Journal of Multimedia Information System VOL. 10, NO. 3, September 2023 (pp. 207-214): ISSN 2383-7632 (Online)
https://doi.org/10.33851/JMIS.2023.10.3.207

Decision-Making for Multi-View Single Object Detection with Graph Convolutional Networks

Ren Wang[1], Tae Sung Kim[2], Tae-Ho Lee[1], Jin-Sung Kim[2*], Hyuk-Jae Lee[1]

Abstract

Aggregating predicted outputs from multiple views helps boost multi-view single object detection performance. Decision-making strategies are flexible to perform this result-level aggregation. However, the relationship among multiple views is not exploited in aggregation. This study proposes a novel decision-making model with graph convolutional networks (DM-GCN) to address this issue by establishing a relationship among predicted outputs with graph convolutional networks. Through training, the proposed DM-GCN learns to make a correct decision by enhancing the contributions from informative views. DM-GCN is light, fast, and can be applied to any object detector with a negligible computational cost. Moreover, a real captured dataset named Yogurt10 with a new metric is proposed to investigate the performance of DM-GCN in the multi-view single object detection task. Experimental results show that DM-GCN achieves superior performance compared to classical decision-making strategies. A visual explanation is also provided to interpret how DM-GCN makes a correct decision.

Key Words: Decision-Making Strategies, Multi-View Single Object Detection, Graph Convolutional Networks.

I. INTRODUCTION

Multi-view representation is the most intuitive and the closest to human perception for understanding 3-dimensional (3D) objects. Since the appearance of a 3D object may change considerably depending on viewpoints, a single view of an object often cannot give sufficient information for classifying the object. When the initial view of an object is very similar to different objects, humans will check other views until an informative view is found. Multi-view-based object detection methods should consider the different contributions from informative and uninformative views, especially when most of the views of an object are highly similar to the other object and only a few views are informative. Therefore, the challenge of multi-view representation is how to make a correct decision by utilizing the informative view when the similarity between different objects leads to serious confusion.

Classical decision-making strategies [1] are commonly used for aggregating detection results from multiple views. However, the contributions of informative and uninformative views have yet to be well-considered in aggregation. This study proposes a novel decision-making model with graph convolutional networks (DM-GCN) that aggregates outputs predicted from multiple viewpoints while considering the contribution of each view. In the proposed method, the output from a more informative view has more influence on the decision than that from a less informative view. Outputs are represented in the non-Euclidean space and aggregated with graph convolutional networks (GCN) [2]. Each predicted output is defined as a node in the input graph, and a relationship is built among the nodes based on the class labels and views. By leveraging the relationship, DM-GCN enhances the contributions of informative views when making a decision. Since the input graph only contains predicted class labels and confidence scores, DM-GCN can be applied to any object detector [3-5].

To investigate the performance of DM-GCN, a real captured multi-view single object detection dataset named Yogurt10 is proposed. Yogurt10 consists of 10 'Yogurt' products with high similarity in shape, size, color, and texture. In addition, a new evaluation metric is proposed to evaluate the performance fairly. Experimental results show that DM-GCN outperforms classical decision-making strategies. It is light and has an insignificant computational cost. The inference latency of a 3-layer DM-GCN is only 0.2 ms on GPU and can be negligible compared to object detectors' latency. Moreover, a visual explanation method is provided to inter-

Manuscript received July 09, 2023; Revised August 01, 2023; Accepted August 02, 2023. (ID No. JMIS- 23M-07-028)
Corresponding Author (*): Jin-Sung Kim, +82-41-530-2232, jinsungk@sunmoon.ac.kr
[1]Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea, wangren@capp.snu.ac.kr, taehov@capp.snu.ac.kr, hjlee@capp.snu.ac.kr
[2]Department of Electronic Engineering, Sunmoon University, Asan, Korea, ts7kim@sunmoon.ac.kr, jinsungk@sunmoon.ac.kr

pret how DM-GCN gives correct multi-view single object detection results.

The rest of this paper is organized as follows. Section 2 briefly reviews classical decision-making strategies and the background of graph convolutional networks. The proposed DM-GCN model is presented in Section 3. Section 4 shows the experimental results of the proposed work. Section 5 concludes this paper.

II. RELATED WORK

2.1. Decision-Making Strategies

Decision-making strategies can be categorized into voting-based rules and score fusion methods. Voting-based rules only consider class labels predicted by the models. The majority voting chooses the class with the highest number of votes. Borda count sorts the classes of each model and assigns different votes to them according to their ranks. The summation of votes across all the models is obtained, and the class with the most votes becomes the final decision. Behavior-Knowledge Space [6] constructs a look-up table to record the frequency of each label combination produced by the models based on the training data. The class with the highest frequency is selected according to the look-up table during the test. Score fusion methods consider both predicted class labels and scores. Algebraic combiners perform the mean, max, or product operations. These methods are simple, non-trainable, and widely adopted in the palmprint recognition task [7-8]. Decision Templates (DT) [9] average the decision profiles of each class based on the training data. The final decision is the class with the highest similarity between decision templates and the decision profile constructed from a test instance. However, these decision-making strategies do not exploit the relationship among multiple views.

2.2. Graph Convolutional Networks

Kipf and Welling [2] introduce graph convolutional networks for graph-based semi-supervised classification. The proposed layer-wise propagation rule is motivated by a localized first-order approximation of spectral graph convolutions. Some further works based on GCN are proposed for many computer vision tasks. Wang and Gupta [10] define the object region proposals from different frames as graph nodes for video action recognition. Yan et al. [11] propose a spatial-temporal GCN to learn spatial and temporal patterns from data for skeleton-based action recognition. Yang et al. [12] construct a scene graph to capture contextual information between objects and their relations via an attentional GCN.

III. PROPOSED METHOD

3.1. Graph Generation

A graph is represented as $G = (V, E)$, where V is the set of nodes, and E is the set of edges. v_i denotes a node and e_{ij} denotes an edge between node v_i and v_j. The adjacency matrix A is a $n \times n$ matrix, where n is the number of nodes. $A_{ij} = 1$ if $e_{ij} \in E$ and otherwise $A_{ij} = 0$. $x_v \in \mathbb{R}^n$ is the feature vector of the node v, where d is the dimension of x_v. $X \in \mathbb{R}^{n \times d}$ represents a node feature matrix.

This study constructs an undirected graph for multi-view single object detection. Each predicted output from an object detector is defined as a node v. The nodes with input scores lower than a threshold s_t are discarded to reduce the computational complexity. e_{ij} is given if node v_i and v_j a re generated from the same view or have the same class label. Otherwise, nodes v_i and v_j are disconnected. Therefore, the adjacency matrix is built as follows:

$$A_{ij} = \begin{cases} 1, & if\ c_i = c_j\ or\ c_i, \quad c_j \in p_k \\ 0, & otherwise \end{cases}. \tag{1}$$

Here, c_i denotes the input class label of the node v_i, and p_k denotes the kth view. x_v contains a confidence score and a class label. A confidence score is normalized between 0 and 1, and the class label is represented in a one-hot manner. Hence, $d = 1 + C$, where C is the number of categories.

3.2. DM-GCN

This study designs three models with different depths as shown in Fig. 1. Following the work of Kipf and Welling [2], the graph convolutional layer is defined as:

$$H^{l+1}(X, A) = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^l(X, A)W^l). \tag{2}$$

Here, $\tilde{A} = A + I_N$ is the adjacency matrix with added self-connections, where I_N is the identity matrix. $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. H^l is the matrix of the activations at the l^{th} layer and

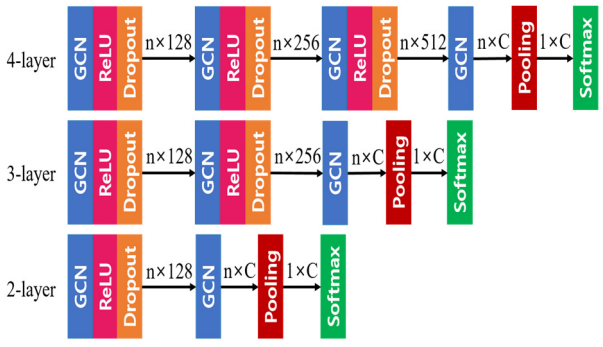

Fig. 1. Illustration of three DM-GCN networks with different depths. The input dimension is $n \times (1 + C)$ and the output dimension of each layer is shown above the arrow.

$H^0 = X$. W^l is a layer-specific trainable weight matrix. σ is an elementwise nonlinear activation function. Let GCN denote a graph convolutional layer without an activation function. σ = ReLU. Dropout [13] with a probability of 0.5 is added after the activation operation to prevent networks from overfitting. Pooling denotes the mean pooling layer and performs a readout operation that generates a graph-level representation based on node-level representations, which can be expressed as:

$$H^m = \frac{1}{n}\sum_{i=1}^{n} H_i^z\,(X, A). \tag{3}$$

Here, H^m is the feature matrix after the mean pooling operation. z is the index of the last graph convolutional layer. The standard cross entropy loss provided by the PyTorch library is adopted as the objective function for the optimization.

3.3. Interpretability of DM-GCN

A visual explanation method is provided to interpret DM-GCN based on Grad-CAM [14]. To obtain the heat map on the input graph, a class-specific weight α^c for the f^{th} feature at layer l is first obtained by

$$\alpha^c = \frac{1}{n}\sum_{i=1}^{n} \frac{\partial y_c}{\partial H_{f,i}^l(X,A)}, \tag{4}$$

where y_c is the score for class c after the softmax layer. Thereafter, the class-specific heat map G^c is obtained by

$$G^c[l,i] = \sigma\left(\sum_{f=1}^{d^l} \alpha^c\, H_{f,i}^l(X,A)\right), \tag{5}$$

where i is the index of the node v. The contribution of the node v_i to the decision of class c is obtained by

$$v_i^c = G^c[l,i] \times s_i, \tag{6}$$

where s_i is the input confidence score of the node v_i.

IV. EXPERIMENTS

4.1. Dataset and Evaluation Metric

Yogurt10 is a real captured dataset proposed for the multi-view single object detection task. As shown in Fig. 2, Yogurt10 comprises two groups of remarkably similar-looking products. Each group contains five products with different flavors of the same brand, which are slightly different in color, text, and patterns. This work collects 500

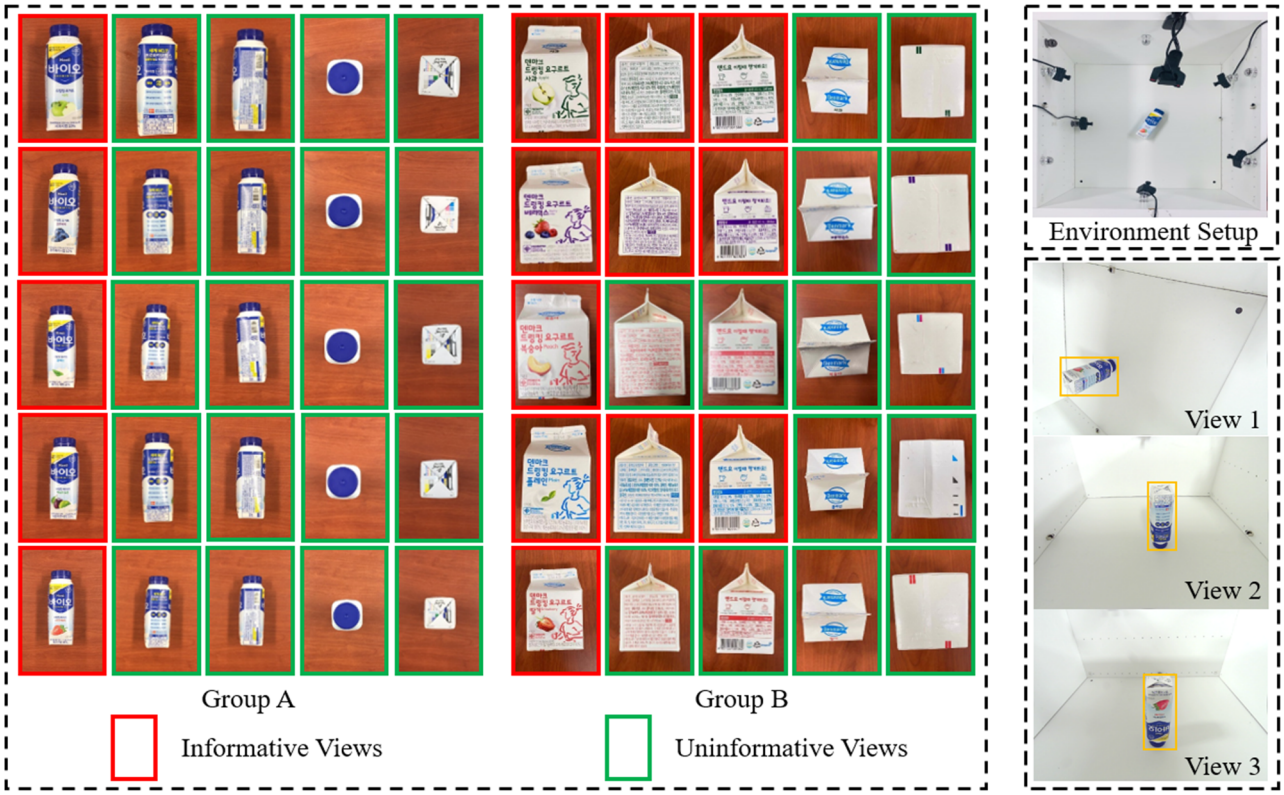

Fig. 2. Illustration of ten yogurt objects and the proposed Yogurt10 dataset. Group A and Group B show 'Yogurt' products. Each row in each group shows five representative views of an object, and different rows mean different objects. Group A: 'AApple', 'ABlueberry', 'APlain', 'APlum', and 'AStrawberry'; Group B: 'BApple', 'BBerry', 'BPeach', 'BPlain', and 'BStrawberry'. Informative views are marked with red rectangles, while uninformative views are marked with green ones. The right top part shows the environment setup for collecting images, and the right bottom part provides an example set of multi-view images for the 'AStrawberry' object.

informative and 3,000 uninformative single-view images with a resolution of 640×480 by multiple randomly placed cameras in a white environment. Two experts annotate the bounding boxes and perform three rounds of double-checking. Subsequently, 1,000 sets of multi-view images are generated. Each set of multi-view images contains three random views of an object, including at least one informative view, to ensure a human-level accuracy of 100%. Each class has the same number of samples: 50 single-view images for fine-tuning the network and 40/60 train/test sets for validating multi-view single object detection methods.

A new evaluation metric is proposed by combining the average precision (AP) and the localization intersection over union (IoU) metric. Specifically, a correct multi-view detection should satisfy two criteria: a correct class label and the average IoU (mIoU) over multiple views is higher than a threshold θ, where $\theta \in [0.50, 0.95]$ with a uniform step size of 0.05. Here, only the bounding box with the maximum confidence score in each view is considered during localization performance evaluation. In this experiment, mvAP_{50}, mvAP_{75}, and mvAP are used as the evaluation metrics, where mvAP is obtained by averaging over all ten mIoU thresholds.

4.2. Training Details

The experiments are performed on one NVIDIA Titan Xp GPU and Intel(R) Xeon(R) Gold 5118 CPU @2.30 GHz. All the models are implemented in PyTorch and trained with CUDA 9.0 and cuDNN 7 as computational back-ends. YOLOv3 [10] is used as the object detector to extract predicted outputs. The post-processing step non-maximum suppression is removed to avoid the loss of detection candidates. The predicted outputs are constructed as the input graphs and adjacency matrices for DM-GCN. DM-GCN is trained for 200 epochs with a warm-up epoch of 4 and a batch size of 1. The initial learning rate is set to 0.001 and gradually decreased to 0 through the cosine annealing strategy. The SGD optimizer is used with a momentum of 0.9 and a weight decay of 0.0005. The input score threshold s_t is set to 0.1, which approximately ensures a recall of 100.

4.3. Results

Table 1 shows the results of four classical decision-making strategies and proposed 3-layer DM-GCN on the Yogurt10 dataset. DM-GCN achieves the best performance and has a significant improvement over the baselines. The accuracy gains are 1.9%, 1.8%, and 1.3% compared with the max rule on mvAP_{50}, mvAP_{75}, and mvAP, respectively. The improvement indicates that DM-GCN gives more weight to outputs from the informative views. For computational efficiency, the model size is independent of the number of views and has only 0.05 million trainable parameters. With an input image size of 640, the inference latency of 3-view YOLOv3 is approximately 104 ms. As a result-level aggregation model, DM-GCN only increases 0.2 ms latency and thus remains the real-time performance of the detector.

Table 1. Comparison with classical decision-making strategies on the Yogurt10 dataset. Results are reported in percentage.

Method	mvAP_{50}	mvAP_{75}	mvAP
Majority voting	90.7	88.5	68.3
Mean rule	92.8	90.7	69.8
Max rule	94.8	92.7	71.3
DT	94.0	91.8	70.7
DM-GCN	**96.7**	**94.5**	**72.6**

4.4. Ablation Studies

4.4.1. Influence of the Pooling Layer

Compared to the mean pooling layer in DM-GCN, using the max pooling layer degrades the performance and only has 95.8% mvAP_{50}. The accuracy drop indicates that DM-GCN prefers to make a correct decision based on the global information of all the nodes rather than the local information of a single node.

4.4.2. Influence of the Depth

The 2-layer DM-GCN has 96.0% mvAP_{50}, 0.8% lower than the performance of the 3-layer one. As a deeper model, the 4-layer one only has a 0.1% accuracy improvement than the 3-layer one, indicating that a deeper model may result in an overfitting problem owing to more propagation among nodes.

4.4.3. Influence of the Threshold

The input score threshold $s_t \in [0.1, 0.9]$ with a uniform step size of 0.1 is applied during the test. To fairly check the performance of DM-GCN, the trained weight obtained at $s_t = 0.1$ is used. As shown in Fig. 3, DM-GCN outperforms the other four strategies under different thresholds. With the increase of s_t, the number of miss detections increases, and mIoU decreases, causing a drop in accuracy. Note that when keeping the same training data, the performance of trainable combination rule DT severely declines with the increase of s_t while DM-GCN is much more robust.

4.5. Visual Explanations

Fig. 4 shows an example set of 3-view images for the 'BStrawberry' object. As shown in Fig. 2, 'BPeach' is highly similar to 'BStrawberry'. In this example, View 1 is informative, while Views 2 and 3 are uninformative. Yellow and orange nodes in the input graph represent 'BPeach'

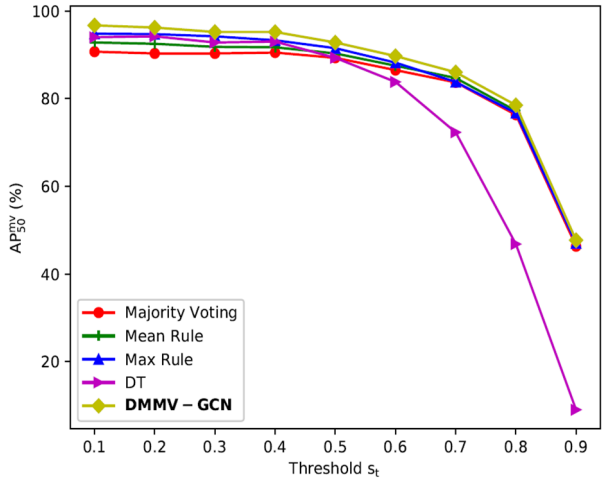

Fig. 3. Influence of the input score threshold s_t.

and 'BStrawberry', respectively. The node size in the input graph and heat map represents its input score and class-specific contribution, respectively. Here, the sum of predicted confidence scores of 'BPeach' is larger than that of 'BStrawberry', and the mean rule will make a wrong decision. Unlike the mean rule, DM-GCN obtains a higher confidence score of 'BStrawberry' than 'BPeach' via a weighted combination of outputs by building the relation among them.

The heat map illustrates that DM-GCN can automatically distinguish which outputs are helpful for the final decision. For the 'BStrawberry' decision made by DM-GCN, all the nodes with the 'BPeach' label are eliminated, indicating that the outputs with a certain class label have no contribution to the decision with a different class label. Moreover, when a single view has outputs with different class labels, DM-GCN will inhibit the contributions of each output. In this example, it is mainly reflected in the reduced sizes of nodes from View 3. Since the outputs from Views 1 and 2 only contain one class label, the node sizes remain the same as those in the input graph. DM-GCN utilizes identical input scores as the mean rule. However, the proposed method makes a correct decision, indicating that the weights for the input scores in View 1 increase in the combination. The process is similar to human perception because when humans recognize the object from these three views, View 1 will play a more critical role in their understanding. The result shows that DM-GCN enhances the contribution of the informative view when making a decision.

V. CONCLUSION

This study proposes a novel decision-making model with graph convolutional networks for multi-view single object detection. The proposed model is light, fast, and performs excellently on the Yogurt10 dataset. DM-GCN can be applied to any 2D object detector, which enables real-world applications. For future work, DM-GCN will be investigated on multi-view multi-object detection tasks.

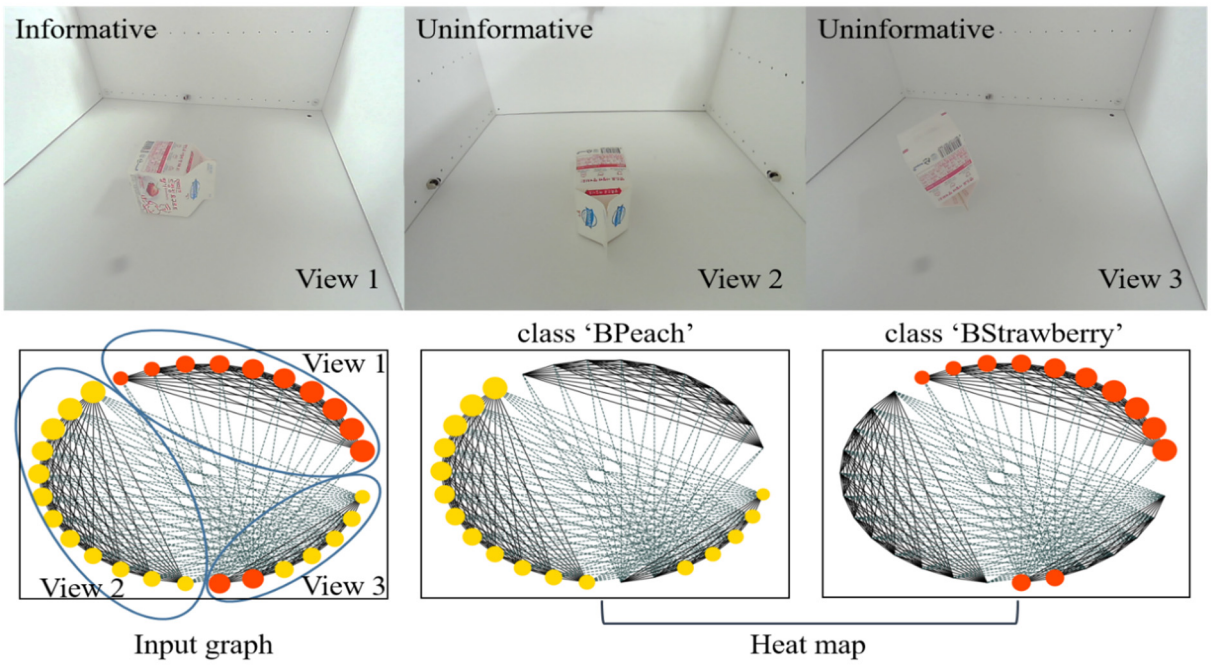

Fig. 4. The circular patterns represent nodes of the input graph, and the colors of nodes mean input class labels of nodes. The solid black line represents an edge connection between two nodes from the same view. The dashed gray line represents an edge connection between two nodes from different views with the same class label.

ACKNOWLEDGMENT

This work was supported by the Sun Moon University Research Grant of 2022.

REFERENCES

[1] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and Systems Magazine*, vol. 6, no. 3, pp. 1-45, 2006.

[2] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the International Conference on Learning Representations,* 2017.

[3] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv Prep. arXiv:1804.02767*, 2018.

[4] Y. Zhang, J. Chu, L. Leng, and J. Miao, "Mask-refined R-CNN: A network for refining object details in instance segmentation," *Sensors*, vol. 20, no. 4, p. 1010, 2020.

[5] J. Chu, Z. Guo, and L. Leng, "Object detection based on multi-layer convolution feature fusion and online hard example mining," *IEEE access*, vol. 6, pp. 19959-19967, 2018.

[6] Y. S. Huang and C. Y. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90-94, 1995.

[7] Z. Yang, L. Leng, and W. Min, "Extreme downsampling and joint feature for coding-based palmprint recognition," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-12, 2020.

[8] L. Leng, Z. Yang, and W. Min, "Democratic voting downsampling for coding-based palmprint recognition," *IET Biometrics*, vol. 9, no. 6, pp. 290-296, 2020.

[9] L. I. Kuncheva, J. C. Bezdek, and R. P. Duin, "Decision templates for multiple classifier fusion: An experimental comparison," *Pattern Recognition*, vol. 34, no. 2, pp. 299-314, 2001.

[10] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proceedings of the European Conference on Computer Vision*, 2018.

[11] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[12] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in *Proceedings of the European Conference on Computer Vision*, 2018.

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

[14] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *Proceedings of the International Conference on Computer Vision*, 2017.

AUTHORS

Ren Wang received the B.S. degree in electrical and computer engineering from Seoul National University, Seoul, Korea, in 2018. He is currently working toward the integrated M.S. and Ph.D. degrees in electrical and computer engineering at Seoul National University, Seoul, Korea. His current research interests include computer vision.

Tae Sung Kim received the B.S degree in electrical electronic engineering from Pusan National University, Pusan, Korea, in 2010 and the M.S. and Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, Korea, in 2013 and 2017. From 2018 to 2021, he worked at the System LSI Division, Samsung Electronics Corporation, Hwaseong, Korea as a staff engineer, where he was involved in image/video processor design development. In 2021, he joined the Department of Electronic Engineering at Sunmoon University, Korea, where he is currently working as an assistant professor. His research interests include the algorithm and architecture design of AV1 and VVC, and deep learning based image/video coding systems.

Tae-Ho Lee received the B.S., M.S., degrees in electronic, electrical, control & instrumentation engineering from the Hanyang University, in 2001 and 2003, respectively. and He received the Ph.D. degrees in electrical & computer engineering from the Seoul National University, Seoul, Korea, in 2018. From 2018 to 2021, he was a Post-Doctoral Researcher with Seoul National University. From 2021 until now, he is a Visiting Assistant Professor in the Next-Generation Semiconductor Convergence and Open Sharing System, Seoul National University. Fields of interest are deep learning object detection, hand gesture recognition, GAN based medical image registration for augmented reality applications.

Jin-Sung Kim received his B.S. and M.S. degrees and his PhD degrees in electrical engineering and computer science from Seoul National University, Seoul, Korea, in 1996, 1998, and 2009, respectively. From 1998 to 2004, and from 2009 to 2010, he was with Samsung SDI Ltd., Cheonan, Korea, as a senior researcher, where he was involved in driver circuits and discharge waveform research. From 2010 to 2011, he was a post-doctoral researcher at Seoul National University. In 2011, he joined the Department of Electronic Engineering, Sun Moon University, Asan, Korea, where he is currently a Professor. His current research interests include deep learning, algorithms and architectures for video compression and computer vision.

Hyuk-Jae Lee (Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Korea, in 1987 and 1989, respectively, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 1996. From 1996 to 1998, he was with the Faculty of the Department of Computer Science, Louisiana Tech University, Ruston, LA, USA. From 1998 to 2001, he was with the Server and Workstation Chipset Division, Intel Corporation, Hillsboro, OR, USA, as a Senior Component Design Engineer. In 2001, he joined the School of Electrical Engineering and Computer Science, Seoul National University, Korea, where he is currently a Professor. He is the Founder of Mamurian Design, Inc., a fabless SoC design house for multimedia applications. His research interests include the areas of computer architecture and SoC design for multimedia applications.