2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Orthogonal Decomposition Network for Pixel-wise Binary Classification

Chang Liu[†], Fang Wan[†], Wei Ke[†],
Zhuowei Xiao[†‡], Yuan Yao[†], Xiaosong Zhang[†] and Qixiang Ye[†§*]

[†]University of Chinese Academy of Sciences, Beijing, China
[‡]Institute of Geology and Geophysics, Chinese Academy of Sciences
[§]Peng Cheng Laboratory, Shenzhen, Guangdong, China
{liuchang615,wanfang13,kewei11,yaoyuan17,zhangxiaosong18}@mails.ucas.ac.cn
xiaozhuowei@mail.iggcas.ac.cn, qxye@ucas.ac.cn

Abstract

The weight sharing scheme and spatial pooling operations in Convolutional Neural Networks (CNNs) introduce semantic correlation to neighboring pixels on feature maps and therefore deteriorate their pixel-wise classification performance. In this paper, we implement an Orthogonal Decomposition Unit (ODU) that transforms a convolutional feature map into orthogonal bases targeting at de-correlating neighboring pixels on convolutional features. In theory, complete orthogonal decomposition produces orthogonal bases which can perfectly reconstruct any binary mask (ground-truth). In practice, we further design incomplete orthogonal decomposition focusing on de-correlating local patches which balances the reconstruction performance and computational cost. Fully Convolutional Networks (FCNs) implemented with ODUs, referred to as Orthogonal Decomposition Networks (ODNs), learn de-correlated and complementary convolutional features and fuse such features in a pixel-wise selective manner. Over pixel-wise binary classification tasks for two-dimensional image processing, specifically skeleton detection, edge detection, and saliency detection, and one-dimensional keypoint detection, specifically S-wave arrival time detection for earthquake localization, ODNs consistently improves the state-of-the-arts with significant margins.

1. Introduction

Pixel-wise binary classification tasks, *e.g.*, skeleton detection, edge detection, and saliency detection, are fundamentally important for computer vision and pattern recognition. Skeleton is one of the most representative visual properties, describing objects with compact but informative curves. Such curves constitute a continuous decomposition of object shapes [25], providing valuable cues for both object representation and recognition. Edge can be converted into descriptive features and spatial constraints, which enforce object grouping [38], semantic segmentation [21], and object localization [13]. Saliency represents the most conspicuous and attractive regions in an image, and saliency detection serves as the first step to a variety of computer vision applications [8].

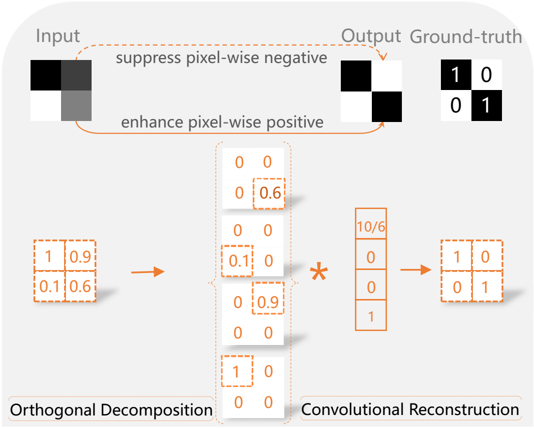

Figure 1: Orthogonal Decomposition Unit (ODU) transforms a convolutional feature map to orthogonal bases, which can be used to perfectly reconstruct complex binary masks (ground-truth) with convolutional reconstruction.

Pixel-wise binary classification tasks have the common goal about predicting a mask of interest given a color input image. In the deep learning era, the fully convolutional neural networks (FCNs) [27] have been widely applied to solve pixel-wise classification problems due to their end-to-end training manner and flexibility to the image size. Recent FCN-based approaches, *e.g.*, holistically-nested edge de-

[*] Corresponding author.

978-1-7281-3293-8/19/$31.00 ©2019 IEEE
DOI 10.1109/CVPR.2019.00622

tection (HED) and side-output residual network (SRN) [11] root in multi-layer feature fusion with the motivation that low-level features focus on details while high-level features are rich in semantics [11]. Linear Span Network (LSN) [19] uses linear span theory to de-correlate the convolutional feature channels and increase their independence.

The multi-layer feature fusion and channel-level de-correlation have been extensively explored. However, pixel-level de-correlation remains unsolved. The backbone CNNs used in pixel-wise binary classification approaches are usually pre-trained on image classification and have pixel-wise semantic correlation which caused by the weight sharing scheme and pooling operations notably deteriorates the network's capability about pixel-wise classification.

In this paper, we propose an Orthogonal Decomposition Unit (ODU) which transforms a convolutional feature map into orthogonal channels, Fig. 1, and targets at alleviating semantic correlation of neighboring pixels on convolutional the feature map. The ODU is based on a mathematical principle that any vector, $e.g.$, a binary ground-truth mask, can be perfectly reconstructed by a set of complete orthogonal bases (orthogonal feature channels). When fusing the decomposed feature maps with a convolutional operation, the convolutional weights can be solved independently, which endows ODU the capability of pixel-wise refinement over feature maps. Considering that neighboring pixels suffer semantic correlation more critical, we utilize incomplete orthogonal decomposition on local feature patches. Besides the de-correlation of local patches, incomplete orthogonal decomposition also decreases the computational cost compared with complete orthogonal decomposition.

The ODU can be added atop the output layer of FCNs and update them to Orthogonal Decomposition Networks (ODNs) which facilitate learning de-correlated and complementary features. In contrast to existing FCNs, $e.g.$, SRN [11], LSN [19], and deeply supervised short-connections (DSS) [8] that focus on learning channel-level complementary features, ODNs pursue pixel-level spatial de-correlation and feature complementarity in a more effective way.

The contributions of this paper include:

- A plug-and-play module named Orthogonal Decomposition Unit (ODU) is designed to partition neighboring pixels on a feature map into orthogonal channels that targets to alleviate the fundamental drawback of CNNs about pixel-level semantic correlation.
- With ODU plugged atop the output layer, successful fully convolutional networks, including VGG [33], HED [39], and SRN [11], are upgraded to Orthogonal Decomposition Networks (ODNs), which facilitate learning de-correlated and complementary features.
- With negligible computation cost, ODNs consistently improves the state-of-the-arts with significant margins on pixel-wise binary classification tasks for two-dimensional image processing, specifically skeleton detection, edge detection, and saliency detection, and one-dimensional keypoint detection, specifically S-wave arrival time detection for earthquake localization.

2. Related Work

In this section, we review approaches about pixel-wise binary classification and methods about learning complementary and/or de-correlated features with CNNs.

Pixel-wise binary classification. The early pixel-wise binary classification approaches rooted in hand-crafted image processing methods [3, 12, 18, 24], which used morphological operations to localize pixels of interest, $e.g.$, edge or skeleton pixels. Recently, learning based methods were proposed for pixel-wise binary classification. The multiple instance learning method [36] was used to learn a true skeleton pixel from a bag of pixels. The structured random forest [35] and subspace multiple instance learning [28] were employed to perform skeleton detection and localization.

With the rise of deep learning, researchers have formulated pixel-wise binary classification as an image-to-mask mapping problem and focusing on fusing the multi-layer convolutional features in an end-to-end manner.

HED [39] introduced deeply supervised side-output network to learn a pixel-wise classifier for edge detection. Multiscale deep features (MSD) [14] extracted and fused features from multiple convolutional layers for saliency detection, and leverages image segmentation to further boost the performance. Fusing scale-associated deep side-outputs (FSDS) [31] learned multi-scale skeleton representation given scale-associated ground-truth. SRN [11] and DSS [8] leveraged the side-output residual units to fit the errors between the object symmetry/skeleton ground-truth and the side-outputs of multiple convolutional layers. To improve the pixel-wise classification accuracy, a segment-wise spatial pooling were proposed for saliency detection [16].

It has been extensively explored that how to fuse multi-layer convolutional features to predict binary mask. Nevertheless, researchers barely investigate de-correlating neighboring pixels in a convolutional layer. With strong semantic correlation, CNN's capability for pixel-wise classification remains limited.

Convolutional feature de-correlation. From a broad view of convolutional feature de-correlation, canonical correlation analysis (CCA) [4, 34] and singular vector decomposition (SVD) [34] were implemented into CNNs with specially designed de-correlation layers. These approaches validated the feasibility of using a single de-correlation layer to push the network learning complementary features.

For object saliency detection, a super-pixel approach was combined with FCNs to reduce the correlation of deep pix-

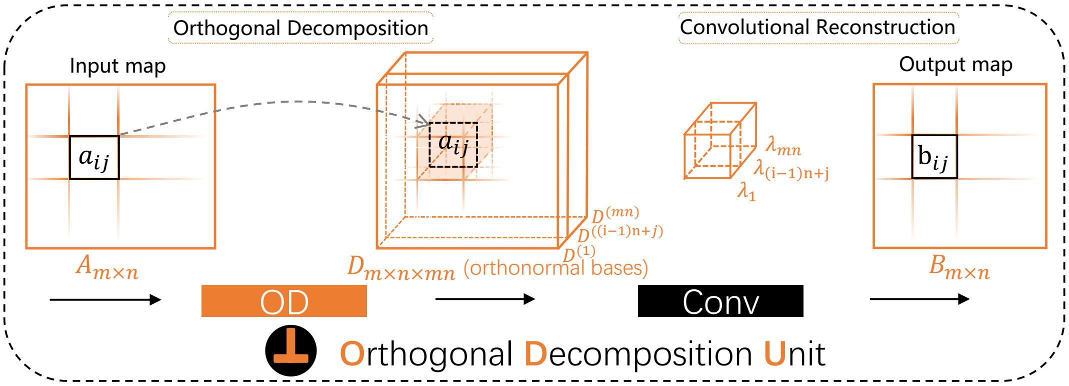

Figure 2: ODU eases the semantic correlation among neighbouring pixels by decomposing the input map to orthogonal channels, which are further fused in the convolutional reconstruction to perform pixel-wise refinement.

els [7]. For multiple pixel-wise classification tasks, SRN [11] and DSS [8] used residual models to "force" the side-output features from different convolutional layers to be complementary. A recent research, linear span network (LSN) [19] linked the multi-layer convolutional features with the linear span theory to reduce the correlation and increase the complementary of multi-channel features. However, the pixel-wise de-correlation problem remains unsolved, despite of the effort on sub-pixel operations [32].

3. Orthogonal Decomposition Unit

An Orthogonal Decomposition Unit (ODU) is made up of two components of orthogonal decomposition and convolutional reconstruction, Fig. 2. Orthogonal decomposition is utilized to decompose an input map to a set of orthogonal channels, which have the same size to the input map. Convolutional reconstruction is then utilized to fuse the orthogonal channels to an output map with pixel-wise refinement.

3.1. Orthogonal Decomposition for Spatial Decorrelation

Formally, we denote an $m \times n$ input map as $A = (a_{ij})$, where $i = 1, 2, \cdots m$ and $j = 1, 2, \cdots n$. a_{ij} is the pixel at i-th row and j-th column on the map. The input map A is transformed to an $m \times n \times mn$ map with complete orthogonal decomposition, denoted as D, on the left of Fig 2. Supposing the k-th channel of D as $D^{(k)} = (d_{ij}^{(k)})$, where $k = 1, 2, \cdots, mn$, the pixel-wise correspondence between A and D is defined as

$$\begin{cases} d_{ij}^{(k)} = a_{ij}, & \text{if } k = (i-1) \times n + j \\ d_{ij}^{(k)} = 0, & \text{else} \end{cases}. \tag{1}$$

According to Eq. 1, it can be easily concluded that complete orthogonal decomposition has two elegant properties.

Property 1: sparsity. Neighboring pixels in the input map are decomposed to different channels with sparse non-zero elements, *i.e.*, a channel has at most one non-zero pixel. The element-wise sum of the orthogonal channels is equivalent to the input map, as

$$A = \sum_{k=1}^{mn} D^{(k)}. \tag{2}$$

Property 2: orthogonality. As the locations of non-zero pixels from any two channels are different with each other, any two channels of the output map are orthogonal to each other, as

$$D^{(i)} \cdot D^{(j)} = 0, 1 \leq i \neq j \leq mn. \tag{3}$$

3.2. Convolutional Reconstruction for Pixel-wise Refinement

In convolutional reconstruction, a 1×1 convolutional layer is inserted atop the orthogonal decomposition layer to fit the ground-truth. Denoting the $m \times n$ output map and binary ground-truth as $B = (b_{ij})$ and $G = (g_{ij})$, the convolutional reconstruction is formulated as

$$\sum_{k=1}^{mn} \lambda_k D^{(k)} = B \approx G, k = 1, 2, \cdots, mn, \tag{4}$$

where $k = 1, 2, \cdots, mn$ and λ_k denotes the convoltuional reconstruction coefficient to be learned. Eq. 4 can be solved as

$$\lambda_{(i-1)n+j} = g_{ij}/a_{ij}, a_{ij} \neq 0, \tag{5}$$

which implies that with back-propagation each channel is assigned an independent weight corresponding to the single non-zero pixel. In this way, we can operate each pixel on the input map in the convolutional reconstruction for pixel-wise refinement. As the condition $a_{ij} \neq 0$ can be easily satisfied in the back-propagation process, ODU can completely reconstruct an arbitrary given binary mask with the orthogonal features.

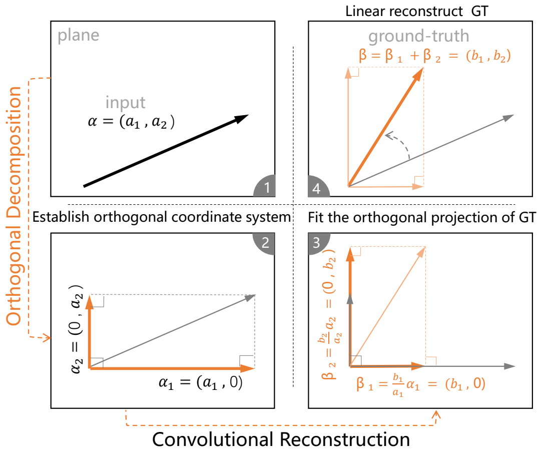
Linear reconstruct GT
plane
input
$\alpha = (a_1, a_2)$
1
ground-truth
$\beta = \beta_1 + \beta_2 = (b_1, b_2)$
4
Orthogonal Decomposition
Establish orthogonal coordinate system
Fit the orthogonal projection of GT
2
$\alpha_2 = (0, a_2)$
$\alpha_1 = (a_1, 0)$
3
$\beta_2 = \frac{b_2}{a_2}\alpha_2 = (0, b_2)$
$\beta_1 = \frac{b_1}{a_1}\alpha_1 = (b_1, 0)$
Convolutional Reconstruction

Figure 3: The orthogonal coordinate system generated from the input map by ODU can span the whole space.

3.3. Geometric Interpretation

To understand ODU, we visualize a two dimensional example, Fig. 3. A single channel input map can be represented as a vector $\alpha = (a_1, a_2)$ in the plane, Fig. 3(1). According to Eq. 1, vector α is decomposed into two orthogonal vectors, *i.e.*, $\alpha_1 = (a_1, 0)$ and $\alpha_2 = (0, a_2)$, in the *orthogonal decomposition*. These two orthogonal vectors establish an orthogonal coordinate system, Fig. 3(2).

As shown in Fig. 3(2), α_1 and α_2 can span the whole plane, where any vector can be perfectly reconstructed by α_1 and α_2. In *convolutional reconstruction*, the reconstruction coefficients can be directly computed by the coordinates of β, *i.e.*, a ground-truth mask. According to Eq. 5, α_1, and α_2 are transformed to β_1 and β_2, which are equivalent to the orthogonal projection of β to α_1 and α_2, Fig. 3(3). β is then reconstructed as $\beta = \beta_1 + \beta_2$, Fig. 3(4). Therefore, ODU can reconstruct β by independently fitting β_1 and β_2 in two orthogonal directions, which ensures the capability of pixel-wise refinement through spacial de-correlation of the input map.

4. Orthogonal Decomposition Network

Considering that neighboring pixels suffer semantic correlation most, we introduce ODU with incomplete orthogonal decomposition (IOD) which focuses on de-correlating the neighboring pixels within local patches. With IOD, we build Orthogonal Decomposition Network (ODNs) using single-ODU or multi-ODU to approximate the sparsity and orthogonality of features.

4.1. Incomplete Orthogonal Decomposition

The incomplete orthogonal decomposition, Fig. 4(a), is proposed to perform spacial de-correlation on local patches. Specifically, we use a $p_w \times p_h$ non-overlap densely sliding window to partition an input map into local patches with the same size and perform the complete orthogonal decomposition defined in Eq. 1 on all patches. With incomplete orthogonal decomposition, the pixel correspondence between the input map and the channels of the output feature map is defined as

$$\begin{cases} d_{ij}^{(k)} = a_{ij}, & \text{if } k = (i^{'} - 1) \times p_w + j^{'} \\ d_{ij}^{(k)} = 0, & \text{else} \end{cases}, \tag{6}$$

where a_{ij} and d_{ij} are defined in Eq. 1, $i^{'} \equiv i(modp_h)$, $j^{'} \equiv j(modp_w)$, $1 \leq i^{'} \leq p_h, 1 \leq j^{'} \leq p_w, 1 \leq i \leq m, 1 \leq j \leq n$.

The incomplete orthogonal decomposition approximates the properties of sparsity and orthogonality defined in Eq. 2 and Eq. 3 while reducing the channels of the output feature map to $p_w \cdot p_h$. Contrarily, according to Eq. 1, an $m \times n$ input map requires mn channels to implement the complete orthogonal decomposition. When m and n are large, there is a curse of dimensionality.

According to Eq. 5, the essence of ODU with complete orthogonal decomposition is refining each pixel with an independent weight. In incomplete orthogonal decomposition, there is weight sharing that non-zero pixels in the same channel share the reconstruction coefficient. Therefore, we appropriately use filter size larger than the size of orthogonal decomposition patch in the convolutional reconstruction to ease the weight sharing.

4.2. ODN Exemplars

By integrating the ODU with incomplete orthogonal decomposition, we update FCNs including VGG [33], HED [39] and SRNs [11] to OD-VGG, OD-HED, and OD-SRN, as shown in Fig. 4(b). For OD-VGG, a single ODU is added atop the last convolutional layer of VGG-16. For OD-HED and OD-SRN, six ODUs are inserted in the five side-output branches and one fusion branch. HED is an effective architecture for multi-scale convolutional feature fusion, while SRN updated the fusion strategy by introducing residual modules between the adjacent side-output branches. With these ODN exemplars, we target at validating the general applicability of ODUs to pixel-wise refinement in popular FCN architectures.

Single-ODU: spatial de-correlation. It is known that neighboring pixels on CNN feature maps have strong semantic correlation for the weight sharing scheme and spatial pooling operations. Such correlation causes falsely classifying background pixels near a true positive to false positives, and vice versa. With the sparsity (**Property 1**) of orthogonal decomposition, convolutional reconstruction can independently pinpoint the label of each pixel. In other words, a single ODU implement pixel-wise refinement through spatial de-correlation.

Multi-ODU: complementary feature learning. State-of-the-art approaches usually perform channel-wise feature

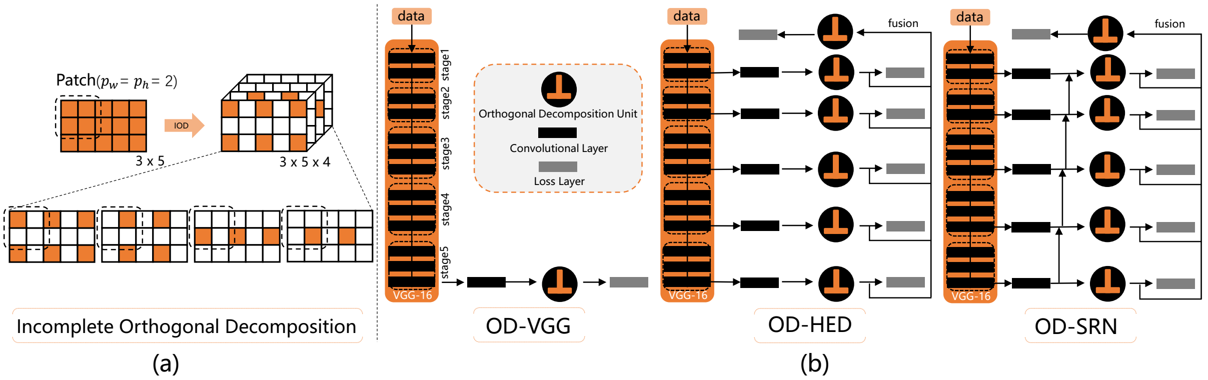

Figure 4: (a) Incomplete orthogonal decomposition with an 3 × 5 input map and 2 × 2 patch size. (b) With ODUs, plug-and-play modules added atop the output layers, VGG-16 [33], HED [39] and SRN [11] are updated to ODNs.

integration, but fail to differentiate positive and negative pixels in the same channels, which inevitably introduces noise to the final outputs. With multiple ODUs, the orthogonality (**Property 2**) drives the network to learn complementary features in the linear span view [19]. Meanwhile, the sparsity (**Property 1**) make the feature fusion procedure more selective, $i.e.$, different stages enhance/suppress pixels in different areas. In this way, the potential of the backbone network to learn complementary feature is further explored which further promote the ODN's capability of pixel-wise refinement.

5. Experiments

In this section, the experimental settings are first introduced. The effects of ODU on feature de-correlation and pixel-level refinement are then validated. Finally, the performance of ODNs is reported on pixel-wise binary classification tasks for two-dimensional image processing, specifically skeleton detection, edge detection, and saliency detection, and one-dimensional keypoint detection, specifically S-wave arrival time detection for earthquake localization.

All the experiments run on a Tesla K40 GPU. The mini-batch size is set to 1, the loss-weight to 1 for each output layer, the momentum to 0.9, the weight decay to 0.002, and the initial learning rate to 1e-6, which decreases one magnitude for every 10,000 iterations.

5.1. ODU Effect

From left to right in each row of Fig. 5, it can be seen that the output maps of OD-HED are refined and much background noise is suppressed. Particularly, the zigzag noise caused by up-sampling is eased and the fused output (the last row of OD-HED) is closer to the ground-truth. By comparing the outputs of all five stages of HED and OD-HED in Fig. 5 from top to down, we conclude that the output maps of OD-HED are more complementary than those of HED. In shallow stages of OD-HED, the fine details are richer than that of HED. The torso of the horse is suppressed while the slim parts including the tail and legs are enhanced. In deep stages of OD-HED, the torso of the horse is enhance while slim parts are suppressed. These validate that ODUs can drive multi-layer learning complementary convolutional features. With t-SNE analysis, it's illustrated that with the features learned by OD-HED, the background and foreground pixels are more separable, validating the effectiveness of ODUs for semantic de-correlation, Fig. 6.

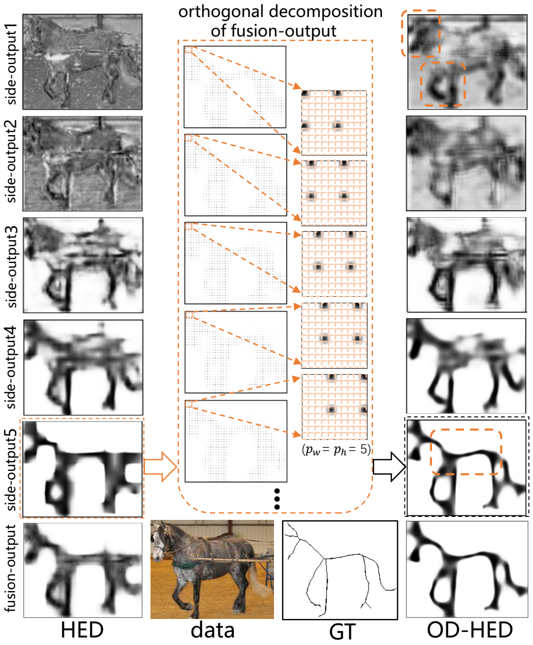

Figure 5: With ODU, each input feature map (first column) is decomposed into orthogonal channels (middle column) that facilitate de-correlated and complementary feature learning and pixel-wise refinement (last column).

We give quantitative comparison of patch sizes and convolutional reconstruction filter size of incomplete orthogonal decomposition on SK-LARGE skeleton detection dataset, Table 1 and Table 2. In Table 1, it can be seen

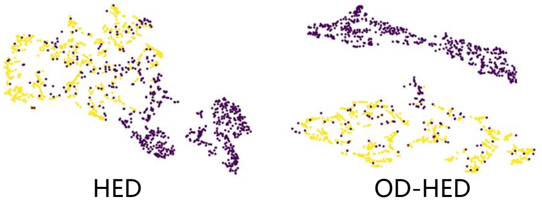

Figure 6: t-SNE of foreground/background pixels shows that the features of OD-HED incorporate less semantic correlation than those of HED. (Best viewed in color)

Table 1: Comparison of patch sizes of incomplete orthogonal decomposition on SK-LARGE with 11 × 11 filter size.

Patch size	w/o	3×3	5×5	7×7
F-measure	0.489	0.604	**0.606**	0.600

Table 2: Comparison of filter sizes for convolutional reconstruction on SK-LARGE with 5×5 patch size.

Filter size	3×3	7×7	11×11	15×15	17×17
F-measure	0.511	0.575	0.606	**0.620**	0.619

Table 3: Skeleton detection on SKLARGE.

Methods	F-measure	Δ F-measure
VGG-16	0.489	
OD-VGG	0.620	0.131
HED [39]	0.495	
OD-HED	0.644	0.149
SRN [11]	0.655	
OD-SRN	**0.676**	0.021

that the F-measure significantly increases after applying the ODU. Compared with VGG-16 without ODU, VGG-16 with incomplete orthogonal decomposition patch size 3×3, achieve 21.5% (0.489 vs. 0.604) performance gain. As semantic correlation mainly exits within local patches, the F-measure increases to 0.606 using patch size 5×5, but stops increase when the size becomes larger. In all the following experiments, we use the patch size 5×5. In Table 2, we evaluate reconstruction convolutional filters under different sizes. From 3×3 to 15×15, the performance keeps increasing, but drops a little at 17×17. The reason for this phenomenon is that larger filter eases the weight sharing of non-zero pixels of the incomplete decomposed orthogonal features while aggravates the weight learning burden. Therefore the 15×15 convolutional filter is selected for reconstruction.

5.2. Skeleton Detection

Five skeleton detection datasets, including SYMMAX [36], WH-SYMMAX [28], SK-SMALL [31], SK-LARGE [30], and Sym-PASCAL [11] are used to evaluate ODNs. SYMMAX contains 200/100 training/test images. SK-SMALL involves skeletons about 16 classes of objects with 300/206 training/test images. Based on SK-SMALL, SK-LARGE is extended to 746/745 training/test images. Sym-PASCAL is derived from the PASCAL-VOC-2011 segmentation dataset [5] which contains 14 object classes with 648/787 images for training and test. By changing threshold values on output masks we get multiple predicted binary skeleton masks, which is compared with the ground-truth pixel-by-pixel to compute precision (P) and recall (R). The F-measure is used to evaluate the performance of the different detection approaches, which is calculated with the optimal threshold values over the whole dataset as $F = 2P \cdot R/(P + R)$.

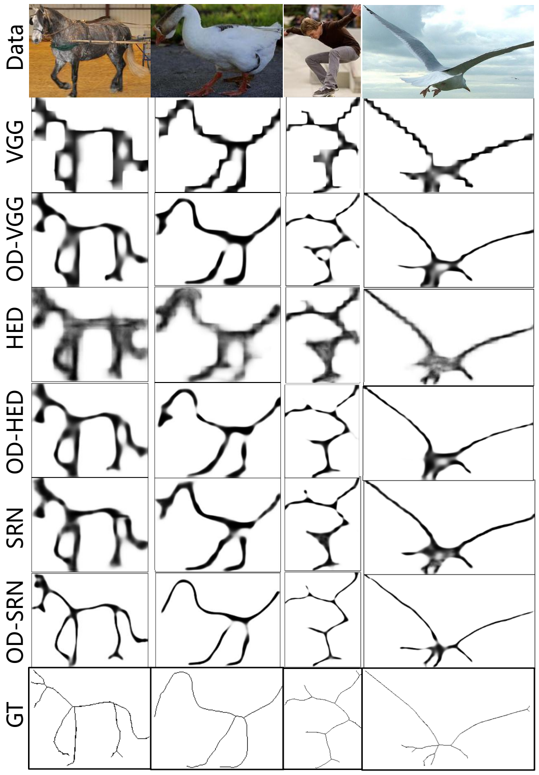

Figure 7: Pixel-wise refinement of skeleton detection examples by ODNs. It can be seen that with ODUs inserted, zigzag noise and blur are suppressed and fine details of object skeletons are detected.

The performance of three ODNs including OD-VGG, OD-HED, and OD-SRN and the comparisons with the baseline networks are shown in Table 3. It can be seen that OD-VGG outperforms VGG by 13.1%, OD-HED outperforms HED by 14.9%. The reason for the large performance gain lies in that a single ODU in VGG, Fig. 4(b), effectively decorrelates convolutional features while multiple ODUs in HED, Fig. 4(b), drive learning multi-layer complementary features. With the capability of pixel-wise label refinement,

Table 4: Performance comparison of state-of-the-art approaches on five commonly used skeleton detection datasets.

	Lindeberg [18]	MIL [36]	HED [39]	FSDS [31]	LSN [19]	SRN [11]	OD-SRN (ours)
WH-SYMMAX [28]	0.277	0.365	0.743	0.769	0.797	0.780	**0.804**
SK-SMALL [31]	0.227	0.392	0.542	0.623	**0.633**	0.609	0.624
SYMMAX [36]	0.360	0.362	0.427	0.467	0.480	0.446	**0.489**
SK-LARGE [30]	0.270	0.293	0.495	–	0.668	0.655	**0.676**
Sym-PASCAL [11]	0.138	0.174	0.369	0.418	0.425	0.443	**0.444**

Table 5: Performance comparison on the BSDS500 edge detection dataset.

Methods	ODS	OIS	AP
DC [29]	0.757	0.776	0.790
HED [39]	0.780	0.797	0.814
SRN [11]	0.782	0.800	0.779
LSN [19]	0.790	0.806	0.618
OD-SRN(ours)	**0.798**	**0.814**	0.782
Human	0.800	0.800	–

OD-SRN outperforms SRN that uses residual modules to learn complementary features, aggregating the performance by 2.1%. The pair-wise comparison of skeleton detection results is illustrated in Fig. 7.

On the five commonly used skeleton detection datasets, Table 4, the OD-SRN respectively outperforms the baseline SRN 2.4%, 1.5%, 4.3%, 2.1%, and 0.1% and beats the state-of-the-art approaches.

5.3. Edge Detection

Edge detection is another typical pixel-wise binary classification task. The BSDS500 [2] dataset that is composed of 200 training images, 100 validation images, and 200 testing images is used to evaluated the ODN. In BSDS 500, each edge mask is manually annotated by five persons on average. For training images, we preserve their positive labels annotated by at least three human annotators. The F-measures when choosing an optimal scale for the entire dataset (ODS) or per image (OIS), and the average precision (AP) are used as the evaluation metrics.

As shown in Table 5, the DeepContour (DC) [29] sets a solid baseline. The HED [39] approach that fuses multi-scale convolutional features reports higher performance with ODS=0.780. The state-of-the-art SRN [11] achieves ODS=0.782, and the LSN [19] achieves ODS=0.790. OD-SRN achieves the best performance, ODS=0.798 and OIS =0.814, which is even comparable to human performance.

5.4. Saliency Detection

We evaluate OD-SRN on five object saliency detection datasets, including MSRA-B [20], ECSSD [40], HKU-IS [15], PASCALS [17], SOD [22] [23]. MSRA-B contains 5,000 images with single objects. ECSSD contains 1,000 images with complex backgrounds. HKU-IS contains 4000 images for multi-object saliency. PASCALS contains 850 images. SOD is a subset of the BSDS dataset and contains 300 images, most of which has more than one salient objects. We train OD-SRN with the MSRA-B dataset and test the model on all five datasets. F-measure and the mean absolute error (MAE) are used as performance metrics [9].

In Table 6, OD-SRN consistently outperforms the baseline SRN on the five datasets. The advantage of our approach lies in pixel-wise label refinement, so it mainly aggregates the saliency detection results at object boundaries, as shown in Fig. 8. For holistic saliency regions, the performance improvement is moderate.

5.5. One-dimensional Keypoint Detection: S-wave Arrival Time Detection

Task and dataset. As a general approach for pixel-wise binary classification, ODN is extended to one-dimensional keypoint detection, $i.e.$, detecting S-wave arrival times of an earthquake in enormous seismograms, Fig. 9. These arrivals are keypoints of time when S-wave reaches seismometers, which are essential for accurate earthquake localization [6] and earth interior imaging [10] [37]. Such arrivals are usually annotated by experts with great efforts [1] [26]. The training dataset includes 20,000 S-wave pick-waveform pairs provided by human experts[1]. The testing dataset includes 1512 records from 782 stations of 97 earthquakes in Japan. Each record is discretized as a 1D vector with 1200 points and two channels corresponding to a radial component R and a transverse component T, Fig. 9.

Performance and analysis. The S-wave arrival time detection performance is evaluated by calculating the proportions of samples under different time deviation between the prediction and the ground-truth, Table 7. It can be seen that OD-SRN outperforms the SRN baseline, $i.e.$, with higher proportion under the same time deviation. Specially, with deviation ≤ 0.2s, OD-SRN achieves 76.7% accuracy, which significantly outperforms SRN by 4.5%. In Fig. 9, SRN reports a false positive (red dotted box), while OD-SRN can precisely detect the arrival time of the S-wave by predicting

[1] Research Center for Prediction of Earthquakes and Volcanic Eruptions, Tohoku University and Hi-net

Table 6: Performance comparison of the state-of-the-art approaches on commonly used saliency detection datasets. (Smaller MAE indicates better performance.)

Methods	MSRA-B [20]		ECSSD [40]		HKU-IS [15]		PASCALS [17]		SOD [22] [23]	
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
SRN [11]	0.888	0.063	0.872	0.084	0.871	0.065	0.771	0.129	0.803	0.132
OD-SRN (ours)	**0.899**	**0.058**	**0.883**	**0.078**	**0.882**	**0.060**	**0.786**	**0.121**	**0.815**	**0.129**

Table 7: Performance comparison on S-wave arrival time detection (sample proportions under time deviation).

Deviation(s)	0.00	0.01	0.02	0.03	0.04	0.05	0.06-0.10	0.11-0.20	≤ 0.20
SRN [11]	0.030	0.065	0.063	**0.073**	0.051	**0.057**	0.192	0.190	0.722
OD-SRN(ours)	**0.035**	**0.072**	**0.079**	0.053	**0.058**	0.054	**0.211**	**0.204**	**0.767**

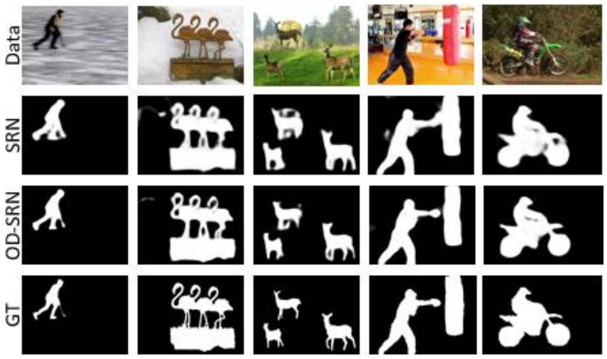

Figure 8: Pixel-wise refinement of saliency by OD-SRN.

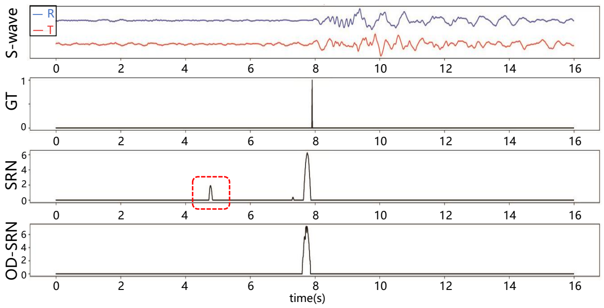

Figure 9: An S-wave arrival time example is detected by SRN and OD-SRN. A false positive (red dotted box) predicted by SRN is suppressed by OD-SRN.

the maximum keypoint, which shows the pixel-level refinement capability of OD-SRN.

Fig. 10 shows S-wave arrival time detection examples predicted by OD-SRN (red lines) which achieves close performance to the ground-truth (blue lines) annotated by human experts.

6. Conclusion

We proposed the Orthogonal Decomposition Unit (ODU) targeting at de-correlating neighboring pixels on convolutional features. In theory, the complete orthogonal decomposition produced orthogonal bases that can perfectly reconstruct any binary mask (ground-truth). In practice, incomplete orthogonal decomposition with proper patch sizes can effectively and efficiently approximate the complete orthogonal decomposition. We updated successful FCNs including VGG-16, HED, and SRN to Orthogonal Decomposition Networks (ODNs) and applied them on typical pixel-wise binary classification tasks including skeleton, edge, and saliency detection to validate the generality of ODUs for pixel-level spacial de-correlation and pixel-wise refinement. The extension of ODN to 1D keypoint detection, $i.e.$, S-wave arrival time detection for earthquake localization provides fresh insight about the application of deep learning in the area of geoscience.

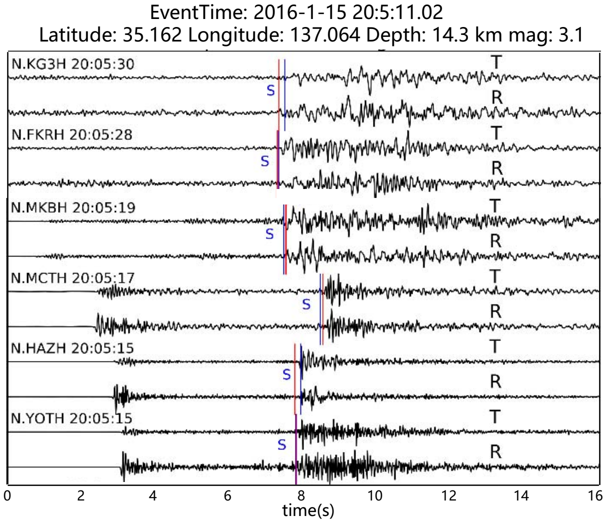

Figure 10: S-wave arrival times in an earthquake detected by OD-SRN . Red lines are arrivals detected by OD-SRN while blue ones are ground-truth. (Best viewed in color)

Acknowledgments. The authors are very grateful to the support by NSFC grant 61836012 and 61671427, and Beijing Municipal Science and Technology Commission grant Z181100008918014.

References

[1] Jubran Akram and David W Eaton. A review and appraisal of arrival-time picking methods for downhole microseismic dataarrival-time picking methods. *Geophysics*, 81(2):KS71–KS91, 2016.

[2] Pablo Arbelaez, Michael Maire, Charless C. Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011.

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

[4] Xiaobin Chang, Tao Xiang, and Timothy M. Hospedales. Scalable and effective deep cca via soft decorrelation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1488–1497, 2018.

[5] Mark Everingham, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[6] Jeanne L Hardebeck and Peter M Shearer. A new method for determining first-motion focal mechanisms. *Bulletin of the Seismological Society of America*, 92(6):2264–2276, 2002.

[7] Shengfeng He, Rynson W. H. Lau, Wenxi Liu, Zhe Huang, and Qingxiong Yang. Supercnn: A superpixelwise convolutional neural network for salient object detection. *International Journal of Computer Vision*, 115(3):330–344, 2015.

[8] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. In *IEEE CVPR*, pages 3203–3212, 2017.

[9] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5300–5309. IEEE, 2017.

[10] Chengxin Jiang, Brandon Schmandt, Steven M Hansen, Sara L Dougherty, Robert W Clayton, Jamie Farrell, and Fan-Chi Lin. Rayleigh and s wave tomography constraints on subduction termination and lithospheric foundering in central california. *Earth and Planetary Science Letters*, 488:14–26, 2018.

[11] Wei Ke, Jie Chen, Jianbin Jiao, Guoying Zhao, and Qixiang Ye. SRN: side-output residual network for object symmetry detection in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 302–310, 2017.

[12] Louisa Lam, Seong-Whan Lee, and Ching Y. Suen. Thinning methodologies - A comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(9):869–885, 1992.

[13] Tom Sie Ho Lee, Sanja Fidler, and Sven J. Dickinson. Learning to combine mid-level cues for object proposal generation. In *IEEE International Conference on Computer Vision*, pages 1680–1688, 2015.

[14] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 5455–5463, 2015.

[15] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5455–5463, 2015.

[16] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 478–487, 2016.

[17] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 280–287, 2014.

[18] Tony Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–156, 1998.

[19] Chang Liu, Wei Ke, Fei Qin, and Qixiang Ye. Linear span network for object skeleton detection. In *European Conference on Computer Vision*, pages 133–148, 2018.

[20] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, 33(2):353–367, 2011.

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[22] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416–423. IEEE, 2001.

[23] Vida Movahedi and James H Elder. Design and perceptual validation of performance measures for salient object segmentation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 49–56. IEEE, 2010.

[24] Punam K. Saha, Gunilla Borgefors, and Gabriella Sanniti di Baja. A survey on skeletonization algorithms and their applications. *Pattern Recognition Letters*, 76:3–12, 2016.

[25] Thomas B. Sebastian, Philip N. Klein, and Benjamin B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):550–571, 2004.

[26] BK Sharma, Amod Kumar, and VM Murthy. Evaluation of seismic events detection algorithms. *Journal of the Geological Society of India*, 75(3):533–538, 2010.

[27] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, 2017.

[28] Wei Shen, Xiang Bai, Zihao Hu, and Zhijiang Zhang. Multiple instance subspace learning via partial random projection tree for local reflection symmetry in natural images. *Pattern Recognition*, 52:306–316, 2016.

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

[30] Wei Shen, Kai Zhao, Yuan Jiang, Yan Wang, Xiang Bai, and Alan L. Yuille. Deepskeleton: Learning multi-task scale-associated deep side outputs for object skeleton extraction in natural images. *IEEE Trans. Image Processing*, 26(11):5298–5311, 2017.

[31] Wei Shen, Kai Zhao, Yuan Jiang, Yan Wang, Zhijiang Zhang, and Xiang Bai. Object skeleton extraction in natural images by fusing scale-associated deep side outputs. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 222–230, 2016.

[32] Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1874–1883, 2016.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *IEEE International Conference on Computer Vision*, 2015.

[34] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. In *IEEE International Conference on Computer Vision, ICCV*, pages 3820–3828, 2017.

[35] Ching Lik Teo, Cornelia Fermüller, and Yiannis Aloimonos. Detection and segmentation of 2d curved reflection symmetric structures. In *IEEE International Conference on Computer Vision, ICCV*, pages 1644–1652, 2015.

[36] Stavros Tsogkas and Iasonas Kokkinos. Learning-based symmetry detection in natural images. In *European Conference on Computer Vision (ECCV)*, pages 41–54, 2012.

[37] Zewei Wang, Dapeng Zhao, Xin Liu, Chuanxu Chen, and Xibing Li. P and s wave attenuation tomography of the japan subduction zone. *Geochemistry, Geophysics, Geosystems*, 18(4):1688–1710, 2017.

[38] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[39] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *IEEE International Conference on Computer Vision*, pages 1395–1403, 2015.

[40] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1155–1162, 2013.