Hindawi
Scientific Programming
Volume 2022, Article ID 2037141, 11 pages
https://doi.org/10.1155/2022/2037141

Research Article

A Robust Convolutional Neural Network for 6D Object Pose Estimation from RGB Image with Distance Regularization Voting Loss

Faheem Ullah,[1] Wu Wei,[1] Yousef Ibrahim Daradkeh ,[2] Muhammad Javed,[3] Ihsan Rabbi ,[3] and Hanan Al Juaid [4]

[1]*School of Automation and Engineering, South China University of Technology, Guangzhou 510000, China*
[2]*Department of Computer Engineering and Networks, College of Engineering at Wadi Ad-Dawasir, Prince Sattam Bin Abdulaziz University, Wadi Ad-Dawasir, 11991, Saudi Arabia*
[3]*Department of Computer Science, University of Science & Technology Bannu, Bannu 28100, Pakistan*
[4]*Computer Sciences Department, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University (PNU), P.O. Box 84428, Riyadh 11671, Saudi Arabia*

Correspondence should be addressed to Hanan Al Juaid; haaljuaid@pnu.edu.sa

Received 23 April 2022; Revised 30 May 2022; Accepted 11 June 2022; Published 8 August 2022

Academic Editor: Sikandar Ali

Six-degree (6D) pose estimation of objects is important for robot manipulation but at the same time challenging when dealing with occluded and textureless objects. To overcome this challenge, the proposed method presents an end-to-end robust network for real-time 6D pose estimation of rigid objects using the RGB image. In this proposed method, a fully convolutional network with a features pyramid is developed that effectively boosts the accuracy of pixelwise labeling and direction unit vector field that take part in the voting process for object keypoints estimation. The network further takes into account measuring the distance between pixel and keypoint, which aims to help select accurate hypotheses in the RANSAC process. This avoids hypothesis deviations caused by the errors due to direction unit vectors in cases of distant pixels from keypoints. A vectorial distance regularization loss function is used to help Perspective-n-Point find 2D-3D correspondences between 3D object keypoints and their estimated corresponding 2D counterparts. Experiments are performed on widely used LINEMOD and occlusion LINEMOD datasets with ADD (-S) and 2D projection evaluation metrics. The results show that our method improves pose estimation performance compared to the state-of-the-art while still achieving real-time efficiency.

1. Introduction

The 6D object pose estimation is challenging due to occlusion and textureless surfaces of objects and becomes even more challenging when estimating 6D object poses from a single RGB image than from RGB-D or stereo images. 6D pose, which is the 3D rotation R and 3D translation T, is a rigid object transformation $(R; T)$ from the coordinates of the rigid object to the coordinates of the camera. The transformation here can be shown as a rigid transformation matrix $[R, T] \in \mathrm{SE}(3)$, where $R \in \mathrm{SO}(3)$ and $T \in R^3$. With the advancement of robot manipulation, navigation, self-driving cars, and augmented reality, 6D object pose estimation has attracted the interest of researchers extensively. In literature, some single-shot approaches are used that regress 6D pose from the image coordinates directly [1], which are not effective in the occluded environment. Recently, two-stage methods have shown progress in this field of research which detects keypoints followed by Perspective-n-Points (PnP).

Some of these two-stage approaches [2–4] detect keypoints first by regressing their image coordinates and then calculating 6D poses, but these keypoints are sparse, due to which these networks also show sensitivity to occlusion. Some approaches like [5, 6] use postrefinement for 6DoF poses like iterative closest point (ICP) [7] after calculating initial 6DoF object poses using deep learning. In recent years, vector field-based keypoints voting methods [5, 8] tackled the issue of occlusion effectively even without postrefinement. We use the vector field-based keypoints voting approach. These approaches introduce pixelwise voting by a vector field from each pixel that votes to detect keypoints using RANSAC [9] and then estimate 6D poses using Perspective-n-Point (PnP) [10]. The keypoint localization is achieved through hypotheses with the highest voting score in the unit vector field [8]; however, these methods do not take into account the distances between the object's pixels and the object's keypoints which also cause errors and deviate hypotheses due to small errors of direction vectors of distant pixels from keypoints. Handling errors in the direction vectors that occur because of the distances from pixels to keypoints is useful and is considered by [7, 11, 12]. These studies have used different approaches to the problem based on PVNet's RANSAC-based voting for keypoints estimation and the PnP solution. Reference [7] proposes atrous spatial pyramid pooling for capturing global context and distance-filtered pixel voting (ASPP-DF-PVNet) to calculate distances between pixels and keypoint. The attention voting network by [12] incorporates a channel-level attention module for adaptive feature fusion called the adaptive fusion attention module (AFAM) into U-Net and calculates distances between pixels and keypoints using prior distance augmented loss (PDAL).

Following these latest approaches, an end-to-end convolutional neural network (CNN) is used that takes an RGB image in 2D where occlusions occur among objects and estimates the 3D translation and 3D rotation, that is, the 6DoF pose of objects. The CNN based on fully convolutional networks (FCN) [13] with features pyramid resembling pyramid scene parsing network (PSPNet) [14] is developed for pixelwise labeling and unit vector field generation that brings robustness to occlusion as most errors occur due to incorrect pixel labels and direction vectors. The unit vectors vote for localizing object keypoints in a RANSAC hypothesis space, and a vector-based distance voting regularization loss function has been incorporated, which helps in the selection of accurate hypotheses in the voting process. Finally, PnP calculates 6D poses for objects using 2D-3D correspondences among object 3D model keypoints and their estimated corresponding 2D objects in the RGB image.

The loss function considers the distances between pixels and keypoints to reduce deviations in hypotheses that occur due to inaccurate direction vectors. As sampling and inliers search take part in the voting process, like RANSAC, it is difficult to differentiate. The difficulties arise while using voted keypoints and their actual ground truths for training networks end-to-end. For that, a proxy hypothesis is employed closely related to [11] but with a different computation technique, the vectorial distance voting loss, to calculate and approximate for each pixel the distance deviations among voted keypoints and their respective ground truths so that pixels produce approximated hypotheses with respect to their ground-truth keypoints. The proposed robust end-to-end network produces better results in heavy occlusion. The illustration of the system is given in Figure 1. Our architecture is as follows:

(i) End-to-end CNN for 6D object pose estimation presenting robust pixelwise labeling that produces an accurate vector field for voting for object keypoints

(ii) Calculating distances among pixels to keypoints to avoid errors due to inaccurate direction vectors

(iii) Proposing a vectorial distance loss for the distance between pixel and keypoint, which is generalizable to any number of dimensions

Experiments performed on LINEMOD and occlusion LINEMOD datasets that are used widely in this area of research show significant performance. These datasets are specially produced for 6D object pose estimation using RGB image. Our end-to-end network achieves real-timing in estimating 6D object poses and achieves high accuracy in cluttered space compared to the state of the art. Our method does not calculate any postrefinement of the 6DoF object pose.

2. Related Work

This section presents previous related works on 6DoF pose estimation based on a single RGB image. The 6D object pose estimation has been achieved using different approaches over the years.

2.1. Template-Based Methods. In this approach, a rigid template is used for scanning the image and calculating at each location in the image a similarity score, and then comparisons of these scores take place to obtain the best match. References [15–21] are the conventional methods based on template matching. In 6D pose estimation, a template is usually obtained through corresponding 3D model rendering. Some deep learning-based object detection approaches basically used for 2D object detection have also been employed for template matching, which has been enhanced for 6D pose estimation [1–3, 22]. This approach works well for textureless object detection but does not work well in a cluttered environment where some objects are occluded. However, [16] has tried to detect 3D objects in occlusion also through multimodalities using a dense depth map with the input image.

2.2. Feature/Keypoint-Based Methods. This approach extracts points of interest or keypoints from images as features to detect the object and then establishes the 2D-3D correspondences of the object to its 3D model to achieve 6D poses. References [23, 24] are the traditional feature-based approaches that use feature engineering and are translation and scale variants and also sensitive to other variations in the scene. Feature-based methods are good at handling occlusions but need textured objects for feature extraction. Several deep learning methods [4, 25–27] have been used to learn textured and textureless object detection features. A few conventional approaches directly regress pixels to the 3D object coordinate for 2D-3D correspondences [28, 29]. Similarly, [30] is a deep learning method for achieving the

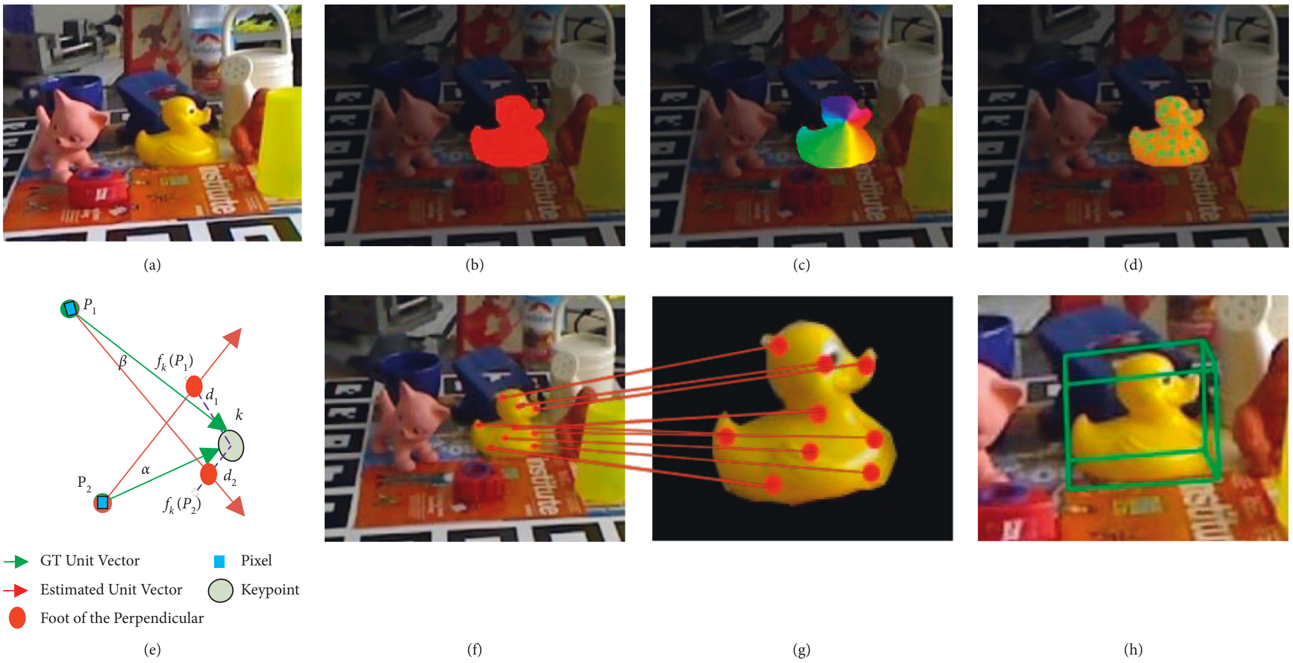

Figure 1: The 3D translation and 3D rotation are estimated through 2D and 3D keypoints correspondences. (b, c) The pixelwise labeling and pixelwise unit vectors field for keypoints voting, respectively. (d, e) The voting process for finding keypoints and calculate the distances among pixels and keypoints that affect the hypotheses. (f, g) The 2D and 3D keypoints correspondences using Perspective-n-Point (PnP), and finally, the 6D poses of objects are estimated (h). (e) p_1 and p_2 are the pixels, k is the keypoint, α and β are the angles between the two ground-truth and estimated unit vectors from pixels to keypoint, $f_k(p_1)$ and $f_k(p_2)$ are the foot of perpendiculars, and d_1 and d_2 are the distances between keypoint and foot of the perpendicular. (a) Input image, (b) pixel labeling, (c) vector field, (d) voting, (e) pixels to keypoints distances, (f) 2D keypoints, (g) 3D keypoints, and (h) 6D object pose.

same task, but these approaches require RGB-D data to regress to 3D coordinates and avoid handling symmetry problems. As local feature extraction can be done through either keypoints or pixels of the objects in the image, some methods do not regress the pose directly from images; instead, they define semantic keypoints sets and for detecting keypoint use deep neural networks. This approach uses a two-stage process where it performs semantic segmentation and then predicts 2D keypoints on objects surface from which it estimates 6D poses via 2D-3D correspondences using Perspective-n-Point. BB8 [29] generates pixelwise labeling for objects and regresses keypoints from each object to predict 3D bounding boxes. References [31–33] regress the 3D coordinates of objects from images directly and further use PnP for 2D-3D correspondences between objects and respective models for final poses. Reference [3] predicts the 2D projections of the corners of the 3D bounding box around the objects. The feature maps are fixed size and cannot handle occlusions well. Few methods solve the occlusion problem by producing pixelwise heatmaps of keypoints [4, 34].

2.3. Voting-Based Methods. In these methods, pixelwise labeling and pixelwise voting together take place for 2D object detection and key feature finding for 2D-3D correspondences to achieve the final 6D poses. References [35–37] use the Hough voting scheme, and [28, 38] use the random forest to predict pixels' 3D coordinates of objects. PoseCNN [5] uses semantic segmentation to localize objects in the RGB image, finds the center point of objects by estimating a vector field pointing towards the center of the object, then employs Hough voting for center prediction, and then predicts the depth to get object poses. Similar to PoseCNN, [6] employs semantic segmentation and objects center point but uses the dense approach for the final rotation quaternions. The 6D poses of the objects are regressed by a subnetwork. PoseCNN also uses depth information and ICP [6] to refine the estimated poses. DOPE [39] does not apply postalignment and uses a simple deep network architecture to infer image coordinates in 2D from the projected 3D bounding boxes and then applies Perspective-n-Point (PnP) [10]. DOPE recovers the final 3D translation and 3D rotation, that is, 6D pose of the object with respect to the camera, from the detected projected vertices of the bounding box. The system is fully trained on simulated data to avoid the generalization problem in PoseCNN, which occurs due to high correlation in real data, rather than only estimating a centroid. PVNet [8], a two-stage deep learning network, votes several features of interest on any object. Using pixelwise labeling and unit vectors from each pixel of the object, RANSAC-based voting hypotheses [9] are employed for finding key points, and then PnP is applied for the final pose estimation. A total of 8 keypoints are selected for each object using the farthest point sampling (FPS) algorithm on the objects' 3D models. DPVL [11] has used a similar approach as PVNet but considers the distance between object pixels and object keypoints. As the RANSAC-like voting process is difficult to differentiate, it uses the proxy hypothesis to calculate and approximate for each pixel the distance deviations among voted keypoints and their respective ground truths. ASPP-DF-PVNet [7] considers global context using atrous spatial pyramid pooling and distances between pixels and keypoints. He et al. [12] incorporate a channel-level attention module for the adaptive feature fusion into U-Net and calculate distances between pixels and keypoints using prior distance augmented loss. Another related architecture based on the channel spatial attention network (CSA6D) is proposed by Chen and Gu [40] to estimate the 6D object pose from RGB-D images.

3. Proposed Method

In this paper, an end-to-end network for 6DoF object poses estimation is proposed, which is effective in a cluttered environment. The purpose of this research work is to handle occlusion, texture, and symmetry so that it can be used for robot manipulation. A voting-based approach is used as this approach is robust towards occlusions and view changes. 6DoF objects pose estimation is object detection in an RGB image and 3D translation and orientation estimation of those objects. A CNN based on FCN [13] with a feature pyramid approach has been used for pixel labeling and vector field prediction, and a voting loss with vector-based distance has been incorporated for selecting accurate hypotheses in the voting process. The RGB input image passes to the network, and it detects objects and calculates the 6D pose through 3D rotation R and 3D translation T accurately without any postrefinement. Here, we assume that the objects are rigid and their 3D models are available. Our method first calculates pixelwise classification and vector field prediction, then votes for 2D keypoints in the object body from the vector fields, and then estimates the 6DoF pose by solving a PnP problem. Due to the use of smooth ℓ_1 loss for learning unit vectors, small errors in the vector may occur, leading later on to large deviation errors of hypotheses as ℓ_1 loss does not consider the distances between pixels and keypoints. That is why we consider the distance between a pixel and a keypoint in order to avoid large deviation errors of hypotheses.

We use an approach that fulfills the pose estimation process in a two-stage pipeline similar to [3, 5, 8, 39], that is, semantic segmentation and 3D orientation and 3D translation that completes the process of 6D object pose estimation.

3.1. Semantic Segmentation and Unit Vectors. Inspired by FCN [13], our proposed multiclass semantic segmentation architecture uses a similar approach exploiting ResNet-50 v2 [41] as the backbone and using multiple scales of feature maps that further generate pixelwise classification and vector fields. This pixel labeling and pixel voting network takes as input an RGB image $(w \times h \times 3)$ and outputs a tensor with similar dimensions, except that the last dimension is the number of classes $(w \times h \times (m+1)$ for m-classes) and a tensor $(w \times h \times (k \times 2 \times m)$ where k is the number of keypoints) for unit vectors. To avoid problems in the early stages due to small receptive fields, our network leverages a larger receptive field as all of its layers are convolutional. The pixelwise classification and unit vector field prediction network is shown in Figure 2.

Taking the $(640 \times 480 \times 3)$ dimensions RGB image as input, the ResNet-50 v2 performs max-pooling two times to get the feature maps of dimensions $(1/4)$ of the original input image. Additional sets of feature maps are generated by using successive $(\text{Conv2 } D - \text{stride} = 2 - \text{BN} - \text{ReLU})$ layers that result in feature maps of dimensions $(1/8, 1/16, 1/32)$ of the original input image $(w \times h)$. We further improved by modifying our semantic segmentation network, further processing each feature map with another convolutional layer, an approach similar to the PSPNet [14]. The feature pyramid is generated after the output of the ResNet. To achieve the size as the first set of feature maps, upsampling of the features' pyramid takes place, which then is concatenated and further applying transposed convolution $(\text{strides} = 2)$ twice of 256 filters, each getting the original image size back. Finally, apply a transposed convolution $(\text{kernel\_size} = 1, \text{filters} = 13)$ with filters equal to the number of object classes leading to softmax to generate the pixelwise prediction. To obtain unit vectors along with the class probabilities, we apply a 1×1 convolution on the final feature map.

Using a simple or some basic CNN architecture for semantic pixel labeling would improve the speed of the system by some margin but would decrease the accuracy accordingly. The proposed architecture for semantic segmentation and vector field generation is robust to occlusion and is inspired by [13, 14] and [8].

3.2. Object Detection and Pose Estimation. After processing the image and doing the pixelwise classification and obtaining unit vectors, our network predicts the 2D locations of 3D keypoints using RANSAC, from which the pose can be obtained using the EPnP algorithm. PoseCNN [5] uses Hough voting for finding the center point of objects, while PVNet [8] and DPVL [11] use RANSAC-based voting for keypoints localization. It is a two-stage process that is robust for occlusion, symmetry, and handling textureless objects. The first stage locates the 2D projection points of predefined 3D keypoints associated with the 3D objects' models, where the keypoint localization is implemented through the RANSAC based on the pixel labeling and vector field representation. In the second stage, the 6D object pose estimation takes place using PnP. Figure 2 shows the complete proposed 6DoF object poses estimation method.

3.2.1. Keypoint Localization. Here, we first present the vector field representation, which is unit vectors directing from each pixel towards each keypoint and then the keypoint localization. To handle the varying objects' sizes during detection, the vector field of direction vectors is estimated with a larger receptive field that covers larger parts of objects. Because of this, even invisible keypoints can be induced by the network from visible parts. Here, $u_k(p)$ is a function of unit direction vector u_k from pixel p to a specific keypoint k.

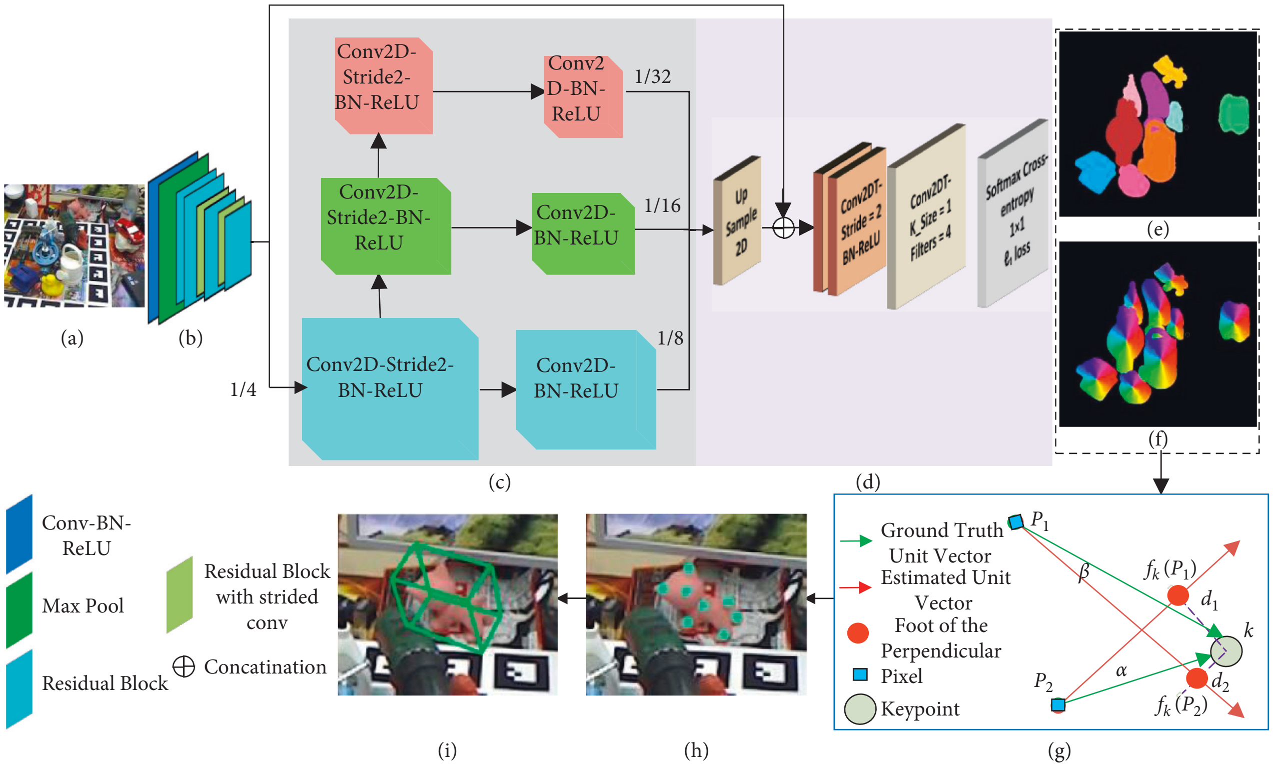

Figure 2: The complete 6D object pose estimation process that performs semantic segmentation, vector field prediction pointing towards keypoints of the object, distances between pixels and keypoints, hypothesis selection, and finally the pose estimation. (b) A ResNet-50 is used that passes higher level feature maps of input image (a–c) feature pyramid for detecting features at different scales. Then, (d) upsampling stage achieves the features map size equal to the input image, and then (e) pixelwise labeling and (f) pixelwise unit vector field are achieved. (g) Pixel to keypoint distances help RANSAC in finding accurate hypothesis, and then (h) keypoints are estimated, based on which (i) 6D poses are calculated. (a) Input image, (b) ResNet-50, (c) feature pyramid, (d) upsampling, (e) pixel labeling, (f) vector field, (g) pixels to keypoints distances, (h) keypoints, and (i) 6D pose.

$$u_k(p) = \frac{k-p}{\|k-p\|_2}. \qquad (1)$$

Key point hypotheses can be generated from semantic labels and unit vectors in a RANSAC-based voting scheme [6]. Given keypoint and its corresponding direction vectors to vote, we generate hypotheses for keypoint k; that is, $H_{k,i} = \{H_{k,i} | i = 1, 2, 3, \ldots, k\}$. We consider initially all the intersections of any two direction unit vectors as candidate points for the final keypoints selection. The hypothesis deviation due to error in the predicted direction vector depends on both the angle and the distance between a pixel and a keypoint. If a pixel is far from a keypoint, a small angle between direction vectors can also generate a large hypothesis deviation. Finally, all the direction unit vectors in the generated vector field vote for choosing the keypoints wherever the deviation angle from the pixel to the keypoint relative to the direction is less than a certain threshold. The candidate points having most of the votes are considered as keypoint hypotheses. This way the voting directions deviating by a large angle from the hypothesis are removed. For this, we take the formula for voting from PVNet [6] given as

$$v_{k,i} = \sum_{p\in M} \mathbb{I}\left(\frac{\left(H_{k,i}-p\right)^T}{H_{k,i}-p_2} u_k(p) \geq \theta\right), \qquad (2)$$

where $v_{k,i}$ is the voting score for the hypothesis $H_{k,i}$ for 2D keypoint k, M is the mask of the object, $\mathbb{I}$ is an indicator function that indicates whether pixel votes for a keypoint hypothesis or not, and θ is the threshold.

3.2.2. Loss Function. Assume that image I and the keypoint locations $K = \{k_i\}$ where $i = 1, 2, 3, \ldots n$ is the number of selected keypoints on the surface of the object. The smooth ℓ_1 loss [42] between predicted and ground-truth direction vectors is used to regress the direction vectors as

$$\mathcal{L}_{vf} = \sum_{k=1}^{K} \sum_{P\in M} \ell_1\left(\left\|u_k(p) - v_k(p)\right\|\right), \qquad (3)$$

where $\mathcal{L}_{vf}$ is the loss of vector field, $v_k(p)$ is the predicted direction vector, and M is the mask of the object. Our network estimate vector fields in a similar way. The pixel segmentation labeling $s(p)$ where $s(p) \in [0,1] \forall M$ is achieved through a softmax cross-entropy loss function as

$$\mathcal{L}_{\text{seg}} = -\sum_{P\in M} \log(s(p)). \qquad (4)$$

The errors that occur due to the estimated direction vectors can cause large deviations in hypotheses even if the errors are small, which affects the pose estimation performance. We consider the hypotheses distributions enforcing all the hypotheses to be more effective and produce fewer errors towards the actual keypoints. To learn the distance between a keypoint k and its respective foot of perpendicular $f_k(p)$ with the direction vector $u_k(p)$ of pixel p, we get the hypothesis that is differentiable. It is given as

$$\mathscr{L}_{pv} = \sum_{k=1}^{K} \sum_{P\in M} \ell_1\left(\|k - f_k(p)\|\right). \tag{5}$$

Here, $f_k(p)$ is the foot of the perpendicular. This equation calculates the distances d between all $f_k(p)$ and keypoints k that need to be minimized to achieve accurate hypotheses for keypoint voting, which will be achieved by minimizing the loss function as follows:

$$\begin{gathered}\mathscr{L}_{pv} = \sum_{k=1}^{n} \sum_{p\in M} \ell_1\left(\sqrt{\left(k^x - p^x - v_k^x\lambda'\right)^2 + \left(k^y - p^y - v_k^y\lambda'\right)^2}\right),\\ \lambda' = \frac{v_k^x k^x - v_k^x p^x + v_k^y k^y - v_k^y p^y}{v_k^{x^2} + v_k^{y^2}},\end{gathered} \tag{6}$$

where $v_k(p) = (v_k^x, v_k^y)$ is the estimated unit vector, $p = (p^x, p^y)$ is the pixel with its x, y coordinates, $k = (k^x, k^y)$ is the keypoint with its x, y coordinates, and λ' is a parameter. The vector-based approach to distance is generalizable to any number of dimensions.

Here, $v_k(p)$ is the result of unit vector estimation by our network, so it needs normalization as the output may not be a unit vector which is why the normalization operation is involved in $\mathscr{L}_{pv}$. Due to $\mathscr{L}_{pv}$, the distance regularization voting loss, the $v_k(p)$ points correctly to keypoints because of its sensitivity towards pixels locations. The final objective is calculated as follows:

$$\mathscr{L} = \lambda_1 \mathscr{L}_{\text{seg}} + \mathscr{L}_{vf} + \lambda_2 \mathscr{L}_{pv}, \tag{7}$$

where $\mathscr{L}$ is the total loss and λ_1 and λ_2 are the hyperparameters for trade-off management between pixel labeling and vector field estimation.

Our method starts at pixelwise labeling and pixelwise unit vectors discussed in Section 3.1 of the methodology, where unit vectors take part in the voting process with $\mathscr{L}_{vf}$ the vector field loss for keypoints localization using RANSAC, and then the $\mathscr{L}_{pv}$ loss for pixel to keypoint distances is used which are shown in the Section 3.2.2 of the methodology. Finally, the PnP is used for 2D object keypoints to 3D object model keypoints correspondences to calculate the final 6D object poses, which is the 3D translation and 3D rotation of rigid objects. Section 3.2.3 shows further details related to the implementation of the system.

3.2.3. Implementation Details. Based on [8, 11], 8 keypoints are selected for each object, and the farthest point sampling is used for this purpose on its 3D model. Initially, we consider the center of the object where the keypoint set is initialized. We apply data augmentation following [8] to the data to avoid overfitting where we achieve in some images a slight truncation due to random cropping. Some other processing performed includes color jittering, rotation, width shift, height shift, shear, zoom, channel shift, and horizontal flip. In training, λ_1 and λ_2 are set to similar values and set to 1. During experiments, λ_2 is set to $1e^{-3}$ and increases to $1e^{-2}$ gradually and then increases λ_1 by a factor of 1.1 each epoch. The learning rate at $1e^{-3}$ provides the best results according to DPVL, so we also set it to $1e^{-3}$ and decay by a factor of 0.75 to $1e^{-5}$ gradually and with a total of 100 training epochs. Adam optimizer has been employed. We train our method on the LINEMOD dataset. We do not perform any postrefinement operations. Our method performs in real time on a GTX 2080 Ti GPU at the input image (480×640).

4. Results and Discussion

This section explores experiments, results, comparisons, and discussion of our method with other related methods of the 6D object pose estimation using state-of-the-art datasets and evaluation metrics.

4.1. Datasets. Very popular datasets for 6D poses have been used for conducting experiments for this research work. The proposed method has been trained on LINEMOD [18] and evaluated using both the LINEMOD and the occlusion LINEMOD [28] datasets. LINEMOD dataset consists of 15783 images, 13 objects, and a total of about 1200 instances of each object with a mask. Each object is provided with its respective 3D model. Similarly, the occlusion LINEMOD dataset consists of 8 objects and 1214 images with occlusions which is more challenging. These datasets have been extensively reported in a number of research articles for comparative analysis of 6D object pose estimation.

4.2. Evaluation Metrics. Two evaluation metrics, the 2D projection metric [29] and the ADD score metric [18], have been used to evaluate our method. The 2D projection metric uses the estimated and ground-truth poses to calculate the average 3D distance of the model points. The correct pose estimated will have a distance of less than 5 pixels.

$$2\ D.\text{Proj} = \frac{1}{m} \sum_{x\in\mathscr{M}} \|K(Rx + T) - K(\widetilde{R}x + \widetilde{T})\|_2, \tag{8}$$

where m is the total number of points on the 3D object model $\mathscr{M} = \{x_i \in R^3 | i = 1, 2, 3, \ldots, m\}$, x is a point or a set of points on the surface 3D object model, R and T are the rotation and translation respectively, $Rx + T$ is the target pose that transforms the point with $\text{SE}(3)$ transformation and vice versa, and K is the camera's intrinsic parameter matrix.

The ADD score calculates the average 3D distance between 3D model points transformed by the estimated and ground-truth poses. Then, the correctly estimated pose will have less than 10 percent distance from the diameter of the 3D model.

$$\text{ADD} = \frac{1}{m} \sum_{x\in\mathscr{M}} \|(Rx + T) - (\widetilde{R}x + \widetilde{T})\|_2. \tag{9}$$

Table 1: The performance on the LINEMOD dataset for objects pose estimation based on ADD (-S) scores.

Methods	Ape	Bench vise	Cam	Can	Cat	Driller	Duck	Egg box	Glue	Hole puncher	Iron	Lamp	Phone	Mean
BB8 [29]	40.40	91.80	55.70	64.10	62.60	74.70	44.30	57.80	41.20	67.20	84.70	76.50	54.00	62.70
SSD6D [1]	65.00	80.00	78.00	86.00	70.00	73.00	66.00	100	**100**	49.00	78.00	73.00	79.00	79.00
YOLO6D [3]	21.62	81.80	36.57	68.80	41.82	63.51	27.23	69.58	80.02	42.63	74.97	71.11	47.74	55.95
DPOD [31]	53.28	95.34	90.36	94.10	60.38	97.72	66.01	99.72	93.83	65.83	99.80	88.11	74.24	82.98
Pix2Pose [33]	58.10	91.00	60.90	84.40	65.00	73.60	43.80	96.80	79.40	74.80	83.40	82.00	45.00	72.40
CDPN [32]	64.38	97.77	91.67	95.87	83.83	96.23	66.76	99.72	99.61	85.82	97.85	97.86	90.75	89.86
PoseCNN [5]	27.80	68.90	47.50	71.40	56.70	65.40	42.80	98.30	95.60	50.90	65.60	70.30	54.60	62.70
PVNet [8]	43.62	99.90	86.86	95.47	79.34	96.43	52.58	99.15	95.66	81.92	98.88	99.33	92.41	86.27
DPVL [11]	69.05	**100**	94.12	98.52	83.13	99.01	63.47	**100**	97.97	88.20	99.90	99.81	96.35	91.50
PDAL + AFAM [12]	69.43	**100**	92.45	99.21	87.72	99.01	67.79	**100**	98.94	86.01	99.38	99.81	95.10	91.91
L+ [7]	65.34	**100**	92.65	97.84	**90.22**	97.72	62.54	99.72	95.56	88.97	99.30	99.53	95.87	91.18
Ours	**76.27**	**100**	**96.80**	**99.38**	87.85	**99.40**	**71.42**	**100**	99.68	**94.72**	**100**	**99.92**	**98.64**	**94.16**

Some objects like glue and egg box are symmetric objects. The bold values given in Table 1 indicate the high value among the compared methods for pose estimation on the LINEMOD dataset with respect to ADD (-S) metric.

ADD (-S) is employed for symmetrical objects, using the closest point distance, and the 3D distance is calculated.

$$\mathrm{ADD}\,(S) = \frac{1}{m}\sum_{x_1\in\mathcal{M}}\min_{x_2\in\mathcal{M}}\left\|\left(Rx_1 + T\right) - \left(\tilde{R}x_2 + \tilde{T}\right)\right\|_2. \qquad (10)$$

4.3. Comparisons with State of the Art. The state-of-the-art PoseCNN has successfully solved the problems of template-based and feature-based approaches for 6D pose estimation; however, postrefinement for the final poses is required for better accuracy. The voting-based keypoints prediction approaches are robust in this regard and can estimate accurate initial 6D poses, so they do not require any post-refinement. Our method follows a similar approach and focuses on providing robust semantic segmentation and vector field prediction, which further predicts object keypoints and distances and angles between each pixel to each keypoint. The robust semantic segmentation shows robustness to occlusions, due to which it provides better accuracy for the final pose estimation, so our method does not need any pose refinement for pose estimation improvement performance. Here, we compare our results with 6D pose estimation approaches using a single RGB image, which are state of the art in this research area. The comparisons have been carried out against PVNet [8], DPVL [11], ASPP-DF-PVNet with *L*+ loss [7], and PDAL-AFAM approach of He et al. (2021) [12] and some previous approaches such as PoseCNN [5], SSD6D [1], YOLO6D [3], BB8 [29], CDPN [32], DPOD [31], Pix2Pose [33], and CSA6D [40]. The results are evaluated using ADD (-S) and 2D-Projection metrics on LINEMOD and occlusion LINEMOD datasets.

4.3.1. Comparisons Using ADD (-S) Metric. Table 1 shows the comparison of our method with several other methods mentioned above for pose estimation on the LINEMOD dataset with respect to ADD (-S) metric. It shows that our method outperforms state-of-the-art methods; especially, our method outperforms our baseline methods PVNet [8] and DPVL [11]. The performance is improved by a margin of 2.66% compared to [11] using ADD (-S) metric. Occluded, textureless, and symmetric objects are the main issues for pose estimation systems. Our method's accuracy has improved significantly for all as for "ape," accuracy has improved by 7.22%, and for "duck," the accuracy has improved by 7.95% using ADD (-S) score. Both the "ape" and the "duck" are textureless objects. The accuracy for "glue," which is a symmetric object, improves by 1.71%.

Table 2 shows the comparisons of our method with the state-of-the-art approaches on occluded LINEMOD dataset in terms of ADD (-S) scores where our method achieves better overall performance. Our method improves the performance of occluded objects by 3.88%, especially the accuracy of "glue" during occlusion improves significantly. The overall results show that our proposed method gives the best performance compared to the state-of-the-art approaches. Figure 3 demonstrates our method's qualitative results visualization on the occlusion LINEMOD dataset. Our method outperforms PVNet and DPVL and the variations of DPVL, the ASPP-DF-PVNet, and the PDAL-AFAM. The robust semantic segmentation and vector field prediction lead the network to better pose estimation under heavy occlusion.

4.3.2. Comparisons Using 2D Projection Metric. We include only those results for comparisons that are provided by other methods as 2D projection-based results are not reported by some methods, so we do not include those in Tables 3 and 4. CSA6D [40] reported only 2D projection-based results on the LINEMOD dataset, so we only include those. Table 3 shows the comparison of our method with a number of other methods for pose estimation on the LINEMOD dataset concerning the 2D projection metric. It provides a 0.28% improvement using the 2D projection metric, which shows that our method outperforms the state-of-the-art methods and also outperforms PVNet and DPVL, which are our baseline methods. Table 4 shows the results of the occlusion LINEMOD dataset using the 2D projection metric. DPVL has

TABLE 2: The performance on the occlusion LINEMOD dataset for pose estimation based on ADD (-S) scores.

Methods	Ape	Can	Cat	Driller	Duck	Egg box	Glue	Hole puncher	Mean
YOLO6D [3]	2.48	17.48	0.67	7.66	1.14	—	10.08	5.45	6.42
Pix2Pose [33]	22.00	44.70	22.70	44.70	15.00	25.20	32.40	49.50	32.00
PoseCNN [5]	9.60	45.20	0.93	41.40	19.60	22.00	38.50	22.10	24.90
PVNet [8]	15.81	63.30	16.68	65.65	25.24	50.17	49.62	39.67	40.77
DPVL [11]	19.23	69.76	21.06	71.58	34.27	47.32	39.65	45.27	43.52
PDAL + AFAM [12]	25.47	68.20	22.26	68.33	32.61	45.28	49.28	47.51	44.87
L+ [7]	**29.58**	69.10	21.74	70.10	32.08	47.58	**55.24**	**52.22**	47.23
Ours	22.44	73.31	24.23	75.07	38.60	51.43	44.08	50.11	**47.40**

The bold values given in Table 2 indicate the high performance on the occlusion among the compared methods for pose estimation. It is also required to bold the values of our proposed method from column number 2 to 6 (Can to Egg box).

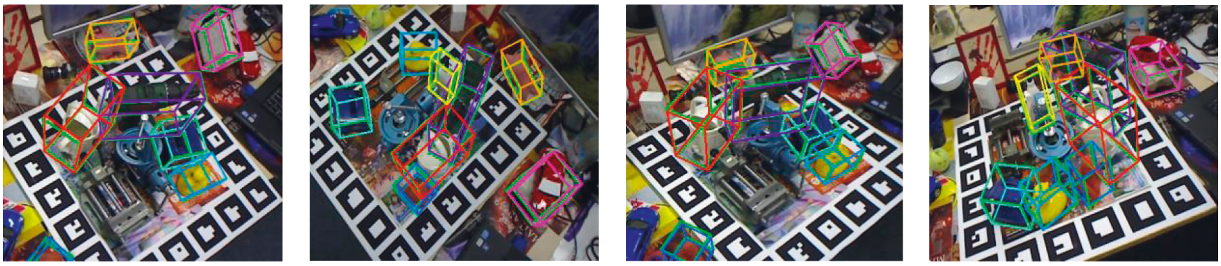

FIGURE 3: The qualitative results on the occlusion LINEMOD dataset. The green 3D bounding boxes around the objects show the ground truths, and the other colors show the proposed system's predictions.

TABLE 3: The performance on the LINEMOD dataset for objects pose estimation based on 2D projection errors.

Methods	Ape	Bench vise	Cam	Can	Cat	Driller	Duck	Egg box	Glue	Hole puncher	Iron	Lamp	Phone	Mean
BB8 [29]	96.60	99.10	86.00	91.20	98.80	80.90	92.20	91.00	92.30	95.30	84.80	75.80	85.30	89.30
YOLO6D [3]	92.10	95.06	93.24	97.44	97.41	79.41	94.65	90.33	96.53	92.86	82.94	76.87	86.07	90.37
CDPN [32]	96.86	98.35	98.73	99.41	99.8	95.34	98.59	98.97	99.23	99.71	97.24	95.49	97.64	98.10
PoseCNN [5]	83.00	50.00	71.90	69.80	92.00	43.60	91.80	91.10	88.00	82.10	41.80	48.40	58.80	70.20
PVNet [8]	99.23	99.81	99.21	99.90	99.30	96.92	98.02	99.34	98.45	**100**	99.18	98.27	99.42	99.00
DPVL [11]	99.04	99.71	99.41	**100**	99.70	98.12	99.06	99.43	99.51	**100**	99.69	99.14	99.42	99.40
L+ [7]	99.05	99.71	**99.61**	99.71	99.81	98.62	98.97	99.44	99.23	99.91	99.80	98.28	99.52	99.00
CSA6D [40]	98.60	95.80	98.80	97.40	99.50	95.10	98.40	99.90	**99.90**	98.20	97.80	95.50	97.60	98.10
Ours	**99.42**	**99.83**	99.55	**100**	**99.86**	**98.83**	**99.59**	**99.84**	99.86	**100**	**99.91**	**99.53**	**99.65**	**99.68**

The bold values given in this Table show the highest value(s) of performance on the LINEMOD dataset for objects pose estimation based on 2D projection errors.

TABLE 4: The performance on the occlusion LINEMOD dataset for objects pose estimation based on 2D projection errors.

Methods	Ape	Can	Cat	Driller	Duck	Egg box	Glue	Hole puncher	Mean
PVNet [8]	69.14	86.09	**65.12**	73.06	61.44	**8.43**	55.37	69.84	61.06
L+ [7]	**67.61**	86.75	62.85	79.91	64.07	3.75	60.47	80.25	63.21
Ours	68.88	**87.02**	64.35	**82.75**	**66.00**	7.27	**60.89**	**83.06**	**65.02**

These values indicate the highest value of performance on the occlusion LINEMOD dataset for objects pose estimation based on 2D projection errors.

not reported these results, but compared to the state-of-the-art ASPP-DF-PVNet with *L*+ loss [7], our network shows a 1.81% improvement. Table 5 shows the number of wins by our method against all the datasets and evaluation metrics which show the robustness of our method. The number of wins shows how many times our method achieves the best score, and actually, it beats all the previous methods in the table.

4.4. Ablation Study. The two-stage processes show better results. The results presented in Section 4, Results and Discussions, show that the pixelwise voting [8] processes are more robust to occlusion compared to the processes that directly regress coordinates of keypoints using convolutional neural networks [3]. By incorporating the distance regularization to decrease the distance error between keypoints and hypotheses, the new method enhances the results further by incorporating a robust pixelwise labeling and vector field prediction network with the hypotheses that consider vectorial distance error between keypoints and pixels. The errors mainly increase due to incorrect pixel labels and direction vectors. Segmenting occluded objects can easily fail

Table 5: The performance of our system in terms of the number of wins on the LINEMOD and occlusion datasets using ADD (-S) and 2D projection errors.

Number of wins	LINEMOD ADD (-S)	Occlusion ADD (-S)	LINEMOD 2D projection	Occlusion 2D projection
SSD6D [1]	1/13	0/8	0/13	0/8
PVNet [8]	0/13	0/8	1/13	2/8
DPVL [11]	2/13	0/8	2/13	0/8
PDAL + AFAM [12]	2/13	0/8	0/13	0/8
L+ [7]	2/13	3/8	1/13	1/8
Ours	**11/13**	**5/8**	**11/13**	**5/8**

The bold values given in this table show the highest numbers of wins on the LINEMOD and occlusion datasets using ADD (-S) and 2D projection errors.

if the segmentation network is not robust, especially if the object looks thin from a specific view in the image. The example is the object "glue" in the dataset when it is partially occluded. The proposed semantic segmentation network is robust to occlusions and can be further optimized by changing the number of filters in the features' pyramid to increase or decrease the number of parameters. The number of levels in the features' pyramid can also be increased or decreased to test its speed and accuracy. Changing the size of the ResNet will affect efficiency and accuracy. The results of our network are improved significantly, which are shown in Tables 1–4, using LINEMOD and occlusion LINEMOD datasets and 2D projection metrics and ADD (-S) metrics for evaluation. Table 5 shows the comparison in terms of the number of wins against both datasets and evaluation metrics in comparison with all the state-of-the-art methods. The qualitative visualized results can be seen in Figure 3. Our method converges faster by using just 100 epochs to train and converge compared to PVNet, which needs 200 epochs during training for proper convergence. Reaching a consensus during voting for keypoints, our method shows robustness too. Other experiments are needed to achieve a further faster, more accurate, and more scalable network for 6D object pose estimation.

5. Conclusions

A network consisting of robust pixelwise classification and voting for keypoints, finding 2D-3D correspondences for the final pose estimation of objects is presented. The proposed pixelwise labeling improves the accuracy of the vector field and the system as a whole. For achieving further accurate voting for keypoints findings, the proposed system considers the distances among pixels and the keypoints that lead to better pose estimation. For this, a vectorial distance-based differentiable loss function is used to solve the problem of deviated hypotheses due to distant pixels from keypoints. The good thing about the vectorial distance function is its generalizable nature to any number of dimensions. The proposed approach speeds up the convergence of the network during training. The results in Tables 1–4 show the robustness of our model compared to the latest preexisting approaches. In terms of the number of wins presented, Table 5 also shows our system's robustness. In future work, we will consider incorporating the suggestions presented in the "Limitations and Future Work" section.

6. Limitations and Future Work

As our work focuses on an efficient robust system for object pose estimation for robot manipulation, we adopt a robust semantic segmentation network and vectorized distance function. Recent works such as [8, 11] use a ResNet-18 as the backbone network that provides a weak segmentation. Hence failures occur in segmentation masks. Our backbone architecture ResNet-50 v2 in combination with FCN and PSPNet model produces comparatively more accurate segmentation masks. Due to sophisticated semantic segmentation architecture, the real-time speed of the complete system for object pose estimation affects slightly. Managing the speed versus accuracy trade-off is the key problem that needs to be solved. From PSPNet to FC-HarDNet-L2, any choice of selecting a segmentation network can be made, but probably the FASSD-Net-L1 and FC-HarDNet-L2 are better options for managing trade-offs between speed and accuracy. A thorough review of semantic segmentation networks has been presented by Rosas-arias and Benitez-Garcia [43]. One possible solution can be using more powerful GPUs or performing more experiments to find out new settings for another model. For more accuracy, training the same network on more data like occlusion LINEMOD and new datasets will also improve performance. Using direction vectors to extract pairwise features and triplet regularization can be another way to be used to see the accuracy of the method. Other approaches for finding loss function may affect the performance and should be tested. Some further postrefinement techniques will also improve the accuracy of the system.

Data Availability

All the data are included within the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Science and Technology Planning Project of Guangdong Province of China under Grant 2019A050520001 and the Princess Nourah Bint Abdulrahman University Research Supporting Project (no. PNURSP2022R54).

References

[1] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: making RGB-based 3D detection and 6D pose estimation great again," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1530–1538, Montreal, BC, Canada, October 2017.

[2] V. Lepetit and M. Rad, "BB8 : a scalable, accurate, robust to partial occlusion method for predicting," in *Proceedings of the IEEE international conference on computer vision*, pp. 3828–3836, Montreal, BC, Canada, March 2017.

[3] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6D object pose prediction," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 292–301, Salt Lake City, UT, USA, June 2018.

[4] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-dof object pose from semantic keypoints," in *Proceedings of the IEEE International Conference On Robotics And Automation*, ICRA, New York, NY, USA, May 2017.

[5] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN a convolutional neural network for 6d object pose estimation in cluttered scenes," in *Proceedings of the Robotics: Science and Systems*, June 2017.

[6] P. J. Besl, N. D. McKay, A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 03 1992.

[7] Y. Zhu, L. Wan, W. Xu, and S. Wang, "ASPP-DF-PVNet: atrous Spatial Pyramid Pooling and Distance-Filtered PVNet for occlusion resistant 6D object pose estimation," *Signal Processing: Image Communication*, vol. 95, Article ID 116268, 2021.

[8] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "PVNET: pixel-wise voting network for 6dof pose estimation," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2019, pp. 4556–4565, 2019.

[9] M. A. Fischler and R. C. Bolles, "Random sample consensus," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[10] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: an accurate O(n) solution to the PnP problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[11] X. Yu, Z. Zhuang, P. Koniusz, and H. Li, "6DoF object pose estimation via differentiable proxy voting loss," in *Proceedings of the British Machine Vision Conference*, BMVC, New York, NY, USA, Feburary 2020.

[12] Y. He, J. Li, X. Zhou, Z. Chen, and X. Liu, "Attention voting network with prior distance augmented loss for 6DoF pose estimation," *IEICE - Transactions on Info and Systems*, vol. 104, no. 7, pp. 1039–1048, 2021.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Boston, MA, USA, June 2015.

[14] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, and S. G. Limited, "Pyramid Scene Parsing Network," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HIUSA, July 2017.

[15] Z. Zhe Cao, Y. Sheikh, and N. K. Banerjee, "Real-time scalable 6DOF pose estimation for textureless objects," in *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2441–2448, Stockholm, May 2016.

[16] S. Holzer, S. Hinterstoisser, S. Cagniart, C. Ilic et al., "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 858–865, Barcelona, Spain, November 2011.

[17] S. Cagniart, C. Ilic, S. Sturm, P. Navab, N. Fua, P. Lepetit, V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 876–888, 2012.

[18] S. Hinterstoisser, V. Lepetit, S. Ilic, and S. Holzer, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Proceedings of the 12th international conference on Computer Vision*, pp. 548–562, New York, NY, USA, October 2012.

[19] C. Re and X. Ren, "Discriminative Mixture-Of-Templates for Viewpoint Classification," in *Proceedings of the Computer Vision - ECCV 2010*, pp. 408–421, New York, NY, USA, December 2010.

[20] R. Rios-Cabrera and T. Tuytelaars, "Discriminatively trained templates for 3D object detection: a real time scalable approach," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, Sydney, NSW, Australia, December 2013.

[21] M. Zhu and K. G. Derpanis, Y. Yang, S. Brahmbhatt, M. Zhang, C. Phillips, M. Lecce, and K. Daniilidis, "Single image 3D object detection and pose estimation for grasping," in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3936–3943, Hong Kong, China, May 2014.

[22] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for CNN: viewpoint estimation in images using CNNs trained with rendered 3D model views," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2686–2694, Santiago, Chile, December 2015.

[23] D. G. Lowe, "Object recognition from local scale invariant features," *Concrete Products*, vol. 20, no. 5, p. 15, 2002.

[24] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *International Journal of Computer Vision*, vol. 66, no. 3, pp. 231–259, 2006.

[25] S. Tulsiani and J. Malik, "Viewpoints and Keypoints," pp. 1–8, Cvpr, 2015, https://arxiv.org/abs/1411.6067.

[26] P. Wohlhart, V. Lepetit, and V. Lepetit, "Learning Descriptors for Object Recognition and 3D Pose Estimation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 99–131, Boston, MA, USA, June 2015.

[27] A. Doumanoglou, V. Balntas, R. Kouskouridas, and T.-K. Kim, "Siamese Regression Networks with Efficient Mid-level Feature Extraction for 3D Object Pose Estimation," Icvl, 2016, http://arxiv.org/abs/1607.02257.

[28] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6D object pose estimation using 3D object coordinates," *Computer Vision - ECCV 2014*, LNCS, vol. 8690, pp. 536–551, 2014.

[29] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3364–3372, Las Vegas, NV, USA, December 2016.

[30] A. Krull, E. Brachmann, F. Michel, M. Y. Yang, S. Gumhold, and C. Rother, "Learning analysis-by-synthesis for 6d pose estimation in RGB-D images," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 954–962, Santiago, Chile, December 2015.

[31] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1941–1950, Seoul, Korea (South), October 2019.

[32] Z. Li, G. Wang, and X. Ji, "CDPN: coordinates-based disentangled pose network for real-time RGB-based 6-DoF object pose estimation," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7677–7686, Seoul, Korea (South), October 2019.

[33] K. Park, T. Patten, and M. Vincze, "Pix2pose: pixel-wise coordinate regression of objects for 6D pose estimation," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7667–7676, Seoul, Korea (South), October 2019.

[34] M. Oberweger, M. Rad, and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3D object pose estimation," *Computer Vision - ECCV 2018*, LNCS, vol. 11219, pp. 125–141, 2018.

[35] J. Liebelt, C. Schmid, and K. Schertler, "Viewpoint-independent object class detection using 3D feature maps," in *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Anchorage, AK, USA, June 2008.

[36] M. Sun, G. Bradski, B. X. Xu, and S. Savarese, "Depth-encoded hough voting for joint object detection and shape recovery," in *Proceedings of the Computer Vision - ECCV 2010*, pp. 658–671, LNCS, New York, NY, USA, September 2010.

[37] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich, "Viewpoint-aware object detection and pose estimation," in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 1275–1282, Barcelona, Spain, November 2011.

[38] F. Michel, A. Kirillov, E. Brachmann, A. Krull, S. Gumhold, B. Savchynskyy, and C. Rother, "Global hypothesis generation for 6D object pose estimation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 115–124, Honolulu, HI, USA, July 2017.

[39] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep object pose estimation for semantic robotic grasping of household objects," *CoRL*, vol. 87, pp. 306–316, 2018.

[40] T. Chen, D. Gu, "CSA6D - Channel-Spatial Attention Networks for 6D Object Pose Estimation," *Cognitive Computation*, vol. 14, no. 2, pp. 702–713, 2022.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *Proceedings of the European Conference on Computer Vision*, pp. 630–645, Spinger, New York, NY, USA, September 2016.

[42] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, December 2015.

[43] L. Rosas-arias and G. Benitez-garcia, "Fast and Accurate Real-Time Semantic Segmentation with Dilated Asymmetric Convolutions," in *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 2264–2271, Milan, Italy, January 2021.