Rotation Ensemble Module for Detecting Rotation-Invariant Features

Dongwon Park, Yonghyeok Seo, Se Young Chun
Ulsan National Institute of Science and Technology (UNIST), Republic of Korea
{dong1, syh4661, sychun}@unist.ac.kr

Abstract

Deep learning has improved many computer vision tasks by utilizing data-driven features instead of using hand-crafted features. However, geometric transformations of input images often degrade the performance of deep learning based methods. In particular, rotation-invariant features are important in computer vision tasks such as face detection, biological feature detection of microscopy images, or robot grasp detection since the input image can be fed into the network with any rotation angle. In this paper, we propose rotation ensemble module (REM) to efficiently train and utilize rotation-invariant features in a deep neural network for computer vision tasks. We evaluated our proposed REM with face detection tasks on FDDB dataset, robotic grasp detection tasks on Cornell dataset, and real robotic grasp tasks with several novel objects. REM based face detection deep neural networks yielded up to 50.8% accuracy in face detection task on FDDB dataset at false rate 20 with IOU 75%, which is about 10.7% higher than the baseline. Robotic grasp detection deep neural networks with our REM also yielded up to 97.6% accuracy in robotic grasp detection on Cornell dataset that is higher than current state-of-the-art performance. In robotic grasp task using a real 4-axis robotic arm with several novel objects, our REM based robotic grasp achieved up to 93.8%, which is significantly higher than the baseline robotic grasps (11.0-56.3%).

1. Introduction

Rotation invariance has been an important topic in many computer vision tasks such as face detection [24], texture classification [5], and character recognition [10], to name a few. Recently, deep learning based approaches have significantly improved many computer vision tasks by utilizing data-driven features instead of using hand-crafted features [12]. However, the importance of rotation invariant properties for computer vision methods still remains. Face detection using deep learning still requires to deal with rotated faces. Robot grasp detection using deep learning also requires to detect robotic grasps regardless of the direction of an object. For pathology images of hospitals, a specimen on a slide can be placed in any direction. Therefore, rotation-invariant feature detectors can greatly boost the performance of various computer vision tasks including face detection and robotic grasp detection.

There have been several deep learning based computer vision works considering rotation invariance. Current CNN models are inefficient in terms of rotation invariance because they require many parameters to train with data augmentation with rotations. Max pooling in CNN often helps alleviating this issue, but since it is usually $2 \times 2$, it is only for images rotated with very small angles. In general, there are a few strategies for dealing with rotation variations including image transformation, weight rotation, enlarged receptive field, and discrete prediction.

Spatial transformer network (STN) and its variant were proposed to transform input images or feature maps for better classification accuracies [8, 15]. There are also methods to use rotation region proposals for recognizing arbitrarily placed texts [17] and to use the polar transform of an input image to extract rotation-invariant features [3]. There are also methods to rotate network weights for rotation-invariance [2, 4]. There are methods to enlarge the receptive field of deep neural networks using dialed CNN [26] or a pyramid pooling layer [7]. There is a way to predict ro-
tations discretely at intervals of 10 degrees. Most previous works are based on rotating an image or a feature map. However, the method of rotating images does not get much efficiency for multiple objects with different angles in one image due to time-consuming computations.

We propose a rotation-invariant rotation ensemble module (REM) using a convolution that bi-linearly rotates network weights. The angular probability for each grid is used as a weight to the feature map that may have a similar effect to the region proposal. Also, since weights are rotated, the same rotation invariant operations are applied during the test time via four networks that pass through. Our proposed “REM” can be plugged into any deep neural network for classification and/or regression. In this paper, we demonstrated the strength of our proposed “REM” by using it with various deep neural networks for predicting rotations by classification, rotation angles by regression and so on that are mostly based on YOLO9000. The angle stepsize for REM was $\pi/4$ and the angle stepsize for the final rotation classification was set to $\pi/18$.

We evaluated our proposed REM for three different tasks: face detection with rotations on a modified version of the FDDB (Face Detection Data set and Benchmark) database with rotations (we will call it rotated FDDB dataset), robotic grasp detection on the Cornell robotic grasp dataset and real robotic grasping tasks with novel objects that were not used in training deep neural networks. Our proposed method was able to perform multi-face detections in blurry environment as shown in Figure 1 (Left) and multi-object, multi-grasp detections as shown in Figure 1 (Right). Our proposed methods yielded up to 89.6% accuracy at false rate 20 with IOU 50% and up to 50.8% accuracy at false rate 20 with IOU 75% for the rotated FDDB dataset. In particular, for false rate 20 with IOU 75%, our proposed REM was able to achieve more than 10% higher accuracy than the baseline and almost 20% higher than the case of using RBF module.

Our proposed REM was also able to achieve up to 97.6% accuracy for the Cornell robotic grasp dataset, which is higher than the state-of-the-art performance of the work of Chu et al. 96.4%. However, our proposed method was able to perform robotic grasp detection task 20 times faster than the work of Chu et al.. In addition, our proposed methods were able to yield up to 3.8% success rate for the real-time robotic grasping tasks with a 4-axis robot arm with novel objects. Since this small robot has a gripper with a maximum range of 27.5 mm, it was critical to use accurate rotation-invariant feature information. Figure 2 summarizes the performances of our proposed method and our in-house implementations of other previous methods (Lenz, Redmon, Ours Reg, Ours Cls).

2. Related work

Efficient parameter utilization. In general, max pooling layers in convolutional neural networks (CNN) are required to alleviate the issue of spatial variance in CNN. Assuming that spatial invariance is important for image classification, Jaderberg et al. proposed spatial transformer network (STN), a method of image (or feature) transformation by learning (affine) transformation parameters so that it can help to improve the performance of inference operations of the following neural network layers. Lin et al. proposed to use STN repeatedly with an inverse composite method by propagating warp parameters rather than image intensities (or features) unlike the original STN and it yielded improved performance over STN.

Esteves et al. proposed a rotation-invariant network by replacing the grid generation part in STN with a polar transform. They transformed the input feature map (or image) into the polar coordinate with the origin that was determined by the center of mass. They showed that polar coordinate allows to predict parameters more stably than the original STN. Cohen and Welling proposed a method to use group equivariant convolutions and group pooling with weight flipping and four rotations with $\pi/2$ stepsize. Patrick et al. proposed to use rotational invariant features that were created using rotational convolutions and pooling layers. By back-rotating features, they proposed a way to return negative angles. Diego et al. proposed RotEqNet with a different set of weights for each local window, without weight rotation.

Detection. Faster-RCNN is a method of reducing region proposal time by using a region proposal network for generating region proposals. YOLO was proposed as a model that is faster but less accurate than the faster R-CNN.
by a way of predicting \( \{x, y, w, h, \text{class}\} \) directly without using the region proposal network [21]. YOLO9000 stabilized the loss by using anchor box inspired by the region proposal network, called Darknet [22]. YOLO9000 yielded much faster object detection outputs than faster R-CNN while its accuracy was comparable to faster R-CNN.

In terms of rotation-invariant object (or face) detection, Shi et al. performed face detection using the progressive calibration network method, which predicted the rotation by \( 180^\circ, 90^\circ, \text{and} [-45^\circ, 45^\circ] \) values after using sliding window [25]. Ma et al. used a rotation region proposal network to transform text regions for classification [17]. After generating proposals reflecting rotations, classification was performed using rotation region-of-interest (ROI) pooling. In summary, for the case of rotations, the rotation angle was predicted using 1) rotation anchor box, 2) direct regression, or 3) angle classification.

For robotic grasp detections, Lenz et al. proposed a model that can predict whether it is graspable or not using a proposed sparse auto-encode (SAE) with sliding windows [14]. Redmon et al. proposed a deep learning regression based robotic grasp detection model based on AlexNet [20]. Recently, Chu et al. proposed to use the region proposal network based on faster R-CNN and a method of predicting angles by classification [11].

3. Method

3.1. Problem setup and reparametrization

The goal of the problem is to predict 5D representations for multiple objects from a color image where a 5D representation consists of location \((x, y)\), orientation \(\theta\), width \(w\), and height \(h\), as illustrated in Figure 3. Multi-object detection often directly estimates 5D representation \(\{x, y, \theta, w, h\}\) as well as its probability (confidence) of being a class (or being graspable) \(z\) for each grid cell. In summary, the 5D representations with its probability are as follows:

\[
\{x, y, \theta, w, h, z\}
\]

There are also region proposal networks based methods such as faster R-CNN [23] or rotation region proposal network [17] that firstly proposes arbitrary-oriented proposal and then adjusts the size using rotation ROI pooling before performing classification. Note that rotation region proposal network classifies rotation anchor boxes with \(\pi/6\) stepsizes first and then regresses angles later.

Inspired by YOLO9000 [22], a better network for object detection than YOLO [21], we propose to use the following reparametrization technique for 5D grasp representation and its probability for rotation object detection or robotic grasp detection as follows:

\[
\{t_i^x, t_i^y, \theta_i, t_i^w, t_i^h, t_i^z\}, \quad i \in \{1, 2\}
\]

where \(x = \sigma(t^x) + c_x, y = \sigma(t^y) + c_y, w = p_w \exp(t^w), h = p_h \exp(t^h)\) and \(z = \sigma(t^z)\). Note that \(\sigma(\cdot)\) is a sigmoid function, \(\exp(\cdot)\) is an exponential function, \(p_h, p_w\) are the predefined height and width of anchor box, respectively, and \((c_x, c_y)\) are the (known) position of the top left corner of each grid cell. Therefore, a deep neural network for rotation object detection or robotic grasp detection of the proposed methods directly estimates \(\{t_i^x, t_i^y, \theta_i, t_i^w, t_i^h, t_i^z\}\) instead of \(\{x, y, \theta, w, h, z\}\). Note that \(i\) is an index for each set of 5D detection representation since there may be more than 1 outlets to generate these parameter sets.

Note also that MultiGrasp that was using this 5D detection representation for robotic grasp detection reparameterized \(\theta\) to be \(c = \cos \theta, s = \sin \theta\) [20]. In other words, 7 parameters \(\{x, y, c, s, w, h, z\}\) were directly estimated using deep learning based regressors. This approach has been used in YOLO [21] for object detection. However, this approach is no longer used for YOLO9000, a better and faster deep neural network for object detection [22]. Thus, we chose not to use this reparametrization technique.

\((x, y)\) coordinates in each grid cell (offset). For \(S \times S\) grid cells, the following locations are defined

\[
(c_x, c_y) \in \{(c_x, c_y) | c_x, c_y \in \{0, 1, \ldots, S - 1\}\},
\]

which are the top left corner of each grid cell \((c_x, c_y)\). Thus, our proposed method does not predict the coordinates \((x, y)\) in a given image, but does estimates the \((x, y)\) offset from the top left corner of each grid cell \((c_x, c_y)\). Thus, for a given \((c_x, c_y)\), the range of \((x, y)\) will be

\[c_x < x < c_x + 1, \quad c_y < y < c_y + 1\]

due to the re-parametrization using sigmoid functions.
$w, h$ coordinates in each cell (anchor box). Anchor box approach has also been used for object detection such as YOLO9000 [22], so we adopt it to our rotation object detection or robotic grasp detection. Due to the re-parametrization using anchor box, estimating $w, h$ is converted into estimating $t_w, t_h$ that are related to the expected values of various sizes of $w, h$. Then, classification is performed to pick the best representation among all anchor box candidates. In other words, this re-parametrization changes regression problems for $w, h$ into regression & classification problems.

3.2. Rotation ensemble module

We propose a Rotation Ensemble Module (REM) using rotation convolution and rotation activation to determine the ensemble weight according to the angle class probability for each grid. REM is used in the latter part of the object detection (or grasp detection) network since rotated weights according to the angle class of each grid in REM become sensitive to the rotated input image, making efficient use of parameters. There is also an ensemble effect of angle if REM is used in the latter part of the network. A typical location for REM in deep neural networks for detection is illustrated in Figure 4 (a).

Rotation convolution. Consider a typical scenario of convolution with input feature maps $f \in \mathbb{R}^{H \times W \times C}$, where $N = H \times W$ is the number of pixels and $C$ is the number of channels. Let us denote $g_l \in \mathbb{R}^{K \times K \times C}$, $l = 1, \ldots, n_f$ a convolution kernel where $K$ are the spatial dimension of the kernel (i.e., $K \times K$ filter) and there are $n_f$ number of kernels in each channel. Let us only consider square shape of kernels for simplicity. Similar to the group convolutions [2], we propose to use $n_r$ rotations of the weights to obtain $n_f \cdot n_r$ rotated weights for each channel. We use bilinear interpolations of four adjacent pixel values for generating rotated kernels. A rotation matrix for 2D kernels can be defined as follows:

$$R(r) = \begin{bmatrix} \cos(r\pi/4) & -\sin(r\pi/4) & 0 \\ \sin(r\pi/4) & \cos(r\pi/4) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $r$ is an index for rotations. Then, the rotated weights (or kernels) are

$$g_{i}^{l} = R(i) \ast g_{l}, i = 1, \ldots, 4, l = 1, \ldots, n_f,$$

where $* \ast$ is a rotation operator with the rotation matrix $R(i)$. Finally, the output of these convolutional layer with rotation operators for the input $f$ is

$$z_{i}^{l} = g_{i}^{l} \ast f, i = 1, \ldots, 4, l = 1, \ldots, n_f.$$

This pipeline of convolutions with rotations is called rotation convolution.

Rotation activation. As illustrated in Figure 4 (b), we designed the REM to have a skip connection structure in the middle of the module to extract network feature map before rotation convolution. This feature map will be used to yield intermediate 5D detection representation prediction values $\{t^x_1, t^y_1, \theta_1, t^w_1, t^h_1\}$ for each grid. We simplify the regression problem of $\theta_1$ to the classification problem to choose one value among a discrete number of angle candidates in $[0, \pi]$. Specifically, we chose $\theta_1 \in \{0, \pi/4, \pi/2, 3\pi/4\}$. Then, we use all values of $\theta_1$ as activation functions for the output of the rotation convolution feature map for each grid.

We propose two rotation activation methods here. The first proposed model is to use angles only as follows:

$$h_l = \sum_{i=1}^{4} \theta^i_1 z^i_l / 4$$
where $\theta_i^z$ is a (activation) value of $\theta_i$ for the $i$th angle. The second model is to use the multiplication of angles and probabilities (confidences) as follows:

$$h_i = \sum_{i=1}^{4} t_i^z \theta_i^z z_i^2 / 4$$

for each grid. Note that there is only one value for $t^z$ for each grid. During the experiment, we observed that REM often decreased the probability of angle classes, especially for the objects with round shape, due to the averaging effect of the multiple possible angles to detect. Thus, we added 'circle class' (meaning no angle that is different from 0°) in angle classification for better performance.

**Initial bounding box for the final output.** In the REM, the intermediate output $\{t_i^x, t_i^y, \theta_i, t_i^w, t_i^h, t_i^z\}$ is used for rotation activation, but it still contains valuable, compressed information about the final output - it could be a good initial bounding box. Thus, we designed our proposed REM to decompress it using convolutions and then to concatenate it at the end of REM as illustrated in Figure 4(b). This pipeline delivers valuable information about $\{t_i^x, t_i^y, \theta_i, t_i^w, t_i^h, t_i^z\}$ indirectly to the final layer and we found that this structure seemed to decrease probability errors.

### 3.3. Loss function for detection with REM

We proposed a novel loss function for object or grasp detection with REM considering the following items.

**Angle in each grid (discretization).** MultiGrasp reparameterized the angle $\theta$ with $c = \cos \theta$ and $s = \sin \theta$ so that estimating $c, s$ yields the estimated $\theta = \arctan(s/c)$ [20]. However, as described in the previous sub-section, we chose to use $\theta$ itself based on YOLO9000. Just like the proposed method in REM, we propose to use classification based $\theta$ determination among finite number of angle candidates instead of conventional regression based $\theta$ estimation. Specifically, we modeled that the final angle $\theta_2 \in \{0, \pi/18, \ldots, 17\pi/18\}$. Along with data augmentation for different angles every epoch, we were able to observe substantial performance improvements by using angle discretization. Similar angle discretization for robotic grasp detection was also used in [6], but with different detection networks.

**Detection (grasp) probability (new ground truth).** Predicting detection (grasp) probability is crucial for multi-box approaches such as MultiGrasp [20]. Conventional ground truth for detection (grasp) probability was 1 (object or graspable) or 0 (not object or not graspable) [20]. Inspired by YOLO9000, we proposed to use IOU (Intersection Over Union) as the ground truth for detection (grasp) probability as follows:

$$z^g = \frac{|P \cap G|}{|P \cup G|}$$

where $P$ is the predicted detection (grasp) rectangle, $G$ is the ground truth detection (grasp) rectangle, and $| \cdot |$ is the area of the rectangle.

**Proposed loss function.** We propose the following loss function using both the intermediate output in the REM and the final output of the detection network to train deep neural networks for (grasp) detection that we will describe further in the next section. For the outputs of the deep neural network $\{t_i^x, t_i^y, \theta_i, t_i^w, t_i^h, t_i^z\}$ and the ground truth $\{x^g, y^g, \theta^g, w^g, h^g, z^g\}$, the total weighted loss is

$$\sum_{r=1}^{2} \rho L(t^x_r, t^y_r, \theta_r, t^w_r, t^h_r, t^z_r)$$

where

$$L(t^x, t^y, \theta, t^w, t^h, t^z) =$$

$$\lambda_{\text{cdd}} \sum_{i=1}^{S^2} \sum_{j=1}^{A} m_{ij}^{\text{obj}} [(z^g_i - z_i)^2 + (y^g_i - y_i)^2] +$$

$$\lambda_{\text{cdd}} \sum_{i=1}^{S^2} \sum_{j=1}^{A} m_{ij}^{\text{obj}} [(w^g_{ij} - w_{ij})^2 + (h^g_{ij} - h_{ij})^2] +$$

$$\lambda_{\text{prob}} \sum_{i=1}^{S^2} \sum_{j=1}^{A} m_{ij}^{\text{obj}} \text{CrossEntropy}(\theta^g_i, \theta_i)$$

where $x_i, y_i, w_{ij}, h_{ij}, z_i$ are functions of $(t^x, t^y, t^w, t^h, t^z)$, respectively, $S^2$ is the number of grid cells and $A$ is the number of anchor boxes (7 in our case). We set $\lambda_{\text{cdd}} = 1, \lambda_{\text{prob}} = 5$ and $\lambda_{\text{cls}} = 1$. We set $m_{ij} = 1$ if the ground truth $(x^g, y^g)$ is in the $i$th cell and $m_{ij} = 0$ otherwise.

### 4. Simulations and Experiments

In this section, we evaluated our proposed methods with REM on three datasets: rotated FDDB [9], Cornell robotic grasp dataset [13, 14] and real robot grasping task with novel objects. We also demonstrated the effectiveness of our proposed REM in terms of accuracy and computational speed time.

For rotated FDDB, we were not able to compare the results of our proposed methods with the results in other literature since the database is different and the evaluation criteria is also different. Thus, we compared our proposed methods with other methods by implementing all in-house such as Ours Reg (regression based detection that is similar to YOLO9000), Ours Cls (Ours Reg with angle classification for better performance), Ours Cls with RBF module that was proposed by [18] to increase the receptive field of object detection.
For robotic grasp detection and real grasping task, we compared our proposed REM based methods with other works such as Lenz et al. [14], Redmon et al. [20], as well as our in-house implementations such as Ours Reg, Ours Cls, and Ours Cls + RBF. Note that since Cornell dataset is common, we were able to perform comparisons based on literature, too.

4.1. Implementation details

Most of our proposed networks as well as other in-house implemented networks are based on the Darknet-19 [19]. We performed several simulations with or without individual ingredients (e.g., with REM, with RBF, with angle classification, with circle class for no angle) so that it becomes clear that what part will affect the performance of rotated object (or grasp) detection most significantly. We placed our proposed REM at the 3rd and the 6th layers from the end of the object detection network. We also placed the RBF [18] at the 6th layers from the end of the network. We also performed simulations with rotation activations using angle only and using angle and probability.

In the case of multiple face detection or multiple robotic grasps detection, boxes were plotted when the probability values were 0.25 or higher. All algorithms were tested on the platform with a single GPU (NVIDIA GeForce GTX1080Ti), a single CPU (Intel i7-7700K 4.20GHz) and 32GB memory. We implemented the work of Lenz et al. [14] and Redmon et al. [20] using TensorFlow. Note that the work of Redmon was based on the AlexNet. We implemented our proposed REM based networks, other in-house networks such as Ours Reg, Ours Cls, and Ours Cls with RBF using PyTorch.

4.2. Benchmark datasets

Rotated FDDB. (Figure 5a) FDDB dataset contains 5,171 labeled faces, which are collected from 2,845 news photographs [9]. FDDB contains large variances in facial appearance, skin color, facial expression, lighting, occlusion, and resolution. However, most faces in FDDB are upright since images in FDDB are collected from news photographs. To evaluate our proposed REM based face detection better, we created a new face database, called Rotated FDDB, by rotating images in FDDB dataset from -180° to 180°. Note that rotation could yield unrecognized face labels once the bounding box is outside the field of view of the image. Then, we randomly divided the Rotated FDDB dataset into 4:1 ratios for training and testing data, respectively. We measured accuracy by defining the successful face detection as follows: if IOU is 50% (or 75%) or larger, then it will be counted as a successful detection. Thus, note that IOU 75% measures more accurate successful detection than IOU 50%. Then, finally, the recall value of 20 failures was measured for accuracy as also used in [25].

Cornell robot grasp detection. (Figure 5b) Cornell robot grasp detection dataset [13,14] is often used for robot grasping. This dataset consists of 885 images (RGB color and depth) of 240 different objects with the ground truth labels of a few graspable rectangles and a few non-graspable rectangles. Note that we did not use depth information in the Cornell dataset, but only used RGB. We cropped the original image to make a 360 × 360 image and performed five-fold cross validation. Then, the mean prediction accuracy was reported for image-wise and object-wise splits. Image-wise split divides Cornell dataset into the train dataset and the test data with the ratio of 4:1 randomly without considering the same or different objects. Object-wise is a method of splitting train data and test data at a 4:1 ratio for each object. Successful grasp detection is defined as follows: if IOU is larger than a certain threshold (e.g., 0.25, 0.3) and the difference between the output orientation θ and the ground truth orientation θg is less than 30° (Jaccard index), then it will be considered as a successful grasp detection. The same metric for accuracy has been used in other previous
works [14, 20, 11].

**Real robot grasping task.** (Figure 5c) We also evaluated our proposed methods with a small 4-axis robot arm (Dobot Magician, Shenzhen YueJiang Tech Co., Ltd, China, Figure 7(i)) and a RGB-D camera (Intel RealSense D435, Intel, USA) attached so that our vision system can have a field-of-view including the robot and its workspace from the top. The following 8 novel objects were used for real grasping tasks as shown in Figure 5c (toothbrush, candy, earphone cap, cable, Styrofoam bowl, L-wrench, nipper, pencil). If the robot arm is able to hold an object for more than 3 sec, it is counted as a successful grasp.

5. Simulation and Experiment Results

5.1. Results on Rotated FDDB

Table 1 summarizes all evaluation results on the Rotated FDDB dataset for all our proposed and other methods. Our proposed methods yielded state-of-the-art performance, accuracies of up to 89.6% at IOU 50% and up to 50.8% at IOU 75%, respectively, compared to other methods such as Ours Reg, Ours Cls, and Ours Cls with RBF. Note that changing angle regression problem (Ours Reg) into angle classification problem (Ours Cls) was important to significantly improved the performance. Further Our proposed method with REM further improved the performance over Ours Cls. In particular, ‘ang×prob / 6’ REM model yielded the best performance at IOU 50%, while ‘ang×prob / 3’ REM model yielded the best performance at IOU 75%. Thus, it seems important to use probability information for rotation activation unit for better performance. ROC (Receiver Operating Characteristic) curves are illustrated in Figure 6, showing that ‘ang×prob / 6’ yielded the best performance over varying false positives.

Table 1: Accuracy (recall rate (%)) at 20 false positives) comparisons among different methods on Rotated FDDB.

| Method         | RBF [16] | false rate 20 |
|----------------|----------|---------------|
| Ours Reg       | X        | 88.0          |
| Ours Cls       | X        | 87.0          |
| Ours Cls+REM   | O        | 87.8          |
| ang×prob / 6   | X        | 89.6          |
| ang×prob / 3   | X        | 88.7          |
| ang / 6        | X        | 89.2          |
| ang / 3        | X        | 88.0          |

Figure 6: ROC curves of rotation-invariant face detectors on Rotated FDDB. The horizontal axis on the ROC is “false positives over the whole dataset”.

5.2. Results on Cornell Grasp Dataset

Table 2 summarizes all evaluation results on the Cornell robotic grasp dataset for all our implemented and proposed methods. Our proposed methods yielded state-of-the-art performance, up to 97.6% prediction accuracy for image-wise split and up to 97.0% for object-wise split, respectively, over reported accuracies of the work of Chu [11] and others. Note that our proposed methods yielded these state-of-the-art performances with the fastest computation time per image, 6 ms. These improvements were better as less tolerant metrics were used (e.g., IOU 35%). All proposed ingredients, 1) REM at the 6th layer from the end of the network, 2) rotation activation with probability, and 3) Circle Class for no angle label, seem to contribute to the improved detection performance especially for tough accuracy metric with IOU 35% and using all of them yielded the most favorable performance for different metrics.

Figure 7 (d), (f), and (h) illustrate robotic grasp detection results. It seems that all methods yielded relatively good grasp detections, but our proposed method (Ours Cls + REM) seems to predict the best robotic grasp information, close to the ground truth. Qualitatively, it was important to use Circle Class for accurate graspability prediction.

5.3. Results with 4-Axis Robot Arm

Figure 7(i) illustrates our robot grasp experiment with a novel “nipper” object using our proposed vision algorithm for grasp detection. Robot grasp detections were performed using grasp detection methods such as Redmon-like, Ours Reg, Ours Cls, and our proposed Ours Cls + REM as shown
in Figure 7(j). All methods have capability for multi-object multi-grasp detections as shown in Figure 7(k), (l), (m).

When real grasping tasks were performed with our 4-axis robot arm with small, novel objects, our proposed REM played an important role to detect reliable grasp locations and angle and our proposed Ours Cls + REM yielded 93.8% grasping success rate, which is 11% higher than previous state-of-the-art method, Ours Cls as shown in Table 3. We believe that it is because our REM helped to predict the best possible angle and accurate angle was important for real grasping tasks.

6. CONCLUSIONS

We proposed REM to efficiently train and utilize rotation-invariant features in a deep neural network for computer vision tasks. REM based face detection networks yielded up to 50.8% accuracy in face detection task on rotated FDDB dataset at false rate 20 with IOU 75%. Robotic grasp detection networks with our REM also yielded up to 97.6% accuracy in grasp detection on Cornell dataset. In real robotic grasp task, our REM based robotic grasp achieved up to 93.8%, which is much higher than baselines.
Table 2: Performance summary on Cornell dataset with IOU metric. Our proposed methods yielded state-of-the-art prediction accuracy in both image-wise and object-wise splits with state-of-the-art computation time. The unit for performance is %.

| Ang predict | REM   | Circle Class | RBF [10] | Image-wise | Object-wise | Time / image (ms) |
|-------------|-------|--------------|----------|------------|-------------|-------------------|
| Reg         | X     | X            | X        | 89.8 87.6 84.9 | 86.7 76.6 64.6 | |
| Cls         | X     | X            | X        | 96.4 93.6 90.6 | 94.1 91.3 86.5 | |
| Cls         | X     | O            |          | 97.1 94.8 91.1 | 95.7 90.5 89.3 | |
| Cls         | ang / 3 | X            | X        | 97.0 95.0 92.4 | 96.5 93.4 89.2 | 6 |
| Cls         | ang×prob / 3 | X            | X        | 97.3 95.1 92.0 | 96.1 93.3 88.1 | |
| Cls         | ang / 6 | X            | X        | 97.1 93.9 90.4 | 95.7 93.8 89.6 | |
| Cls         | ang×prob / 6 | X            | X        | 97.3 95.4 91.1 | 96.1 93.3 88.1 | |
| Cls         | ang / 6 | O            | X        | 97.6 95.8 92.5 | 97.0 95.0 91.2 | |
| Cls         | ang×prob / 6 | O            | X        | 97.1 95.4 92.8 | 96.9 95.2 92.0 | |

Table 3: Performance summary of real robotic grasping for 8 novel, small objects with 8 repetitions. For Redmon and Lenz, our in-house implementations (modifications) were used after validating their performance with the Cornell dataset. Ours Reg is a model that predicts rotation as regression. Ours Cls is a model that predicts angle by classification in $\pi/18$ units. Ours Cls + REM placed REM at the 6th layer from the end of the network, used ang×prob for rotation activation, and also used Circle Class as no angle label.

| Object       | Lenz* | Redmon* | Ours Reg* | Ours Cls* | Ours Cls + REM |
|--------------|-------|---------|-----------|-----------|----------------|
| toothbrush   | 80.0% | 87.5%   | 62.5%     | 100%      | 100%           |
| candy        | 0.0%  | 37.5%   | 0.0%      | 75.0%     | 100%           |
| earphone cap | 40.0% | 25.0%   | 37.5%     | 75.0%     | 87.5%          |
| cable        | 0.0%  | 0.0%    | 37.5%     | 75.0%     | 87.5%          |
| styrofoam bowl| 0.0%  | 37.5%   | 37.5%     | 87.5%     | 87.5%          |
| L-wrench     | 80.0% | 75.0%   | 62.5%     | 75.0%     | 100%           |
| nipper       | 20.0% | 12.5%   | 0.0%      | 62.5%     | 75.0%          |
| pencil       | 100%  | 37.5%   | 37.5%     | 100%      | 100%           |
| Average      | 40.0% | 39.1%   | 37.5%     | 82.8%     | 93.8%          |

Acknowledgments

This work was supported partly by the Technology Innovation Program or Industrial Strategic Technology Development Program (10077533, Development of robotic manipulation algorithm for grasping/assembling with the machine learning using visual and tactile sensing information) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea) and partly by a grant of the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (grant number : HI18C0316).

References

[1] F.-J. Chu, R. Xu, and P. A. Vela. Real-World Multiobject, Multigrasp Detection. *IEEE Robotics and Automation Letters*, 3(4):3355–3362, Oct. 2018.
[2] T. Cohen and M. Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning (ICML)*, pages 2990–2999, 2016.
[3] C. Esteves, C. Allen-Blanchette, X. Zhou, and K. Daniilidis. Polar transformer networks. In *International Conference on Learning Representations (ICLR)*, 2018.
[4] P. Follmann and T. Bottger. A rotationally-invariant convolution module by feature map back-rotation. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 784–792, 2018.
[5] H. Greenspan, S. J. Belongie, R. M. Goodman, P. Perona, S. Rakshit, and C. H. Anderson. Overcomplete steerable
pyramid filters and rotation invariance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 222–228, 1994.

[6] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi. A hybrid deep architecture for robotic grasp detection. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1609–1614, 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European conference on computer vision*, pages 346–361, 2014.

[8] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems 28*, pages 2017–2025, 2015.

[9] V. Jain and E. Learned-Miller. FDDB: A benchmark for face detection in unconstrained settings. Technical report, Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[10] W.-Y. Kim and P. Yuan. A practical pattern recognition system for translation, scale and rotation invariance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 391–396, 1994.

[11] S. Kumra and C. Kanan. Robotic grasp detection using deep convolutional neural networks. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 769–776, 2017.

[12] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[13] I. Lenz, H. Lee, and A. Saxena. Deep Learning for Detecting Robotic Grasps. In *Robotics: Science and Systems*, page P12, June 2013.

[14] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, Apr. 2015.

[15] C.-H. Lin and S. Lucey. Inverse compositional spatial transformer networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] S. Liu, D. Huang, and A. Wang. Receptive field block net for accurate and fast object detection. In *European Conference on Computer Vision (ECCV)*, 2018.

[17] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 2018.

[18] D. Marcos, M. Volpi, N. Komodakis, and D. Tuia. Rotation equivariant vector field networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5058–5067, 2017.

[19] J. Redmon. Darknet: Open source neural networks in c. [http://pjreddie.com/darknet/](http://pjreddie.com/darknet/) 2013–2016.

[20] J. Redmon and A. Angelova. Real-time grasp detection using convolutional neural networks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1316–1322, 2015.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.