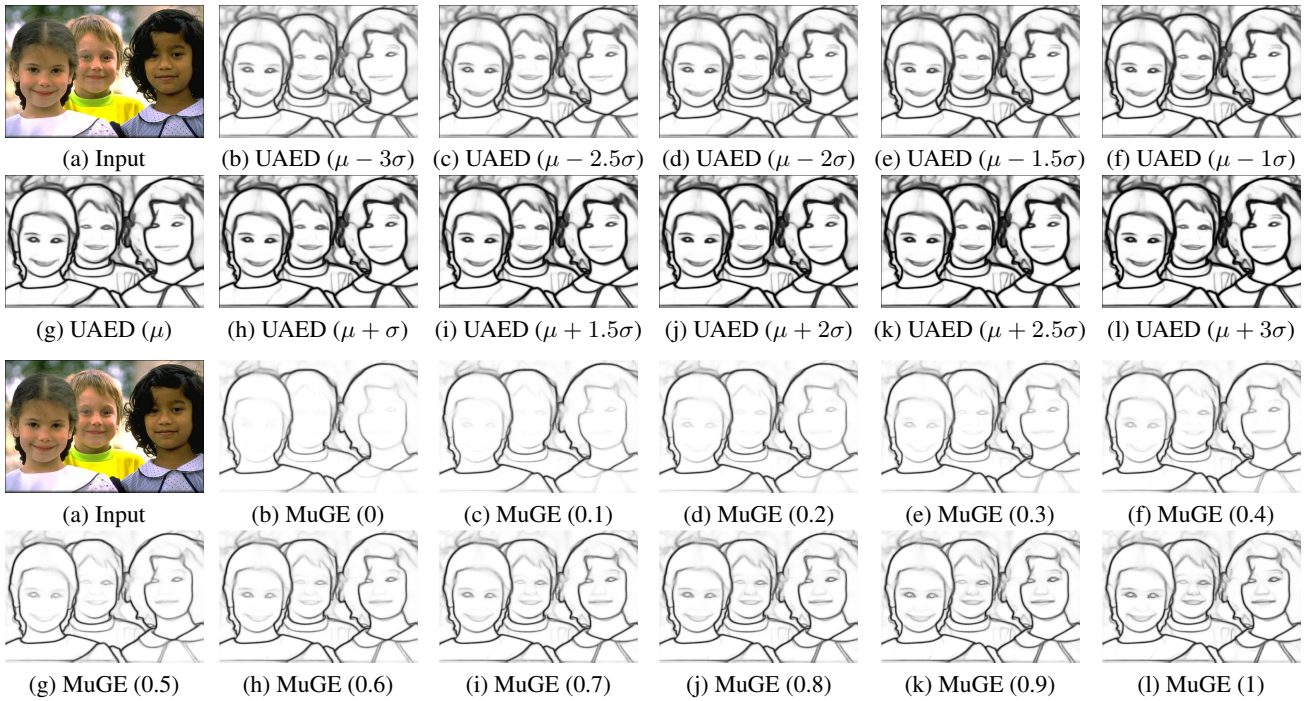

Figure 9. Qualitative diversity comparisons of UAED and our proposed MuGE in the BSDS500 test set under the MS-VOC setting.

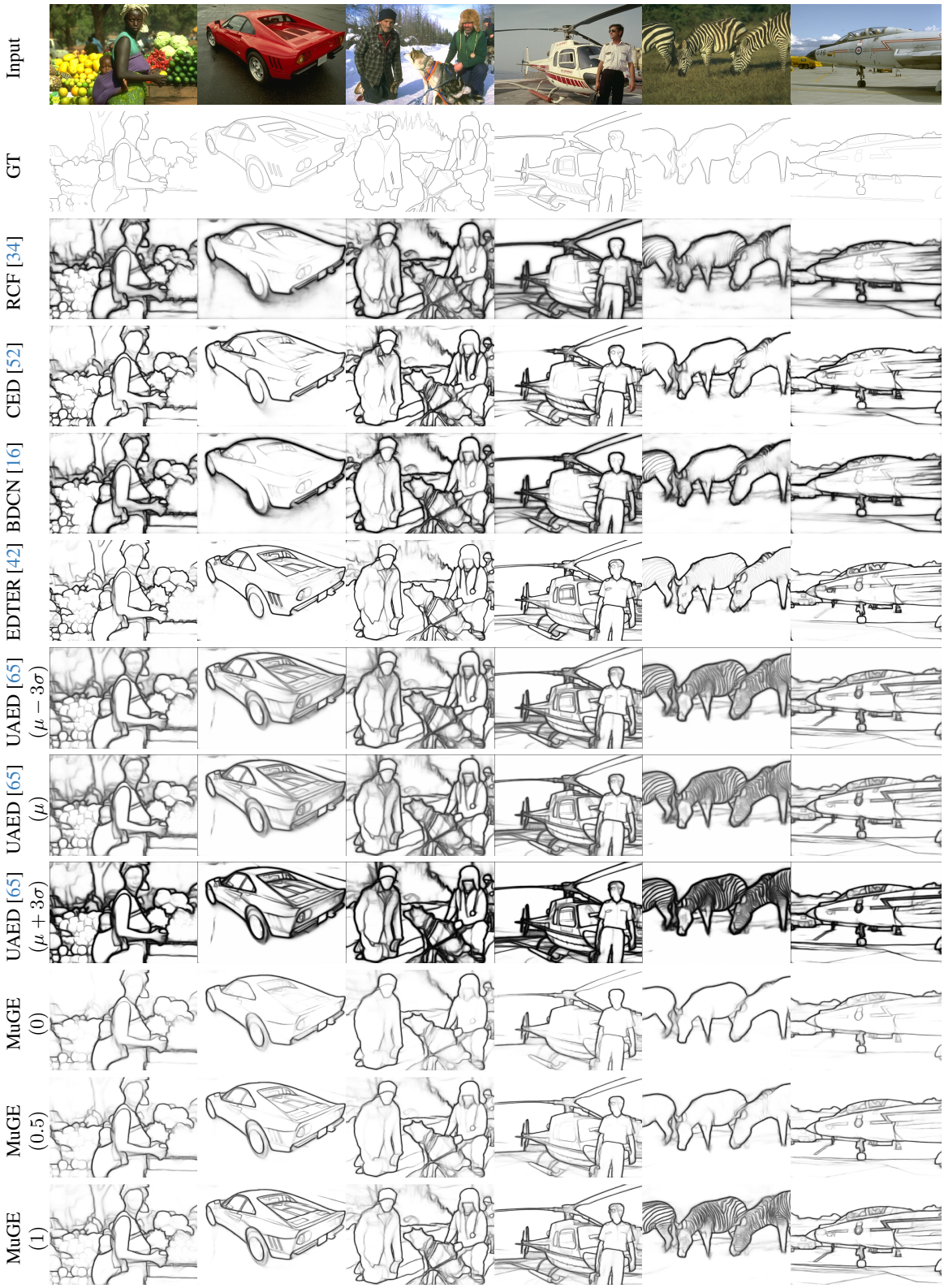

Figure 10. Qualitative comparisons on challenging samples in the BSDS500 test set under the MS setting. Note that UAED samples from the learned distribution with $\mu - 3\sigma$, μ, and $\mu + 3\sigma$, respectively, and MuGE produces diverse results with edge granularity of 0, 0.5, and 1, respectively.

(a) Image (b) Ground truth (c) MuGE (0) (d) MuGE (0.5) (e) MuGE (1)

Figure 11. Qualitative comparisons on the testing set of BSDS500 under the MS-VOC setting.

(a) Input (b) GT-Boundary (c) UAED-Boundary (d) MuGE-Boundary (0) (e) MuGE-Boundary (1)

Figure 12. Qualitative results with different edge granularity on the Multicue Boundary.