Learning to Decode Contextual Information for Efficient Contour Detection

Ruoxi Deng
Wenzhou University
ruoxii.deng@gmail.com

Shengjun Liu
Central South University
shjliu.cg@csu.edu.cn

Jinxin Wang
Wenzhou University
jxwang@stu.wzu.edu.cn

Huibing Wang
Dalian Maritime University
huibing.wang@dlmu.edu.cn

Hanli Zhao
Wenzhou University
hanlizhao@wzu.edu.cn

Xiaoqin Zhang*
Wenzhou University
xqzhang@wzu.edu.cn

ABSTRACT

Contour detection plays an important role in both academic research and real-world applications. As the basic building block of many applications, its accuracy and efficiency highly influence the subsequent stages. In this work, we propose a novel lightweight system for contour detection that achieves state-of-the-art performance while keeps ultra-slim model size. The proposed method is built on an efficient encoder in a bottom-up/top-down fashion. Specially, we propose a novel decoder which compresses side features from an encoder and effectively decodes compact contextual information for high-accurate boundary localization. Besides, we propose a novel loss function that is able to assist a model to produce crisp object boundaries.

We conduct extensive experiments to demonstrate the effectiveness of the proposed system on the widely adopted benchmarks BSDS500 and Multi-Cue. The results show that our system achieves the same best performance , yet only consumes **3.3%** computational cost (**16.45GFlops** VS. **499.15GFlops**) and **2.35%** model size (**1.94M** VS. **82.43M**) of the SOTA detector *RCF-ResNet101*. In the meantime, our method outperforms a large portion of the recent top edge detectors by a clear margin.

CCS CONCEPTS

• **Computing methodologies** → *Scene understanding*; **Image segmentation**.

KEYWORDS

contour detection, edge detection, compact neural networks, lightweight contour detection

ACM Reference Format:
Ruoxi Deng, Shengjun Liu, Jinxin Wang, Huibing Wang, Hanli Zhao, and Xiaoqin Zhang*. 2021. Learning to Decode Contextual Information for Efficient Contour Detection. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475593

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475593

1 INTRODUCTION

Contour detection aims to precisely localize object boundaries in images and videos. Since it can produce important visual cues like geometry shape and semantic boundary information, it is often employed as the basic block of multimedia applications such as icon generation [34] in banners and homepages, object colorization [14], object segmentation and tracking [20]. Recently, contour detection methods equipped with deep learning techniques [6, 7, 23, 36, 37] achieved state-of-the-art performances on the datasets such as BSDS500, which demonstrates the effectiveness of deep convolutional neural networks (DCNN).

However, most of the recent works for contour detection seem to only focus on chasing the best performances on benchmarks, yet lack of exploration on system efficiency. While the trend may be disadvantaged to the spread of DCNN-based edge detectors, since the majority of edge devices can not satisfy high computational requirements. In the meantime, simply duplicating efficient models from related studies like efficient semantic segmentation maybe suboptimal to contour detection. Therefore, designing efficient system is of great importance for the development of contour detection study.

In this work, we present a novel compact model for both accurate and efficient object contour detection. Making system computational efficient indicates reducing a large number of weights, which would cause severe performance degradation. To avoid such issue, we propose three effective ways to build a high-performance system. Firstly, we discuss the contributions of different basic network structure to system performance; Secondly, we build an effective decoder to extract contextual information from side features that significantly advancing system performance;

Unlike common light-weight methods relying on transferring dark knowledge from computational expensive model to achieve good performance [11], our compact model needs no knowledge distillation and still outperforms a large portion of the recent top edge detectors while keeping ultra-slim model size. To sum up, our main contributions are as follow

1) We propose a novel compact model dedicating to efficient contour detection, which effectively leverages multi-scale, multi-level features to produce high-quality object contours.
2) We present an effective decoder net to significantly improve model performance; besides, we propose a novel loss function that further advance the system.

* Corresponding author

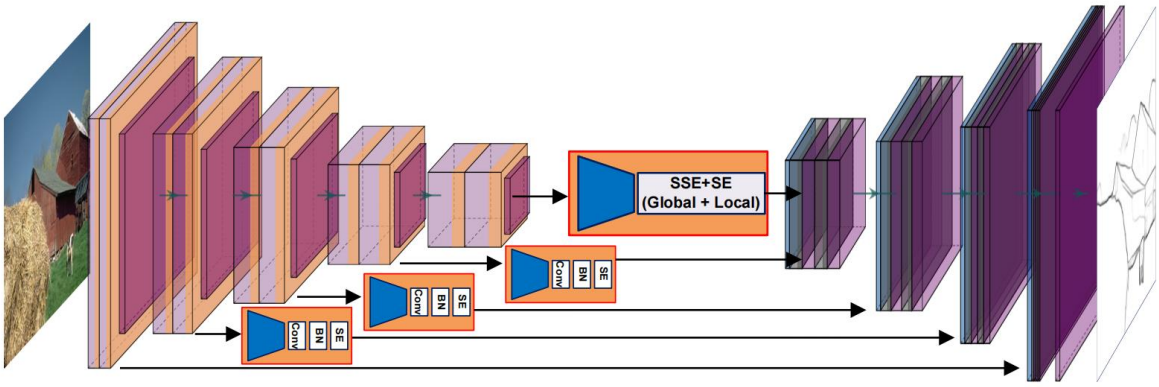


Figure 1: Illustration for the overall structure of the proposed method. our compact model is a convolutional encoder-decoder network. We adopt existing efficient networks such as SqueezeNet, MobileNetV2, and RegNetX, to be the encoders that provides reliable multi-scale and multi-level visual cues. Specially, the model armed with the proposed hyper modules can effectively enhance their representation power by exploring compact global and local contextual information. The details are described in Sec. 3.3

3) Our compact model only takes up 1.94M weights and cost 16.45G MAC operations for a 480×320 image, however, achieves state-of-the-art performances on BSDS500 and Multi-Cue datasets without knowledge distillation. To our best, the proposed method is the first efficient system to perform SOTA against its top competitors.

2 RELATED WORK

The history of edge detection study can trace back forty years ago [17]. Early edge detectors such as the Sobel [17] and the Canny detectors [4] rely on calculating image gradients to produce edges [9, 27]. Although the unsupervised methods suffer from drawbacks of noisy pixels and lack of semantic contours, they are still widely used in the tasks like image segmentation [31], object feature extraction [24] and human pose estimation [33]. As time goes on, learning-based methods [1, 8**?** **?**] became popular. This kind of methods make use of different low-level features and train a classifier to generate object-level contours. Although these methods achieve significant progress compared to traditional methods, they still rely on hand-craft features, which limits their room for improvements.

Recently, SOTA edge detectors [3, 18, 23, 32, 37] are mainly built on deep convolutional neural networks [19**?**]. The methods achieve good performances and some of them even outperform humans on the BSDS500 dataset. HED [37] is an end-to-end fully convolutional neural network, which takes an image as input and directly outputs the prediction. It is able to produce accurate predictions by presenting a weighted cross-entropy loss and a skip-layer structure. RCF [23] improves the skip-layer structure of HED by making independent predictions from each convolutional layer of the VGG model and achieves better results. CED [36] applies a novel decoder structure in the encoder-decoder network to generate crisp object boundaries. LPCB [7] explains why CNNs tend to produce blurry edges and proposes a solution to make CNNs directly predict crisp boundaries without post-processing. DSCD [6] proposes a novel loss function and three different scales of dilated convolution blocks to increase the ability of high-level feature extraction. BANet [10] presents a novel Bidirectional encoder-decoder network for accurate boundary detection.

3 THE PROPOSED METHOD

3.1 Problem Formulation

Assuming C is a computational model to extract object contours from images or videos. The input of the model is an RGB image $I \in \mathbb{R}^{H \times W \times 3}$, the output is a mask $M \in \mathbb{R}^{H \times W}$. Each pixel in the mask has the range of $[0, 1]$ representing the probability of being an edge. For any given resources constraint, the objective is to maximum the prediction quality of model C, which can be formulated as an optimization problem [35]

$$\begin{aligned} \max \quad & Quality(C) \\ s.t. \quad & \textbf{Memory}(C) \leq target\_memory \\ & \textbf{FLOPs}(C) \leq target\_flops \end{aligned} \tag{1}$$

The quality of prediction is in two-folds: the localization accuracy and the sharpness of boundary shape[7]. For current state-of-the-art edge detection system [23, 36, 37], the model C can be representated as a convolutional encoder-decoder network, which composes of a base network $\mathcal{E}$ to produce multi-scale and multi-level side features, and a decoder network $\mathcal{D}$ to fuse the features and reconstruct the mask M.

Given the objective 1, the aim of our work is to develop a contour detection system which can produce both crisp and accurate object boundaries. In the meantime, the system should be as efficient as possible with slim model size, which allows it to be deployed in broad computing platforms. Since the system is built from an encoder E and a decoder D, the overall resources constraint can also be decomposed into $\text{Memory}(C) = \text{Memory}(\mathcal{E}) + \text{Memory}(\mathcal{D})$, $\text{FLOPs}(C) = \text{FLOPs}(\mathcal{E}) + \text{FLOPs}(\mathcal{D})$. Therefore, we can separately optimize the performance, memory consumption and computational cost of $\mathcal{E}$ and $\mathcal{D}$ in order to solve the objective 1. In the next

subsections, we introduce the details of encoder, the proposed novel and effective decoder and a new loss function for contour detection. Fig. 1 illustrates the overall structure of our detector.

3.2 Encoder

An encoder is a fundamental component and the core of a contour detection system. It provides multi-scale and multi-level object features that determine the overall performance. In this work, we do not design a base network. Instead, we test our system on three different light-weight encoders, i.e., SqueezeNet[15], MobileNetV2 [30] , and the recent RegNetX [29], By doing so, the advantages are in two aspects: 1. these light-weight networks have been proven to be effective in both academic research and real applications. Their efficiency and computational cost are carefully optimized. The quality of generated features are trustworthy; 2. it allows the proposed method to demonstrate the effectiveness on different lightweight networks. Note that, SqueezeNet has only four convolutional stages while MobileNetV2 and RegNetX have five. Thus the refinement stages of our decoder will vary according to the encoder.

3.3 Decoder

A decoder is another core component in an edge detector. It aims to gradually refine and fuse side features, then produces the mask. A normal decoder shares the same channel number with side features of an encoder, as shown in SegNet [2] or SharpMask [28]. It can keeps the representation power of the side features, however, would result in a large model size and heavy computational cost if we followed such design in a light-weight decoder. For light-weight edge detector, reduce side features is inevitable, especially when the channels of side features are huge.

There are two major trends to squeeze side information in current light-weight models. The first one is to keep all the refinement stages yet only reduce the feature channels. In this way, a majority of channels would be squeezed, e.g., RCF [23] squeezes each VGG16 side features to 21 channels and HED [37] reduce the number to only one channel. The method made the decoders of the detectors ultra-efficient. The other way is to reduce the refinement stages. DeepLabV3+ [5] select this way and reduce the refinement stages from four to two, which allows the model to remove a large number of weights. In this work, we select the four-refinements structure as the topology of the proposed decoder, and compare the second way as one of baseline models. Four refinements allows an edge detector to fully explore the potential of encoder features with different scales and different levels. To sum up, assuming an encoder $\mathcal{E}$ with five convolutional stages produces the side features $\{\mathbf{f}_i | i = 1, ..., 5\}$, where $\mathbf{f}_i \in \mathbb{R}^{H^i \times W^i \times N^i}$, H^i, W^i, N^i represent different height, width, channel number of a side feature at different stage index i. These features are taken from the location before the downsampling layers, e.g., the max-pooling layers or the $stride = 2$ convolutional downsampling layers. Note that, a majority of recent encoders tend to double reduce the feature resolution from the first convolutional layer, which is disadvantaged for the accuracy of the final mask. Thus we revise the stride of the first convolutional layer in the encoders from two to one. Given a side feature $\mathbf{f}^i$ and a mask-encoding $\mathbf{m}^i \in \mathbb{R}^{H^i \times W^i \times N^i}$, a refinment module R^i is going to fuse $\mathbf{f}^i$, $\mathbf{m}^i$ and upsample the fusion result, which is given by

$$\begin{aligned} \mathbf{m}^{i-1} &= R^i(\mathbf{f}^i, \mathbf{m}^i) \\ &= U^i(F^i(\mathbf{f}^i, \mathbf{m}^i)). \end{aligned} \tag{2}$$

F^i, U^i are the fusion and upsample function, respectively. F^i performs the mapping $\mathbb{R}^{H^i \times W^i \times 2N^i} \rightarrow \mathbb{R}^{H^i \times W^i \times N^{i+1}}$ and U^i employs the transformation $\mathbb{R}^{H^i \times W^i \times N^{i+1}} \rightarrow \mathbb{R}^{2H^i \times 2W^i \times N^{i+1}}$. Since the refinement process is in a top-down manner, the index i upon a mask-encoding is inversed. Eq.2 represents the operation that each refinement module functions in a common decoder.

However, in the proposed light-weight decoder, we expand the inner functions of a refinement module from two to four, i.e., the squeeze function S^i, the feature extraction function E^i, the fusion function and the upsampling function, which are

$$\begin{aligned} \mathbf{m}^{i-1} &= R^i(\mathbf{f}^i, \mathbf{m}^i) \\ &= U^i(F^i(E^i(S^i(\mathbf{f}^i, \mathbf{m}^i)))). \end{aligned} \tag{3}$$

Compared to Eq. 2, our method augments the standard processing with information squeezing and feature abstraction, which aims to simultaneously compress the decoder size and enhance the representation ability of a lightweight model. Given the Eq. 3, the main framework is clearly shown and we can introduce the details of each component and discuss their contribution.

3.3.1 Feature compressing. The first component is the function S^i that aims to effectively compress the side information. S^i only compress the feature channels to a fixed number k, where $N^5 >> k$. N^5 denotes the channel number of the output of the last convolutional stage. All the side features will be squeezed to the shape of (H^i, W^i, k), respectively. The choice of k is important since it determines the compressing ratio. We set it as

$$k = \alpha N^1. \tag{4}$$

α is the scaling factor and N^1 is the channel number of the output of the first convolutional layer or stage. In experiments, we set $\alpha = 1$, which balances the model size and the performance. We use 1×1 *conv.* layer as the function S^i to perform the compressing with no activation. We show an ablation study on the choice of α in the experimental section.

3.3.2 Enhancing performance by exploring compact global and local information on features. Directly squeezing feature channels is advantaged for reducing the decoder size, however, has the risk of losing too much information.

To conquer the issue, we propose a novel method named Spatially Squeeze-and-extraction (SSE) module, which intends to explore global contextual information and calibrates high-level features in spatial dimensions. Given a feature $\mathbf{f} \in \mathbb{R}^{H \times W \times K}$, where H, W, K represent the height, width and channel number, we first employ the transfomation $U : \mathbb{R}^{H \times W \times K} \rightarrow \mathbb{R}^{L \times L \times K}$ that bilinearly interpolates the height and width of the feature to the same size L. L is a fixed number. Since latter the module will equip with linear layer, this operation allows arbitrary size of input to feed in our detector. After resize the feature, we extract the attention map $\mathbf{a} \in \mathbb{R}^{L \times L}$ of $\mathbf{f}$, which is

$$\mathbf{a} = \frac{\sum_{k=0}^{K-1} \mathbf{f}(k)}{K}. \tag{5}$$

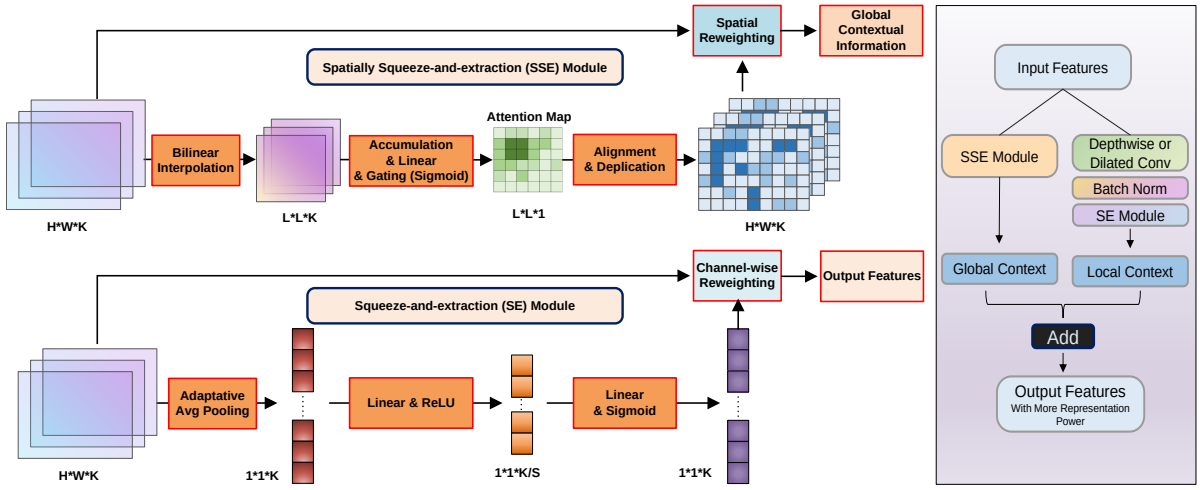


Figure 2: Illustration for extracting compact contextual information. In this work, a novel Spatially Squeeze-and-extraction (SSE) module is proposed to explore global contextual information and calibrate high-level features in spatial dimensions.

Eq. 5 denotes a feature accumulates itself across the channel dimension. Each pixel in the attention map can represent a group of pixels with the shape of $1 \times 1 \times K$ in the original feature.

We then build the global relationship by capturing mutual dependencies among pixels in the feature $\mathbf{a}$. To achieve the goal, we opt to employ a linear transformation followed by a simple gating mechanism

$$\mathbf{s} = \sigma(\mathbf{W}\mathbf{a}), \tag{6}$$

where $\mathbf{W} \in \mathbb{R}^{L \times L}$ and σ refers to the sigmoid function. The result $\mathbf{s}$ has the shape of $L \times L$, which is going to re-weight the feature $\mathbf{f}$. But before that, we have to align its shape with the feature. By using bilinear upsampling again, we first resize the attention map $\mathbf{s}$ from shape (L, L) to (H, W), then duplicate it K times to form a new tensor $\hat{\mathbf{s}}$ with the shape of (H, W, K). Finally, we apply dot-production between the tensor and the feature

$$\hat{\mathbf{f}_g} = \hat{\mathbf{s}}\mathbf{f}, \tag{7}$$

where $\hat{\mathbf{f}}$ is the calibrated feature.

The above approach is the global context branch of our SSE module. It differs from SE method [13] in the way of computing dependencies. SE computes the channel-wise dependencies which squeezes spatial information into a simple scala, yet our method compresses the whole channel information into a scala and calculates the spatial dependencies among pixels.

The local context branch of our SSE module aims to enhance the representation power of a side feature by probing its local context. Our method is given by

$$\hat{\mathbf{f}_l} = SE(BN(DC(\mathbf{f}))), \tag{8}$$

where SE refers to the squeeze-and-extraction [13], BN refers to batch normalization [16] and DC refers to depthwise convolution [12]. Specially, DC is also dilated convolution. We find that enlarging the receptive field of a layer can boost the performance.

Finally, the output of the SSE module is the sum of the global contextual feature and the local contextual feature, which is given by

$$\hat{\mathbf{f}} = \hat{\mathbf{f}_g} + \hat{\mathbf{f}_l}. \tag{9}$$

To sum up, given a compressed side feature $\mathbf{f}$, the proposed SSE module explores its contextual information and generates a new feature $\hat{\mathbf{f}}$ that shares more representation power. The feature $\hat{\mathbf{f}}$ would replace the original $\mathbf{f}$ to participate in the refinement module. Fig. 2 depicts the details of the SSE module.

3.3.3 Fusion. In each refinement process, We apply 1×1 convolution on both side feature $\hat{\mathbf{f}}$ and mask-encoding $\mathbf{m}$, which is given by

$$F(\hat{\mathbf{f}}, \mathbf{m}) = ReLU(BN(Conv_{1\times1}(\hat{\mathbf{f}}) + BN(Conv_{1\times1}(\mathbf{m}))), \tag{10}$$

where ReLU refers to the rectified linear unit[26]. Eq. 10 produces a new mask-encoding for the next refinement.

3.3.4 Upsampling. At the last of most refinements, we adopt learned pointwise transposed convolution to upsample the new mask-encoding. We found that using transposed convolution, the model performs better than employing bilinear or nearest upsampling. We do not utilize upsampling at the last refinement. The output of the decoder will go through a convolution head to generate the final mask which is the result of our detector.

We emphasize that the cause of the high performance of our light-weight decoder is the way that we explore global and local contextual information described in Sec. 3.3.2. Besides, four refinements structure, instead of two, allows our method fully accesses all levels of side features, which is another reason for improvements.

3.4 Optimization

Loss function for contour detection highly influences both the localization accuracy and the boundary sharpness [7, 37] of a model. In this work, we propose a novel loss function to additionally improve model performance.

Contour detection is a task that deals with data imbalanced issues. In a contour ground-truth, a majority of pixels are background.

Therefore the loss function oughts to be carefully designed to handle the issue as well as pursuit high performance. We follow the thought of [7] to make constraint on both pixel-level distance and image-level similarity, yet pay more attention to hard examples. The first term of our loss function is given by

$$\mathcal{L}_p = -\sum_{i}^{N}((1-\mathbf{p}_i)^{\gamma}\mathbf{g}_i\log(\mathbf{p}_i) + \mathbf{p}_i^{\gamma}(1-\mathbf{g}_i)\log(1-\mathbf{p}_i)), \quad (11)$$

where $\mathbf{p} \in \mathbb{R}^{H\times W}, \mathbf{g} \in \mathbb{R}^{H\times W}$ refer to the predction and the corresponding label, respectively. i refers to the index of each pixel. This term calculates and accumulates the pixel-level distances between prediction and groundtruth. $(1-\mathbf{p}_i)^{\gamma}$ is a modulating factor and γ is a tunable focusing parameter which is motivated by the focal loss [21]. A contour detector often encounters many hard examples that an edge pixel is predicted with very low confidence. The factor $(1-\mathbf{p}_i)^{\gamma}$ can effectively penalize this situation and forces the model to optimize the prediction. In experiments, we set the focal parameter $\gamma = 2$. The remaining factors in Eq. 11 is the cross-entropy loss. Unlike HED and RCF using a balancing weight on the terms, we find that only employing the focal parameter on the cross-entropy loss with the global similarity constraint works better.

The second term is the Dice loss [7] which computes the similarity distance of two binary maps. The term is given by

$$\mathcal{L}_g = \frac{\sum_{i=1}^{N}\mathbf{p}_i^2 + \sum_{i=1}^{N}\mathbf{g}_i^2}{2\sum_{i=1}^{N}\mathbf{p}_i\mathbf{g}_i}. \quad (12)$$

The term implicitly solves the data imbalanced issue by penalizing the global differences at image similarity. Owing to the Dice loss, our prediction show crisp object boundaries. Finally, the overall loss function is given by

$$\mathcal{L} = \mathcal{L}_g + \alpha\mathcal{L}_p, \quad (13)$$

where $\alpha = 0.01$ controls the influence of the local constraint. The proposed loss function effectively solves the data distribution issue and forces the detector to produce crisp boundaries. Besides, introducing the focal factor further optimizes the model performance. We conduct a comparative experiment on the balanced cross-entropy loss [37], the Dice loss [7] and the proposed loss function in the next section. The results demonstrate the effectiveness of our method.

4 EXPERIMENTS

In this section, we conduct extensive experiments on the widely used BSDS500 dataset and Multi-Cue dataset. We first introduce the basic setting of experiments including hyper-parameters, data augmentation, and evaluation metrics. We then report two ablation studies on the model structure and backbone type. Thirdly, we compare our lightweight model with current SOTA edge detectors.

4.1 Training hyper-parameters

We train our network using two Nvidia T4 cards. The training hyper-parameters of our model contains: batchsize (20), training epochs (21), learning rate (1e-4), weight decay (4e-5). Data augmentation is performed by randomly upscaling or downscaling images and labels, rotating them to 12 different angles, cropping the least square regions and resizing all the samples to a fixed size. Specially, we follow the suggestion of [7] by using all the annotations in training.

Table 1: Ablation study on compressing ratio. MobileNetV2 is adopted for the test. Channels refer to the input channels for a decoder. Size refers to the amount of model weights. ODS here denotes the ODS F-score. The experiment is conducted on BSDS500.

Method	Channels	Ratio	Size	ODS
TW	[96,192,960]	\	1.86M	.779
FRK	[96,144,192,576,960]	\	2.65M ↑	.792 ↑
FRC-96	[96,96,96,96,96]	1	1.77M ↓	.790 ↓
FRC-16	[16,16,16,16,16]	1/6	1.55M ↓	.780 ↓

In two datasets, each image is annotated by five people, thus we build the one-to-five relationship by creating five different image-label pairs for data augmentation. This is an effective way to enlarge the training samples.

4.2 Evaluation metrics

We follow the prior works [6, 7, 23, 36, 37] to use F-score as the evaluation metric. The F-score (also known as F-measure, F1-score) can be obtained by $2\times P\times R/(P+R)$ where $P = TP/(TP+FP)$, P, TP, FP denotes the Precision, True Positive and False Positive, respectively; $R = TP/(TP+FN)$, R and FN denotes the Recall and False Negative. Note that there are three kinds of F-score used in the evaluation, i.e., Optimal Dataset Scale (ODS), Optimal Image Scale (OIS), and Average Precision (AP). All the metrics are the higher, the better.

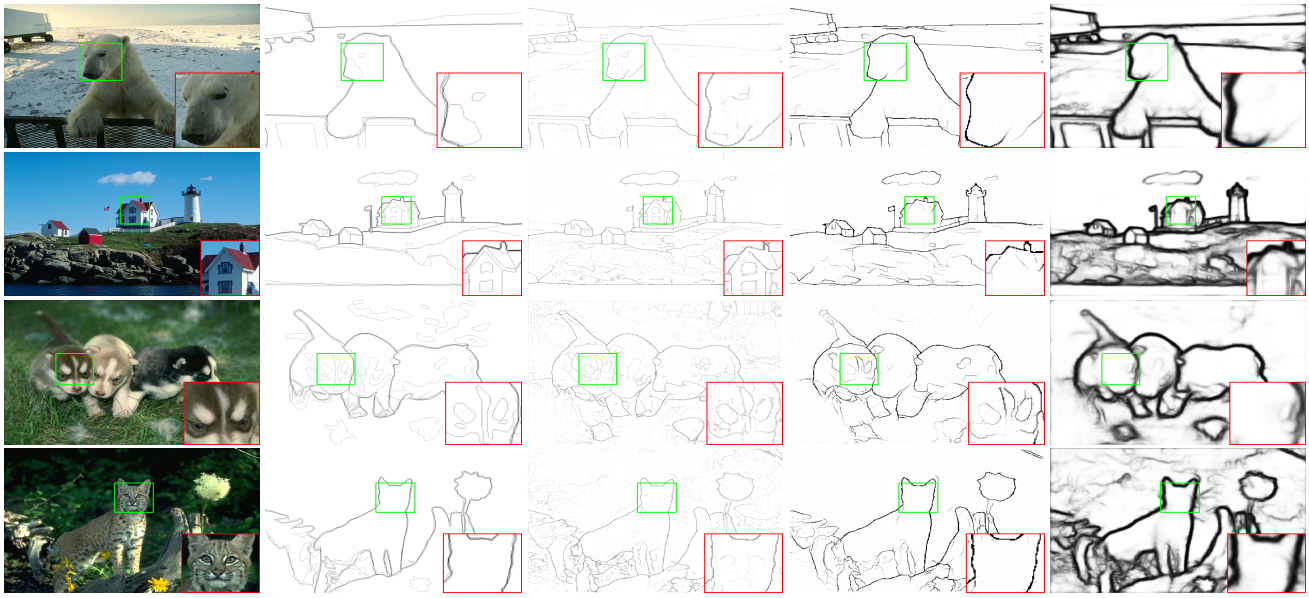
Table 2: Ablation study on each proposed model component. MS refers to the model size. CC refers to the computational cost (The input resolution is 480×320). FRC-96 refers to the base detector that is also the third baseline in Tab. 1. Computational cost refers to the sum of multiply–accumulate operations. The experiment is conducted on BSDS500.

Baselines	MS	CC	ODS	OIS	AP
FRC-96	1.77M	16.35G	.790	.808	.810
+Local Branch	1.78	16.43G	.794 ↑	.810 ↑	.823 ↑
+Global Branch	1.94M	16.45G	.797↑	.812↑	.826 ↑
+Mixed Loss	1.94M	16.45G	.799 ↑	.816 ↑	.837 ↑

4.3 Ablation study

We perform three ablation studies to verify the compressing ratio α mentioned in Sec. 3.3.1, the effectiveness of each component in our model, and the influence of different lightweight backbone. All the experiments are conducted on BSDS500.

4.3.1 Ablation study on compressing ratio. We adopt MobileNetV2 as the backbone and extract its side features to test the relation between compressing ratio and the performance. The channel numbers of the extracted features are [96,144,192,576,960]. We use a naive decoder that directly upsamples and fuses side features and mask-encodings. The naive decoder adopts transposed convolution to performs upsampling. We propose four baselines, i.e., two refinments structure (TW), four refinements keeping feature channels



(p) Input Image (q) The Groundtruth (r) Ours (s) RCF (t) CED

Figure 3: State-of-the-art comparisons on BSDS500. We can clearly observe that our method is good at producing crisp boundaries and keep finer object details.

Table 3: Ablation study on different backbones. All the models are trained with the proposed loss. These results have clearly demonstrated the effectiveness of the proposed light-weight system.

Methods	MS	CC	ODS	OIS	AP
SqueezeNet-Naive	6.49M	18.71G	.767	.786	.817
SqueezeNet-Proposed	1.62M ↓	11.84G ↓	**.775** ↑	**.790** ↑	**.820** ↑
MobileNetV2-Naive	2.65M	24.91G	.792	.809	.828
MobileNetV2-Proposed	1.94M ↓	16.45G ↓	**.799** ↑	**.816** ↑	**.837** ↑
RegNetX(400M)-Naive	7.73M	11.48G	.793	**.809**	.823
RegNetX(400M)-Proposed	4.97M ↓	10.95G ↓	**.794** ↑	.808↓	**.828** ↑

unchanged (FRK), four refinements compressing feature channels to 96 ($\alpha = 1$, FRC-96), four refinements compressing feature channels to 16 ($\alpha = 1/6$, FRC-16). The results are listed in Tab 1.

Although two refinements structure shows a reasonable performance ODS F-score .779 and slim model size, without any assistant method to improve performance, the performance of this kind of light-weight structure falls far behind existing CNN-based detectors. FRK model is improved by a large margin and achieved ODS F-score .789, however, due to a large number of side features, it also has the largest model size. FRC-96 shows a slightly worse performance (ODS F-score .790) and much smaller model size. By first compressing size features to a fixed size (96 channels), the model reduce large weights of the decoder, and still achieves good performance. In contrast, although FRC-16 has the smallest size, its performance decreases severely by losing too much side feature information. From the results, we find that compressing too much low-level information maybe harmful to the localization accuracy of a low-level task. Considering all the factors, we choose the compressing ratio $\alpha = 1$ for our light-weight methods.

Table 4: Results on the BSDS500 dataset. S refers to the multi-scale testing. VOC-aug refers to training with extra PASCAL VOC context data.

Method	MS	CC	ODS	OIS	AP
HED [37]	14.71M	131.58G	.788	.808	.840
CED [36]	21.4M	138.8G	.794	.811	.847
CED-S [36]			.803	.820	.871
CED-S-VOC-aug [36]			.815	.833	.889
LPCB	17.01M	121.50G	.800	.816	
LPCB-VOC-aug			.808	.824	
LPCB-S-VOC-aug			.815	.834	.827
DSCD [6]	34.07M	135.30G	.802	.817	.826
DSCD-VOC-aug			.813	.836	.847
DSCD-S-VOC-aug			**.822**	**.859**	**.863**
RCF-VGG16-aug [23]	14.8M	119.9G	.806	.823	
RCF-VGG16-aug-s [23]			.811	.830	.846
RCF-ResNet50-aug	40.88M	299.73G	.808	.825	
RCF-ResNet50-aug-S			.814	.833	.849
RCF-ResNet101-aug	82.43M	499.15G	.812	.829	
RCF-ResNet101-aug-S			.819	.838	.847
Ours-MobileNetV2	**1.94M**	**16.45G**	.799	.816	.837
Ours-VOC-aug			.812	.826	.857
Ours-S-VOC-aug			.819	.834	.860

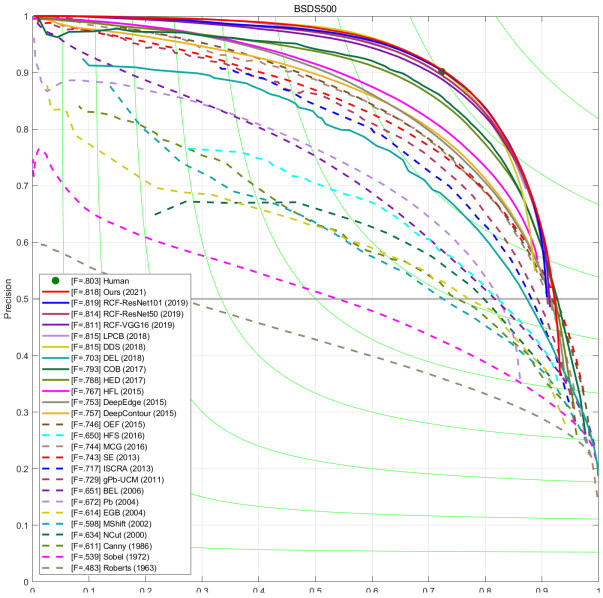


Figure 4: State-of-the-art comparisons on the BSDS500 dataset. Note that, our method is the only lightweight method while the rest of top detectors are computational expensive. However, our performance outperforms most of the methods.

4.3.2 Ablation study on model components. The second ablation study is to examine the effectiveness of each proposed component. We still use the MobileNetV2 as the base net and extract side features from the same location with the last experiment. Our first baseline is the FRC-96; the second baseline is the FRC-96 equipped with the local context branch (Eq. 8), which puts the dilated *Conv.*, batch normalization, and SE module on side features; the third baseline is the one adding the global context branch (Eq. 7); the last one is our model trained with the proposed loss function Eq. 13. Most of the models trained with the balanced cross-entropy loss [37] except the last one using the proposed loss. The results are shown in Tab. 2.

The performances show consistent improvements along the adding of the components. In the meantime, the model size and the computational cost are increased in a limited range. Note that, the global branch of the hyper module is only added in the high-level side feature to control the model size. The length L for BSDS500 image is setted to 20. These results have demonstrated the effectiveness of our method.

4.3.3 Ablation study on different backbones. The last ablation experiment is to test if the proposed method can work on different backbones. The backbones are SqueezeNet[15], MobileNetV2 [30], and the recent RegNetX [29]. Among these networks, SqueezeNet and MobileNetV2 are designed to be efficient, while the RegNetX has many variants from computational efficient to computational expensive. We choose the RegNetX-400M for comparison, since it has the similar ImageNet TOP-1 accuracy (72.278%) with VGG-16 and MobileNetV2. Our detailed setting is as follow: for each backbone, we test two different decoders, i.e., a naive decoder that directly fuses and upsamples side features and the proposed decoder. The naive decoder adopts transposed convolution to performs upsampling. The results are shown in Tab. 3.

The results have clearly demonstrated the effectiveness of the proposed decoder on keeping the slim model size and achieving high performance simultaneously. For the SqueezeNet-based detector, using the proposed decoder manages to significantly reduce 75% model size and 36.7% computational cost. However, its performance has a significant improvement from .767 to .775. For MobileNetV2, the model size has a 26.8% decrease and the computational cost is reduced by 34.0%. For RegNetX, two models share very similar performance, yet the second model is much smaller than the naive one (4.97M VS. 7.73M).

4.4 BSDS500

Berkeley Segmentation Dataset (BSDS 500) [1] is a widely used dataset which contains 200 training images, 100 validation images, and 200 testing images. The resolution of a BSDS500 image is 321×481. Each image is annotated by multiple people.

4.4.1 SOTA comparisons. Since we have already reported three ablation experiments on BSDS500, we now compare our method with top edge detectors. These SOTA methods include HED [37], RCF [22, 23], CED [36], LPCB [7] and DSCD [6]. Following their training strategy, we first pretrain our method on Pascal Voc Context dataset then finetune the model on the BSDS500 training dataset. We also perform multi-scale testing that resizes the input in three scales (0.5,1,1.5) to output and average the results. The qualitative results are shown in Fig.3 and the quantitative results are shown in Fig.4 and Tab.4.

From the quantitative results, we can clearly see that the proposed method outperforms most of the recent top edge detectors, yet only consumes limited computational resources. Our MobileNetV2-based system only has 1.94M weights and consumes 16.45GFlops for an 480×320 image, which is 13.2% and 12.5% of HED model size and computational cost, repectively; 4.8% and 11.9% of CED; 11.4% and 13.5% of LPCB; 13.1% and 13.7% of RCF-VGG16. Yet our method are better at contour alignment than these competitors, according to the results of benchmark. Especially comparing with RCF-ResNet101, although we share the same ODS F-score, our method only comsumes **3.3%** RCF-ResNet101's computational cost and **2.35%** model size, which demonstrates the superiority of our method. Although DSCD performs better than our method, it needs to take 17.6× more weights and 8.2× more computational cost, which is much worse at the system efficiency and the overall performance.

From the qualitative results, we can find that our predictions are crisp and better at localizing the inner visual salient edges. From the results in Fig. 3, one can find that all the predictions capture main contour structures of objects and suppress useless inner and background texture, which indicates that our light-weight model represents the same detection ability, compared to the SOTA methods. However, our method performs better at producing object inner salient edge. In the first example, Polar bear's eyes are well detected by our method, however, ignored by RCF and CED detectors; in the second example, our method manages to produce the contours

(g) Input Image (h) The Groundtruth (i) Ours

Figure 5: Examples of our predictions on the Multi-cue dataset. The predictions show consistent performances, which depict crisp boundaries and keep finer details of object boundaries.

of the windows. While other methods generate false negative at the same pixels. Besides, our prediction are crisp due to the advantage of the proposed loss function. These results demonstrate the effectiveness of our method.

4.5 Multi-Cue dataset

Multi-cue dataset [25] is a novel human-labeled object boundary detection dataset. It consists of 100 short binocular video clips of different scenes captured by a stereo camera. The dataset includes both object edge-groundtruth and boundary-groundtruth for each annotated image. Unlike BSDS500 dataset, each image has much larger resolution of (1280×720), compared to the BSDS500 image (321×481). We follow the preprocessing step of HED and RCF to first randomly crop 500×500 patch-label pairs for training. Then these patch-label pairs are randomly horizontal flipped, rotated (90, 180, and 270 degrees). The training hyper-parameters are the same as the setting of BSDS500 dataset. All the methods are trained separately on the boundary data and the edge data. Following the method of HED, we randomly split train/test images (80/20) three times, test our method, then report the average results. We show the results in Tab.5 and Fig.5.

In the Multi-Cue dataset, our system show consistent performances at producing crisp and accurate object boundaries. The qualitative results reveal that our predictions achieve the state-of-the-art results and outperforms computational expensive models. The qualitative results depict that the proposed detector is good at keep finer details of objects.

5 CONCLUSION

In this work, we present a highly efficient and accurate contour detection system. The proposed model can cost only around 10× less model weights and computational cost against most of the recent top edge detectors, however, produce better object boundaries at both accuracy and crispness. The success of our system is benefited from three aspects: full-refinements structure in a bottom-up/top-down manner, an ultra-efficient decoder that can decode compact contextual information, and a novel loss function. The basic network structure is the cornerstone and the efficient decoder is the core to the efficiency as well as the high accuracy. Our system has the great potential to favor broad computer vision applications. In the future, we will integrate our edge detector into other applications to further demonstrate the effectiveness of our method.

Table 5: Edge and boundary detection results on the Multi-cue dataset. Our method achieves SOTA performance and significantly outperforms other competitors.

Method	ODS	OIS
Multi-cue-Boundary	.720(.014)	-
HED-Boundary	.814(.011)	.822(.008)
RCF-Boundary	.817(.004)	.825(.005)
Ours-Boundary	**.839** (.004)	**.853**(.006)
Multi-cue-Edge	.830(.002)	-
HED-Edge	.851(.014)	.864(.011)
RCF-Edge	.857(.004)	.862(.004)
Ours-Edge	**.881**(.012)	**.893**(.011)

6 ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China [Grant No.61922064, U2033210], in part by the Hunan Science Fund for Distinguished Young Scholars [Grant No. 2019JJ20027], in part by the Zhejiang Provincial Natural Science Foundation [Grant No. LR17F030001, No. LZ21F020001], in part by the Project of science and technology plans of Wenzhou City [Grant No. C20170008].

REFERENCES

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. 2011. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence* 33, 5 (2011), 898–916.

[2] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. 2015. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293* (2015).

[3] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. 2015. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4380–4389.

[4] John Canny. 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), 679–698.

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 801–818.

[6] Ruoxi Deng and Shengjun Liu. 2020. Deep Structural Contour Detection. In *Proceedings of the 28th ACM International Conference on Multimedia*. 304–312.

[7] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. 2018. Learning to Predict Crisp Boundaries. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VI*. 570–586. https://doi.org/10.1007/978-3-030-01231-1_35

[8] Piotr Dollár and C Lawrence Zitnick. 2015. Fast edge detection using structured forests. *IEEE transactions on pattern analysis and machine intelligence* 37, 8 (2015), 1558–1570.

[9] J. R Fram and E. S Deutsch. 1975. On the Quantitative Evaluation of Edge Detection Schemes and their Comparison with Human Performance. *Computers IEEE Transactions on* C-24, 6 (1975), 616–628.

[10] Lianli Gao, Zhilong Zhou, Heng Tao Shen, and Jingkuan Song. 2020. Bottom-up and Top-down: Bidirectional Additive Net for Edge Detection. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020 [scheduled for July 2020, Yokohama, Japan, postponed due to the Corona pandemic]*. 594–600.

[11] Tong He, Chunhua Shen, Zhi Tian, Dong Gong, Changming Sun, and Youliang Yan. 2019. Knowledge Adaptation for Efficient Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 578–587.

[12] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[13] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[14] Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu. 2005. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th annual ACM international conference on Multimedia*. 351–354.

[15] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016).

[16] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR, 448–456.

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

[18] Iasonas Kokkinos. 2015. Pushing the boundaries of boundary detection using deep learning. *arXiv preprint arXiv:1511.07386* (2015).

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[20] Gerald Kühne, Stephan Richter, and Markus Beier. 2001. Motion-based segmentation and contour-based classification of video objects. In *Proceedings of the ninth ACM international conference on Multimedia*. 41–50.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[22] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Jia-Wang Bian, Le Zhang, Xiang Bai, and Jinhui Tang. 2019. Richer convolutional features for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 8 (2019), 1939–1946.

[23] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. 2016. Richer Convolutional Features for Edge Detection. *arXiv preprint arXiv:1612.02103* (2016).

[24] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* 60, 2 (2004), 91–110.

[25] David A Mely, Junkyung Kim, Mason McGill, Yuliang Guo, and Thomas Serre. 2016. A systematic comparison between visual cues for boundary detection. *Vision research* 120 (2016), 93–107.

[26] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. 807–814.

[27] Pietro Perona and Jitendra Malik. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence* 12, 7 (1990), 629–639.

[28] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. 2016. Learning to refine object segments. In *European Conference on Computer Vision*. Springer, 75–91.

[29] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. 2020. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10428–10436.

[30] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4510–4520.

[31] N Senthilkumaran and R Rajesh. 2009. Edge detection techniques for image segmentation–a survey of soft computing approaches. *International journal of recent trends in engineering* 1, 2 (2009), 250–254.

[32] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. 2015. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3982–3991.

[33] Matheen Siddiqui and Gérard Medioni. 2010. Human pose estimation from a single view point, real-time range sensor. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 1–8.

[34] Tsai-Ho Sun, Chien-Hsun Lai, Sai-Keung Wong, and Yu-Shuen Wang. 2019. Adversarial Colorization Of Icons Based On Structure And Color Conditions. *arXiv preprint arXiv:1910.05253* (2019).

[35] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 6105–6114.

[36] Yupei Wang, Xin Zhao, and Kaiqi Huang. 2017. Deep Crisp Boundaries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3892–3900.

[37] Saining Xie and Zhuowen Tu. 2015. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*. 1395–1403.