



# Semantic Line Detection Using Mirror Attention and Comparative Ranking and Matching

Dongkwon Jin<sup>✉</sup>, Jun-Tae Lee<sup>✉</sup>, and Chang-Su Kim<sup>(✉)</sup>

School of Electrical Engineering, Korea University, Seoul, Korea  
[dongkwonjin,jtlee}@mc1.korea.ac.kr](mailto:{dongkwonjin,jtlee}@mc1.korea.ac.kr), [changsukim@korea.ac.kr](mailto:changsukim@korea.ac.kr)

**Abstract.** A novel algorithm to detect semantic lines is proposed in this paper. We develop three networks: detection network with **mirror attention** (D-Net) and comparative ranking and matching networks (R-Net and M-Net). D-Net extracts semantic lines by exploiting rich contextual information. To this end, we design the mirror attention module. Then, through pairwise comparisons of extracted semantic lines, we iteratively select the most semantic line and remove redundant ones overlapping with the selected one. For the pairwise comparisons, we develop R-Net and M-Net in the Siamese architecture. Experiments demonstrate that the proposed algorithm outperforms the conventional semantic line detector significantly. Moreover, we apply the proposed algorithm to detect two important kinds of semantic lines successfully: dominant parallel lines and reflection symmetry axes. Our codes are available at <https://github.com/dongkwonjin/Semantic-Line-DRM>.

**Keywords:** Semantic lines · Line detection · Attention · Ranking · Matching

## 1 Introduction

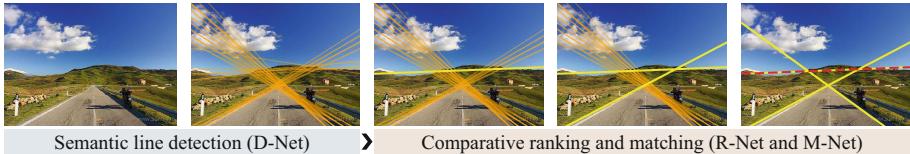
A *semantic line* [28] can be roughly defined as a dominant line, separating different semantic regions in a scene, which is reasonably approximated by an end-to-end straight line, as exemplified in Fig. 1.



**Fig. 1.** Examples of various semantic lines.

---

**Electronic supplementary material** The online version of this chapter ([https://doi.org/10.1007/978-3-030-58565-5\\_8](https://doi.org/10.1007/978-3-030-58565-5_8)) contains supplementary material, which is available to authorized users.



**Fig. 2.** Illustration of the proposed DRM algorithm: First, D-Net extracts semantic lines (orange). Second, R-Net selects the most semantic line (yellow). Third, M-Net removes redundant lines overlapping with the selected one. The second and third steps are iteratively applied. In this example, three semantic lines are selected. The first one is called the primary semantic line (dashed red). (Color figure online)

Semantic lines are essential components in high-level image understanding [15, 19, 20, 29, 55, 57]. In photography, photographic composition rules, such as horizontal, diagonal, and symmetric ones, are described by semantic lines [15, 29]. Under perspective projection, dominant parallel lines in the 3D world are projected to semantic lines in 2D images, intersecting at vanishing points and conveying depth impressions [57]. Also, in autonomous driving systems [19, 20], the boundaries of road lanes, sidewalks, or crosswalks are important semantic lines. However, it is difficult to detect semantic lines, which are often unobvious and implied by complex boundaries of semantic regions.

Although many techniques have been developed to detect lines by exploiting low-level cues [3, 11, 34, 45] or deep features [23, 49, 56], they extract many short (possibly noisy) line segments or rather obvious lines in man-made environments. Also, several attempts [27, 48, 52] have been made to detect unobvious horizon lines. However, horizons are just a specific type of semantic lines. Recently, SLNet, which is a general semantic line detector, was proposed in [28]. Although SLNet provides promising results, it tends to detect many redundant lines near the boundaries of semantic regions.

In this paper, we propose a novel semantic line detection algorithm, called DRM, which consists of three networks: detection network with mirror attention (D-Net) and comparative ranking and matching networks (R-Net and M-Net). In Fig. 2, D-Net first extracts semantic lines by classifying and regressing candidate lines. For effective detection, we design the mirror attention module and the region pooling layer in D-Net. Then, by comparing semantic lines in pairs, R-Net selects the most semantic line and M-Net removes redundant lines overlapping with the selected one. This comparative ranking and matching process is performed iteratively. In Fig. 2, three iterations are performed to yield three semantic lines. Experimental results demonstrate that the proposed DRM algorithm outperforms the conventional SLNet [28] significantly.

This work has the following major contributions:

- We develop D-Net to detect semantic lines, in which the mirror attention module and the region pooling layer extract discriminate features effectively.
- We propose two Siamese networks, R-Net and M-Net, for pairwise ranking and matching of semantic lines.

- We construct a challenging dataset (SEL\_Hard) of semantic lines, which are highly implied in cluttered scenes.<sup>1</sup>
- We also apply the proposed algorithm to two important line detection tasks: dominant parallel lines and reflection symmetry axes.

## 2 Related Work

### 2.1 Line Detection

Lines are geometrically important cues to describe the layouts or structural information of images. In line segment detection [3, 11, 34, 45], many short line segments are detected using low-level cues (*e.g.* image gradients). However, this approach may not discriminate meaningful lines from noisy ones. To utilize higher-level cues, deep learning methods have been proposed [23, 40, 49, 56]. In [23], two networks were used to predict, respectively, a line heat map and junctions in man-made environments. Then, the wireframe, summarizing the scene, was obtained by connecting the junctions based on the heat map. In [56], a network verified whether a candidate line was salient or not, where the candidate was also generated by connecting two junctions. In [49], the line segment detection was posed as the dual problem of region coloring to address local ambiguity and class imbalance. In [40], a network was trained to yield the coordinates of a bounding box, whose diagonal was the resultant line segment. However, these methods [23, 40, 49, 56] detect rather obvious lines in man-made environments.

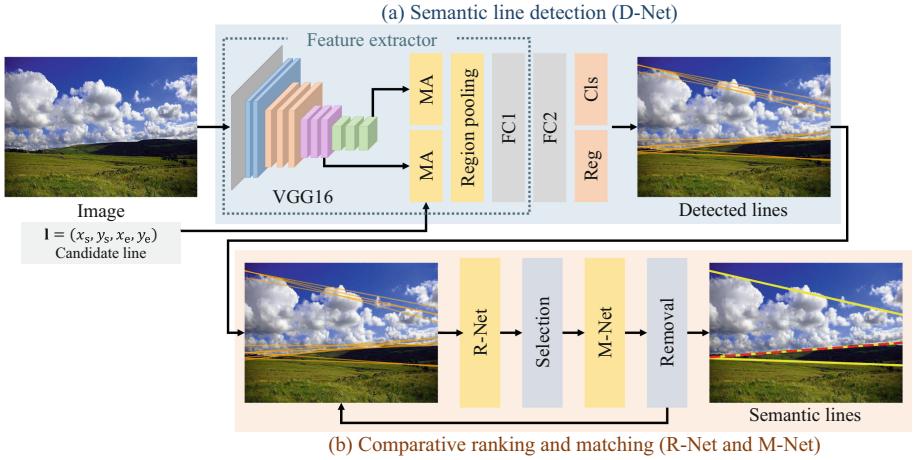
Meanwhile, several methods [12, 28, 48, 52] have been developed to detect implied lines. In [48], horizon lines were directly estimated by CNNs, without requiring geometric constraints. In [52], horizons were detected similarly to [48], but their locations were refined by exploiting vanishing points. In [12], soft labels of horizon line parameters were used to train the regression network. In [28], the first semantic line detector was proposed, which can detect general, semantically meaningful lines. Semantic lines, located near the boundaries of semantic regions, represent the layout and composition of an image, even when the boundaries are not straight lines.

### 2.2 Attention Mechanisms in CNNs

Human visual system pays more attention to salient parts of a scene for efficiency [10, 24]. Similarly, attempts have been made to bias the processing resource of a neural network towards more informative parts of input data [4, 43, 50, 51]. Also, attention mechanisms have been developed to improve the representation power of convolutional layers in CNN-based vision tasks [22, 35, 46, 47, 53]. In [46], Wang *et al.* adopted an encoder-decoder structure to obtain a pixel-wise attention mask of a convolutional feature map. In [22], to address the interdependencies of filter responses, Hu *et al.* used the average-pooled feature at each channel to compute the channel-wise attention. In [35, 47], this channel-wise

---

<sup>1</sup> SEL\_Hard is available at <https://github.com/dongkwonjin/Semantic-Line-DRM>.



**Fig. 3.** An overview of the proposed algorithm: (a) D-Net detects semantic lines, by classifying and regressing candidate lines, based on mirror attention (MA). (b) R-Net selects the most meaningful semantic line and M-Net removes redundant lines alternately through pairwise comparisons.

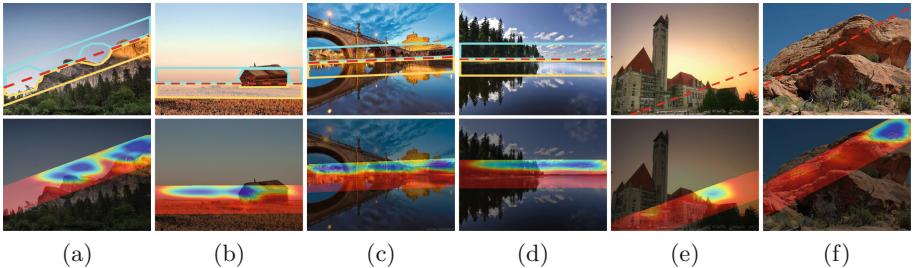
attention module has been modified to obtain both spatial and channel-wise attention. In [25], multiple attention maps were obtained from intermediate convolutional layers, and then the ensemble of those maps was applied to the last layer. In [53], Zhao and Wu applied spatial attention to lower layers to focus on local details and channel-wise attention to higher layers to capture contextual cues.

### 2.3 Metric Learning and Order Learning

Metric learning [38, 39] constructs an appropriate feature embedding space, where similar objects are located tightly while dissimilar objects are far from one another. In contrast, in order learning [30], embedded features are ordered according to the ranks or priorities of objects. Both the similarity and order relationships depend on target applications and are implicitly defined by user-provided examples. Accordingly, the learned metric is useful for matching similar objects, *e.g.*, in image retrieval [17, 21], person re-identification [7, 13], and few-shot learning [41, 44]. On the other hand, the learned order can be used to rank or sort objects, as done in image quality assessment [26, 31], object detection [37, 42], and age estimation [5, 6].

## 3 Proposed Algorithm

We propose a novel semantic line detection algorithm, called DRM, which is composed of three networks: D-Net, R-Net, and M-Net. Figure 3 is an overview



**Fig. 4.** [Top] A semantic (or candidate) line is shown in red, while two regions producing the line are in cyan and yellow. [Bottom] The attention mask is color-coded: red and blue depict big and small values. Note that the two regions are semantically different from each other in (a) and (b) and symmetric in (c) and (d). In (e) and (f), the candidate lines are not semantic. (Color figure online)

of the proposed DRM algorithm. First, we generate candidate lines by connecting two points, uniformly sampled on image boundaries [28]. Second, D-Net extracts semantic lines by classifying and regressing the candidate lines. For discriminative feature extraction, we design the mirror attention module and the region pooling layer. Third, through pairwise comparisons, we iteratively select the most meaningful semantic line and remove the other semantic lines overlapping with the selected one. For this purpose, we develop R-Net and M-Net in the Siamese architecture.

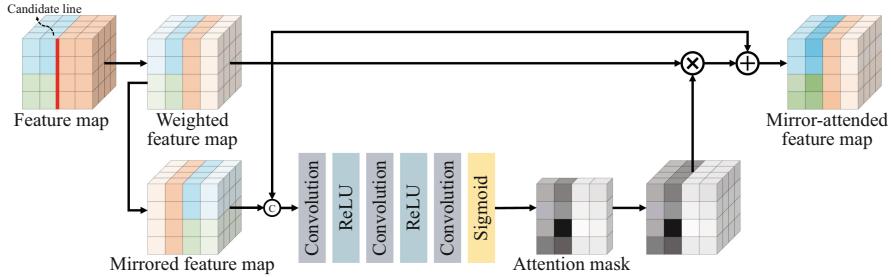
### 3.1 D-Net: Semantic Line Detection with Mirror Attention

**Mirror Attention:** A semantic line separates a region into two distinct sub-regions. Those two regions can be semantically different as in Fig. 4(a), (b), or symmetric around the line as in Fig. 4(c), (d). In the former case, if a region is mirrored along the line, it should contain quite different objects from the other region. In the latter case, it should be almost identical with the other region. To summarize, the line is semantic because of the mirrored dissimilarity (heterogeneity of two regions) or mirrored similarity (symmetry). Based on this observation, we develop the mirror attention module in Fig. 5. Given a feature map, the mirror attention module generates an attention mask, which is then used to reweight the feature map to make it more discriminative.

We apply the mirror attention module to a convolutional feature map  $X = [X^1, X^2, \dots, X^C] \in \mathbb{R}^{H \times W \times C}$ , where  $H$ ,  $W$ , and  $C$  denote the height, the width, and the number of channels. Since pixels near a candidate line are more relevant for semantic line detection, we first obtain a weighted feature map  $Y = [Y^1, Y^2, \dots, Y^C] \in \mathbb{R}^{H \times W \times C}$ , where  $X^c(k)$  for pixel  $k$  is weighted by

$$Y^c(k) = \omega(d_k) \times X^c(k). \quad (1)$$

Here,  $d_k$  is the distance of pixel  $k$  from the candidate line and  $\omega(\cdot)$  is the Gaussian weighting function.



**Fig. 5.** Illustration of the mirror attention process.

To analyze the mirror relationships around the candidate line, we obtain the mirrored feature map  $\tilde{Y}$  by flipping  $Y$  across the line. If a flipped pixel is outside the feature map, it is set to zero. Then, we concatenate  $Y$  and  $\tilde{Y}$  and obtain an initial mask  $A_0 \in \mathbb{R}^{H \times W}$  by

$$A_0 = f_0([Y, \tilde{Y}]) \quad (2)$$

where  $f_0$  is a convolutional layer using a single filter of size  $n \times n \times 2C$ . We set  $n$  to 3 empirically. To increase the receptive field and capture the semantics from a wider region, we use two more convolution layers to yield  $A_2 = f_2(f_1(A_0))$ , where  $f_1$  or  $f_2$  uses a single filter of size  $(2n+1) \times (2n+1) \times 1$ . Finally, we obtain the attention mask  $A = \sigma(A_2)$  where  $\sigma(\cdot)$  is the sigmoid activation function.

We apply the mirror attention module to two deep layers of D-Net, as shown in Fig. 3. The attention mask may over-suppress the values in the weighted feature map  $Y$ . To prevent this, we adopt the residual attention scheme [46]. More specifically, we obtain the attended feature map  $Y_{\text{att}} = [Y_{\text{att}}^1, Y_{\text{att}}^2, \dots, Y_{\text{att}}^C]$ , where  $Y_{\text{att}}^c$  is given by

$$Y_{\text{att}}^c = (1 + A) \otimes Y^c \quad (3)$$

for each  $1 \leq c \leq C$ .

Figure 4 shows examples of attention masks. In Fig. 4(a)–(d), there are roughly two semantic regions around the candidate line. We see that one region is attended with small weights, while the other with big weights. Thus, the feature difference between the two regions is emphasized, facilitating the semantic line detection. Note that the mirror attention module is trained in an end-to-end manner such that emphasizing masks are generated in both cases of mirrored dissimilarity (Fig. 4(a) or (b)) and mirrored similarity (Fig. 4(c) or (d)). On the other hand, in Fig. 4(e) or (f), a less informative mask is generated because the candidate line is not semantic.

**Region Pooling:** Whereas the conventional algorithm [28] uses a line pooling layer, we design a region pooling layer to extract more discriminated features from the mirror-attended feature map  $Y_{\text{att}}$ . We set two adjacent regions  $\mathcal{U}$  and  $\mathcal{V}$  along the candidate line, which contain pixels whose distances from the line are

less than a threshold, respectively. Then, we aggregate the regional information of  $\mathcal{U}$  and  $\mathcal{V}$  into  $\mathbf{u}$  and  $\mathbf{v} \in \mathbb{R}^C$ ;

$$\mathbf{u} = \frac{1}{|\mathcal{U}|} \sum_{k \in \mathcal{U}} Y_{\text{att}}(k) \quad \text{and} \quad \mathbf{v} = \frac{1}{|\mathcal{V}|} \sum_{k \in \mathcal{V}} Y_{\text{att}}(k). \quad (4)$$

Then,  $\mathbf{u}$  and  $\mathbf{v}$  are concatenated to form the feature vector of the candidate line.

**D-Net Architecture:** We plug the proposed mirror attention module and region pooling layer into the classification-regression framework of [28]. Hence, D-Net takes an image and a candidate line, parameterized by  $\mathbf{l} = (x_s, y_s, x_e, y_e)$ , and yields classification and regression results. Figure 3(a) shows its architecture. We use the 13 convolution layers of VGG16 [36] as the backbone, and implement two mirror attention modules after Conv10 and Conv13, respectively. From each mirror-attended feature map, the region pooling layer extracts the feature vector. The two vectors are concatenated and fed into fully connected layers FC1 and FC2. Finally, D-Net branches into two parallel output layers: one for classifying the candidate line (Cls), and the other for computing regression offsets for the line parameters (Reg). Cls computes the softmax vector  $\mathbf{p} = (p, q)$ , where  $p$  is the probability that the candidate line  $\mathbf{l}$  is semantic. Reg outputs a line offset  $\Delta\mathbf{l}$ . When  $p > 0.5$ , D-Net declares that the regressed line  $\mathbf{l} + \Delta\mathbf{l}$  is semantic.

To train D-Net, when a candidate line is annotated by  $\bar{\mathbf{p}}$  and  $\Delta\bar{\mathbf{l}}$ , we minimize the loss

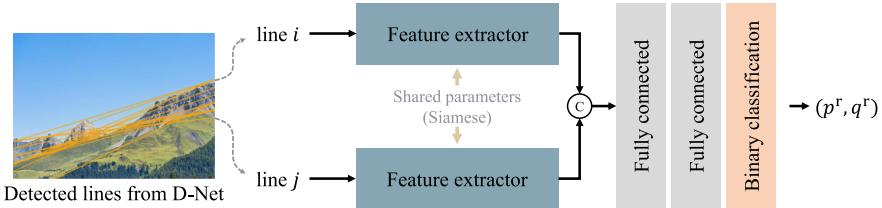
$$L(\mathbf{p}, \bar{\mathbf{p}}, \Delta\mathbf{l}, \Delta\bar{\mathbf{l}}) = L_{\text{cls}}(\mathbf{p}, \bar{\mathbf{p}}) + \lambda L_{\text{reg}}(\Delta\mathbf{l}, \Delta\bar{\mathbf{l}}) \quad (5)$$

where  $L_{\text{cls}}(\mathbf{p}, \bar{\mathbf{p}})$  and  $L_{\text{reg}}(\Delta\mathbf{l}, \Delta\bar{\mathbf{l}})$  are the classification loss and the regression loss, respectively, and  $\lambda$  is a balancing parameter.  $L_{\text{cls}}$  is the cross-entropy loss over the two classes (semantic and non-semantic).  $L_{\text{reg}}(\Delta\mathbf{l}, \Delta\bar{\mathbf{l}}) = \eta(\Delta\mathbf{l} - \Delta\bar{\mathbf{l}})$ , where  $\eta$  is the smooth  $L_1$  loss [18].

### 3.2 R-Net and M-Net: Comparative Ranking and Matching

Note that D-Net detects many semantic lines from densely sampled candidate lines, as illustrated in Fig. 2. Since each candidate is tested independently, semantically identical lines are detected closely. For example, in Fig. 2, there are three groups, each of which contains semantically identical lines. From each group, we select the most reliable line, while removing the other redundant ones. To this end, we develop the **comparative ranking and matching networks**, referred to as R-Net and M-Net, respectively. Given a pair of semantic lines, R-Net finds which one is more reliable and M-Net determines whether they are semantically identical or not. Thus, R-Net is related to *priority* in order learning [30], while M-Net is to *similarity* in metric learning [38, 39].

Both R-Net and M-Net are implemented as binary classifiers in the Siamese architecture. Figure 6 shows R-Net, which yields a softmax probability vector  $\mathbf{p}^r = (p^r, q^r)$ . Here,  $p^r$  or  $q^r$  is the probability that line  $i$  is more reliable or less reliable than line  $j$ , respectively. Since D-Net is well-trained for semantic line



**Fig. 6.** Siamese architecture for R-Net: Given a pair of detected lines from D-Net, R-Net decides whether one line is more reliable or less reliable than the other.

detection, we truncate it before the FC2 layer and use it as the feature extractor of R-Net. Then, the features of the two lines are concatenated, propagated into two fully connected layers, and categorized into one of the binary classes. The cross-entropy loss is used to train R-Net. Also, M-Net is implemented in the same way as R-Net, except that it yields a softmax vector  $\mathbf{p}^m = (p^m, q^m)$ , where  $p^m$  or  $q^m$  is the probability that the two lines are semantically identical or not, respectively.

Using R-Net and M-Net, we perform the selection and removal of semantic lines alternately. At step  $t$ , we measure the reliability of each semantic line. Specifically, the reliability  $r_i$  of semantic line  $i$  is defined as

$$r_i = \sum_{j=1, j \neq i}^{N_t} p_{ij}^r \quad (6)$$

where  $N_t$  is the number of available semantic lines at step  $t$ , and  $p_{ij}^r$  is the probability that line  $i$  is more reliable than line  $j$ . Then, we select the most reliable line  $i^*$  by

$$i^* = \arg \max_i r_i. \quad (7)$$

We then remove the lines that are semantically identical with line  $i^*$ . Specifically, line  $j$  is removed if the matching probability  $p_{i^*j}^m$  from M-Net is higher than 0.5. We iteratively perform the alternate selection and removal until  $N_t = 0$ . In the example of Fig. 2, three iterations are performed to select the three resultant lines. The firstly selected line is called the primary line.

We configure the training data for R-Net and M-Net as follows. Note that a detected semantic line is declared to be correct if its mean intersection over union (mIoU) ratio with the ground-truth is higher than 0.85 [28]. After training D-Net on the SEL dataset [28], we use the correctly detected semantic lines in the training images to train R-Net and M-Net. For R-Net, a ground-truth semantic line and one of its detection results are used as an input pair. To encode the one-hot vector  $\bar{\mathbf{p}}^r = (\bar{p}^r, \bar{q}^r)$ , the ground-truth line is used as the more reliable one than the detection result. For M-Net, we use two detected lines as input. To encode  $\bar{\mathbf{p}}^m = (\bar{p}^m, \bar{q}^m)$ , if the two lines correspond to the same ground-truth, they are regarded as semantically identical.

## 4 Experimental Results

### 4.1 Datasets

**SEL:** The semantic line (SEL) dataset [28] contains 1,750 outdoor images in total, which are split into 1,575 training and 175 testing images. Each semantic line is annotated by the coordinates of the two end-points on an image boundary. If an image has a single dominant line, it is set as the ground truth primary semantic line. If an image has multiple semantic lines, the line with the best rank by human annotators is set as the ground-truth primary line, and the others as additional ground-truth semantic lines. In SEL, 61% of the images contain multiple semantic lines.

**SEL\_Hard:** In addition to the SEL dataset, we construct a more challenging test dataset, called SEL\_Hard. Its semantic lines are more implied (or less obvious), are more severely occluded, and are in more cluttered scenes. We collect 300 images from the ADE20K image segmentation dataset [54], manually annotate semantic lines, and then also select primary lines. Notice that SEL\_Hard is constructed for testing semantic line detectors and is not used for training them. The supplemental document describes the annotation process in detail and provides example images.

### 4.2 Semantic Line Detection Results

We assess primary and multiple semantic line detection performances on the SEL and SEL\_Hard datasets.

We measure the accuracy for primary semantic line detection and the precision and recall rates for multiple semantic line detection, based on the mIoU metric [28]. A semantic line is regarded as correctly detected if its mIoU score with the ground-truth is greater than a threshold  $\tau$ . Then, the accuracy of the primary semantic line detection is defined as

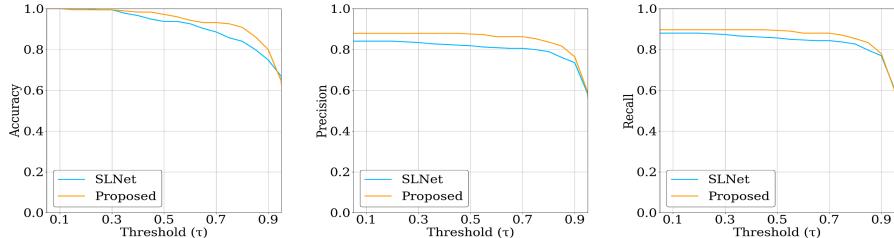
$$\text{Accuracy} = \frac{N_c}{N} \quad (8)$$

where  $N_c$  is the number of the test images whose primary semantic lines are correctly detected, and  $N$  is the number of all test images. For the multiple semantic line detection, the precision and the recall are computed by

$$\text{Precision} = \frac{N_l}{N_l + N_e}, \quad \text{Recall} = \frac{N_l}{N_l + N_m} \quad (9)$$

where  $N_l$  is the number of correctly detected semantic lines,  $N_e$  is the number of false positives, and  $N_m$  is the number of false negatives.

Figure 7 compares the accuracy, precision, and recall curves of the proposed DRM algorithm and the conventional SLNet algorithm [28] on the SEL dataset. The proposed DRM outperforms SLNet in all three curves in the entire range of



**Fig. 7.** Comparison of the accuracy, precision, and recall curves of the proposed DRM and the conventional SLNet in terms of the threshold  $\tau$  on the SEL dataset.

**Table 1.** Comparison of the AUC scores (%) on the SEL and SEL\_Hard datasets.

	SEL			SEL_Hard		
	AUC_A	AUC_P	AUC_R	AUC_A	AUC_P	AUC_R
SLNet	92.00	80.44	83.50	73.59	74.22	70.68
Proposed DRM	94.54	85.44	87.16	80.68	87.19	77.69

the threshold  $\tau$ . Table 1 reports the area under curve (AUC) performances of the accuracy, precision, and recall curves in Fig. 7, which are denoted by AUC\_A, AUC\_P, and AUC\_R, respectively. DRM provides higher AUC\_A, AUC\_P, and AUC\_R than SLNet by 2.54, 5.00, and 3.66, respectively. Table 1 also compares the performances on SEL\_Hard. For this comparison as well, we use the same DRM and SLNet, which are trained using the training images in the SEL dataset. Since SEL\_Hard consists of more challenging images, the performances are lower than those on SEL. Nevertheless, on SEL\_Hard, DRM outperforms SLNet by significant margins 7.09, 12.97, and 7.01 in terms of AUC\_A, AUC\_P, and AUC\_R, respectively.

Figure 8 compares detection results qualitatively. Compared to SLNet, the proposed DRM detects implied, as well as obvious, semantic lines more precisely. Also, DRM suppresses redundant lines more effectively. More detection results are available in the supplemental document.

### 4.3 Ablation Studies

We conduct ablation studies to analyze the efficacy of the proposed D-Net, R-Net, and M-Net on the SEL dataset.

**Efficacy of Mirror Attention Model:** Table 2 compares the performances of several ablated methods. First, to demonstrate the impacts of the mirror attention module in D-Net, we do not use the comparative ranking and matching (R-Net and M-Net). Instead, we adopt the non-maximum suppression (NMS) scheme in [28], which removes overlapped semantic lines based on low-level edge features. Method i uses no attention module. Method ii uses the attention module



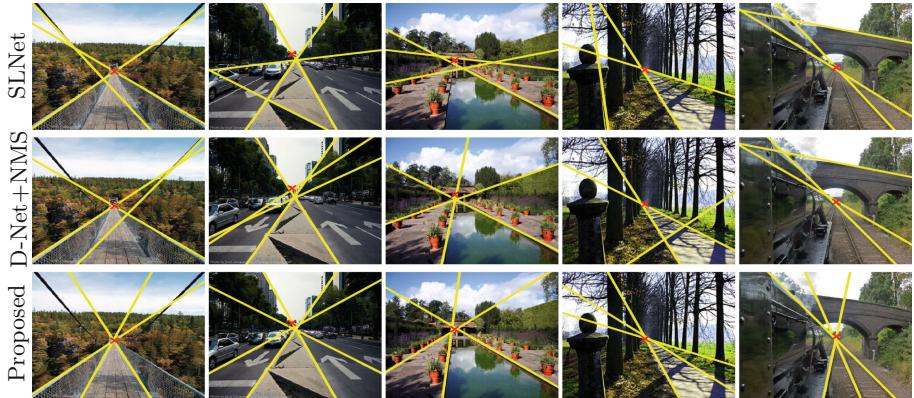
**Fig. 8.** Comparison of semantic line detection results. The left two images are from SEL, and the other three from SEL.Hard. Primary and multiple semantic lines are depicted in dashed red and solid yellow, respectively. (Color figure online)

**Table 2.** The ablation studies in terms of the mirror attention module and comparative ranking and matching. AUC scores (%) of primary and multiple semantic line detection are compared on the SEL dataset.

		AUC_A	AUC_P	AUC_R
i	D-Net(without attention)+NMS	92.48	83.37	84.76
ii	D-Net(with attention, no flipped feature map)+NMS	93.24	81.33	84.86
iii	D-Net(with spatial-channel attention)+NMS	92.85	81.74	85.84
iv	D-Net(with mirror attention)+NMS	93.38	83.98	86.23
v	D-Net(with mirror attention)+R-Net+M-Net	94.54	85.44	87.16

in Fig. 5 but without concatenating a flipped feature map. Method iii replaces the mirror attention module with the spatial-channel attention in [47]. As compared with no attention in i, the two attention schemes in ii and iii improve the accuracy and recall scores but lower the precision score. On the contrary, the proposed mirror attention model in iv improves all three scores and also outperforms the two alternative schemes in ii and iii. Also, by comparing iv with ii, we see that the mirroring of feature maps across semantic lines is effective for emphasizing informative regions. More specifically, the mirroring improves AUC\_A, AUC\_P, and AUC\_R by 0.14, 2.65, and 1.37, respectively.

**Efficacy of R-Net and M-Net:** By comparing methods iv and v, we see that the proposed DRM algorithm provides 1.16, 1.46 and 0.93 higher AUC\_A, AUC\_P, and AUC\_R scores, by employing R-Net and M-Net instead of NMS. This indicates that the proposed comparative ranking and matching is an effective approach to select reliable semantic lines and remove redundant ones.



**Fig. 9.** Detection results of dominant parallel lines. For reference, the ground-truth vanishing points are depicted by red cross symbols. (Color figure online)

**Table 3.** AUC-A scores (%) in the dominant parallel line detection, according to the number  $K$  of detected lines.

$K$	1	2	3	4	5
SLNet [28]	46.83	41.73	38.58	37.22	36.44
D-Net+NMS	52.62	48.93	46.24	44.71	43.30
Proposed	56.31	54.36	52.42	51.17	50.72

## 5 Applications

We apply the proposed DRM algorithm to detect two kinds of semantically important lines: dominant parallel lines and reflection symmetry axes.

### 5.1 Dominant Parallel Lines

When projected onto a 2D image, dominant parallel lines in the 3D world convey depth impressions, intersecting at a vanishing point (VP) [57]. Despite researches on the VP detection [55, 57], a single VP is less informative for conveying the depth information than those projected parallel lines are. Hence, we apply the proposed DRM algorithm to detect dominant parallel lines in a 2D image. We first extract semantic lines using D-Net. To this end, D-Net is trained to detect semantic lines passing through VPs. Next, using R-Net, the primary semantic line is selected. In this application, M-Net is trained to declare a pair of lines, which are parallel in the 3D space, as ‘matched.’ Then, we repeatedly select the semantic line yielding the highest ‘matched’ score to the primary semantic line. To avoid overlapping, we remove the semantic lines whose mIoUs with an already selected one are greater than 0.85 after each selection.

We assess the proposed algorithm on the AVA landscape dataset [57]. It contains 2,275 training and 275 test landscape images. For each image, a dominant

**Table 4.** Comparison of AA scores (%) for the dominant VP detection.

	AA1°	AA2°	AA10°
Zhou <i>et al.</i> [57]	18.5	33.0	60.0
NeurVPS [55]	<b>19.6</b>	<b>35.9</b>	65.9
Proposed	8.6	22.9	<b>68.3</b>

VP and two dominant parallel lines are annotated. We declare a detected line as correct, when its distance to the ground-truth VP is smaller than a threshold. Then, we compute AUC\_A scores by varying the threshold. Table 3 compares the AUC\_A scores of the conventional SLNet [28], ‘D-Net+NMS,’ and the proposed algorithm according to the number  $K$  of detected lines in each image. As more lines are selected in an image, the accuracy score is lowered. However, for every  $K$ , D-Net+NMS outperforms SLNet, which indicates that D-Net detects dominant parallel lines more precisely. Moreover, by employing R-Net and M-Net, the proposed algorithm further improves the performances. Figure 9 shows that the proposed algorithm detects dominant lines that pass through VPs accurately.

Next, we detect a VP as the intersecting point of the first two selected lines. Table 4 compares this VP detection scheme with the existing methods[55,57]. Angle accuracies AA1°, AA2°, AA10° are used as the performance metrics, as done in [55]. Two performances of NeurVPS are reported in [55]. Table 4 includes their accuracies when the same training data as the proposed algorithm are used. Note that the proposed algorithm focuses on the detection of dominant lines and provides VPs as side results. In contrast, the existing methods are tailored for the VP detection. Therefore, when the tolerance angles are small ( $1^\circ$  or  $2^\circ$ ), the proposed algorithm yields poorer accuracies than the existing methods. However, when the tolerance angle is  $10^\circ$ , the proposed algorithm outperforms them. This indicates that the proposed algorithm can detect rough locations of VPs with a high recall rate, although it lacks the precision of the existing methods.

## 5.2 Reflection Symmetry Axes

Reflection symmetry is a common, but important visual property in various scenes, such as landscapes and man-made structures [32]. However, since reflection symmetry axes are often highly implied or even invisible, their detection should exploit semantic regions around the axes. Accordingly, we train the proposed algorithm to detect the reflection symmetry axis of an image as the primary semantic line. More specifically, we train D-Net to extract symmetry axes as semantic lines, and R-Net to prioritize those axes among the detected lines. We empirically find that the Gaussian weighting in (1) and the residual attention in (3) are less effective in this task. Hence, we exclude those operations from the mirror attention module in Fig. 5. Then, we extract semantic lines using D-Net, and choose the most reliable one as the symmetry axis via (7) using R-Net.



**Fig. 10.** Detection results of symmetry axes. The ground-truth axes are in red, the detection results of Loy and Eklundh [33] are in green, and those of the proposed algorithm are in yellow. (Color figure online)

**Table 5.** Comparison of AUC\_A scores (%) of the symmetry axis detection.

	ICCV	NYU	SYM_Hard
Cicconet <i>et al.</i> [8]	80.80	82.85	68.99
Elawady <i>et al.</i> [14]	87.24	83.83	73.90
Cicconet <i>et al.</i> [9]	87.38	87.64	81.04
Loy & Eklundh [33]	89.77	90.85	81.99
Proposed	<b>90.60</b>	<b>92.78</b>	<b>84.73</b>

We test the proposed algorithm on three datasets: ICCV [16], NYU [9], and SYM\_Hard. ICCV provides 100 training and 96 test images, and NYU contains 176 test images. In SYM\_Hard, we collect 45 images from photo sharing websites [1, 2], each of which includes a reflection symmetry axis. The axis is implied, and the neighboring regions are not exactly symmetric. Thus, its detection is challenging. Since the proposed algorithm detects symmetry axes as primary lines, we compare the proposed algorithm with the existing methods [8, 9, 14, 33] using the AUC\_A metric. We train the proposed algorithm using the ICCV training images and use it to assess the performances on all three datasets. Table 5 compares the results. On ICCV, NYU, and SYM\_Hard, the proposed algorithm outperforms the existing methods by at least 0.83, 1.93, and 2.74, respectively. Figure 10 compares detection results of the proposed algorithm with those of Loy and Eklundh [33]. The proposed algorithm detects symmetry axes more robustly. More experimental results are available in the supplemental document.

## 6 Conclusions

We proposed a novel semantic line detector using D-Net, R-Net, and M-Net. First, D-Net extracts semantic lines using the mirror attention module. Second, R-Net selects the most semantic line through ranking. Third, M-Net removes redundant lines overlapping with the selected one through matching. The second and third steps are alternately performed to yield reliable semantic lines as output. Experimental results demonstrated that the proposed DRM algorithm outperforms the conventional SLNet significantly. Moreover, it was shown that the proposed algorithm can be applied to successfully detect two important kinds of semantic lines: dominant parallel lines and reflection symmetry axes.

**Acknowledgement.** This work was supported in part by the Agency for Defense Development (ADD) and Defense Acquisition Program Administration (DAPA) of Korea under grant UC160016FD and in part by the National Research Foundation of Korea (NRF) through the Korea Government (MSIP) under grant NRF-2018R1A2B3003896.

## References

1. Flickr. <https://www.flickr.com/>
2. GoogleImage. <https://www.google.com/>
3. Akinlar, C., Topal, C.: EDLines: a real-time line segment detector with a false detection control. *Pattern Recogn. Lett.* **32**(13), 1633–1642 (2011)
4. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Proceedings of ICLR (2015)
5. Chen, S., Zhang, C., Dong, M.: Deep age estimation: from classification to ranking. *IEEE Trans. Multimedia* **20**(8), 2209–2222 (2017)
6. Chen, S., Zhang, C., Dong, M., Le, J., Rao, M.: Using ranking-CNN for age estimation. In: Proceedings of IEEE CVPR (2017)
7. Chu, R., Sun, Y., Li, Y., Liu, Z., Zhang, C., Wei, Y.: Vehicle re-identification with viewpoint-aware metric learning. In: Proceedings of IEEE ICCV (2019)
8. Cicconet, M., Birodkar, V., Lund, M., Werman, M., Geiger, D.: A convolutional approach to reflection symmetry. *Pattern Recogn. Lett.* **95**, 44–50 (2017)
9. Cicconet, M., Hildebrand, D.G., Elliott, H.: Finding mirror symmetry via registration and optimal symmetric pairwise assignment of curves: algorithm and results. In: Proceedings of IEEE ICCV Workshops (2017)
10. Corbetta, M., Shulman, G.L.: Control of goal-directed and stimulus-driven attention in the brain. *Nat. Rev. Neurosci.* **3**(3), 201–215 (2002)
11. Desolneux, A., Moisan, L., Morel, J.M.: Meaningful alignments. *Int. J. Comput. Vis.* **40**(1), 7–23 (2000)
12. Diaz, R., Marathe, A.: Soft labels for ordinal regression. In: Proceedings of IEEE CVPR (2019)
13. Ding, S., Lin, L., Wang, G., Chao, H.: Deep feature learning with relative distance comparison for person re-identification. *Pattern Recogn.* **48**(10), 2993–3003 (2015)
14. Elawady, M., Ducottet, C., Alata, O., Barat, C., Colantoni, P.: Wavelet-based reflection symmetry detection via textural and color histograms. In: Proceedings of IEEE ICCV Workshops (2017)
15. Freeman, M.: *The Photographer’s Eye: Composition and Design for Better Digital Photos*. Focal Press, Waltham (2007)
16. Funk, C., et al.: 2017 ICCV Challenge: Detecting symmetry in the wild. In: Proceedings of IEEE ICCV (2017)
17. Gao, X., Hoi, S.C., Zhang, Y., Wan, J., Li, J.: SOML: sparse online metric learning with application to image retrieval. In: Proceedings of AAAI (2014)
18. Girshick, R.: Fast R-CNN. In: Proceedings of IEEE ICCV (2015)
19. Guo, C., Yamabe, T., Mita, S.: Robust road boundary estimation for intelligent vehicles in challenging scenarios based on a semantic graph. In: Intelligent Vehicles Symposium. IEEE (2012)
20. Hillel, A.B., Lerner, R., Levi, D., Raz, G.: Recent progress in road and lane detection: a survey. *Mach. Vis. Appl.* **25**(3), 727–745 (2014)

21. Hoi, S.C., Liu, W., Chang, S.F.: Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Trans. Multimed. Comput. Commun. Appl.* **6**(3), 1–26 (2010)
22. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of IEEE CVPR (2018)
23. Huang, K., Wang, Y., Zhou, Z., Ding, T., Gao, S., Ma, Y.: Learning to parse wireframes in images of man-made environments. In: Proceedings of IEEE CVPR (2018)
24. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(11), 1254–1259 (1998)
25. Jetley, S., Lord, N.A., Lee, N., Torr, P.H.S.: Learn to pay attention. In: Proceedings of ICLR (2018)
26. Kong, S., Shen, X., Lin, Z., Mech, R., Fowlkes, C.: Photo aesthetics ranking network with attributes and content adaptation. In: Proceedings of ECCV (2016)
27. Koo, H.I., Cho, N.I.: Skew estimation of natural images based on a salient line detector. *J. Electron. Imaging* **22**(1), 013020 (2013)
28. Lee, J.T., Kim, H.U., Lee, C., Kim, C.S.: Semantic line detection and its applications. In: Proceedings of IEEE ICCV (2017)
29. Lee, J.T., Kim, H.U., Lee, C., Kim, C.S.: Photographic composition classification and dominant geometric element detection for outdoor scenes. *J. Vis. Commun. Image Represent.* **55**, 91–105 (2018)
30. Lim, K., Shin, N.H., Lee, Y.Y., Kim, C.S.: Order learning and its application to age estimation. In: Proceedings of ICLR (2020)
31. Liu, X., van de Weijer, J., Bagdanov, A.D.: RankIQA: learning from rankings for no-reference image quality assessment. In: Proceedings of IEEE ICCV (2017)
32. Liu, Y., Hel-Or, H., Kaplan, C.S., Van Gool, L.: Computational symmetry in computer vision and computer graphics. *Found. Trends Comput. Graph. Vision* **5**(1–2), 1–195 (2010)
33. Loy, G., Eklundh, J.O.: Detecting symmetry and symmetric constellations of features. In: Proceedings of ECCV (2006)
34. Matas, J., Galambos, C., Kittler, J.: Robust detection of lines using the progressive probabilistic hough transform. *Comput. Vis. Image Understand.* **78**(1), 119–137 (2000)
35. Park, J., Woo, S., Lee, J., Kweon, I.S.: BAM: bottleneck attention module. In: Proceedings of BMVC (2018)
36. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proceedings of ICLR (2015)
37. Singh, K.K., Lee, Y.J.: You reap what you sow: using videos to generate high precision object proposals for weakly-supervised object detection. In: Proceedings of IEEE CVPR (2019)
38. Sohn, K.: Improved deep metric learning with multi-class n-pair loss objective. In: Proceedings of NIPS (2016)
39. Song, O.H., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: Proceedings IEEE CVPR (2016)
40. Sun, Y., Han, X., Sun, K.: Sem-LSD: A learning-based semantic line segment detector. arXiv preprint [arXiv:1909.06591](https://arxiv.org/abs/1909.06591) (2019)
41. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: relation network for few-shot learning. In: Proceedings of IEEE CVPR (2018)
42. Tan, Z., Nie, X., Qian, Q., Li, N., Li, H.: Learning to rank proposals for object detection. In: Proceedings of IEEE ICCV (2019)

43. Vaswani, A., et al.: Attention is all you need. In: Proceedings of NIPS (2017)
44. Vinyals, O., Blundell, C., Lillicrap, T., kavukcuoglu, k., Wierstra, D.: Matching networks for one shot learning. In: Proceedings of NIPS (2016)
45. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: a fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(4), 722–732 (2008)
46. Wang, F., et al.: Residual attention network for image classification. In: Proceedings of IEEE CVPR (2017)
47. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: convolutional block attention module. In: Proceedings of ECCV (2018)
48. Workman, S., Zhai, M., Jacobs, N.: Horizon lines in the wild. In: Proceedings of BMVC (2016)
49. Xue, N., Bai, S., Wang, F., Xia, G.S., Wu, T., Zhang, L.: Learning attraction field representation for robust line segment detection. In: Proceedings of IEEE CVPR (2019)
50. Yu, A.W., et al.: QANet: combining local convolution with global self-attention for reading comprehension. In: Proceedings of ICLR (2018)
51. Zambaldi, V., et al.: Deep reinforcement learning with relational inductive biases. In: Proceedings of ICLR (2019)
52. Zhai, M., Workman, S., Jacobs, N.: Detecting vanishing points using global image context in a non-manhattan world. In: Proceedings of IEEE CVPR (2016)
53. Zhao, T., Wu, X.: Pyramid feature attention network for saliency detection. In: Proceedings of IEEE CVPR (2019)
54. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ADE2020K dataset. In: Proceedings of IEEE CVPR (2017)
55. Zhou, Y., Qi, H., Huang, J., Ma, Y.: NeurVPS: neural vanishing point scanning via conic convolution. In: Proc. NIPS (2019)
56. Zhou, Y., Qi, H., Ma, Y.: End-to-end wireframe parsing. In: Proceedings of IEEE ICCV (2019)
57. Zhou, Z., Farhat, F., Wang, J.Z.: Detecting dominant vanishing points in natural scenes with application to composition-sensitive image retrieval. *IEEE Trans. Multimedia* **19**(12), 2651–2665 (2017)