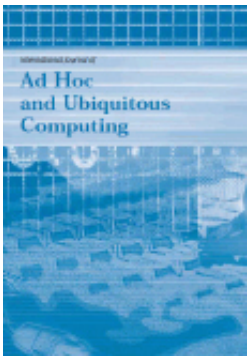


International Journal of Ad Hoc and Ubiquitous Computing

ISSN online: 1743-8233 - ISSN print: 1743-8225
https://www.inderscience.com/ijahuc

Multiscale hierarchical attention fusion network for edge detection

Kun Meng, Xianyong Dong, Hongyuan Shan, Shuyin Xia

DOI: 10.1504/IJAHUC.2023.10052674

Article History:

Received:	15 December 2021
Last revised:	11 January 2022
Accepted:	25 January 2022
Published online:	16 December 2022

Multiscale hierarchical attention fusion network for edge detection

Kun Meng

School of Computer Science and Technology,
Chongqing University of Posts and Telecommunications,
Chongqing 400065, China
Email: s190231175@stu.cqupt.edu.cn

Xianyong Dong*

China Three Gorges Construction Engineering Corporation,
1 Liuhe Road, Jiangan District, Wuhan, China
Email: 1184705866@qq.com
*Corresponding author

Hongyuan Shan and Shuyin Xia

School of Computer Science and Technology,
Chongqing University of Posts and Telecommunications,
Chongqing 400065, China
Email: s190231046@stu.cqupt.edu.cn
Email: 18844073573@163.com

Abstract: Edge detection is one of the basic challenges in the field of computer vision. The results of most recent methods produce thick edges and background interference. The images generated by these networks must be postprocessed with non-maximum suppression (NMS). To tackle the problem, we propose a novel edge detection model that allows the network to concentrate on learning the contextual features of an image, thereby obtaining more accurate pixel edges. To obtain abundant multi-granularity features of image high-level features, we introduce multi-scale feature stratification module (MFM). Then, we increase the constraint between pixels through the edge attention module (EAM), so that the model can obtain stronger feature extraction ability. These new approaches can improve the ability of describing edges of models. Evaluating our method on two popular benchmark datasets, the edge image predicted by this method is superior to existing edge detection methods in subjective perception and objective evaluation indexes.

Keywords: edge detection; deep learning; multiscale; attention network; non-maximum suppression; NMS; multi-scale feature stratification module; MFM; edge attention module; EAM.

Reference to this paper should be made as follows: Meng, K., Dong, X., Shan, H. and Xia, S. (2023) 'Multiscale hierarchical attention fusion network for edge detection', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. 42, No. 1, pp.1–11.

Biographical notes: Kun Meng received his BA in Internet Engineering from the Southwest Petroleum University of Science and Technology in Sichuan, China in 2015. He is currently pursuing his MS in Computer Technology at the Chongqing University of Posts and Telecommunications in Chongqing, China. His research interests include deep learning computer vision.

Xianyong Dong studied hydrology and water resources at Sichuan University from September 1996 to June 2000. He studied hydrology and water resources at Sichuan University from September 2002 to June 2005. He studied hydraulics and river dynamics at China Institute of Water Resources and Hydropower Research from 2007 to September 2010.

Hongyuan Shan received his BA in Communication Engineering from the Changchun University in Jilin, China in 2015. He is currently pursuing his MS in Computer Technology at the Chongqing University of Posts and Telecommunications in Chongqing, China. His research interests include deep learning computer vision.

Shuyin Xia received his BS degree in 2008 and MS degree in 2012, both in Computer Science and both from the Chongqing University of Technology in China. He received his PhD degree from the College of Computer Science at Chongqing University in China. Since 2015, he has been with the Chongqing University of Posts and Telecommunications in Chongqing, China, where he is currently an Associate Professor and PhD Supervisor.

This paper is a revised and expanded version of a paper entitled 'Multiscale hierarchical attention fusion network for edge detection' presented at International Conference on Applied Machine Learning, Yunnan of China, 23–25 July 2021.

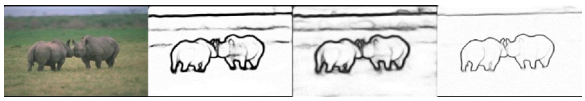
1 Introduction

With the rapid development of artificial intelligence, computer vision marks the arrival of Industry 4.0. The core of Industry 4.0 is efficient human-computer interaction, and computer vision, as the eyes of industrial robots, is of great significance to the Industry 4.0. Image edge detection which aims to find a collection of pixels with sharp brightness changes from natural images is considered a basic task in the field of computer vision (Canny 1986; Senthilkumaran and Rajesh, 2009). In computer vision applications, e.g., object detection (Ullman and Basri, 1991; Ferrari et al., 2007), image segmentation (Arbelaez et al., 2010; Abdel-Basset et al., 2020, 2021; Dhiman et al., 2021) and saliency detection (Zhao and Wu, 2019), artificial intelligence's security is an important guarantee for the development of Industry 4.0. Edge detection has obvious effect on countering attacks. Edge detection methods from the traditional manual feature extraction method (Canny, 1986; Kittler, 1983; Arbelaez et al., 2011) to the current end-to-end automatic feature extraction method using CNNs have attracted the attention of many researchers. In the past few years, deep learning has achieved impressive results in edge detection, Xie and Tu (2015), Liu et al. (2017) proposed some excellent deep CNN models [holistically-nested edge detection (HED) (Xie and Tu, 2015) and richer convolutional features (RCF) (Liu et al., 2017), respectively, with VGG-16 network (Simonyan and Zisserman, 2014) as the backbone network].

Although CNN-based methods (Xie and Tu, 2015; Liu et al., 2017; Yang et al., 2016; Yu et al., 2017) can efficiently generate image edges with semantic information, most of the current CNN-based methods will produce thick edges and background interference. As shown in Figure 1, the background interference in (c) is the most serious, and (b) and (c) have thicker edges. Therefore, most deep learning methods require postprocessing of the images generated by the model, which will increase the hardware overhead. Some researchers have proposed corresponding solutions to this phenomenon. Wang et al. (2017) proposed an enhanced feedforward network with a subpixel convolution layer, which uses features of different scales to learn distinct edges. Deng et al. (2018) proposed using the dice loss function as the auxiliary loss function to solve the problem of data imbalance between the positive and negative data of the edge detection image, which makes the network learn positive sample edges more effectively. The method they proposed partially solves the problem of excessively thick edges extracted by deep learning; however, it does not effectively combine high-level and low-level semantic information.

The main contribution of our research work proposes a novel edge detection model, named the multiscale hierarchical attention fusion network (MHANet). The network makes use of the fusion method of an edge attention module (EAM) and a multi-scale feature-layering module (MFM) to tackle the problem of the thick edges and background interference of deep neural network boundary prediction. In the context of the rapid development of artificial intelligence in Industry 4.0, edge images that conform to visual inspection are more conducive to features to understand images and generate data. Inspired by the scale-invariant feature transform (SIFT) (Lowe, 2004), we designed an MFM to extract the multi-granularity features of each layer. We take advantage of the attention module for low-level features and fuse them with multi-granularity features with an MFM to achieve the screening and use of low-level features. This method improves the visual effect of edge detection and the performance based on edge detection evaluation indicators. Our method that has not been processed by non-maximum suppression (NMS) achieves good performance on the BSDS500 dataset.

Figure 1 (a) Is an image in the BSDS500 dataset, (b) and (c) show the experimental results of the RCF and HED detectors, respectively, and (d) is the experimental result of our method. The edge contours in (d) are clearer than those in (b) and (c) without excessive background information interference (see online version for colours)



(a) (b) (c) (d)

Note: Postprocessing was not applied to any of the predicted edges.

We summarise our contributions as follows:

- We propose a novel model that focuses on learning the contextual features of images. It can obtain more accurate pixel edges without postprocessing.
- We propose a cross-space self-attention mechanism to obtain the spatial position relations between distant pixels and increase the constraints between pixels.

- The method we proposed that does not conduct NMS postprocessing outperforms the existing methods and still has good performance after NMS. Information as a service: providing remotely hosted information services.

2 Related work

In the past decades, lots of excellent works on edge detection have emerged. Early works (Sobel and Feldman, 1973; Duda and Hart, 1973; Torre and Poggio, 1986; Perona and Malik, 1990) focused on the image texture gradient, colour or other artificial visual features to extract edges. For example, the popular Sobel detector (Canny, 1986) and Prewitt (1970) detector use the greyscale difference between the upper and lower pixels and the left and right neighbours to detect edge pixels that reach the maximum value. The canny detector (Kittler, 1983) obtains the maximum gradient of the image through NMS, and then a dual threshold algorithm is used to detect the contours of finer pixels. However, the lack of advanced semantic features in these traditional methods limits the ability to obtain precise edges.

Recently, models based on deep learning have made remarkable achievements in extracting image edges. Ganin and Lempitsky (2014) proposed N4-Fields, which combines a CNN with traditional machine learning-based nearest neighbour search methods; Bertasius et al. (2015) proposed the DeepEdge network where the target related features are used for advanced clue detection of the contour and the patch extracted by the canny detector (Kittler, 1983) is input into the network model to determine the effective edge of the detection patch. Xie and Tu (2015) and Liu et al. (2017) adopted an end-to-end network model to capture richer semantic expression features through the supervision of features at different layers; Yu et al. (2017) proposed CASENet, which associates each edge pixel with more than one edge class and uses a multilabel loss function to supervise the activation fusion; Yang et al. (2016) proposed a novel encoding and decoding network (cedn) to deeply supervise the high-level semantic contour. He et al. (2019) used a two-way cascaded network to supervise the single-layer output of the labelling edge. Gao et al. (2020) proposed for the first time a two-way overlay network combining top-down and bottom-up approaches and used channel weighting mechanism to filter useless feature maps.

The above CNN-based methods have obtained excellent experimental results in image edge detection. However, they all need to perform postprocessing (NMS) on the generated images to eliminate the interference of thick edges and background to achieve a higher evaluation metric. As stated by Wang's (Yu et al., 2017) work, although thick edges can achieve high scores through NMS, when the maximum tolerance distance is reduced, the score will drop very drastically, which also shows that thick edges cannot be aligned with the image boundary in the real environment. In response to this phenomenon, we propose a hierarchical fusion of low-level and high-level features to effectively improve the thick edges. The difference between our previous work focusing on the thicker edge problem is that we add spatial attention (SA) to the image edge task for the first time to make the model focus to the contextual features of an image. We will detail our approach below.

3 Multiscale hierarchical attention fusion network

In this paper, we propose a MHANet model to reduce the effect of thick and background content interference in edge detection tasks. The MHANet is based on the VGGNet (Simonyan and Zisserman, 2014) as the backbone network, removing fully connected layers. The MHANet contains multiple MFMs to capture high level features and EAMs to fuse low-level features. The overall structure is shown in Figure 2.

3.1 Multiscale feature-layering module

The extraction and fusion of multiscale features are critical to many visual tasks (He et al., 2019). Multiscale features contain not only overall global information but also detailed local information. Inspired by the SIFT (Lowe, 2004) and the latest DeepLab series of works (Chen et al., 2018), we designed a novel MFM to extract and fuse the image's multiscale feature information. Similar to the DeepLab work, we use dilated convolutions to obtain multiscale features.

Specifically, the MFM is shown in Figure 3. We use dilated convolutions with different dilation rates to obtain the multiscale features of the high-level features. Then, the 1 × 1 convolved feature map and feature maps with different scales are used as residuals (He e al., 2016) to prevent overfitting and the vanishing gradients problem and increase the flow of information. Finally, the multiscale features and 1 × 1 convolution features from the residuals are combined through cross-channel connections. In this way, the model fully extracts and integrates multiscale features and uses them as the output of the MFM.

Generally, we let the input be $f \in \mathbb{R}^{(H \times W \times C)}$, which is a feature map with C channels. The formula of the MFM component is as follows:

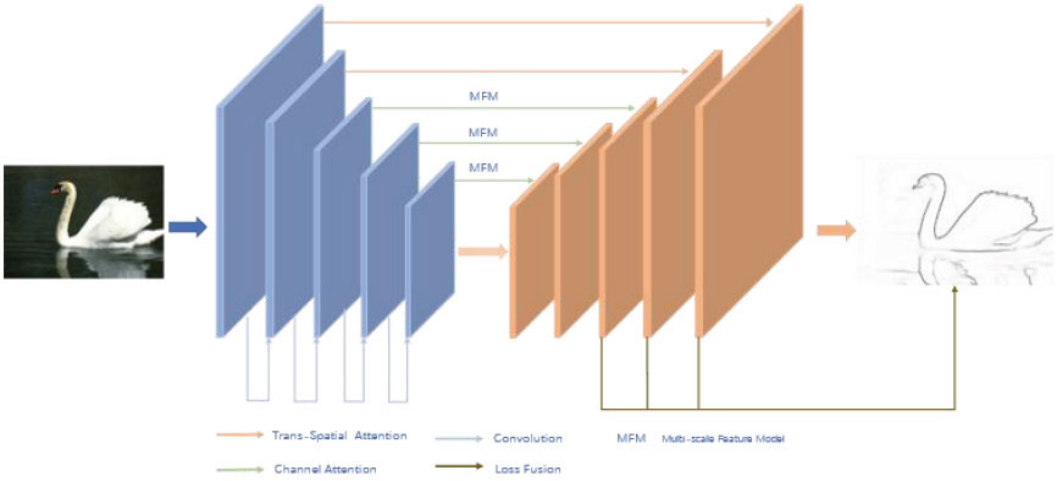
$$x_0 = H_0(f) \tag{1}$$

$$x_i = \sum_{i=1}^{k} H_i(f) + x_0 \tag{2}$$

H_0 represents an ordinary convolution, the size of the convolution kernel is 1 × 1

H_i represents convolutions with different dilation rates.

Figure 2 The model structure of the MHANet (see online version for colours)



Notes: Its core is composed of an MFM and an EAM. The EAM is composed of a CA module and an SA module.

Figure 3 Multiscale feature layering module (see online version for colours)

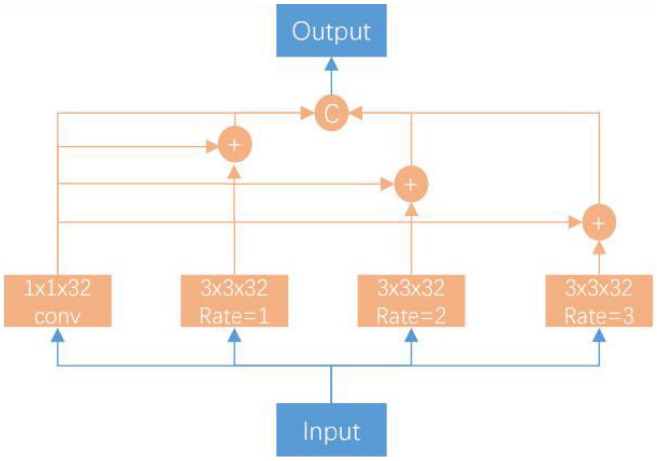


Figure 4 Channel attention of EAM (see online version for colours)

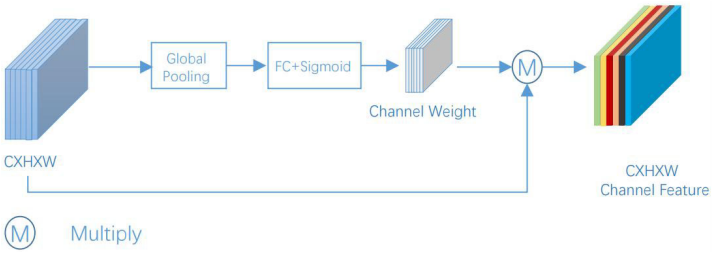
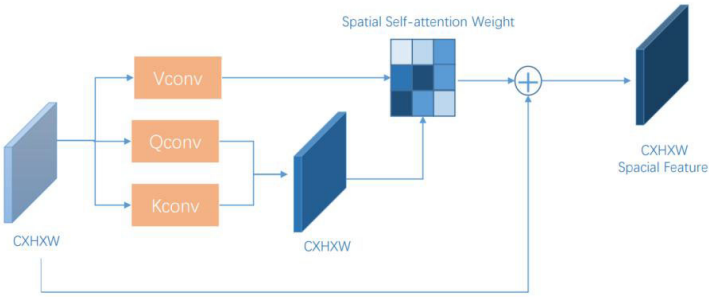


Figure 5 Trans-spacial self-attention of EAM (see online version for colours)



Convolutions receive residuals from ordinary convolutions to increase the flow of information. So the final output of the *MFM* is as follows:

$$MGM(f) = \left(\left[x_0, x_1, x_2, x_3 \right] \right) \tag{3}$$

where [] means concatenating the matrix in the specified dimension.

3.2 Edge attention module

We use the MFM to obtain the multiscale features of high-level features. In image edge detection, the saliency maps generated by different features will have information redundancy (He et al., 2019), and information that is similar to noise may cause performance degradation or even mispredictions. The MFM uses an attention mechanism to weight the information, effectively alleviating information redundancy. According to the characteristics of different levels of the VGGNet, we use a CA mechanism for the third, fourth and fifth layers and an SA mechanism for the first and second layers. Generally, low-level features have some noise, but they contain more spatial information. At the same time, there is almost no semantic difference between channels and channels in shallow channels, so the underlying features only use SA. Although most high-level features are coarse, they carry more semantic information. High-level features have a large quantity of channels, and there will be information redundancy. Therefore, only the CA mechanism is used for high-level features.

3.2.1 Channel attention

Different feature channels in the CNN respond to different semantics. After the MFM, we add CA to the different feature channels after multiscale features to weight those edge feature maps with higher correlation. In this section, we introduce CA in detail, and its structure is shown in Figure 4.

Similarly, let $f \in \mathbb{R}^{(H \times W \times C)}$ denote a high-level feature map with the C channel. First, apply the global pooling layer to f to obtain a vector $v \in \mathbb{R}^c$ containing global information f. Then, the weight relationship between the channels is obtained through two consecutive fully connected layers. The sigmoid activation function normalises the weight to between 0 and 1, and this process can be expressed by the following formula:

$$w = F(v, W) sigmoid\left(fc_2\left(\sigma\left(fc_1\left(v, W_1 \right) \right), W \right) \right) \tag{4}$$

where w represents the weight of each channel, σ represents the ReLU activation function, fc_1 and fc_2 represent the fully connected layers, and v represents the vector of f after global pooling. The final definition of CA is expressed as:

$$CA(f) = w * f + f \tag{5}$$

3.2.2 Trans-spacial self-attention

The underlying features usually contain rich detailed foreground and complex background, which increase the uncertainty of image representation and increase the difficulty of model training (Zhao and Wu, 2019). In edge detection task, we hope to obtain a detailed boundary between the object and the background. Therefore, we need to correlate the relationship between more adjacent pixels, so as to filter the spatial information of the underlying features that contain rich details, instead of considering all spatial positions equally, which helps to generate finer edges. Inspired by self-attention mechanism (Vaswani et al., 2017; Devlin et al., 2018), we proposed trans-spacial self-attention to get more relevant pixel relations shown in Figure 4.

Let $f \in \mathbb{R}^{(H \times W \times C)}$ be the lower-level feature map with C channels. Inspired by the SA mechanism (Ferrari et al., 2007), the input feature maps are respectively passed through three convolutional layers, the first layer of convolution kernel is $1 \times k$, the second layer of convolution kernel is $k \times 1$, and the last layer of convolution kernel is $k \times k$, where k is set to 3, which is used to receive the global information of the underlying features without adding parameters. Finally, the output feature maps are S_Q, S_k, S_v (see Figure 5), Then S_Q is multiplied by the transposition of S_k to obtain the transposed matrix, and finally the Softmax function normalises the encoded spatial feature maps to [0, 1].

$$S_Q = Q_{conv}(CA(fi)) \tag{6}$$

$$S_K = K_{conv}(CA(fi)) \tag{7}$$

$$S_K = V_{conv}(CA(fi)) \tag{8}$$

where $CA(f)$ represents the CA module in the previous section, Q_{conv}, K_{conv}, and V_{conv} represent three convolutional layers, so the final output of SA is expressed as:

$$SA\left(fi, S_Q, S_K, S_V \right) = fi\left(Soft\max\left(SQ \cdot S_K^T \right) * S_V \right) \tag{9}$$

3.2.3 Network optimisation

Loss function: in the publicly available dataset, the number of pixels in the background area is much larger than that in the edge area, which will cause a serious imbalance of positive and negative samples (Gu et al., 2007; Tang and Liu, 2005; Blagus and Lusa, 2010; Haider et al., 2014; Lee et al., 2018). Therefore, we use the class-balanced cross-entropy loss to train our network based on previous work (Liu et al., 2017; Yang et al., 2016). The loss is defined as:

$$L_{BCE} = (W') = -\beta \sum\nolimits_{T_i \in E} \log P\left(T_i = 0 \middle| I; W' \right) \# \tag{10}$$

where I is a natural image, E represents all pixels, E_+ represents all edge pixels in the image, E_- represents all non-edge points in the image, and T is the predicted image

of the model. $\beta = \frac{|E_-|}{|E|}$ is the proportion of the target pixel in image pixels, which represents the weight of evaluating positive and negative samples when calculating the loss. W' is the trainable parameter of the network. Following previous work (Deng et al., 2018; Milletari et al., 2016), the class-balanced cross-entropy loss combined with the dice loss (Dice, 1945) can generate clear-edge mapping. The dice loss function is defined as:

$$L_{dice}(W') = \frac{\sum_i^N p_i^2 + \sum_i^N t_i^2}{2\sum_i^N p_i t_i} \tag{11}$$

where p_i is the predicted value of the i-th edge pixel, and t_i is the true value of the i-th edge pixel. The final loss function is expressed by the formula:

$$L(W') = L_{BCE}(W') + \lambda\left(L_{Dice}(W')\right) \tag{12}$$

where λ is a hyperparameter that balances L_{BCE} and L_{Dice} and is set to 0.01.

Training strategy: we use the output containing high-level features to calculate the loss to reduce the impact of low-level features on high-level features. We use the output of the last three layers (P_3, P_4, P_5) and the fusion P_{fuse} of the three-layer output as the total cumulative loss. P_i is obtained through an EAM and an $N \times 1 \times 1$ convolutional layer, as shown in Figure 2. To generate enough semantic information, we use only the Dice loss for P_{fuse}. The loss function strategy is expressed by the formula:

$$L_{total}\left(P_3, P_4, P_5, P_{fuse}\right) = L\left(P_{fuse}\right) + \lambda\sum_{i=3}^{5} L_{BCE}\left(P_i\right) \tag{13}$$

P_{fuse} represents the final output of the fusion layer, and P_i represents the output of the i-th layer of the network.

4 Experiments

We will discuss the specific parameters of the experiment and briefly describe the data set. Then, we conducted ablation experiments on the proposed method and reported the performance of the method.

4.1 DataSets

We will evaluate the excellent performance of the proposed method on two publicly available datasets widely used in previous work: BSDS500 (Arbelaez et al., 2010) and the NYU Depth dataset, version 2 (NYUDv2) (Silberman et al., 2012).

The BSDS500 is used by the University of Berkeley for image segmentation and object edge detection tasks. It contains 200 training images, 100 verification images and 200 test images. Each image is annotated manually by many annotators, and the final GroudTrue images are the average annotation of the annotator.

The NYUDv2 dataset is very challenging. It is used by some jobs for edge detection tasks. The NYUDv2 dataset is composed of 1,449 densely labelled aligned RGB and depth images. Gupta et al. (2013, 2014) divided the NYUDv2 dataset into 381 training images, 414 verification images, and 654 test images. We will follow their data division method to train the network.

4.2 Experimental details

We use the PyTorch (Paszke et al., 2017) to implement our method. We use the pretrained VGGNet model as the initial backbone network of the MHANet model. Unlike previous work (Xie and Tu, 2015; Liu et al., 2017; Ganin and Lempitsky, 2014; Bertasius et al., 2015) we used the Adam (Gupta et al., 2014) optimiser to optimise the network. The initial learning rate is set to 0.01, the batch size of all the experiments is set to 8, the weight attenuation is set to 1e-4, the number of training epochs is set to 20, and the λ in the loss function is set to 0.01. Based on previous work, appropriate data enhancement will effectively improve the model's performance; therefore, we performed data enhancement on each dataset. First, the dataset was randomly rotated by ten different angles, and then appropriate brightness and gamma correction were applied on the basis of these ten angles. Finally, an enhanced dataset of more than 50 k images was obtained. We cropped the images in the dataset to 320 × 320 size and then input them into the network model.

It is worth emphasising that unlike some previous works (Xie and Tu, 2015; Liu et al., 2017; Wang et al., 2017; Xu et al., 2017). We did not perform NMS on the final edge map. The final output image is a complete end-to-end generation of clear-edge images, which is more consistent with human visual inspection. At the same time, the work of Wang et al. (2017) showed that after NMS, even if the edge is slightly offset, a good score can be obtained. To evaluate the edge performance more accurately, we did not use NMS postprocessing. To make a fair comparison with other works (Xie and Tu, 2015; Liu et al., 2017; Wang et al., 2017), we used popular evaluation indicators, namely, the average precision (AP), optimal dataset scale (ODS), and optimal image scale (OIS), to measure the edge detection performance.

In the experiment to test the performance, we set the maximum tolerance distances (Xie and Tu, 2015; Gupta et al., 2014; Dollár and Zitnick, 2014) of the BSDS500 dataset and NYUDv2 dataset to 0.0075 and 0.011, respectively.

4.3 Ablation experiments

In this section, we perform ablation experiments to verify the influence of the MFM and the EAM on the network on the BSDS500.

First, we verify the parameter's influence of the MFM on the network. Our baseline model replaces the dilated convolutions in the MFM with ordinary convolutions. We also verified the influence of different dilation rates on the laboratory results.

Table 1 The influence of the MFM on the edge detection results under different parameters

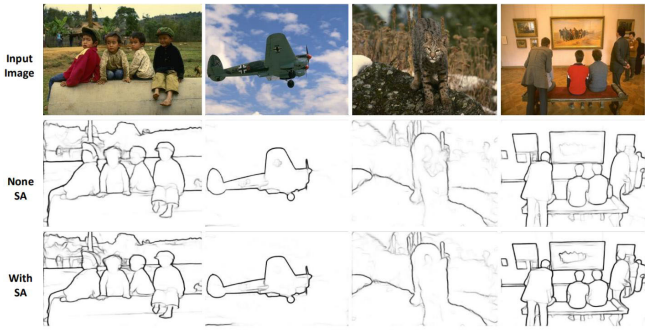
r_0	*Rate*	*ODS*	*OIS*	*AP*
0	1, 1, 1	0.769	0.784	0.741
1	*1, 2, 3*	*0.781*	*0.799*	*0.766*
2	2, 4, 6	0.773	0.787	0.745
3	4, 8, 12	0.769	0.776	0.705
4	8, 16, 24	0.767	0.784	0.742

Table 2 Verification of the validity of CA and SA in MHANet, where NoCA represents not adopting the CA mechanism and NoSA represents not adopting the SA mechanism

Variant	*ODS*	*OIS*	*AP*
Baseline	0.716	0.732	0.699
CA + SA	0.759	0.775	0.705
CA + MFM	0.768	0.783	0.729
SA + MFM	0.770	0.786	0.737
CA + SA + MFM	*0.781*	*0.799*	*0.766*

From Table 1, we can see that when the rates are 1, 2, and 3, the MFM can achieve the highest score. In our model, using a lower dilation rate improves the performance. Similarly, other models (Yang et al., 2016; He et al., 2019) also show that different models have different performances for the dilation rate, so it is also very important to choose the dilation rate that suits the model.

Figure 6 The different effects of THE BSDS500 data set with and without SA (see online version for colours)



Note: It can be shown that the model using SA can better capture the edge pixels of the detailed features of the image, and at the same time remove part of the noise of the image background.

Second, we verify that the EAM improves the network performance. The EAM is composed of two parts, and we need to verify the influence of the CA mechanism and SA mechanism separately. In the baseline model, we removed the CA, SA and MFM modules. The number of convolutions is consistent with the number of convolutions of the EAM. Table 2 shows that when the CA mechanism and MFM module are adopted, the ODS, OIS, and AP improve by 5.2%, 5.0%, and 3.0%, respectively, compared to the baseline model. When the SA mechanism and MFM module are adopted, they increase by 6.6%, 5.4%, and 3.8%, respectively, compared to the baseline model. Using the CA mechanism, SA mechanism and MFM module at the same time increases the final experimental effects of the model by 6.5%, 6.7%, and 6.7%, respectively.

We visualised a part of the effect map without SA, as shown in Figure 6. It can be shown that the NoneSA model cannot capture the edges of the subtle features of the image, and the WithSA model can not only effectively filter background noise, but also the edge pixels of the detailed features of the image can be extracted.

4.4 Comparison with other works

4.4.1 BSDS500 dataset

We will compare our results with the results of previous excellent work (Xie and Tu, 2015; Liu et al., 2017; Wang et al., 2017; He et al., 2019). We show the visual effects of the images generated by the MHANet and the work of others in Figure 7.

Table 3 Test results on the BSDS500 before applying NMS

Method	*ODS*	*OIS*	*AP*
Canny (Canny, 1986)	0.600	0.640	0.580
EGB (Felzenszwalb and Huttenlocher, 2004)	0.610	0.640	0.560
Mshift (Comaniciu and Meer, 2002)	0.601	0.644	0.493
gPb-owt-ucm (Abdel-Basset et al., 2020)	0.726	0.757	0.696
ISCRA (Ren and Shakhnarovich, 2013)	0.724	0.752	0.783
Sketch tokens (Lim et al., 2013)	0.727	0.746	0.780
DeepNets (Bertasius et al., 2015)	0.738	0.759	0.758
MCG (Arbelaez et al., 2014)	0.747	0.779	0.759
SE (Dollár and Zitnick, 2014)	0.746	0.767	0.803
OEF (Hallman and Fowlkes, 2015)	0.749	0.772	0.817
LEP (Zhao, 2015)	0.757	0.793	*0.828*
N4-Fields (Ganin and Lempitsky, 2014)	0.753	0.769	0.784
HED_BeforeNMS (Xie and Tu, 2015)	0.644	0.635	——
RCF_BeforeNMS (Liu et al., 2017)	0.773	0.789	0.633
LPCB_BeforeNMS (Deng et al., 2018)	0.693	0.700	——
BDCN_BeforeNMS (He et al., 2019)	0.777	0.794	0.471
Ours_BeforeNMS	*0.781*	*0.799*	0.766

It can be shown from Figure 7 that the output results of our model have smaller edges, are closer to the real-label images, and have better visual effects than the results of other excellent work. In the real Industry 4.0 scenario, the original image has strong background interference. During

the training process, our EAM can adaptively adjust the weight of each region of the image and effectively filter out other information except contours. Therefore, the background of the output image of our model is cleaner than that of other research work.

It is worth emphasising that we did not perform NMS operations or other image postprocessing operations on the images generated by the model. For different data sets, NMS needs to use different parameters; in other words, NMS functions are not universal. In this research and the following research, we evaluated the experimental results of each model twice (before applying NMS and after applying NMS). At the same time, we also give the increasing and decreasing trends in the MHANet, RCF and BDCN at different maximum tolerance distances, as shown in Figure 8. By doing presenting these trends, we can comprehensively evaluate the correctness of the edge and accurately locate the edge pixels.

Table 4 Test results on the BSDS500 after applying NMS

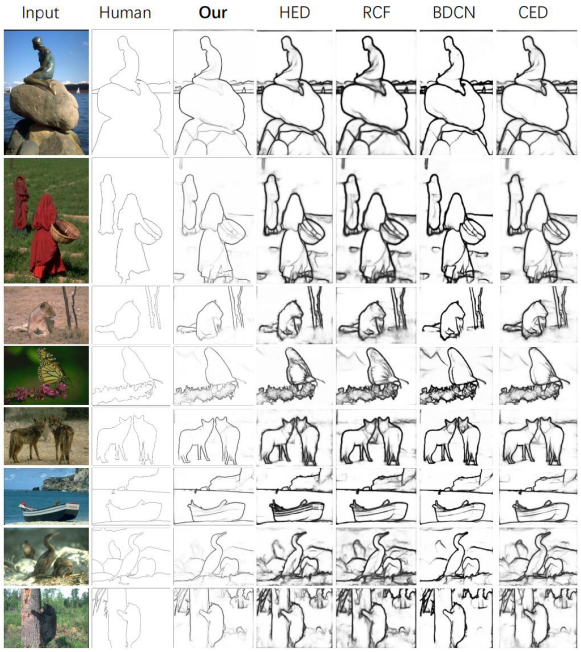
Method	*ODS*	*OIS*	*AP*	*Year*
DeepEdge (Bertasius et al., 2015)	0.753	0.772	0.807	2015
CSCNN (Hwang and Liu, 2015)	0.756	0.775	0.798	2015
DeepContour (Shen et al., 2015)	0.756	0.773	0.797	2015
CEDN (Yang et al., 2016)	0.788	0.804	0.834	2016
HED-fusing (Xie and Tu, 2015)	0.782	0.804	0.833	2017
HED-late-merging (Xie and Tu, 2015)	0.788	0.808	0.840	2017
COD (Shen et al., 2015)	0.793	0.820	0.859	2017
CED (Wang et al., 2017)	0.803	0.820	*0.871*	2017
RCF_AfterNMS (Liu et al., 2017)	0.798	0.815	——	2017
LPCB_AfterNMS (Deng et al., 2018)	0.800	0.816	——	2018
BDCN_AfterNMS (He et al., 2019)	0.806	0.826	0.847	2019
DexiNed_AfterNMS (Poma et al., 2020)	0.729	0.745	0.583	2020
DSCD_AfterNMS (Deng and Liu, 2020)	0.802	0.817	——	2020
PiDiNet (Su et al., 2021)	0.807	0.823	——	2021
Ours_AfterNMS	*0.808*	*0.828*	0.861	

We adopted the same strategy for excellent deep learning models from recent years and obtained the experimental results before applying NMS (Table 3). As shown in Table 3, that under the same experimental conditions before applying NMS, compared with previous research work, our proposed method has significantly improved in terms of the evaluation indicators. The ODS and OIS increased by 0.4% and 0.5%, respectively. Since the edge result of our model's output is clearer and closer to the real edge image, there is a large improvement in the AP. For a better comparison, we performed corresponding NMS operations on the networks. From Table 4, the proposed model not only achieved state-of-the-art performance before applying NMS but also performed well after applying NMS. Although the ODS and OIS indicators are slightly lower than those of the BDCN, we still achieve a 1.3% improvement in the AP.

Table 5 Test results on the NYUDv2 dataset before applying NMS

Method	*ODS*	*OIS*	*AP*
Silberman (Silberman et al., 2012)	0.658	0.661	——
MCG-B (Arbelaez et al., 2014)	0.652	0.681	0.613
SE (Dollár and Zitnick, 2014)	0.685	0.699	0.679
HED_AfterNMS_RGB (Xie and Tu, 2015)	0.720	0.734	*0.734*
RCF_BeforeNMS_RGB (Liu et al., 2017)	0.720	0.732	0.567
BDCN_BeforeNMS-RGB (He et al., 2019)	0.716	0.731	0.577
Ours_BeforeNMS-RGB	*0.730*	*0.744*	0.666

Figure 7 Comparison of results on the BSDS500 test set (see online version for colours)



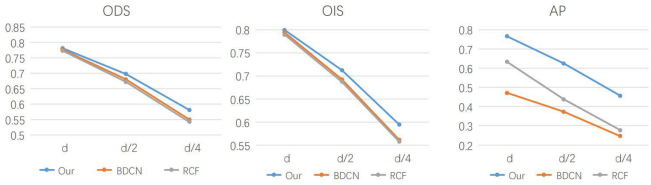
Note: All the test results are before applying NMS.

4.4.2 NYUDv2 dataset

We compare our method with advanced methods on the NYUDv2 dataset. Similarly, we initially did not apply NMS to the predicted images from the model, nor did we perform any postprocessing. Compared with the experimental results of the existing methods, our experimental results are improved in terms of the ODS, OIS, and AP. The comparison results are shown in Table 5. Compared with the RCF (Liu et al., 2017) and BDCN (He et al., 2019), our

model achieves 1%, 0.8%, and 8.9% improvements on the OIS, ODS, and AP, respectively. Our indicators have exceeded the results of the RCF (Liu et al., 2017) after applying NMS. This finding indicates that in our model, NMS postprocessing may not be necessary.

Figure 8 On the BSDS500, the MHANet, RCF and BDCN have different maximum tolerance distances d for the increasing and decreasing trends in the evaluation indexes (ODS, OIS, and AP) (see online version for colours)

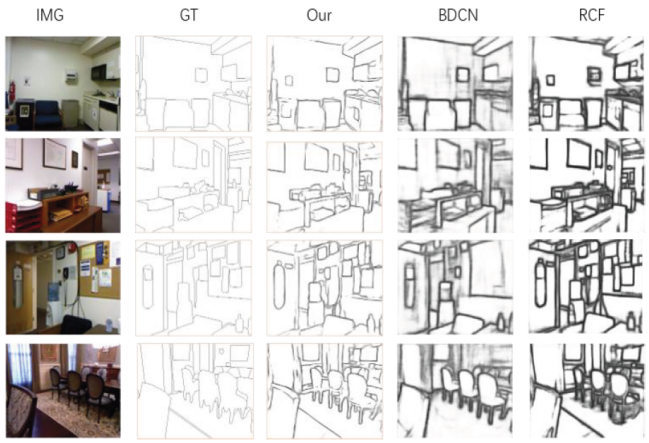


Note: As *d* decreases, the ODS, OIS, and AP performance gaps increase from 0.8% to 3.7%, from 1% to 3.7%, and from 1.33% to 17.9%, respectively.

Table 6 Test results on the NYUDv2 dataset after applying NMS

Method	*ODS*	*OIS*
gPb-UCM (Arbelaez et al., 2010)	0.632	0.661
gPb-NG (Gupta et al., 2013)	0.687	0.716
OEF (Hallman and Fowlkes, 2015)	0.651	0.667
SE (Gupta et al., 2014)	0.695	0.708
SE+NG+ (Dollár and Zitnick, 2014)	0.706	0.734
LPCB (Deng et al., 2018)	0.739	0.754
AMH-Net-ResNet50 (Xu et al., 2017)	0.744	*0.758*
HED_RGB (Xie and Tu, 2015)	0.720	0.734
RCF_RGB (Liu et al., 2017)	0.729	0.742
Ours_RGB	*0.745*	0.753

Figure 9 Comparison of results on the NYUDv2 test set (see online version for colours)



Note: All the test results are before applying NMS.

5 Conclusions

In this paper, we propose a novel MHANet which fully realises the integration of low-and high-level features. An MFM for high-level features is designed to obtain a larger receptive field. Moreover, our MHANet uses an EAM to suppress background noise to obtain a clearer edge. In the Industry 4.0 era, clear edges and noise immunity are more conducive to the deployment of real scenes. Experimental results show that this method has excellent performance without conducting NMS, which means that in edge detection tasks, NMS postprocessing is not necessary. And it is superior to other work in terms of human visual inspection. Pursuing further improvement in terms of the evaluation indicators is our future plan. Another future direction is that, in addition to edge detection, we also hope to apply the proposed method to other computer vision tasks and image pattern recognition in big data (Zerdoumi et al., 2018) in Industry 4.0 era.

Acknowledgements

This work was supported in part by National Key Research and Development Program of China (2019QY(Y)0301, the National Natural Science Foundation of China under Grant Nos. U2040217, 62176033 and 61936001, and the Natural Science Foundation of Chongqing No. cstc2019jcyjcxttX0002.

References

Abdel-Basset, M., Chang, V. and Mohamed, R. (2020) 'HSMA_WOA: a hybrid novel slime mould algorithm with whale optimization algorithm for tackling the image segmentation problem of chest X-ray images', *Applied Soft Computing*, Vol. 95, p.106642.

Abdel-Basset, M., Chang, V. and Mohamed, R. (2021) 'A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems', *Neural Computing and Applications*, Vol. 33, No. 17, pp.10685–10718.

Arbelaez, P., Maire, M., Fowlkes, C. and Malik, J. (2010) 'Contour detection and hierarchical image segmentation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 5, pp.898–916.

Arbelaez, P., Maire, M., Fowlkes, C. and Malik, J. (2011) 'Contour detection and hierarchical image segmentation', *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 33, No. 5, pp.898–916.

Arbelaez, P., Pont-Tuset, J., Barron, J., Marques, F. and Malik, J. (2014) 'Multiscale combinatorial grouping', in *Proc. Structured c. IEEE Conf. Comput. Vis. Pattern Recognit.*, June, pp.128–140.

Bertasius, G., Shi, J. and Torresani, L. (2015) 'DeepEdge: a multi-scale bifurcated deep network for top-down contour detection', in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June, pp.4308–4389.

Blagus, R. and Lusa, L. (2010) 'Class prediction for high-dimensional class-imbalanced data', *BMC Bioinformatics*, Vol. 11, No. 1, pp.1–17.

Canny, J. (1986) 'A computational approach to edge detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp.679–698.

Chen, L-C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L. (2018) 'Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 40, No. 4, pp.834–848.

Comaniciu, D. and Meer, P. (2002) 'Mean shift: a robust approach toward feature space analysis', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, pp.603–619.

Deng, R. and Liu, S. (2020) 'Deep structural contour detection', *Proceedings of the 28th ACM International Conference on Multimedia*, pp.304–312.

Deng, R., Shen, C., Liu, S., Wang, H. and Liu, X. (2018) 'Learning to predict crisp boundaries', in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, September, pp.562–578.

Devlin, J., Chang, M.W., Lee, K. et al. (2018) *Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding*, arXiv preprint arXiv: 1810.04805.

Dhiman, G., Chang, V., Kant Singh, K. and Shankar, A. (2021) 'Adopt: automatic deep learning and optimization-based approach for detection of novel coronavirus covid-19 disease using xray images', *Journal of Biomolecular Structure and Dynamic*s, pp.1–13.

Dice, L.R. (1945) 'Measures of the amount of ecologic association between species', *Ecology*, Vol. 26, No. 3, pp.297–302.

Dollár, P. and Zitnick, C.L. (2014) 'Fast edge detection using structured forests', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, No. 8, pp.1558–1570.

Duda, R. and Hart, P. (1973) *Pattern Classification Scene Analysis*, Wiley, Hoboken, NJ, USA.

Felzenszwalb, P.F. and Huttenlocher, D.P. (2004) 'Efficient graph-based image segmentation', *International Journal of Computer Vision*, Vol. 59, No. 2, pp.167–181.

Ferrari, V., Fevrier, L., Jurie, F. and Schmid, C. (2007) 'Groups of adjacent contour segments for object detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 1, pp.36–51.

Ganin, Y. and Lempitsky, V. (2014) 'N4-Fields: neural network nearest neighbor fields for image transforms', in *ACCV*, pp.536–551.

Gao, L., Zhou, Z., Shen, H.T. and Song, J. (2020) 'Bottom-up and top-down: bidirectional additive net for edge detection', *Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence (IJCAI-PRICAI-20*).

Gu, J., Zhou, Y. and Zuo, X. (2007) 'Making class bias useful: a strategy of learning from imbalanced data', in *International Conference on Intelligent Data Engineering and Automated Learning*, Springer, Berlin, Heidelberg, December, pp.287–295.

Gupta, S., Arbelaez, P. and Malik, J. (2013) Perceptual organization and recognition of indoor scenes from RGB-d images', in *CVPR*.

Gupta, S., Girshick, R., Arbeláez, P. et al. (2014) 'Learning rich features from RGB-D images for object detection and segmentation', *European Conference on Computer Vision*, Springer, Cham, pp.345–360.

Haider, A.H., Schneider, E.B., Sriram, N., Dossick, D.S., Scott, V.K., Swoboda, S.M., Losonczy, L., Haut, E.R., Efron, D.T., Pronovost, P.J. et al. (2014) 'Unconscious race and class bias: its association with decision making by trauma and acute care surgeons', *Journal of Trauma and Acute Care Surgery*, Vol. 77, No. 3, pp.409–416.

Hallman, S. and Fowlkes, C.C. (2015) 'Oriented edge forests for boundary detection', in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June, pp.1732–1740.

He, J., Zhang, S., Yang, M., Shan, Y. and Huang, T. (2019) 'Bi-directional cascade network for perceptual edge detection', in *CVPR*, pp.3828–3837.

He, K., Zhang, X., Ren, S. and Sun, J. (2016) 'Deep residual learning for image recognition', in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp.770–778.

Hwang, J-J. and Liu, T-L. (2015) 'Pixel-wise deep learning for contour detection', in *Proc. IEEE Int. Conf. Learn. Represent.*, pp.1–2.

Kittler, J. (1983) 'On the accuracy of the Sobel edge detector', *Image and Vision Computing*, Vol. 1, No. 1, pp.37–42.

Lee, J., Hong, B., Jung, S. and Chang, V. (2018) 'Clustering learning model of CCTV image pattern for producing road hazard meteorological information', *Future Generation Computer Systems*, Vol. 86, pp.1338–1350.

Lim, J.J., Zitnick, C.L.and Dollár, P. (2013) 'Sketch tokens: a learned mid-level representation for contour and object detection', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Liu, Y., Cheng, M-M., Hu, X., Wang, K. and Bai, X. (2017) 'Richer convolutional features for edge detection', in *CVPR*.

Lowe, D.G. (2004) 'Distinctive image features from scale in variant keypoints', *International Journal of Computer Vision*, Vol. 60, No. 2, pp.91–110.

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

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A. (2017) 'Automatic differentiation in pytorch', in *NIPS Workshop*.

Perona, P. and Malik, J. (1990) 'Scale-space and edge detection using anisotropic diffusion', *IEEE Trans. Pattern Anal. Mach. Intell.*, July, Vol. 12, No. 7, pp.629–639.

Poma, X.S., Riba, E. and Sappa, A. (2020) 'Dense extreme inception network: towards a robust CNN model for edge detection', in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, March, pp.1923–1932.

Prewitt, J.M.S. (1970) 'Object enhancement and extraction', in Lipkin, B. and Rosenfeld, A. (Eds.): *Picture Processing and Psychopictorics*, pp.75–149, Academic, New York.

Ren, Z. and Shakhnarovich, G. (2013) 'Image segmentation by cascaded region agglomeration', in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June, pp.2011–2018.

Senthilkumaran, N. and Rajesh, R. (2009) 'Image segmentation-a survey of soft computing approaches', in *2009 International Conference on Advances in Recent Technologies in Communication and Computing*, IEEE, October, pp.844–846.

Shen, W., Wang, X., Wang, Y., Bai, X. and Zhang, Z. (2015) 'DeepContour: a deep convolutional feature learned by positive-sharing loss for contour detection', in Proc. *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June, pp.3982–3991.

Silberman, N., Hoiem, D., Kohli, P. and Fergus, R. (2012) 'Indoor segmentation and support inference from rgbd images', in *European Conference on Computer Vision*, Springer, Berlin, Heidelberg October, pp.746–760.

Simonyan, K. and Zisserman, A. (2014) *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv preprint arXiv: 1409.1556.

Sobel, I. and Feldman, G. (1973) 'A 3x3 isotropic gradient operator for image processing', in *Pattern Classification and Scene Analysis*, pp.271–272, John Wiley and Sons.

Su, Z., Liu, W., Yu, Z., Hu, D., Liao, Q., Tian, Q. and Liu, L. (2021) 'Pixel difference networks for efficient edge detection', in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp.5117–5127.

Tang, L. and Liu, H. (2005) 'Bias analysis in text classification for highly skewed data', in *Fifth IEEE International Conference on Data Mining*, IEEE, p.4.

Torre, V. and Poggio, T. (1986) 'On edge detection', *IEEE Trans. Pattern Anal. Mach. Intell.*, February, Vol. PAMI-8, No. 2, pp.147–163.

Ullman, S. and Basri, R. (1991) 'Recognition by linear combinations of models', *IEEE Trans. Pattern Anal. Mach. Intell.*, October, Vol. 13, No. 10, pp. 992–1006.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N. and Polosukhin, I. (2017) 'Attention is all you need', in *Advances in Neural Information Processing Systems*, pp.5998–6008.

Wang, Y., Zhao, X. and Huang, K. (2017) 'Deep crisp boundaries', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.3892–3900.

Xie, S. and Tu, Z. (2015) 'Holistically-nested edge detection, in *Proceedings of the IEEE International Conference on Computer Vision*, pp.1395–1403.

Xu, D., Ouyang, W., Alameda-Pineda, X., Ricci, E., Wang, X. and Sebe, N. (2017) 'Learning deep structured multi-scale features using attention-gated crfs for contour prediction', in *NIPS*, pp.3964–3973.

Yang, J., Price, B., Cohen, S., Lee, H. and Yang, M.H. (2016) 'Object contour detection with a fully convolutional encoder-decoder network', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.193–202.

Yu, Z., Feng, C., Liu, M-Y. and Ramalingam, S. (2017) 'CASENet: deep category-aware semantic edge detection', in *IEEE Conf. Comput. Vis. Pattern Recog.*, pp.21–26.

Zerdoumi, S., Sabri, A.Q.M., Kamsin, A., Hashem, I.A.T., Gani, A., Hakak, S. and Chang, V. (2018) 'Image pattern recognition in big data: taxonomy and open challenges: survey', *Multimedia Tools and Applications*, Vol. 77, No. 8, pp.10091–10121.

Zhao, Q. (2015) 'Segmenting natural images with the least effort as humans', in *Proc. Brit. Mach. Vis. Conf.*, March, p.110.

Zhao, T. and Wu, X. (2019) 'Pyramid feature attention network for saliency detection', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.3085–3094.