Learning Affordance Segmentation: An Investigative Study

Chau Nguyen Duc Minh, Syed Zulqarnain Gilani, Syed Mohammed Shamsul Islam and David Suter
Department of Computing and Security, School of Science
Edith Cowan University, Joondalup, WA 6027, Australia
{c.nguyenducminh, s.gilani, syed.islam, d.suter}@ecu.edu.au

***Abstract*—Affordance segmentation aims at recognising, localising and segmenting affordances from images, enabling scene understanding of visual content in many applications in robotic perception. Supervised learning with deep networks has become very popular in affordance segmentation. However, very few studies have investigated the factors that contribute to improved learning of affordances. This investigation is essential to improve precision and balance cost-efficiency when learning affordance segmentation. In this paper, we address this task and identify two prime factors affecting precision of learning affordance segmentation: (1) The quality of features extracted from the classification module and (2) the dearth of information in the Region Proposal Network (RPN). Consequently, we replace the backbone classification model and introduce a novel multiple alignment strategy in the RPN. Our results obtained through extensive experimentation validate our contributions and outperform the state-of-the-art affordance segmentation models.**

I. Introduction

Affordance, a concept coined by well-known influential psychologist James J. Gibson [1, 2], is defined as 'interaction between a living being and an object through the environment' [2]. In the realm of computer vision, affordance is defined as functional interaction between objects and humans [3, 4]. For instance, the affordance of the inner top rim of a cup is to "`contain`". Thus, a region of pixels belonging to a part of an object is considered as an affordance, if it shares the functionality of that part. Affordance is arguably useful to robotic vision and the concept is ubiquitous in human machine interactions and autonomous visual systems.

The goal of this work is to examine two main factors that affect *learning in affordance segmentation*, namely the quality of features extracted from the classification module and the dearth of information in the Region Proposal Network (RPN). To achieve this, first we explore the relationship between semantic, instance and affordance segmentation. Note that classification is common to these three segmentation tasks (see Fig. 1) and depends largely on the quality of features. It is thus natural that a better feature extraction module will improve the accuracy of learning how to segment affordances. To provide empirical evidence for our claim, we conduct a range of experiments by varying the type of feature extraction module and thereby the quality of features. Our results in experiments based on high quality of feature extraction module demonstrate high precision in affordance segmentation.

We use AffordanceNet [4], a state-of-the-art affordance segmentation network, as our baseline model. This network consists of only one feature extraction module [5], which is both insufficient and computationally expensive [5]. Therefore, we take some inspiration from popular image segmentation models that utilize more efficient convolutional neural networks (CNNs) as feature extraction modules [6–10]. These modules are robust and flexible in visual classification tasks and hence can be used for learning affordance segmentation.

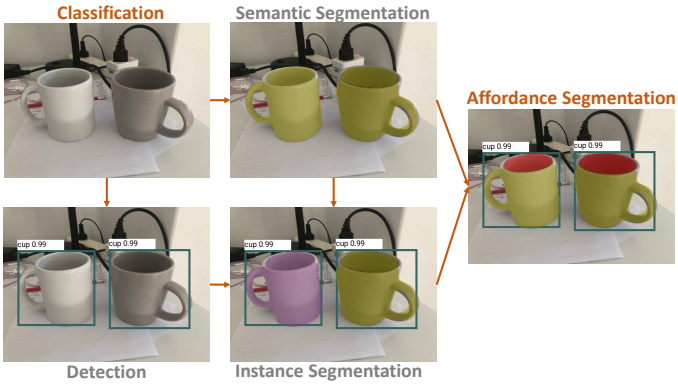

Fig. 1. Pictorial depiction of the relationship between semantic, instance and affordance segmentation and their dependence on classification. Classification is the simplest task but is prerequisite for all segmentation and detection tasks.

The second factor affecting the learning of affordance segmentation is the dearth of information extracted from the features in the Region Proposal Network (RPN). We address this by harnessing multiple deep features which also favour pixel level segmentation [11–13]. More specifically, we generate RPNs are different levels of convolution (of our classification backbone model) and align them with their respective levels in the Deconvolutional Network (see Fig. 2 for details). This method, called Multiple Alignment, differs from classical semantic segmentation [11–16] and instance segmentation [17–19] methods as it builds a bridge between the features of classification task and those of affordance segmentation.

To summarize our contributions, we replace the feature extraction module in the classification network to explore its effect on precision of learning affordance segmentation. We also provide a novel mechanism of multiple alignments around the RPN module. Our results based on these approaches significantly outperform the state-of-the-art affordance segmentation model.

The rest of this paper is organised into six sections. Section II summarizes the relevant works, Section III formulates the problem of affordance segmentation, Section IV introduces the methodology to solve that problem, Section V gives details of the experiments carried out while Section VI provides the conclusion for this work.

II. Background

A. Related work

Semantic, instance and affordance segmentation Image segmentation has witnessed a considerable increasing trend in the use of deep networks since the release of FCN [11]. FCN, an example of semantic segmentation, harnessed deconvolutional/upsampling operation to classify pixel-wise level within an image [11]. Similarly, image semantic segmentation was also mentioned by ParseNet [12], Conv & DeconvNet [14], DeepLab [15], PSPNet [13] and EncNet [16]. Other works developed for semantic segmentation to instance segmentation: FPN [17], Mask R-CNN [18] and PAPNet [19]. Instance segmentation can be viewed as semantic segmentation once it is repeatable in a number of region proposals within an image. To combine these two distinct areas, a work in [20] proposed **panoptic segmentation** as the unified concept. Moreover, a work in [4] claimed an additional form of segmentation: affordance segmentation by leveraging semantic and instance segmentation. They constructed AffordanceNet [4] which is an end-to-end deep network that segments multiple affordances of an object. Their backbone model is limited by VGG16 [5]. However, there are others which had not been considered: ResNet [6], ResNeXt [7], FPN [17], SENet [8], NASNet-A [21], SKNet [9], EfficientNet [10] and recently Noisy-Student [22]. We hypothesise that these backbone models can have high implications for segmentation.

Multiple deep convolutional layers A range of methods applied multiple deep features to backbone models. FCN [11] achieved semantic segmentation by fusing the output from not only final features but also intermediate ones. ParseNet [12] concatenated multiple features from different levels. U-Net [23] exploited shortcut connection that correspond each level of feature to each level of spatial size of output prediction. Furthermore, FPN [17, 24] inherited lateral connections and top-down forward to make predictions at each level stage and then concatenated all predictions. Although these methods used skip connections for gaining advantages of multiple deep features, they have not fused interstate features and deepest features into deconvolutional layer.

III. Problem formulation

The goal of affordance segmentation consists of three tasks: localisation, recognition and segmentation. The problem is to minimise the objective function between a prediction $\hat{\boldsymbol{Y}}$ and a target $\boldsymbol{Y}$ for these tasks. Let $\boldsymbol{X} = (x_{ij}) \in \mathbb{R}^{b_h \times b_w}$ denote a matrix of a region proposal within an image, each x_{ij} is associated with each pixel value, where (b_h, b_w) is the height and the width of the region proposal. The matrix $\boldsymbol{X}$ corresponds to these three ground-truth: y^o for classification, $\boldsymbol{y}^b$ for localisation and $\boldsymbol{Y}^a$ for affordance segmentation task. Firstly, let $y^o \in O$ denote the corresponding object label; where O denote a set of object classes plus a background class: $O = \{o_0, o_1, ..., o_{n_o} | o_i \in \mathbb{N}\}$, n_o is the number of object class labels. Secondly, let $\boldsymbol{y}^b = [y^{b_x}, y^{b_y}, y^{b_h}, y^{b_w}]$ denote a vector of coordinates bounding-box for $\boldsymbol{X}$, where $y^{b_{i=x,y,h,w}} \in \mathbb{R}$, (y^{b_x}, y^{b_y}) represents the coordinates of the top-left corner point of a ground-truth box and (y^{b_h}, y^{b_w}) represents its height and width respectively. Finally, let $\boldsymbol{Y}^a = (y^a_{ij}) \in A^{b_h \times b_w}$ denote a matrix of corresponding affordance label (each element y^a_{ij} is associated with each x_{ij}); where A is a set of affordance classes plus a background class: $A = \{a_i\}_0^{n_a} \in \mathbb{N}$ and n_a is the number of affordance class labels. In general, a tuple $(\boldsymbol{X}, y^o, \boldsymbol{y}^b, \boldsymbol{Y}^a)^{(i)}$ is called as the i^{th} training example. The goal of three tasks are finding mapping functions f_{cls} for classification (cls), f_{reg} for regression (reg) localisation, f_{aff} for affordance (aff) segmentation. Accordingly, $f_{cls} : \boldsymbol{X} \rightarrow y^o$; $f_{reg} : \boldsymbol{X} \rightarrow \boldsymbol{y}^b$ and $f_{aff} : x_{ij} \rightarrow y^a_{ij}$. Concretely, these can be placed into a single multi-task mapping function given by:

$$f : \boldsymbol{X} \rightarrow (y^o, \boldsymbol{y}^b, \boldsymbol{Y}^a) \tag{1}$$

Relationship between an object and affordances Suppose R is a relation set of "`tuples`" from O to A, where each first element comes from O and from each second element comes from A. This is given by: $R = \{r_1, r_2, ..., r_{n_o}\}$, where $r_{i=1 \rightarrow n_o} = (o_i, a_j, ..., a_{n_a})$, where $o_{i=1 \rightarrow n_o} \in O$ and $a_{j=1 \rightarrow n_a} \in A$. Instance segmentation can be viewed as a special case of affordance segmentation. This is because the set A becomes equal to set O: $a_i = o_i$ and $n_o = n_a$. R in this case is a relation set of "`pairs`" from O to A: $R = \{r_i\}_1^{n_o}$, where $r_{i=1 \rightarrow n_o} = (o_i, a_i)$. This relation is also known as an injective surjective function or bijection. Semantic segmentation is similar to affordance segmentation. One highlighting difference between these types of segmentation is that the former applies only to the whole image while the latter applies to a range of Regions of Interest within that image (see Fig. 1).

IV. Methodology

Our network architecture of affordance segmentation is developed upon the work of [18] and [4]. There are basically three main modules within the architecture (each has several related sub-stages): the first module as feature extraction, Region Proposal Network and Region of Interest alignment.

Our work mainly utilise diverse feature extraction modules to explore the effect on performance of segmenting affordances (see Figure 2 and Table I). Accordingly, we select these modules: RESNET50, RESNET101 [6], RESNeXt50-SE, RESNeXt101-SE and SE154 [8] due to their high quality of extracting features. These modules are designed in distinct patterns but their standard structure is divided in these common stages: $conv1$, $conv2$, $conv3$, $conv4$, $conv5$ for simplifying the forward-propagation process (see Fig. 2). Each stage comprises a series of convolutional interstate-layer that carrying features. The feature at the current layer is straightforwardly propagated to the next layer through the

backbone. We consider the feature has spatial size of height and width in addition to the size of channel volume. Generally, the deeper the layer is, the smaller the height and width it has but the wider channel volume are.

The second module is a Region Proposal Network (RPN) [25] for generating proposals. The input of this module is taken from the output of $conv4$ or $conv5$ due to their sufficiently deep features. The number of proposals at this stage is highly large. Therein, many proposals do not contain any foreground while the others can be overlapped by each other. This consumes high memory and computational cost. To overcome this challenge, we use the algorithm Non-Maximum Suppression from [25] to remove overlapping proposals. This algorithm prunes non-essential proposals, which consequently reduces the total numbers of proposals and lessens computationally cost.

After Non-Maximum Suppression (NMS), the next module is to map features with Region of Interest (RoI) by the alignment process. Two parallel branches are: the R-CNN (Convolutional Network) branch as in [4] and an affordance mask branch (Deconvolutional Network).

At this step, we provide a novel multiple alignment technique which aggregates features from deep stages to the affordance mask branch. This is due to the fact that the existing framework [4] aligned only one stage of feature extraction to the mask branch, e.g. $conv5$ stage. This single alignment process can loose information of features and even bias the major effect of the deepest stage, compared to features in earlier stages. This can lead to inefficiency in segmenting affordances. To address this issue, we propose multiple alignment strategy for features at each stage, such as $conv2$, $conv3$ and $conv4$ (see Figure 2 and Table II). Particularly, $conv5$ is not involved in the multiple alignment process with our approach of feature extraction replacement with RESNET50, RESNET101, RESNeXt50-SE, RESNeXt101-SE or SE154. This $conv5$ is rather moved to the R-CNN branch after RPN (see Fig. 2) because facilitating this stage before RPN is inefficient for aforementioned feature extraction modules. This critical point was also debated by the work in [18] through their empirical evidence.

The rest of this section detail modules of the network architecture which are conceptually clarified as follows:

Forward-propagation in the first module We consider the feature $\boldsymbol{Z}^l_{\langle H^l,W^l,C^l\rangle}$ at layer l^{th} as a tensor. (H^l, W^l) is the height and width for the spatial dimension, and C^l is the volume channel dimension. At layer $l=0: \boldsymbol{Z}^{l=0} = \boldsymbol{I}_{\langle H,W,3\rangle}$, where $\boldsymbol{I}$ is an input image. Let $\mathcal{F}^l$ be a function at layer l^{th} within a convolutional network (ConvNet) (e.g. $\mathcal{F}^l$ can be an activation function or a combination of non linear functions like activation, dropout, batch-normalisation, etc.). Thus, $\boldsymbol{Z}^l = \mathcal{F}^{1\leq l\leq n_l}(\boldsymbol{Z}^{l-1})$, where n_l is the total number of forward layers. A ConvNet (suppose $conv1$, $conv2$, $conv3$, $conv4$ or $conv5$) at layer $j^{th} \geq l^{th}$ can be represented by a function composition: $\boldsymbol{Z}^{l\leq j\leq n_l} = \mathcal{F}^j \odot ... \odot \mathcal{F}^2 \odot \mathcal{F}^1(\boldsymbol{I}) = \bigodot^{l=1...j} \mathcal{F}^l(\boldsymbol{I})$ or $\mathcal{F}: \boldsymbol{I} \rightarrow \boldsymbol{Z}^j$ or simply $\mathcal{F}: \mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^{H^j\times W^j\times C^j}$:

$$\boldsymbol{Z}^j_{\langle H^j,W^j,C^j\rangle} = \bigodot^{l=1...j} \mathcal{F}^l(\boldsymbol{I}_{\langle H,W,3\rangle}) \quad (2)$$

$$\text{s.t. } (H^{j-1}, W^{j-1}) \geq (H^j, W^j) \;\&\; C^{j-1} \leq C^j \quad (3)$$

In this module, we replace the backbone (including $conv1$, $conv2$, $conv3$, $conv4$ and $conv5$, mainly used by [4]), by other efficient ones (see §V-B).

Region proposals generation at RPN At layer j^{th}, the feature $\boldsymbol{Z}^j_{\langle H^j,W^j,C^j\rangle}$ is filtered by a window sliding with a filter size $(f \times f)$ for generating region proposals, where $f < W^j \leq H^j$. Sliding each window into feature extraction $\boldsymbol{Z}^j$ have totally n_w sliding windows: $n_w = \lfloor (H^j - f + 2p)/s + 1 \rfloor \times \lfloor (W^j - f + 2p)/s + 1 \rfloor$, where p is padding and s is stride; e.g. $(f = 3, p = 1, s = 1)$ or $(f = 1, p = 0, s = 1)$ then $n_w = H^j \times W^j$. Each region proposal has a size different than others. To tackle this issue, each sliding window incorporates a number of anchors n_{anc} in various sizes: $n_{anc} = n_{sca} \times n_{rat}$, where n_{sca} and n_{rat} are a number of scales (sca) and aspect ratios (rat) respectively as default hyper-parameters (see [25]). Consequently, the total number of region proposals are $n_p = n_w \times n_{anc}$. The anchor box at this stage is commonly known as a region proposal.

RPN output, its ground-truth and RPN loss Each sliding window also incorporates two predicted outputs: one is the predicted coordinate for each anchor box $\hat{\boldsymbol{b}} = [\hat{b}_x, \hat{b}_y, \hat{b}_h, \hat{b}_w]$ and the other is predicted score $\hat{p}$ for each anchor having an object or not. Accordingly, $\hat{\boldsymbol{b}}$ is transformed to regression offset boxes: $\hat{\boldsymbol{t}} = [\hat{t}_x, \hat{t}_y, \hat{t}_h, \hat{t}_w]$. The corresponding ground-truth of $\hat{\boldsymbol{t}}$ is $\boldsymbol{t} = [t_x, t_y, t_h, t_w]$. We refer to the terminology in [25, 26] for the estimation of transformation $\boldsymbol{t}$ and $\hat{\boldsymbol{t}}$:

$$\hat{t}_x = (\hat{b}_x - b^a_x)/b^a_w, \;\; \hat{t}_y = (\hat{b}_y - b^a_y)/b^a_h, \quad (4)$$

$$\hat{t}_h = log(\hat{b}_h/b^a_h), \;\; \hat{t}_w = log(\hat{b}_w/b^a_w) \quad (5)$$

$$t_x = (b_x - b^a_x)/b^a_w, \;\; t_y = (b_y - b^a_y)/b^a_h, \quad (6)$$

$$t_h = log(b_h/b^a_h), \;\; t_w = log(b_w/b^a_w) \quad (7)$$

Where $\hat{\boldsymbol{b}}, \boldsymbol{b^a}$ and $\boldsymbol{b}$ represents predicted box, anchor box and ground-truth box respectively; these subscripts (x, y) and (h, w) represents notation for coordinates of the top-left corner point of bounding box and its height and width respectively. A vector for ground-truth box and one for anchor box: $\boldsymbol{b} = [b_x, b_y, b_h, b_w]$ and $\boldsymbol{b^a} = [b^a_x, b^a_y, b^a_h, b^a_w]$ respectively; where $\boldsymbol{b}$ is for each nearby ground-truth box compared to each anchor box. Next, $\hat{p}$ has its corresponding ground-truth p. The algorithm for estimating the value of each element within p refers to [25] according to the computation of Intersection-over-Union (IoU). Particularly, $p = 1$ if an anchor has the highest IoU overlapping with any nearby ground-truth boxes, or has IoU higher than a threshold t^{iou} (usually $t^{iou} = 0.7$ in experiment); $p = 0$ if an anchor has the IoU lower than a threshold $1 - t^{iou}$; $p \in \emptyset$ if neither positive nor negative anchor. These anchors are not involved in the training process. In terms of the method to optimise RPN loss, we refer to the work in [25].

Non maximum suppression (NMS) and RoI There are totally n_p region proposals at this stage. After processing by the

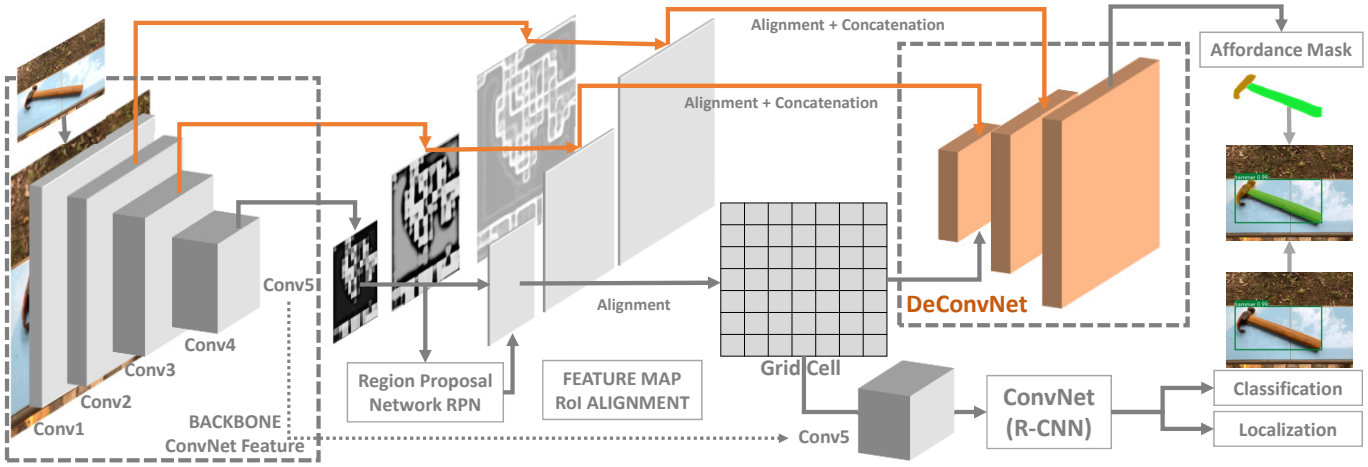

Fig. 2. Network architecture of segmenting affordances. The first module is related to backbone model with $conv1$, $conv2$, $conv3$, $conv4$ convolutional stages, the second module performs RPN and alignment process between RoI and features, the third module is separated to two branches: one is R-CNN with $conv5$ for predicting classification and localisation output and the other is Deconvolutional Network for predicting affordance mask. The solid grey arrows perform forward process, inherited from [4] while the orange solid arrows represent multiple alignment process. The dash square surrounding grey tensor illustrates convolutional/downsampling process while one surrounding orange tensor indicates deconvolutional/upsampling process.

algorithm NMS according to a default threshold $t^{nms} \in (0,1)$, the number of proposals n_p reduces to n_p^{nms}. Once NMS is completed, a region proposal is usually called a Region of Interest (RoI). Next, there are two types of selective RoI to propagate to next branches: foreground RoI and background RoI. The former has totally n_{fg} RoIs while the latter has n_{bg} RoIs: $n_{fg}, n_{bg} \leq n_{fg} + n_{bg} < n_p^{nms} < n_p$. Filtering foreground and background RoIs is according to a threshold t^{fg} and an interval threshold $[t_{low}^{bg}, t_{high}^{bg}]$. RoI is a foreground if $0 < t^{fg} \leq IoU$ (empirically $t^{fg} = 0.5$) while RoI is a background if $0 < t_{low}^{bg} \leq IoU < t_{high}^{bg} < 1$ (empirically $[t_{low}^{bg}, t_{high}^{bg}] = [0.1, 0.5]$). After filtering, RoI aligned with features forward two branches : one is similar to R-CNN [27] for both classification and localisation and the other is for affordance segmentation (see Fig. 2). Inspired by [4], all foreground and background RoIs $n_{fg} + n_{bg}$ are aligned with features at RoI-Alignment layer [18] for the R-CNN branch [26] while only foreground RoIs n_{fg} are aligned at RoI-Alignment layer [18] for the affordance mask branch.

RoI alignment at R-CNN branch: This module is to align all RoIs (foreground and background RoIs) with corresponding feature maps $\boldsymbol{Z}^j$ into a default grid-cell $(g \times g)$ size (empirically $g = 7$, see Fig. 2). Thus, $f : \boldsymbol{Z}^j_{\langle h^{roi}, w^{roi}, c^j \rangle} \rightarrow \boldsymbol{Z}^j_{\langle g,g,c^j \rangle}$. Inspired by [18], the alignment process is bilinear interpolation. This mapping feature $\boldsymbol{Z}^j_{\langle g,g,c^j \rangle}$ is then forward to the R-CNN branch similar to [27] for classification and localisation task. Firstly, the classification task outputs prediction for object classes $\boldsymbol{z} = [z_{o \in O}]$. Then, $\boldsymbol{z}$ is forward through a soft-max layer to compute the condition probability for each object class given by an RoI: $\hat{p}_{o \in O} = \exp^{z_o} / \sum_{i \in O} \exp^{z_i}$. The corresponding ground-truth of $\hat{p}_o$ is p_o. Secondly, the localisation task outputs an offset bounding-box regression: $\hat{\boldsymbol{t}}^o = [\hat{t}_x^o, \hat{t}_y^o, \hat{t}_h^o, \hat{t}_w^o]$ given in [26]. The corresponding ground-truth box regression for $\hat{\boldsymbol{t}}^o$ is $\boldsymbol{t}^o = [t_x^o, t_y^o, t_h^o, t_w^o]$. The computation of $\hat{\boldsymbol{t}}^o$ and $\boldsymbol{t}^o$ is refer to the work in [26]. The multi-task loss $\mathcal{L}_{1,2}$ is as follows [26]:

$$\mathcal{L}_{1,2} = \lambda_1 \times \mathcal{L}_{cls}(\hat{p}_o, p_o) + I[o \geq 1] \lambda_2 \mathcal{L}_{reg}(\hat{\boldsymbol{t}}^o, \boldsymbol{t}^o) \quad (8)$$

$$\mathcal{L}_{cls} = -\log(\hat{p}_o); \; \mathcal{L}_{reg} = \sum_{i \in \{x,y,h,w\}} L_1^{smooth}(\hat{t}_i^o - t_i^o), \quad (9)$$

$$\text{where } L_1^{smooth}(x) = \begin{cases} 0.5x^2 & \text{if } |x| \leq \delta \\ \delta|x| - 0.5\delta^2 & \text{otherwise} \end{cases}$$

Where $o \in O$ (see §III) and the indicator activation $I[o \geq 1] = 1$ if $o \geq 1$ and 0 otherwise. This prevents the background class 0 from being involved in the loss. Particularly, L_1^{smooth} [27] can be originally known as Huber Loss [28] if $\delta = 1$, where δ is a fine-tuning hyper-parameter. λ_1 and λ_2 are default hyper-parameters to weight the classification loss $\mathcal{L}_{cls}$ and regression loss $\mathcal{L}_{reg}$ respectively.

RoI alignment for affordance mask branch This module aims to align all foreground (fg) RoIs (positive RoIs) with corresponding feature maps $\boldsymbol{Z}^j$ into the grid-cell $(g \times g)$ size, $f : \boldsymbol{Z}^j_{\langle h^j, w^j, c^j \rangle} \rightarrow \boldsymbol{Z}^j_{\langle g,g,c^j \rangle}$. Similar to the R-CNN branch, the alignment process is also bilinear interpolation.

Multiple alignment In the conventional approach, each feature from the last convolutional stage, usually $conv4$ or $conv5$, are aligned with each mapping RoI. In our approach, we utilise $conv4$ rather $conv5$ due to expensive computation of $conv5$ in Deconvolutional Network. Apart from taking advantaging of $conv4$, we align each feature from $conv3$ with each corresponding RoI. This is the second alignment in addition to the first one as conventional approach. After this alignment, we fuse the features output into the first layer of Deconvolutional Network. Moreover, we also attempts to process the third alignment between each feature from $conv2$ with each RoI, then fuse them to the second layer of Deconvolutional Network (see Fig. 2 and §V-B).

Affordance mask loss After Deconvolutional Network, the output is a tensor $\boldsymbol{Z}^a = (z_{ijk}) \in \mathbb{R}^{n_a \times m \times m}$, where m is

TABLE I

PERFORMANCE OF AFFORDANCE SEGMENTATION WITH DIFFERENT FEATURE EXTRACTION MODULES IN FULL CONFIGURATION (THE MAXIMUM SCALE UP TO 1000). THE METRIC BENCHMARKING FOR EACH MODEL IS $F_\beta^w * 100$ [29]. THE ABBREVIATION 4C-RPN-5C ILLUSTRATES THE PROCESS OF REPLACING BACKBONE MODELS WITH FIVE CONVOLUTIONAL STAGES. ACCORDINGLY, 4C IS THE FIRST FOURTH STAGES BEFORE RPN WHILE 5C IS THE FIFTH STAGE AFTER RPN (SEE FIG. 2). THIS REPLACEMENT IS SLIGHTLY DIFFERENT THAN [4] AS THEIR WORK UTILISED 5C BEFORE RPN INSTEAD OF AFTER RPN. THE ABBREVIATION (32X4D) IS THE VERSION NAME OF RESNEXT101-SE [7]. SE REPRESENTS SQUEEZE AND EXCITATION [8].

Feature extraction	Description	contain	cut	display	engine	grasp	hit	pound	support	w-grasp	Average
VGG16	AffordanceNet [4]	79.61	75.68	77.81	77.50	68.48	70.75	69.57	69.81	70.98	73.35
RESNET50	4C-RPN-5C	79.03	75.39	78.04	78.30	69.88	72.34	70.97	71.39	72.69	74.23
RESNET101	4C-RPN-5C	80.56	77.68	80.09	80.29	72.55	74.82	73.70	74.15	75.40	76.58
RESNEXT50-SE	4C-RPN-5C (32x4d)	79.55	76.58	79.03	79.36	72.34	74.64	73.60	73.81	75.34	76.03
RESNEXT101-SE	4C-RPN-5C (32x4d)	81.43	78.97	81.56	82.10	75.14	77.46	76.58	77.30	78.76	78.81
SE154	4C-RPN-5C	82.01	80.25	82.84	83.39	76.56	78.73	77.93	78.68	80.32	80.08

the resized affordance mask (m is aka the number of pixels within a RoI) and n_a is the number of affordance classes (see §III). Each resized prediction of pixel can be represented as $\boldsymbol{z} = [z_{a\in A}] \in \mathbb{R}^{n_a \times 1 \times 1}$. This is then forward to a soft-max layer to compute corresponding $\hat{p}_a$. This $\hat{p}_a$ is the condition probability for each affordance class given by a resized pixel within a RoI: $\hat{p}_{a\in A} = \exp^{z_a} / \sum_{i \in A} \exp^{z_i}$. To optimise the affordance loss $\mathcal{L}_{aff}$, the corresponding ground-truth for $\hat{p}_a$ is p_a. The loss $\mathcal{L}_3$ for this branch is as follows [4]:

$$\mathcal{L}_3 = I[a \geq 1] \times \lambda_3 \times \mathcal{L}_{aff}(\hat{p}_a, p_a), \tag{10}$$

$$\text{where } \mathcal{L}_{aff} = -\frac{1}{m} \sum_{j \in RoI} \log(\hat{p}_a^j) \tag{11}$$

Where $a \in A$, the indicator activation $I[a \geq 1] = 1$ if $a \geq 1$ and 0 otherwise and λ_3 is a default hyper-parameter to weight the affordance segmentation loss $\mathcal{L}_{aff}$. The total loss $\mathcal{L}$ for those branches is: $\mathcal{L} = \mathcal{L}_{1,2} + \mathcal{L}_3$. (see Eq. [8, 10]).

V. EXPERIMENT

A range of experiments are conducted for these two approaches: feature extraction module replacement and multiple alignment on top of backbone replacement. All networks are trained on the IIT-AFF dataset [30]. In this section, we discuss configuration for the training and inference process. We also explain the evaluation metric used to estimate the performance of affordance segmentation for each approach. We provide an in-depth quantitative and qualitative analysis of results and finally visualise the output from real images and IIT-AFF dataset (see Fig. 3).

A. Dataset

IIT-AFF dataset has a total of $8,835$ images [30]. While around 40% of the total images were taken by the authors of [30], the rest of dataset was taken from ImageNet [31]. There were 10 classes for object detection and 9 affordance classes for affordance segmentation. Additionally, there is a "`background`" class for both object and affordance labels. Thus, the total object classes are $10+1$ and the total affordance classes are $9+1$. This dataset consisted of nearly 14,642 bounding boxes and 24,677 affordance pixel-wise masks. We use splitting ratio of the dataset into training and deployment set as approximately $7:3$. Hence, there are a total of 6,184 and 2,651 images for the training and inference process respectively. Particularly, we also facilitate an augmentation method, similar to [4], which flips all training images and stack these flipped images into original ones during the training process. This augmentation is beneficial for the network to learn complex and hierarchical structures of an object. Therefore, the total number of training images becomes double $(6,184 \times 2)$.

B. Experimental design

Feature extraction module replacement AffordanceNet utilised VGG16 [5] as a feature extraction module. We replace VGG16 by the following models: RES50, RES101 [6], and a series of Squeeze and Excitation (SE) networks, including RESNeXt50-SE (32x4d), RESNeXt101-SE (32x4d) and SE-154 [7, 8]. Then, we conduct a range of experiments using these feature extraction modules. We use a learning rate of 0.001 for all experiments except for RESNeXt101-SE and SE154. These exceptional cases are trained with the learning rate of 0.0005. This is because such a low learning rate can address the issue of exploding gradient. We optimise the model by Stochastic Gradient Descent (SGD). Although SGD is less efficient than ADAM [32], we use this optimisation method for fair comparison with AffordanceNet [4] which was trained using SGD. We set the hyper-parameter as follows: $(\lambda_1, \lambda_2, \lambda_3) = (3, 2, 3)$ (see Eq. [8, 10]) as similar to [4]. We train each model with a batch size of 16 for 200,000 iterations and record consistent results at the last iteration.

Multiple alignment approach We facilitate the learning network on top of these backbone models: RES50, RES101, RESNeXt101-SE and SE154 separately. The baseline for each backbone model is the one with the original feature extraction module. Accordingly, we attempt to utilise the network by

TABLE II

PERFORMANCE OF AFFORDANCE SEGMENTATION WITH AND WITHOUT MULTIPLE ALIGNMENT. THE EVALUATION METRIC IS $F_\beta^w * 100$ [29]. MA2_1F IS DOUBLE MULTIPLE ALIGNMENT WITH ONE FUSION AND MA3_2F IS TRIPLE MULTIPLE ALIGNMENT WITH TWO FUSIONS (SEE §V-B).

Feature extraction	Multi-align	contain	cut	display	engine	grasp	hit	pound	support	w-grasp	Average
RESNET50	w/o MA	79.18	75.72	77.90	77.90	68.56	70.93	69.46	69.84	71.19	73.41
	MA2_1F	79.14	75.79	77.73	77.68	68.91	70.81	69.68	69.99	71.14	73.43
RESNET101	w/o MA	80.06	77.14	79.50	79.69	71.79	73.95	72.78	73.19	74.52	75.85
	MA2_1F	80.52	77.74	79.97	80.04	72.30	74.42	73.40	73.85	75.16	76.38
	MA3_2F	81.05	78.04	80.13	80.37	72.69	74.81	73.90	74.29	75.64	76.77
RESNEXT101-SE	w/o MA	81.43	78.97	81.56	82.10	75.14	77.46	76.58	77.30	78.76	78.81
	MA2_1F	81.45	79.23	81.83	82.47	75.49	77.80	76.73	77.48	79.11	79.07
SE154	w/o MA	82.01	80.25	82.84	83.39	76.56	78.73	77.93	78.68	80.32	80.08
	MA2_1F	81.90	79.84	82.54	83.21	77.17	79.17	78.21	78.87	80.55	80.16

adding multiple alignment process from a feature extraction layer to a upsampling layer of Deconvolutional network. There are two ablation studies: double multiple alignment with one fusion and triple multiple alignment with two fusions. The former, named MA2_1F, is to add one alignment of *conv3* layer with RoI and then fuse them to *deconv1* layer by a concatenation operator. The latter, named MA3_2F, additionally enhances the former by adding another alignment between *conv2* layer and RoI, then fuses them to *deconv2* layer by a concatenation operator (see Table II).

TABLE III

COMPARISON OF THE NUMBER OF PARAMETERS BETWEEN A MODEL WITH AND WITHOUT MULTIPLE ALIGNMENT. THE NETWORK ARCHITECTURE OF VGG16 IS DISTINCT THAN THE REST OF BACKBONES (*conv5* BEFORE RPN WHILE THE OTHERS HAVE *conv5* AFTER RPN). THEREFORE, NONE OF MULTIPLE ALIGNMENT STRATEGIES ARE APPLIED IN VGG16.

Backbone/Approach	without MA	MA2_1F		MA3_2F	
	Parameters	Parameters	%change	Parameters	%change
RESNET50	38,423,104	41,077,312	6.91	43,575,872	13.41
RESNET101	57,415,232	60,069,440	4.62	62,568,000	8.97
RESNEXT101-SE	61,786,944	64,441,152	4.30	66,939,712	8.34
SE154	127,904,192	130,558,400	2.08	133,056,960	4.03
VGG16	146,654,400	None	None	None	None

Inference During the testing process, we set the IoU threshold for NMS algorithm at 0.7 at the RPN and at 0.3 after the RPN for suppressing overlapping bounding-boxes. In the backbone replacement experiments, the number of region proposals before and after RPN is 2000 and 300 respectively for all cases except for experiments in RESNeXt50-SE, RESNeXt101-SE and SE154. These exceptional cases are deployed under 300 and 100 proposals to reduce the expensive computation and make the inference time more efficient. With regard to the approach of multiple alignment, all experimental scenarios are inferred by 300 proposals before RPN and 100 after RPN. We train and deploy all models in similar scaling configuration as of [4]: the maximum scaling of image is 1000 during the training process. We implement and record results in all experiments by a GPU of NIVIDA Quadro RTX 8000.

Evaluation Metric We benchmark all experiments based on the F_β^w [29] similar to [4] for fair comparisons:

$$F_\beta^w = (1+\beta^2)\frac{\text{Precision}^w \times \text{Recall}^w}{\beta^2\text{Precision}^w + \text{Recall}^w} \tag{12}$$

Where the terminology Precision^w and Recall^w is weighted Precision and weighted Recall (w is a notation), borrowed from [29]. We not only estimate the average performance across all affordance classes, but we also evaluate performance within each affordance class (see Tables I and II). This metric estimates the similarity between a predicted foreground and a ground-truth affordance mask in pixel-to-pixel. This metric is related to F_β that is commonly known as F_1 when $\beta = 1$.

C. Results

Feature extraction module replacement We record the performance of affordance segmentation in the approach of backbone replacement as shown in Table I. All experimental scenarios achieve significant improvement over the state-of-the-art baseline in [4]. Accordingly, the average performance in the approach with RESNET50 [6] is around 74.23, which is slightly higher than the one in AffordanceNet [4]. With regard to RESNET101 [6], RESNeXt50-SE [8], RESNeXt101-SE [8] and SE154 [8], the average $F_\beta^w * 100$ surpasses considerably the AffordanceNet, particularly by around up to 7.0 points for SE154. The approach with SE154 demonstrates the highest performance among others. Note that SE154 is the deepest feature extraction module with more convolutional layers than others, and RES50 is the most efficient deep feature extraction module with less convolutional layers than others but it can surpass the performance of baseline. The results of feature extraction modules with "*Squeeze and Excitation (SE)*" technique [8] plus aggregation technique [7]

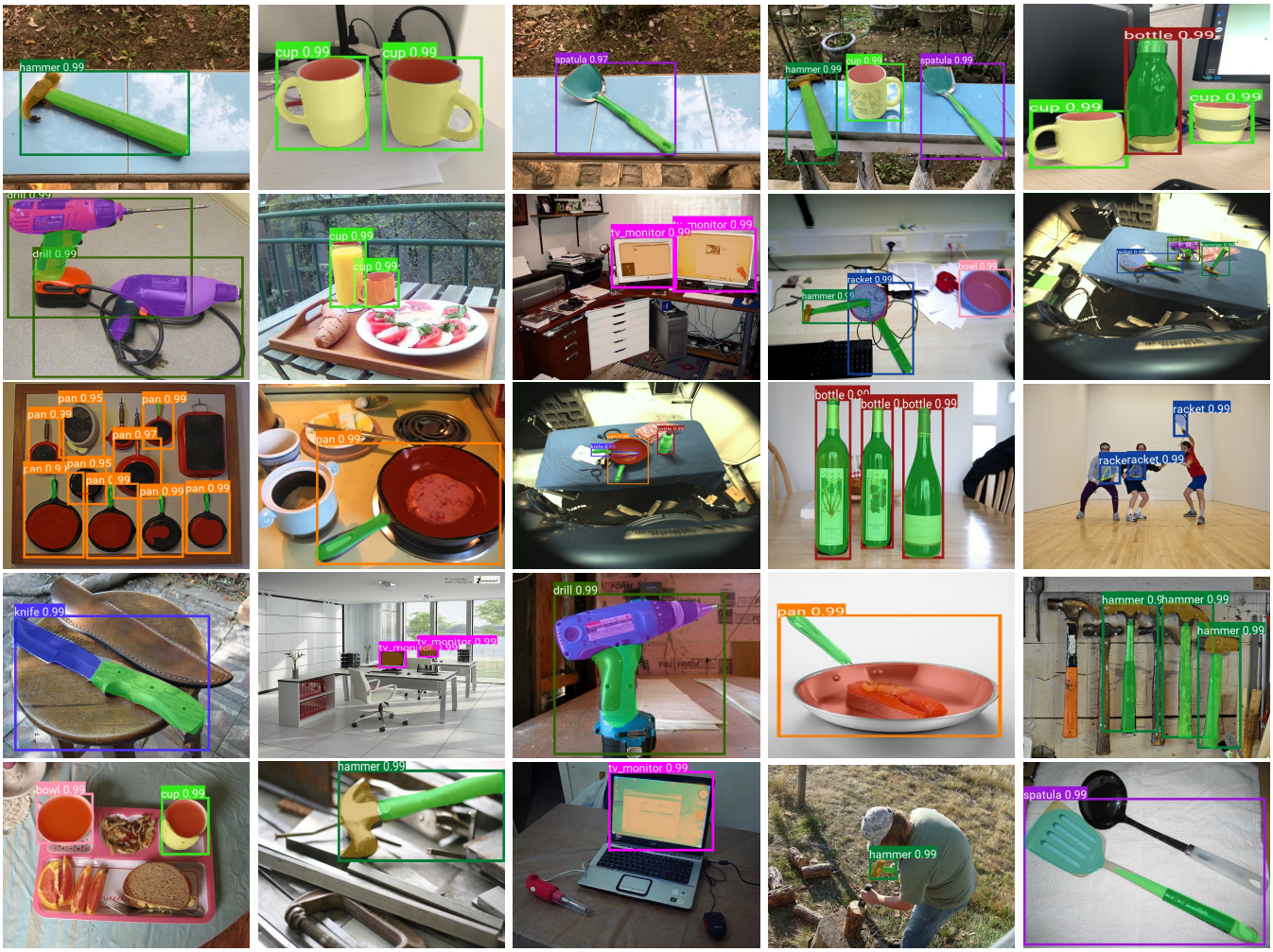

Fig. 3. Some examples of segmenting affordances. **First row:** images taken from real scene. **Other rows:** images from IIT-AFF dataset. Visualisation was done by implementing the trained network with RESNET101 as the feature extraction module.

(RESNeXt50-SE and RESNeXt101-SE) underline higher effect of deeper feature extraction modules, compared to these without SE as well as without aggregation technique (RESNET50 and RESNET101). Accordingly, the average performance of affordance segmentation in RESNeXt50-SE and RESNeXt101-SE record 76.03 and 78.81 of $F_\beta^w * 100$ that are higher than RESNET50 (74.23) and RESNET101 (76.58) respectively. To this end, these achievements highlight the importance of deep feature extraction module. The deeper the feature extraction module, the more improvement of affordance segmentation is.

Multiple alignment Although there are no significant difference with the baseline, the variant of MA2_1F has a non-negative impact on the performance on affordance segmentation (see Table II). Accordingly, there are no significant difference between with and without MA2_1F in the approach of RES50. The same trend is also observed in the case of SE-154. However, there are a slight difference between with and without MA2_1F in the case of RES101, by around 0.5 of $F_\beta^w * 100$. With regard to MA3_2F in RES101, there are improvement in between with and without multiple alignment, by around 1.0 of $F_\beta^w * 100$. There is a small change in point of performance from without and with MA2_1F of RESNeXt101-SE, by nearly 0.3. To this end, the difference is noticeable in the variant of MA3_2F in RES101. Furthermore, we also attempt to estimate the number of parameters in the approach of with and without multiple alignment (see Table III). We set the threshold for the cost of increasing parameters, at 10 % for balancing the cost-efficiency. Therefore, we conduct experiments in MA2_1F, and only one experiment in MA3_2F for RES101. This is due to the fact that the percentage change in parameters of RES50 with MA3_2F is greater than the threshold cost while RESNeXt101-SE and SE154 are expensive in computation. Therefore, none of strategies related to MA3_2F has been done in the case of RES50, RESNeXt101-SE and SE154.

Qualitative analysis To perform the effectiveness of the approach of backbone replacement, we visualise the output of affordance segmentation as shown in Fig. 3. The deployment is inferred from the testing images of IIT-AFF dataset and some real images. There are still some mistaken affordances in the image at first row and fifth column. One reason can be high

correlation of similar affordances. For instance, there is a slight difference between affordance `wrap-grasp` and `grasp` for manipulating an object as they have a functionality `holding` in common, therefore these affordances might exchanged to one another in real time testing. One suggested solution for this problem is to utilise the weight scaling factor of these high-correlated affordances in order to increase the discrepancy between them during the training process. Regardless of this limitation, the task of affordance segmentation have been considerably achieved in the rest of images, for all affordances defined in IIT-AFF dataset. In particular, the system for affordance segmentation performs simultaneously the task of detecting not only a single object but also multiple objects and segmenting multiple affordance classes.

VI. Conclusion

We provided an in-depth explanation of differences between semantic, instance and affordance segmentation. We hypothesised feature extraction module has a high effect on performance of affordance segmentation. Accordingly, we constructed a dynamic network of replacing the feature extraction module with the high quality one. Thus, we provided a method of multiple alignment and compared this novel approach with each feature extraction module case-by-case. We achieved significant performance improvements in the approach of feature extraction replacement while there are slight difference between the approach of feature extraction baseline and multiple alignment. These results support considerably empirical evidence in the importance of deep feature extraction module, transferring learning from classification task to affordance segmentation.

Acknowledgment

The authors would like to thank Department of Computing and Security, School of Science, Edith Cowan University (ECU), ECU Higher Degree by Research Scholarship for funding facilities and resources to complete this research.

References

[1] J. J. Gibson, *The senses considered as perceptual systems.* Houghton Mifflin, 1966.

[2] ——, "The theory of affordances," *Hilldale, USA*, vol. 1, no. 2, 1977.

[3] C. Castellini, T. Tommasi, N. Noceti, F. Odone, and B. Caputo, "Using object affordances to improve object recognition," *IEEE Transactions on Autonomous Mental Development*, vol. 3, no. 3, pp. 207–215, 2011.

[4] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 1–5.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[8] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[9] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 510–519.

[10] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[11] 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*, 2015, pp. 3431–3440.

[12] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *arXiv preprint arXiv:1506.04579*, 2015.

[13] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[14] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[15] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[16] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7151–7160.

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[19] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[20] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 9404–9413.

[21] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.

[22] Q. Xie, E. Hovy, M.-T. Luong, and Q. V. Le, "Self-training with noisy student improves imagenet classification," *arXiv preprint arXiv:1911.04252*, 2019.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[25] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[26] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[27] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[28] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in statistics*. Springer, 1992, pp. 492–518.

[29] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps?" in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 248–255.

[30] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Object-based affordances detection with convolutional neural networks and dense conditional random fields," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5908–5915.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.