DeepPano: Deep Panoramic Representation for 3D Shape Recognition

Baoguang Shi, *Student Member, IEEE*, Song Bai, *Student Member, IEEE*,
Zhichao Zhou, and Xiang Bai, *Senior Member, IEEE*

***Abstract*—This letter introduces a robust representation of 3D shapes, named DeepPano, learned with deep convolutional neural networks (CNN). Firstly, each 3D shape is converted into a panoramic view, namely a cylinder projection around its principle axis. Then, a variant of CNN is specifically designed for learning the deep representations directly from such views. Different from typical CNN, a row-wise max-pooling layer is inserted between the convolution and fully-connected layers, making the learned representations invariant to the rotation around a principle axis. Our approach achieves state-of-the-art retrieval/classification results on two large-scale 3D model datasets (ModelNet-10 and ModelNet-40), outperforming typical methods by a large margin.**

***Index Terms*—3D shape, classification, retrieval, panorama, convolutional neural networks**

I. Introduction

THREE-DIMENSIONAL shapes carry rich information of real-world objects, and are important cues for object recognition. The analysis of 3D shapes is a fundamental problem, and has a wide range of applications in medical imaging, computer aided design (CAD), virtual reality, *etc.*. One of the most important challenges in 3D shape analysis is to obtain a good representation for shapes. The performance of many tasks, including shape classification and shape retrieval, heavily depend on the quality of the representation.

In this letter, we propose a 3D shape descriptor called *DeepPano* for 3D shape classification and retrieval, which is directly learned from the panoramic views of 3D models. The panoramic view is a cylinder projection of a 3D model around its principle axis. Therefore, the panoramic views are in the form of 2D images, which can be considered as a holistic representation of 3D models. We use convolutional neural network [1] (CNN) to learn a deep representation from such views. To make the learned deep features invariant to the rotation around the principle axis, a special layer named *Row-Wise Max-Pooling* (RWMP) layer is presented and inserted between the convolution layers and the fully-connected layers. This layer takes the maximum value of each row in the convolutional feature maps. Consequently, the output feature vector is not affected by the shift of the panoramic view, caused by the rotation of 3D shape.

Authors are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, P.R.China 430074. (e-mail:{shibaoguang, songbai, zzc, xbai}@hust.edu.cn).

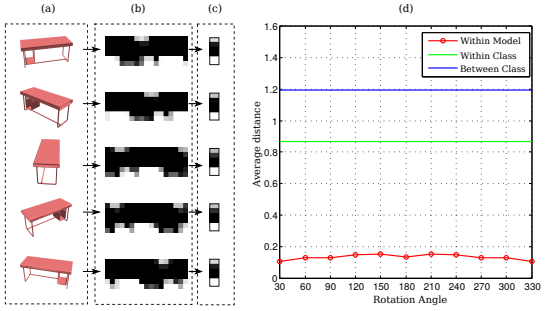

Figure 1. **Rotation invariance of DeepPano:** (a) 3D shapes of the same model, but rotated to different angles; (b) Convolutional feature map for each 3D shape; (c) Output vectors of the RWMP layer; (d) Comparisons among within model distances, within class distances and between class distances (Refer to Section III-E for details).

The previous methods on 3D shape analysis can be coarsely categorized into model-based and view-based methods. Model-based methods calculate a set of features directly from the 3D shape mesh or its rendered voxels. Such methods include the Shape Histogram descriptor [2] and the Spin Images [3]. View-based methods represent 3D shapes by a set of views [4]–[10]. The views can be 2D projections of the shape or the panoramic view. We extract the shape representation from the panoramic view. However, different from most of the methods mentioned above that use hand-crafted features, we learn the representation from data with a variant of CNN. Deeply learned representations are widely used and have achieved superior performance in many pattern recognition and signal processing tasks [11]–[13]. There are other attempts that represent 3D shapes by deep features. Recently, Wu *et. al* [14] proposes the 3D ShapeNets, a Convolutional Deep Belief Network for shape representation. Different from [14] which performs 3D convolutions on the voxels, we extract the representation of a 3D shape from 2D images. Compared with [14], our method achieves better performances on both classification and retrieval tasks (refer to Section III), and is simpler to implement using any open source framework.

Our method is related to previously introduced PANORAMA [6]. In [6], Panagiotis *et al.* proposed to represent a 3D shape by the Discrete Fourier Transform and Discrete Wavelet Transform descriptors calculated from a set of panoramic views. However, the panoramic view shifts when the 3D shape rotates along its principle axis. In [6], this problem is alleviated by pose normalization. As illustrated in Figure 1, the convolutional feature maps extracted from panoramic views shifts when the 3D shape

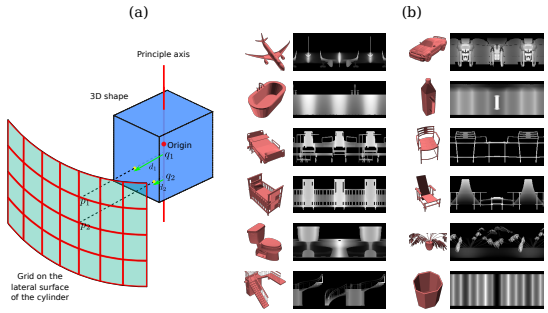

Figure 2. **Panoramic view construction:** (a) Illustration of the panoramic view construction process. p, q and d are respectively the grid point, the corresponding point on the axis and the value assigned to that grid point; (b) 3D shapes and their corresponding panoramic views (with some padding as describe in Section II-B).

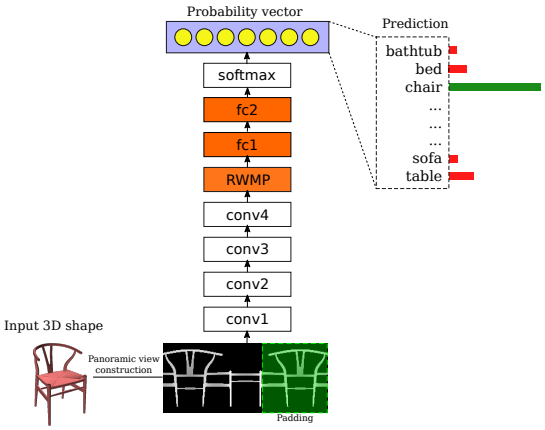

Figure 3. The network for learning and extracting shape representation. The network takes the padded panoramic view as the input. On the top it outputs a probability vector representing class probabilities. The 3D shape representation can be extracted from the highlighted layers, namely RWMP, fc1 or fc2. (fc means fully-connected layer, conv means convolution layer).

rotates. We pool the the responses of each row so that the resulting representation is not affected by this kind of shift. As a result, the representation is invariant to the 3D shape rotation.

To summarize, the key contribution of this letter is the deep panoramic representation that is rotation-invariant to the principle axis. The experiments on large-scale 3D shape datasets show that this representation is effective in both classification and retrieval tasks, outperforming previous methods by a large margin.

The rest of this letter is organized as follows. Section II introduces the learning and extraction processes of the representation in detail. In Section III we verify the rotation invariance of the representation, and evaluate its performance on classification and retrieval tasks. Conclusions are drawn in Section IV.

II. Methodology

Our method consists of two main steps: (1) Generate the panoramic views (Section II-A); (2) Learn and extract the rotation-invariant representation from the views (Section II-B). The representation is used for both classification and retrieval tasks (Section II-C). Throughout this letter, we assume that 3D models are upright oriented, so that the rotation is along a axis that is also upright oriented. This assumption is satisfied in many real-world model repositories, such as the 3D Warehouse [15].

A. Panoramic view construction

To construct the panoramic view, the 3D shape is projected onto the lateral surface of a cylinder whose axis is parallel to the principle axis of the 3D shape. With the upright-orientation assumption, we simply set $p_o = (x_o, y_o)$ as the origin point and take z-direction as the orientation to obtain the principle axis. In our approach, p_o is calculated by the weighted average of all triangles on the model mesh, where the weights are triangle areas.

The projection process is illustrated in Figure 2. We discretize the lateral surface of the cylinder by a dense grid of points, represented by their coordinates $\{(\theta_u, h_v)\}$, where θ is the polar angle and h is the height. For each grid point $p = (\theta_u, h_v)$, a corresponding point q is found by the Cartesian coordinate(x_o, y_o, h_v), which is the point on the cylinder axis with the same height as p. A ray is cast from q to p, intersecting with none, one or several triangles of the 3D shape. The distances between q and the intersection points are recorded as $\mathcal{D} = \{d_1, d_2, \ldots, d_I\}$, $I = 0, 1, 2, \ldots$. For each grid point, we assign a value d that is the max value in $\mathcal{D}$, or zero when $\mathcal{D}$ is an empty set:

$$d = \begin{cases} 0 & \mathcal{D} = \emptyset, \\ \max \mathcal{D} & \text{otherwise.} \end{cases}$$

After the projection, the lateral surface is unfolded from certain angle (we choose $\theta = 0$) into the 2D panoramic view, whose pixel values are the ones assigned to the grid points. To avoid the impact of scale changes of the shape, we choose the cylinder to have the same height as the 3D shape. In addition, the panoramic view is subtracted by its mean and divided by its standard deviation before further processing. Consequently, the constructed panoramic view is not affected by the size of the shape.

B. Representation learning and extraction

The panoramic view keeps most of the information of the 3D shape. Therefore, a 3D shape can be described by the 2D descriptor extracted from its panoramic view. A straightforward method is to train a CNN on the panoramic views of all training data, and extract the representation from it. However, the view shifts when the 3D shape rotates. This shift will greatly affect the representation produced by the CNN, although the CNN provides some form of translation invariance. Moreover, unfolding the lateral surface creates two boundaries on the left and right sides of the panoramic view. The boundaries cause artifacts in the convolutional feature maps, thus affecting the representation extracted.

In our approach, a variant of CNN is created to learn and extract the representation, handling the issues mentioned above. As illustrated in Figure 3, firstly, to avoid boundary artifacts,

the panoramic view is padded on one side. The padded area is cloned from the other side of the map. Specifically, we adopt a $h \times h$ padding size where h is the height of the view.

To obtain rotation-invariance, the representation has to be shift-invariant to the input panoramic view. The first few layers of a typical CNN, namely the convolution layers and the max-pooling layers produce feature maps that shift together with the input view. Between these layers and the fully-connected layers, we insert a layer called the *row-wise max-pooling* layer (RWMP), which takes the maximum value of each row in the input map and concatenate them into the output vector. The output of the RWMP layer is not affected by the shift of the input map, thus its output is invariant to the rotation of the 3D shape. The network is trained on a dataset consisting of pairs of panoramic views and class labels, using the back propagation algorithm [16]. Finally, the representation can be extracted from the RWMP layer, or any fully-connected layer after it.

C. Recognition with the representation

The extracted representation can be used for both classification or retrieval tasks. Since the network for learning the representation is itself a classifier, we direct adopt it for classification tasks. The softmax layer on the top of the network outputs class probabilities, and the class with the highest probability is taken as the prediction (as illustrated in Figure 3). For retrieval tasks, we define the similarity between a pair of 3D shapes as the Euclidean distance between their L_2-normalized descriptors. Since each 3D shape is represented by a fixed-length vector and Euclidean distance is used for retrieval, we can perform fast retrieval on large-scale datasets, particularly when adopting some approximate nearest neighbor search schemes, *e.g.* [17].

III. Experiments

A. Datasets

Princeton ModelNet is a large-scale 3D CAD model datasets that contains 127,915 CAD models in 662 object categories. Two subsets of it are used for training and testing, namely 1) **ModelNet-10**, a 10-categories subset consisting of 4,899 CAD models. All models in this subset are manually cleaned and their orientations are aligned. Among them 3,991 are used for training and the rest 908 are used for testing; 2) **ModelNet-40**, which has 12,311 models in 40 categories. 9,843 models are used for training and 2,468 are used for testing. The models in ModelNet-40 are cleaned but not manually aligned. Still, most models in this subset satisfy the upright assumption.

B. Implementation details

In our implementation, each 3D shape is projected into a 64×96 panoramic view. A 64×64 patch is padded on the right side, so that the input size to the network is 64×160. The architecture of the network is specified in Figure 3. For convolutional layers (conv1$\sim$conv4), the numbers of feature maps are respectively 96, 256, 384, 512 and the kernel sizes are respectively 5, 5, 3, 3. A 2×2 max-pooling layer is inserted after each convolutional layer. The network is trained using the stochastic gradient descent (SGD) with the momentum set to 0.9. The dropout technique [18] is adopted on both fully-connected layers (fc1 and fc2) in order to reduce overfitting.

We implement the GPU-accelerated network within the Torch7 [19] framework. The construction of panoramic views is implemented separately in MATLAB. Running on a machine with Intel Core-i5 CPU, NVIDIA GTX780 GPU and 8GB RAM, the training process takes less than 4 hours to fully converge. Rendering the panoramic view for each 3D shape takes less than 1s with an unoptimized CPU implementation, and should be accelerated a great deal by a GPU implementation.

Table I
Classification accuracies for various methods on the ModelNet-10 and ModelNet-40 datasets. Best results are marked in bold font.

Method	ModelNet-10	ModelNet-40
SPH [20]	79.79%	68.23%
LFD [7]	79.87%	75.47%
3D ShapeNets [14]	83.54%	77.32%
DeepPano	**88.66%**	**82.54%**

C. 3D shape classification

To evaluate our method on 3D shape classification tasks, we train the classification network starting from random initialization. The trained network outputs class probabilities from its softmax layer. The class with the highest probability is taken as the prediction. The performance is evaluated by the average category accuracy. We compare our method with the LightField descriptor [7] (LFD, 4,700 dimensions), the Spherical Harmonics descriptor [20] (SPH, 544 dimensions) and the 3D ShapeNets [14].

Table I summarizes the results. Our method outperforms all the other methods by a large margin. In comparison with the hand-crafted LFD, SPH and PANORAMA, which are also designed to be rotation-invariant, our deeply-learned representation clearly shows a stronger discriminative power. Compared with the 3D ShapeNets, our method performs remarkbly better. One of the reason is that we hard-code the rotation invariance into the network architecture, making the representation rotation invariant.

D. 3D shape retrieval

In retrieval tasks, we extract the 3D shape representations from the RWMP, the fc1 or the fc2 layer in the network. For each query from the test samples, we rank the rest test samples based on the Euclidean distances between their L_2 normalized descriptors. The performance is evaluated under two metrics: (1) mean area under precision-recall curve (AUC); (2) mean average precision (MAP). DeepPano is compared against the the SPH [20], the LFD [7], the PANORAMA [6] and the 3D ShapeNets [14]. In addition, the representations extracted from different layers are compared. Seen from the results summarized Table II and the precision-recall curves plotted

Table II
RETRIEVAL RESULTS ON MODELNET-10 AND MODELNET-40. BEST RESULTS ARE MARKED IN BOLD FONT.

Method	ModelNet-10		ModelNet-40	
	AUC	MAP	AUC	MAP
SPH [20]	45.97%	44.05%	34.47%	33.26%
LFD [7]	51.70%	49.82%	42.04%	40.91%
PANORAMA [6]	60.72%	60.32%	45.00%	46.13%
3D ShapeNets [14]	69.28%	68.26%	49.94%	49.23%
DeepPano (RWMP)	70.76%	69.88%	56.68%	56.14%
DeepPano (fc1)	84.47%	83.52%	74.57%	73.92%
DeepPano (fc2)	**85.45%**	**84.18%**	**77.63%**	**76.81%**

in Figure 4, our representations consistently achieves higher or comparable performances, compared with other methods. Specifically, the representation extracted from fc1 and fc2 performs remarkably better than previous state-of-the-art [14] on both datasets. The RWMP representation performs slightly better than 3D ShapeNets on ModelNet-10, but the gap is still evident on the larger ModelNet-40 dataset. In Figure 5 we show some examples.

E. Discussion

1) Rotation invariance: To verify the rotation invariance of the proposed representation, we take the L_2-normalized output of the RWMP layer as the 3D shape representation. For each of the 100 randomly picked 3D shapes, we generate 11 shapes by rotating it 30, 60, ..., 330 degrees, and calculate the distances between the shape and its rotated shapes. The average distances for all the 11 angles are plotted in Figure 1 (within model).

Our network is a combination of CNN and RWMP layer. To verify the effectiveness of the two parts separately, we evaluate a conventional CNN, which has the the same architecture as the proposed network, except that it does not contain an RWMP layer. On ModelNet-10, the retrieval AUC and MAP are respectively 85.25% and 84.16%. It is very close to our approach (85.45% and 84.18%) since that ModelNet-10 is well aligned and is not much affected by shape rotation. On the other hand, ModelNet-40 contains models in different angles. The AUC and MAP of the CNN are respectively 69.17% and 68.26%, and our rotation-invariant approach shows significant higher performance (77.63% and 76.81%).

2) Within/between-class distance: We pick up 100 pairs of same-class shapes (within class) and 100 pairs different-class shapes (between class). The distances between these pairs are calculated and Figure 1 plots the average. Comparing the distances, we can see that the representation changes a little when the shape rotates, therefore rotation-invariance is obtained. Besides, as can be seen in Figure 1, the representation exhibits significantly larger between-class distances than within-class distances, making it suitable for retrieval tasks.

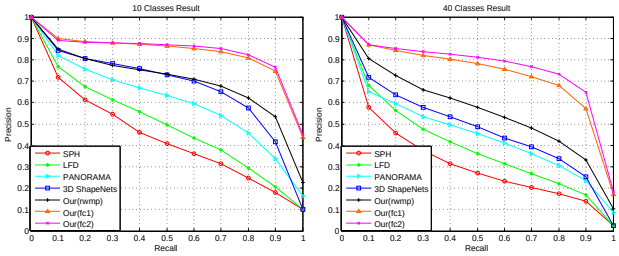

Figure 4. Precision-recall curves for various methods on the ModelNet-10 dataset (left) and ModelNet-40 dataset (right).

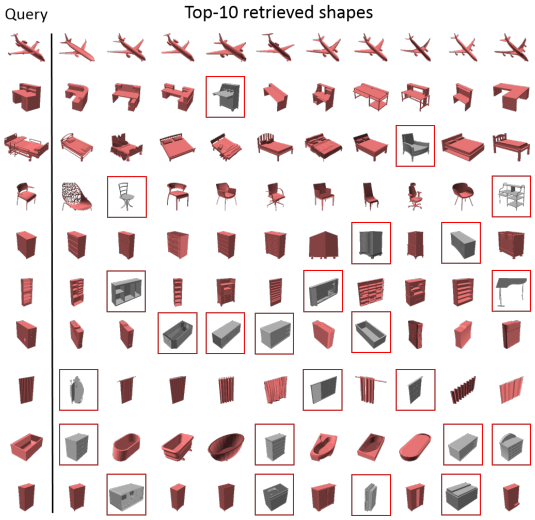

Figure 5. Examples showing the given query and the top 10 retrieved 3D shapes, sorted by matching distance. Mistakes are highlighted in red.

IV. CONCLUSION

In this letter, we have introduced DeepPano, a rotation-invariant deep representation for 3D shape classification and retrieval. Panoramic views are constructed from 3D shapes and representations are learned and extracted from them. DeepPano outperforms previous methods by a large margin, on both classification and retrieval tasks. We have also experimentally verified the rotation invariance of the representation. The limitation of our method is similar to many previous view-based approaches, requiring the principle axes of 3D models, which may fail to recognize the 3D models with serious non-rigid deformation. In the future, some sequence prediction techniques [21], [22] might be used for exploring more contextual information, in order to further improve the performance of shape recognition, as a panoramic view can be considered as a map of feature sequence. In addition, to establish the robust alignments/correspondence [23], [24] between different panoramic views is another direction that is worthy of being studied.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (NSFC) (no. 61222308), NSFC (no. 61573160), Excellent Talents in University (no. NCET-12-0217).

REFERENCES

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[2] M. Ankerst, G. Kastenmüller, H. Kriegel, and T. Seidl, "Nearest neighbor classification in 3d protein databases," in *Proc. of Int. Conf. on Intell. Syst. for Mol. Biol.*, 1999, pp. 34–43.

[3] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, 1999.

[4] D. V. Vranic, "DESIRE: a composite 3d-shape descriptor," in *Proc. of ICME*, 2005, pp. 962–965.

[5] X. Bai, S. Bai, Z. Zhu, and L. J. Latecki, "3d shape matching via two layer coding," *IEEE Trans. Pattern Anal. Mach. Intell. (Accepted)*, 2015.

[6] P. Papadakis, I. Pratikakis, T. Theoharis, and S. J. Perantonis, "PANORAMA: A 3d shape descriptor based on panoramic views for unsupervised 3d object retrieval," *Int. J. Comput. Vision*, vol. 89, no. 2-3, pp. 177–192, 2010.

[7] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung, "On visual similarity based 3d model retrieval," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 223–232, 2003.

[8] X. Bai, C. Rao, and X. Wang, "Shape vocabulary: A robust and efficient shape representation for shape matching," *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3935–3949, 2014.

[9] B. Song, B. Xiang, L. Wenyu, and R. Fabio, "Neural shape codes for 3d model retrieval," *Pattern Recognition Letters*, 2015.

[10] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai, "Deep learning representation using autoencoder for 3d shape retrieval."

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. of NIPS*, 2012, pp. 1106–1114.

[12] Y. Xu, J. Du, L. Dai, and C. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 65–68, 2014.

[13] F. Richardson, D. A. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1671–1675, 2015.

[14] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proc. of CVPR*, 2015, pp. 1912–1920.

[15] "3d warehouse," https://3dwarehouse.sketchup.com/.

[16] Y. LeCun, L. Bottou, G. B. Orr, and K. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade - Second Edition*, 2012, pp. 9–48.

[17] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. of VISAPP*, 2009, pp. 331–340.

[18] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[19] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, no. EPFL-CONF-192376, 2011.

[20] M. M. Kazhdan, T. A. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3d shape descriptors," in *Proc. of SGP*, 2003, pp. 156–164.

[21] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *CoRR*, vol. abs/1507.05717, 2015.

[22] D. Tao, X. Lin, L. Jin, and X. Li, "Principal component 2-d long short-term memory for font recognition on single chinese characters," *IEEE Transactions on Cybernetics*, to appear in 2015.

[23] J. Ma, W. Qiu, J. Zhao, Y. Ma, A. L. Yuille, and Z. Tu, "Robust L_2E estimation of transformation for non-rigid registration," *IEEE Transactions on Signal Processing*, vol. 63, no. 5, pp. 1115–1129, 2015.

[24] J. Ma, Y. Ma, J. Zhao, and J. Tian, "Image feature matching via progressive vector field consensus," *IEEE Signal Processing Letters*, vol. 22, no. 6, pp. 767–771, 2015.