Machine Vision and Applications (2019) 30:23–40
https://doi.org/10.1007/s00138-018-0966-3

ORIGINAL PAPER

A hybrid image dataset toward bridging the gap between real and simulation environments for robotics

Annotated desktop objects real and synthetic images dataset: ADORESet

Ertugrul Bayraktar[1] · Cihat Bora Yigit[2] · Pinar Boyraz[2]

Received: 4 December 2017 / Revised: 4 June 2018 / Accepted: 14 July 2018 / Published online: 1 August 2018

Abstract

The primary motivation of computer vision in the robotics field is to obtain a perception level that is as close as possible to human visual system. To achieve this, the inclusion of large datasets is necessary, sometimes involving less-frequent and seemingly irrelevant data to increase the system robustness. To minimize the effort and time in forming such extensive datasets from real world, the preferred method is to utilize simulation environments, replicating real-world conditions as much as possible. Following this solution path, the machine vision problems in robotics (i.e., object detection, recognition, and manipulation) often employ synthetic images in datasets and, however, do not mix them with real-world images. When the systems are trained only using the synthetic images and tested within the simulated world, the tasks requiring object recognition in robotics can be accomplished. However, the systems trained using this procedure cannot be directly used in the real-world experiments or end-user products due to the inconsistencies between real and simulation environments. Therefore, we propose a hybrid image dataset including annotated desktop objects from real and synthetic worlds (ADORESet). This hybrid dataset provides purposeful object categories with a sufficient number of real and synthetic images. ADORESet is composed of colored images with the dimension of 300 × 300 pixels within 30 categories. Each class has 2500 real-world images acquired from the wild web and 750 synthetic images that are generated within Gazebo simulation environment. This hybrid dataset enables researchers to implement their own algorithms for both real-world and simulation environment conditions. ADORESet is composed of fully annotated object images. The limits of objects are manually specified, and the bounding box coordinates are provided. The successor objects are also labeled to give statistical information and the likelihood about the relations of the objects within the dataset. To further demonstrate the benefits of this dataset, it is tested in object recognition tasks by fine-tuning the state-of-the-art deep convolutional neural networks such as VGGNet, InceptionV3, ResNet, and Xception. The possible combinations regarding the data types for these models are compared in terms of time, accuracy, and loss values. As a result of the conducted object recognition experiments, training with all-real images yields approximately 49% validation accuracy for simulation images. When the training is performed with all-synthetic images and validated using all-real images, the accuracy becomes lower than 10%. If the complete ADORESet is employed for training and validation, the hybrid dataset validation accuracy reaches approximately to 95%. This result proves further that including the real and synthetic images together in the training and validation sessions increases the overall system accuracy and reliability.

Keywords Object dataset · Object recognition · Deep convolutional neural network · Deep learning-based robot vision · Synthetic image data · Labeled image data

✉ Ertugrul Bayraktar
bayraktare@itu.edu.tr

[1] Department of Mechatronics Engineering, Graduate School of Science Engineering and Technology, Istanbul Technical University, 34496 Maslak, Istanbul, Turkey

[2] Department of Mechanical Engineering, Istanbul Technical University, Inonu Cd. No:65, 34437 Beyoglu, Istanbul, Turkey

1 Introduction

Recent advancements in technology made intelligent robotic systems indispensable for people to sustain daily tasks and activities. The aspiration of most robotic applications is to advance the perception, movement, and cognition system as close as possible to the capabilities of a human. The key element to achieve this in robotic perception is largely related to computer vision. As a crucial part in visual perception, computer vision in robotics is mainly employed for object detection, recognition, segmentation, and manipulation. The conventional computer vision approach involves feature matching process using a detector and descriptor. Furthermore, these fundamental techniques may require additional steps such as scale-space representation, key point localization at different scales, assigning an orientation to the key points, and acquiring the description of the key points. All these steps absorb a great amount of computational capacity while yielding only insignificant performance increases in terms of accuracy and reliability. Due to their inadequate capabilities regarding the accuracy and speed that affects real-time performance, the feature detector and descriptor methods are currently not favored in real-time robotics applications [9]. In recent works, there has been a massive trend toward deep convolutional neural network models because of their relative advantages in both real-time requirements and accuracy [38,41]. Although the rise of deep convolutional neural networks (CNN) structures has happened, this revolution in machine learning comes with two requirements to be met successfully: (i) specific hardware that enables their implementation in parallel processing and (ii) large and labeled image datasets with appropriate number of images in each class for training, validation, and testing of the resultant networks.

The large image dataset necessity of the deep CNNs has been answered by many research groups with real and synthetic images. The primary reason behind this necessity is to train the deep CNN models with as much as possible various images for the same class to learn maximum possible distinctive features. In other words, the weakness caused by the rotation and scale dependence of deep CNNs unlike classical feature matching techniques is overcome by large number of training data samples and data augmentation. The robotics community has started forming and using these datasets to train their object recognition systems based on deep CNN structures. However, most of the studies use these two groups of images (i.e., real world and simulation) separately [26,32].Namely, the object recognition system trained using only real-world images is tested in the simulation environment or the systems trained using only synthetic images are tested in real-world applications [10,11,36]. As a result of the inconsistencies in two data groups, most of such robotic applications based on object recognition are not functioning at their highest possible performance. On the other hand, there exists some exceptional studies, which are trained and tested on the same type of data domain regarding real-world [12,21,22,24,28] and simulation environment [3,6,14].

Instant object recognition is a process of calling knowledge about object identities that are stored as prior information, which is previously mapped to consistent memory segments. In computer vision, the efforts behind clarifying the questions of where the object of interest is in the image or what exists in the whole frame in terms of localizing and recognizing have become obsolete so that the current situation implies further endeavor to extract meaningful information from data utilizing various approaches. The latest improvements in hardware, algorithms, and software make it possible for robots to acquire semantic relations and generate inferences by learning from data with deep neural networks. Achieving semantic intelligence enables the machines to answer the content, function, and location of the object. However, the object localization and class information by itself are not sufficient for robots to extract semantic knowledge and object-based relationships. For this reason, additional object attributes beyond class labels play an important role for semantic content extraction. Moreover, the successor object information among the main objects in the images prepares the framework for establishing the relationships between objects. Thus, the successor objects contribute to the acquirement of semantic knowledge as well as increasing the existent object recognition performance.

In this work, we introduce a hybrid image dataset to alleviate this problem increasing the accuracy rates and reliability of the object recognition algorithms. We propose ADORESet, which contains data from both real-world and simulation environment, and it helps to eliminate the inconsistency problems when the researcher goes from simulation environment to real world or vice versa for development and testing purposes. ADORESet has 2500 real and 750 synthetic images for each category of 30 classes, and all of them are colored images with the dimension of 300 × 300. Our experiments are composed of training and test sessions for only real images, only synthetic images, and lastly using hybrid images, separately by fine-tuning VGGNet [19], InceptionV3 [20], ResNet [8], and Xception [3] models. The performance results are compared in terms of accuracy, training and test periods, and model size. Moreover, all images of our dataset are properly labeled and the bounding boxes of the main objects for each class are manually specified. ADORESet images are ready to be used for supervised learning tasks such as object recognition and localization. The dataset objects are selected so that they can be commonly found on office desktops or indoor environments. The selection process also included the group of objects that are movable and can be the natural focus of interaction with humans in daily life.

This paper is organized as follows: First, the previous studies with the similar aim are reviewed and the motivation behind building the hybrid image dataset is given in Sect. 2. Then, in Sect. 3, the technical properties of ADORESet are explained in detail with preprocessing tools. Next, statistical analysis of the hybrid dataset and the semantic relation between different objects are given in Sect. 4. In Sect. 5, the testing of the dataset using most accepted deep CNN structures and relevant performance results are presented. Finally, in Sect. 6, the conclusions are drawn and the future work is presented.

ADORESet and additional information can be found at: http://adoreset.itu.edu.tr/.

2 Motivation and related work

Image datasets can be considered in two categories: labeled and unlabeled/raw, which are relevant for supervised learning (classification) and unsupervised learning (clustering) tasks, respectively. Furthermore, semi-supervised and reinforcement learning algorithms can be applied to both types of datasets. Additionally, much more effort is required to obtain labeled image datasets than unlabeled ones. Robotics research problems involving machine vision are generally carried out using real and simulation images, separately. The main motivation for ADORESet, as a hybrid image dataset containing both real and synthetic images, can be summarized as follows:

(i) A task-specific and context-specific image database is much needed in robotics community to obtain better object localization, recognition, and manipulation algorithms that can be trained for higher accuracy and real-time performance.
(ii) Most of the available databases have either real-world images or synthetic images. The robotics community needs a hybrid image database so that the trained algorithms can work reliably in both simulated and real-world environments and scenarios.
(iii) Most of the available databases are not well annotated, and simple preprocessing tools are not provided. There are often no semantic or probabilistic connection/relation maps provided for the images.

In this section, the motivation for proposing ADORESet and related preprocessing tools is justified by two separate literature reviews. First, to locate the hybrid image database ADORESet among already existing large image databases, a brief overview of similar databases is given in Sect. 2.1. Secondly, as the large image databases are almost always used for training deep CNN structures and their utility is tested using machine learning algorithms, another subsection is devoted to informing the reader on the state-of-the-art deep NN research in Sect. 2.2.

2.1 Overview of existing image datasets

The two important arguments, why deep neural networks have been skyrocketed in recent years, can be traced back to the development at hardware (especially GPUs) and various datasets which consist a huge amount of data. Consequently, new algorithms and applications have arisen which have revolutionized the ways that we evaluate the data. One of the most popular datasets in the last years, particularly in the field of deep neural networks, is ImageNet [30], which is related to a competition called ImageNet Large Scale Visual Recognition Challenge (ILSVRC) organized every year under the topics of object localization , object detection, object detection from video, scene classification, and scene segmentation. ImageNet is constructed according to WordNet [25] hierarchy, and the nouns in this word dataset are employed to label the objects. Even though ImageNet has many more images and categories, 1.2 million images and 1000 categories are used for the challenges as standard. Similar to ImageNet, another competition is run annually using Microsoft Common Objects in Context (MS COCO) [23] dataset including features such as object segmentation, recognition in the context, multiple objects per image by having more than 300,000 images, for 2 million instances, 80 object categories, and 5 captions per image. PASCAL Visual Object Classes (VOC) [7] is another dataset, which was held from 2005 to 2012 as a yearly challenge, assessing performance on object class recognition. Caltech 101 [8] and Caltech 256 [13] consist 101 and 256 classes, respectively, and each class includes various numbers of labeled images ranging about 40 to 800. CIFAR [18] is derived from 80 million tiny images dataset [35] by labeling 60,000 for 10 classes called CIFAR-10, and another 60,000 for 100 classes, which are composed of 5 classes under 20 superclasses called CIFAR-100. One of the biggest publicly available image datasets [35] contains approximately 80 million colored images with the dimension of 32 × 32 pixels with weak labels which are listed within WordNet hierarchy. Yale–CMU–Berkeley (YCB) [2] dataset presents 77 classes of objects relevant to robotic manipulation research. YCB contains 600 high-resolution colored images, 600 colored depth images, and five sets of textured three-dimensional geometric models with mass values of objects per category. ModelNet [37] consists of 151,128 3D computer-aided design (CAD) models belonging to 660 categories, which is created by downloading models from the web. In conjunction with being a purely synthetic dataset, each class of ModelNet

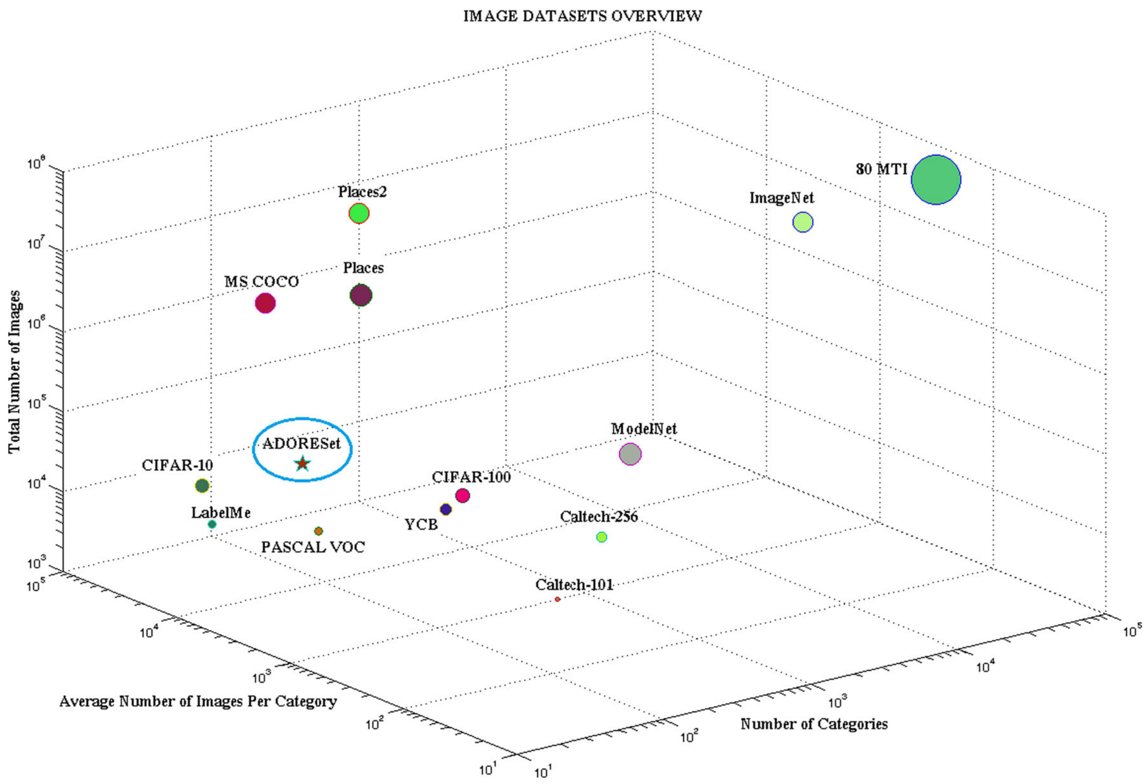

Fig. 1 Comparison of datasets in logarithmic scale according to total number of images, average number of images per class, and total number of classes

has numerous instances. Additionally, Places [43], Places2 [42], and LabelMe [31] datasets are created using outdoor images which contain labeled, weak labels logarithmically scaled 3D space. In Fig. 1, the comparisons of the given datasets are illustrated, where the vertical axis shows the total number of images and the horizontal axes display the average number of images per class and the total number of classes, respectively.

The content and statistical information about the existing image datasets and ADORESet are given in Table 1.

Moreover, [16] provides a synthetic image generator and introduces a pipeline to achieve better results than real-world data when using only synthetic images. However, [16] compares the results solely for vehicle detection tasks. Similarly, [29] contains only synthetic outdoor images, which are obtained from a virtual world with pixel-level labels. The results in [29] show that hybrid dataset approach also contributes to semantic segmentation of objects. With its hybrid and robust structure, ADORESet provides possibilities of transition and flexibility for real-world and simulation environment applications. As a consequence, having richly annotated 3250 images per category and containing an equal number of real (2500) and synthetic (750) images individually per category puts ADORESet one step forward among others.

2.2 Overview of current deep convolutional neural network models

Deep NNs for machine learning are not very different from the previous methods in terms of the necessary steps. These are: (i) gathering data, (ii) processing raw data in view of cleaning and putting into desired order, (iii) building the model by selecting the best algorithm after evaluation, and in the end (iv) transforming algorithm outputs into presentable results, as seen in Fig. 2. When a specific classification or recognition problem is defined, we collect raw data, preprocess, and then ameliorate existing algorithms by fine-tuning methods [27,39] depending on data to acquire plausible results. Our study differs with its labeled successor objects and the relationships between object classes in preparation and representation of the dataset, respectively. Antecedent deep learning methods and the related applications are explained in [20], which gives deeper insight mostly on the subject of CNNs considering object detection/recognition. Additionally, it gives brief information about recurrent neural networks (RNNs) and its usage areas mainly in text processing. [20] states that CNNs are more appropriate for image, video, speech, audio processing applications, and RNNs are for text and speech processing. AlexNet [19] is accepted as one of the milestones in deep learning applications in terms of

Table 1 General specifications of the image datasets

Name	# Total images	# Categories	# Images per category	Label type	Dimension
Caltech 101	8765	101	Various (40–800)	Labeled	300 × 200
Caltech 256	30,607	256	Various (At least 80)	Labeled	300 × 200
CIFAR-10	60,000	10	6000	Labeled	32 × 32
CIFAR-100	60,000	100	600	Labeled	32 × 32
ImageNet	14,197,122	21,841	Various	Labeled, unlabeled	Various
ILSVRC	~ 1,200,000	1000	Various	Labeled	Various
LabelMe	30,369	183	Various	Weakly labeled	Various
MS COCO	Detection (200,000)	80			
	Key point (250,000)	–	Various	Labeled	Various
	Segmentation (55,000)	91 + 1			
PASCAL VOC	11,530	20	Various	Labeled	Various
Places	2,448,873	205	Various (5000–30,000)	Categoric labels	Various
Places2	~ 8,000,000	365	Various (4000–40,000)	Categoric labels	Various
80 MTI	79,302,017	75,062	Various	Weakly labeled	32 × 32
ADORESet	Real (75,000)	30	2500	Densely labeled	300 × 300
	Synthetic (22,500)	30	750		

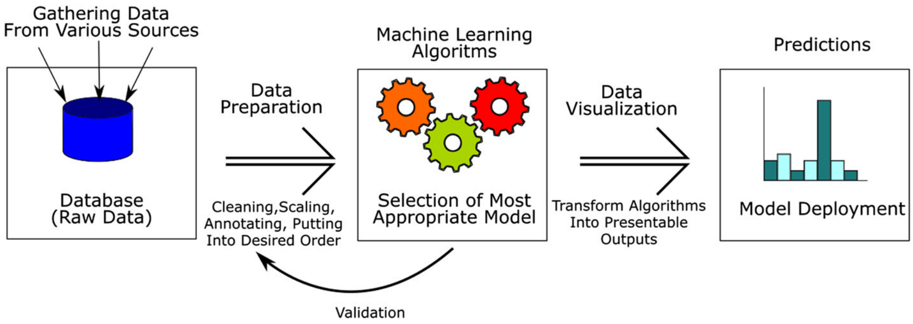

Fig. 2 The general process flow of machine learning systems

object detection/recognition/localization. The importance of this study comes from winning ILSVRC12 for the first time with a CNN architecture. Until then the winner algorithms are based on handcrafted (or hand-engineered) features. Dropout method has been introduced at the same study which proposes to prevent overfitting by randomly eliminating the units and their weights during training. AlexNet has 8 hidden layers, 5 of them are convolutional layers, and the rest are fully connected (FC) layers. ZFNet [40] is constructed based on AlexNet architecture which proposes a new technique to visualize the behaviors of the hidden layers in order to achieve a better understanding of CNNs. This study makes it possible to see how features act during training. This model helps us to have a better intuition about working principles of CNNs. ZFNet uses deconvolutional network (deconvNet) architecture to reconstruct the input image from feature activations to pixel space. They used ImageNet, Caltech 101, Caltech 256, and PASCAL VOC2012 for their experiments. One of the winners of ILSVRC14 is the team GoogleNet with the architecture called Inception [34] by having 12 times fewer parameters than AlexNet. The architecture submitted to ILSVRC14 is composed of 22 layers excluding 5 pooling layers. The aim of the Inception architecture is to obtain sparse structures from dense components of CNN features. This is achieved by concatenating the independent convolutional and/or pooling blocks. In ILSVRC14, they got the winner title by 6.67% error rate for top-5 predictions for classification task and for detection task the method had 43.9% mean average precision (mAP). Another winner of ILSVRC14 is VGGNet [33] as they claim because they realized their architecture gives better results than [34] after submitting it to the competition. They have 5 CNNs with different layer numbers from 11 to 19. Their intention is to investigate the effects of the depth to the improvement of the results in terms of accuracy. Therefore, they fix the parameters of the CNNs and the depth is increased by adding 3 × 3 convolutional filters. Once the improvement is achieved by small sized filters and strides, then this is densely trained and

Table 2 Performance results for VGGNet, ResNet, InceptionV3, and Xception

	Top-1 pred acc %	Top-5 pred acc %	mAP %	# of params
VGGNet	71.5	90.1	84.00	144.3M
ResNet	77.0	93.3	83.80	60.2M
InceptionV3	78.2	94.1	93.50	23.6M
Xception	79.0	94.5	93.22	22.9M

tested on the whole image. VGGNet also obtains good results on the other benchmarking datasets. ResNet [15] is the winner for both detection and localization tasks of ILSVRC15 and MS COCO with the deepest architecture comparing the previous CNNs with 152 layers but it has fewer parameters than VGGNet. Even though it is thought that the deeper the network the better the results, in this study the degradation problem is addressed to the depth of the network. ResNet solves degradation by shortcuts which perform identity mappings to some layers by adding them to the outputs of the stacked layers. As being an expansion of modified Inception [34] model called InceptionV3 (42-layered CNN), Xception [4] (48-layered CNN which is composed of 36 convolutional layers along with pooling and optional FC layers) architecture changes Inception modules with depthwise separable convolutions by having same number of parameters as Inception and taking over its performance slightly in ImageNet dataset. These state-of-the-art base models are mostly fine-tuned to detect and classify objects in particular tasks using smaller datasets. Further applications such as [44,45], which are fine-tuned by training classifiers on top of base models [34] and a combination of [19,33,34], respectively, to recognize objects using particular datasets, achieve successful accuracy rates higher than 90%. In Table 2, the performance results of [4,15,33,34] are presented that are achieved at ILSVRCs.

In this study, images are gathered from wild web and Gazebo simulation environment (GSE). After labeling all images, object recognition performance measures are presented as the outputs of models. Our main contributions are as follows;

(i) A new richly annotated hybrid dataset, ADORESet, is introduced, which consists of 97,500 colored images for 30 categories. It contains 75,000 real-life images and 22,500 synthetically generated simulation images. Real images are acquired from wild web by querying 7 image search engines with 390 words/word pairs.

(ii) ITUrk GUI (image annotation with bounding box specifying tool for large number of images) and synthetically generated images are provided.

(iii) Statistical analysis of the dataset and semantic relations between objects is given.

(iv) Performance results of CNN models on ADORESet including accuracy and loss values (i.e., negative log-likelihood and residual sum of squares for classification and regression, respectively), time per epoch for combinations of real images and synthetic images in terms of being training and testing images are evaluated, which reveal the importance of hybrid dataset.

3 ADORESet

Even if the emphasis in machine learning field is often toward algorithm development, the quality of data has a great influence on resulting models and their performance. The factors affecting the quality of the datasets can be related to the quantity, labeling procedures, missing samples, variations, noise, outliers, invalid instances. Therefore, it is important to form datasets that have the minimum number of such problems. As an answer to this quest, densely annotated ADORESet provides a satisfactory number of images for each class for machine vision-based problems in robotics such as object detection, recognition, localization, tracking, and manipulation. This dataset contains real and synthetic images maintaining flexibility in terms of developing models for both real world and simulations. This enables, in turn, the fast and direct deployment of algorithms developed in simulation to the real-world experiments. ADORESet should be of interest to the field of robotics researchers by means of its hybrid form and its suitability to robotics applications such as detection, recognition, localization, grasping, and dexterous manipulation of objects. To construct ADORESet, we start by downloading instances obtained using image search engines. Afterward, an adequate number of images of relevant classes are generated within the simulation environment. The annotated and resized data obtained from both sources are processed using ITUrk graphical user interface (GUI). The successor objects are also labeled to retrieve statistical information about the probabilistic relations between the objects in terms of coexistence in the same context within the dataset. For example, the relation between monitor, keyboard, and mouse can be directly inferred using this information. Figure 3 presents the flowchart of the construction process for the ADORESet.

3.1 Gathering images from wild web and preprocessing

The object categories in ADORESet, which are given in Table 3, are specified considering the robotics applications. It is unquestionable that these objects have been part of everyday life in the last three decades. With the ambition

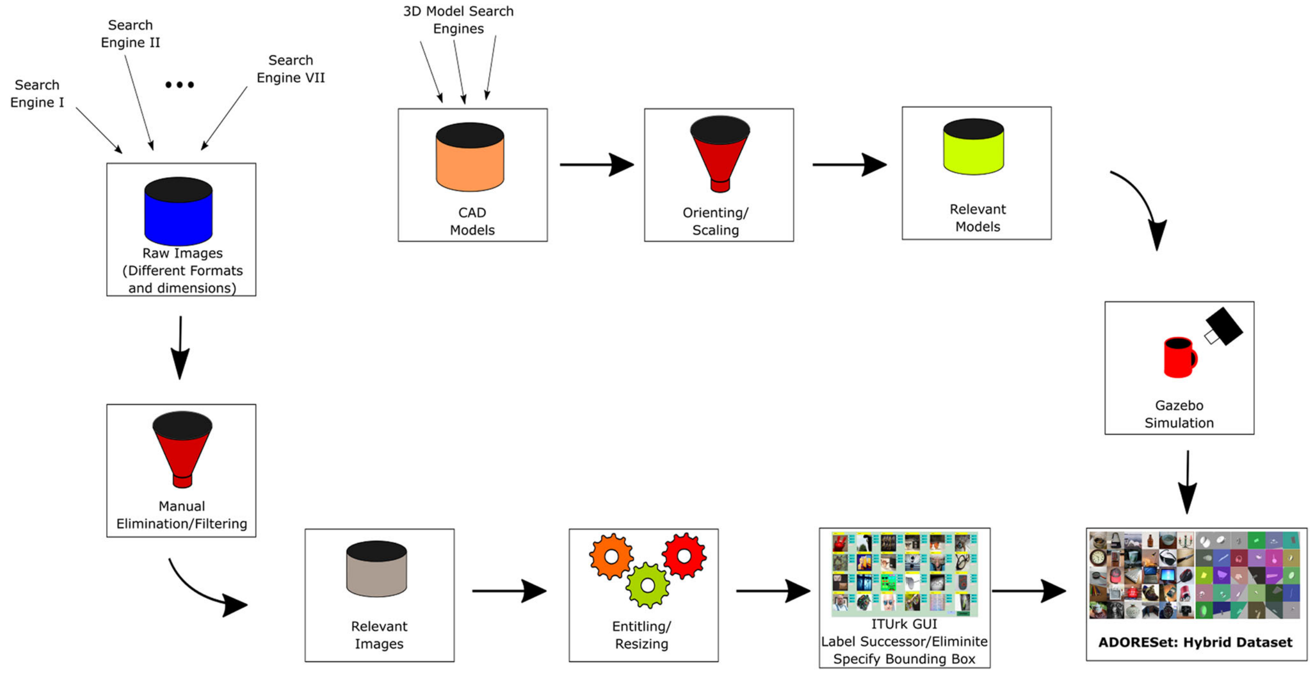

Fig. 3 ADORESet construction pipeline

Table 3 Object categories of ADORESet

1 Ashtray	2 Bag	3 Book	4 Bottle	5 Bowl
6 Can	7 Candlestick	8 Clock	9 CookingPot	10 Cup
11 DeskLamp	12 Eyeglass	13 ForkSpoonKnife	14 FryingPan	15 HeadWear
16 Keyboard	17 Laptop	18 Monitor	19 Mouse	20 Pen(cil)
21 PhotoFrame	22 Shoe	23 SmartPhone	24 Speaker	25 Teapot
26 Telephone	27 Vase	28 Wallet	29 WebCam	30 WristWatch

of building this dataset using the wild web, we utilized about 390 query word(s) or word pairs via seven image search engines. Principally, the multi-language wild web search is performed according to the brand, gender, model, type, color, age, season, material, state, and relation. In the next step, inappropriate raw images are eliminated manually with regard to the parameters such as the light effects and conditions, noise, distance and angle, visibility which determine the dataset quality. Then, the rest of the images are labeled with the following rule: *The first three numbers indicate the category starting with* 0, *and the last five digits display the index number of the image in that category starting with* 0, *e.g.,* 01700754 *is the* 754. *image of the laptop class.* Then, all images are resized to the same dimensions. As a result, ADORESet is a new richly labeled dataset consisting of 75,000 colored real images with the dimension of 300 × 300 pixels for 30 classes including the bounding box coordinates of all objects. Real images that hold approximately 1.3 gigabytes in the hard drive are stored in JPEG compression format.

3.2 Image generation from simulation world

Similar to the process of gathering the real images, image generation from simulation world starts with downloading computer-aided design (CAD) models of the objects from the wild web. For each object class, five different CAD models are downloaded and their file formats are converted to STL which is also appropriate to use together with universal robot description files (URDF). Since they are acquired from various sources, their orientation, scale, and origins are not properly defined. Initially, every model has oriented in a way that normal vector of the meaningful side of the object is parallel with the z-axis. Next, the objects are scaled to their real-world dimensions. Lastly, the origins are relocated to bottom centers of the CAD models. The textures are not attached to the models, and the colors are allowed to change with the color of the simulation world light source. After this compilation, ADORESet includes 750 synthetically generated images per category having the same properties as real images. There are two important variables in the simulation

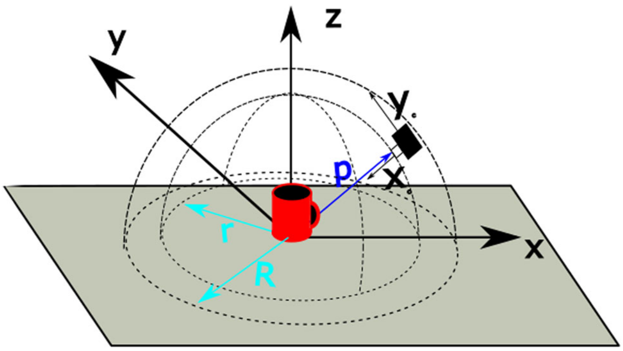

Fig. 4 Schematic view of simulation environment with frames and variable definitions

world which affects variations and the quality of the images, light color and 6D pose of the camera. In GSE, the light is adjusted with a light source model. Thirty images captured for each light source–object couple. After completing the image acquisition, old light source model is deleted and a new one with random color values is created. The second factor, 6D camera pose, consists of three position and three orientation variables. It is assumed that two virtual half spheres are created around the object with radius of r and R and the camera is located between their surfaces. Therefore, the distance between the camera and the object is similar for each object class depending on its average dimensions. For instance, the minimum distance (r) between the camera and the object is set to 0.2 m for wristwatch, while it is 0.4 m for bowls. The environment with half sphere is drawn schematically in Fig. 4.

To calculate a random point on the half sphere surface, a random unit vector **s** is defined as given in Eq. 1 where rand denotes the random function between given argument values. It is worth to note that z vector is restricted for positive numbers which restrains the position of the camera on the upper half of the sphere.

$$\mathbf{s} = [\mathrm{rand}(-1, 1), \mathrm{rand}(-1, 1), \mathrm{rand}(0, 1)]^{\mathrm{T}} \tag{1}$$

The position vector of the camera **p** can now be easily calculated by using known values of r and **s** as in Eq. 2. The constant c_1 defines the maximum distance between the object and the camera R.

$$\mathbf{p} = (c_1 \mathrm{rand}(-1, 1) + r) \cdot \mathbf{s} \tag{2}$$

The opposite direction of the position vector defines the pointing direction of the camera orientation $\mathbf{x_c}$. To use the vector in frame definition, normalization is applied as in Eq. 1.

$$\mathbf{x_c} = -\frac{\mathbf{p}}{\|\mathbf{p}\|} \tag{3}$$

Because the calculated $\mathbf{x_c}$ vector guarantees that the object is on the image plane, other orientation vectors can be selected as any arbitrary vectors meeting orthonormal condition. So $\mathbf{y_c}$ is calculated ensuring the dot product with $\mathbf{x_c}$ results zero as in following equation. Three components of the $\mathbf{x_c}$ vector are denoted with x_{cx}, x_{cy} and x_{cz}.

$$\mathbf{y_c} = [c_2 x_{cy} + c_3 x_{cz}, -c_2 x_{cx}, -c_3 x_{cx}]^T \tag{4}$$

Last vector to form the orientation or rotation matrix is $\mathbf{z_c}$. It has to be a perpendicular vector to the other two and is calculated as given in Eq. 5.

$$\mathbf{z_c} = \mathbf{x_c} \times \mathbf{y_c} \tag{5}$$

Random light source spawning and 6D pose generation are implemented in a ROS node. Every image is acquired from a unique 6D pose. The light source is changed for every 30 images because of the low speed of light source deleting and spawning. Five different CAD models are used for each object which gives total 750 image for each object class. Example pictures of every class are shown in Fig. 5.

3.3 ITUrk GUI

Although the wild web supplies an excessive amount of data, it may cause problems when it is used with deep learning algorithms directly due to lack of quality. In fact, many images tagged with inconsistent keywords or indistinguishably tiny sized objects exist within images. To overcome these obstacles, in most situations, crowdsourcing tools are employed to label the data. There are such mechanisms that are produced for a more general social experimental task which are also known for annotating data called Amazon Mechanical Turk (AMT) [1]. Furthermore, the aforesaid software can be arranged to collect a more wide range of information than only labeling, so that the gathered information can be extended to have a knowledge of the position of the tagged object in the image plane and specify successor objects. In this work, a simple GUI is designed and implemented to obtain annotation of the data samples, the bounding box position, and the successor object category.

The GUI is designed to have 24 images on a page to increase the processing speed while keeping them visible enough for the user. Each object class is loaded to GUI first. Then, the user is asked to delete irrelevant images about the object class by selecting them on delete buttons over the images. At the same time, user clicks on the related object name if a successor object exist. Three most expected successor names are readily given as the buttons. However, the user can add more related items by writing the name of it to the text box placed under the given successor names. After completing the elimination and labeling successor objects,

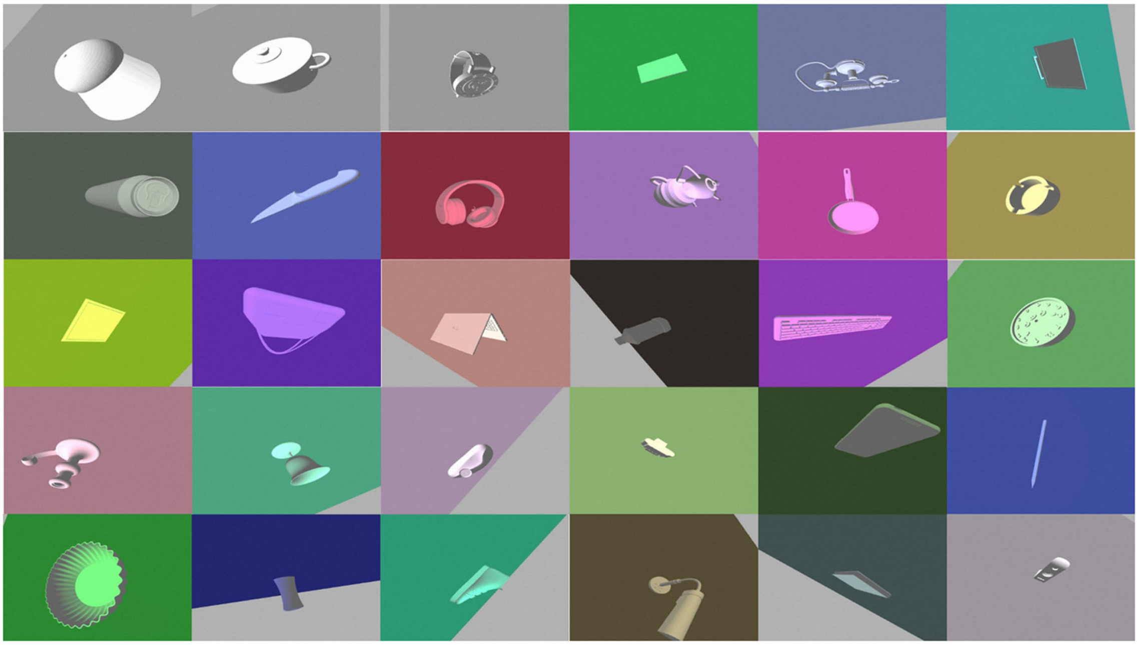

Fig. 5 Example images for all object categories generated in GSE

the continue button starts the bounding box selection process. The user selects the left/top uppermost bounding point with the mouse left click. Similarly, the right/bottom uppermost bounding point is chosen with the right mouse click and it finishes the bounding box selection for the active image. The active images are marked with red delete buttons. When the bounding box selection of an image is finished, the next undeleted image becomes active. Finally, completing bounding box selection starts a new page with new 24 images. The GUI is implemented in MATLAB. The screenshot of the GUI is given in Fig. 6.

In total, 75000 real images belonging to 30 object classes are filtered through ITUrk as convenient images for deep learning algorithms. Images are resized to a dimension of 300 × 300 pixels which is same with the images from simulation world. The user can process 24 images in one page within two minutes. First 40 s is spent in annotating and successor labeling part and remaining time is spent for bounding box selection. Moreover, perspectives and cylindrical objects may reduce the speed of process and cause the failure of the human bounding box specifiers. Example images from each of the object classes are shown in Fig. 7.

3.4 Distinctive properties of ADORESet

The underlying philosophy behind the machine learning systems requires having a dataset which has as many variations as possible and then to build intuition using supervised, unsupervised, or reinforcement learning algorithms from the data. As an applied field of such learning systems, the robotics for non-industrial daily use and humanoid robots are increasing in the last years. Both real and simulation world trials of such robotic systems give successful results in perception, recognition, gripping, grasping, moving, and manipulating of the objects. To make these systems more intelligent and robust, the training data must be compatible with the environments where the test sessions will be conducted. Accordingly, taking these requirements into account, ADORESet is composed of hybrid images for 30 object classes, which may exist mostly on desktops and indoor environments. Following the labeling and elimination operations, some images are exposed to distortions because of resizing that provided extra variations for the dataset which is one of the desired properties as long as the deep CNNs are not robust to scale and rotation invariance. Because there is enough number of images per category, each class of ADORESet is also convenient for sub-category classification. Unlike the datasets mentioned before, which consist of single and centered objects per image, ADORESet contains complicated images including multiple objects, which makes it a more challenging dataset, besides comprising different forms of objects that have been transformed in decades. In addition, our dataset includes a sufficient number of centered and salient images that can be easily separable from the background. Moreover, ADORESet is richer than the existing datasets because it provides information about the probabilistic relations between different objects in couples. Thus, the relation information between objects enables the machine vision systems to construct a further perception than only recognizing or localizing objects in the scene. This type of information could be particularly useful in semantic recognition.

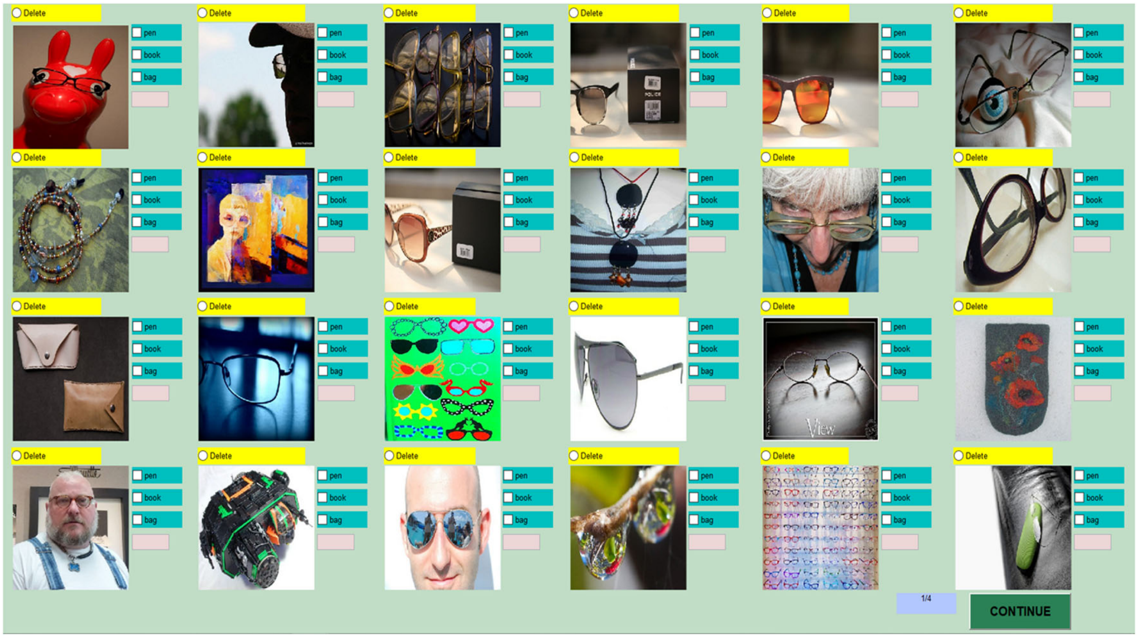

Fig. 6 ITUrk GUI with the images from eyeglass category

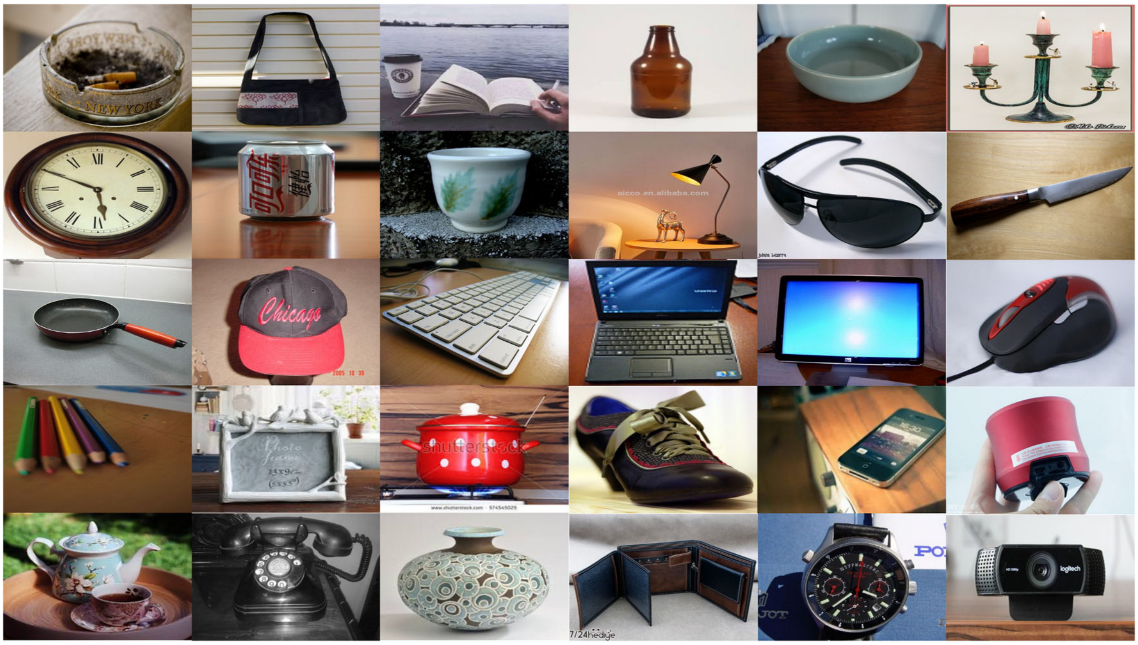

Fig. 7 Resized and labeled wild web images with instances from all categories

4 Statistical analysis of ADORESet and semantic relation between objects

The object classes, which are included in the ADORESet, are chosen from commonly used items in everyday life and mostly located around or on desktops. In addition to this, the objects are related with each other depending on their usage area, appearance similarity, and typical locations. Some of them are used for similar or completely same purposes. For instance, an old dial-based *telephone* and a *smartphone* are used for communication objectives, and a *pot* is used for cooking like a *pan*. Additionally, some tasks include multiple objects which completes each other, such as *mouse–keyboard*, *cup–teapot*, *cup–bottle*. Besides, the physical appearance is another important issue, and for some object class couples, it is occasionally indistinguishable as in the

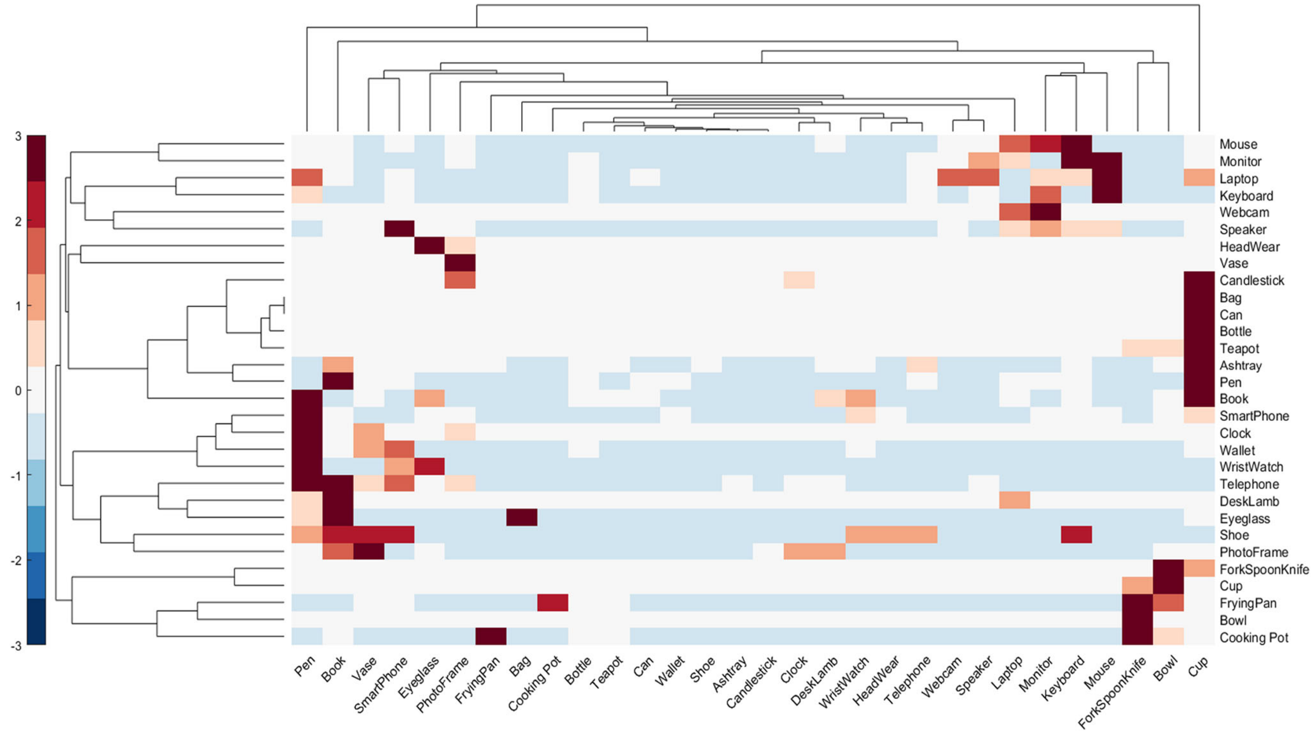

Fig. 8 Relations between object categories (darker color means more relationship between objects)

case of *bowl–vase* and *pan–pot*. Furthermore, specific items are generally placed close to each other. For example, it is strongly probable that a *fork* may be seen near to a *bowl* or a *cup* in the dining table context. It is worth to consider that the object classes consist of not only one object but also multiple very similar objects. For example, the cutlery item object class has *Fork/Spoon/Knife*, which aggregates three eating utensils. The successor objects are detected in randomly collected images from the wild web to identify the semantic relations between them. It may provide useful information to researchers from the robotics field particularly in semantic recognition and manipulation planning. To present the information, existence frequencies of successors for each object class are illustrated as a color matrix in Fig. 8.

The main object classes are given in row entities, and their successor images are given in columns in Fig. 8. Since the object is not a successor for itself, the appearance frequency is assumed to be zero. The columns and rows are arranged in an order so that the mostly related objects are closely aligned. All values are standardized along the rows to emphasize the relations. Using this standardization, relation scores of the objects are colored according to colorbar given on the left side of Fig. 8. Thus, for example, the *bowl* is the most frequent object in the *cup* images. On the other hand, it is worth to notice that the graph is not necessarily symmetrical. Therefore, the *cup* is not the most existent object for the *bowl* class. Using this type of information as semantic cues, a robot can interpret that if a *cup* is in the scene probably a *bowl* can be seen, probably a *bowl* can be seen; however, if a *bowl* is seen in an image, it cannot be said that a *cup* is in the area.

The statistical analysis helps to represent the relation between the object classes in numbers. Robots empowered with vision make use of this much required information to enhance the intuitive capabilities of object search, having an artificial anticipation function. In addition, it can contribute toward the accuracy of object detection under the influence of poor lighting or occlusion. The vision algorithms may estimate where to look for a certain object in a large operation space. An occluded object can be identified more precisely with the assist of detected successor objects. The analysis facilitates manipulation and planning tasks by the means of clustering similar objects as well. The statistical results can be also employed as a guide for the robot to place the complementary items together in a meaningful way.

5 Performance evaluation of CNNs

The way for detecting and recognizing objects in deep neural networks is through training for many times with a sufficient amount of data until reaching the redefined performance criteria. In this section, to reveal the benefits of the hybrid dataset on object recognition task, the performance results of all possible combinations of real and synthetic images

Table 4 Data configurations for experiments using ADORESet including data types and number of images

Type of training data	# Images	Type of validation data	# Images
Real images	1775	Real images	725
Real images	2000	Real + synthetic images	500 + 500
Real images	1500	Synthetic images	750
Synthetic images	750	Real images	375
Synthetic images	600	Real + synthetic images	150 + 150
Synthetic images	500	Synthetic images	250
Real + synthetic images	750 + 750	Real images	750
Real + synthetic images	1775 + 500	Real + synthetic images	925 + 250
Real + synthetic images	375 + 375	Synthetic images	375

Table 5 Performance results if training data consist of only real images

Model	Data type and amount		Train acc (%)	Val. acc (%)	Time per epoch (s)	Batch size
	Train	Validation				
VGGNet	R 1775	R 725	84.82	80.44	435.40	32
	R 2000	R 500 + S 500	98.32	50.86	660.66	32
	R 1500	S 750	98.23	9.30	483.88	32
InceptionV3	R 1775	R 725	96.81	86.54	1634.7	32
	R 2000	R 500 + S 500	97.17	50.97	2657.4	32
	R 1500	S 750	98.23	10.77	872.16	32
ResNet	R 1750	R 750	97.00	86.01	472.9	32
	R 2000	R 500 + S 500	97.72	49.54	1094.2	32
	R 1500	S 750	97.85	7.87	415.02	32
Xception	R 1775	R 725	97.44	85.64	1706.50	32
	R 2000	R 500 + S 500	97.67	49.46	2667.60	16
	R 1500	S 750	97.61	7.04	1974.90	32

R stands for real images, and S stands for simulation images. The numbers near R and S denote the number of images

as being training and validation data are given. These combinations with regard to the types of data for training and validation with the number of images are given in Table 4. Hence, 36 performance results are obtained for nine types of data and four deep CNN methods in terms of time, accuracy and loss values. The number of frozen layers, which are kept same with the weight values of base models, of deep CNNs [4,15,33,34] is varied depending on the number of data. The number of epochs is fixed to 50, which ensures the convergence of performance measures to stable values. Rectified linear unit (ReLU) function is chosen as the activation function for all configurations. Stochastic gradient descent [5] is used as optimization method while fine-tuning [33], and Adam [17] is used for the rest of the architectures. To calculate the probability of the output in the classification layer, *softmax* regression is applied to all models. The batch size is varied with respect to the memory capacity of the system running on 64-bit Ubuntu 14.04 equipped with an NVIDIA GTX 1080 GPU, an Intel i7 CPU 920@2.67GHz × 8, 6GB RAM, and 1TB hard drive spins at 7200RPM.

5.1 Experiments with real-world images as training data

The first three experiments are performed using only real images as training data and combinations of real and synthetic images as validation set. The performance results are given in Table 5. In addition to general performance of the recognition experiment, the progress of accuracy and loss values throughout 50 epochs of training and validation is given in Fig. 9. As can be seen from both Table 5 and Fig. 9, the highest validation accuracy rates are achieved when the real images are used for the training and validation. InceptionV3 is slightly better regarding the validation accuracy than other models, while VGGNet is trained in the shortest time. The batch size of all configurations is set to 32, except the case that the real and synthetic images are used for validation by Xception model because of the memory issue, which is handled by setting the batch size to 16 for this configuration. The training accuracy values for all methods in all data pair cases give acceptable results at around 95%, but not in

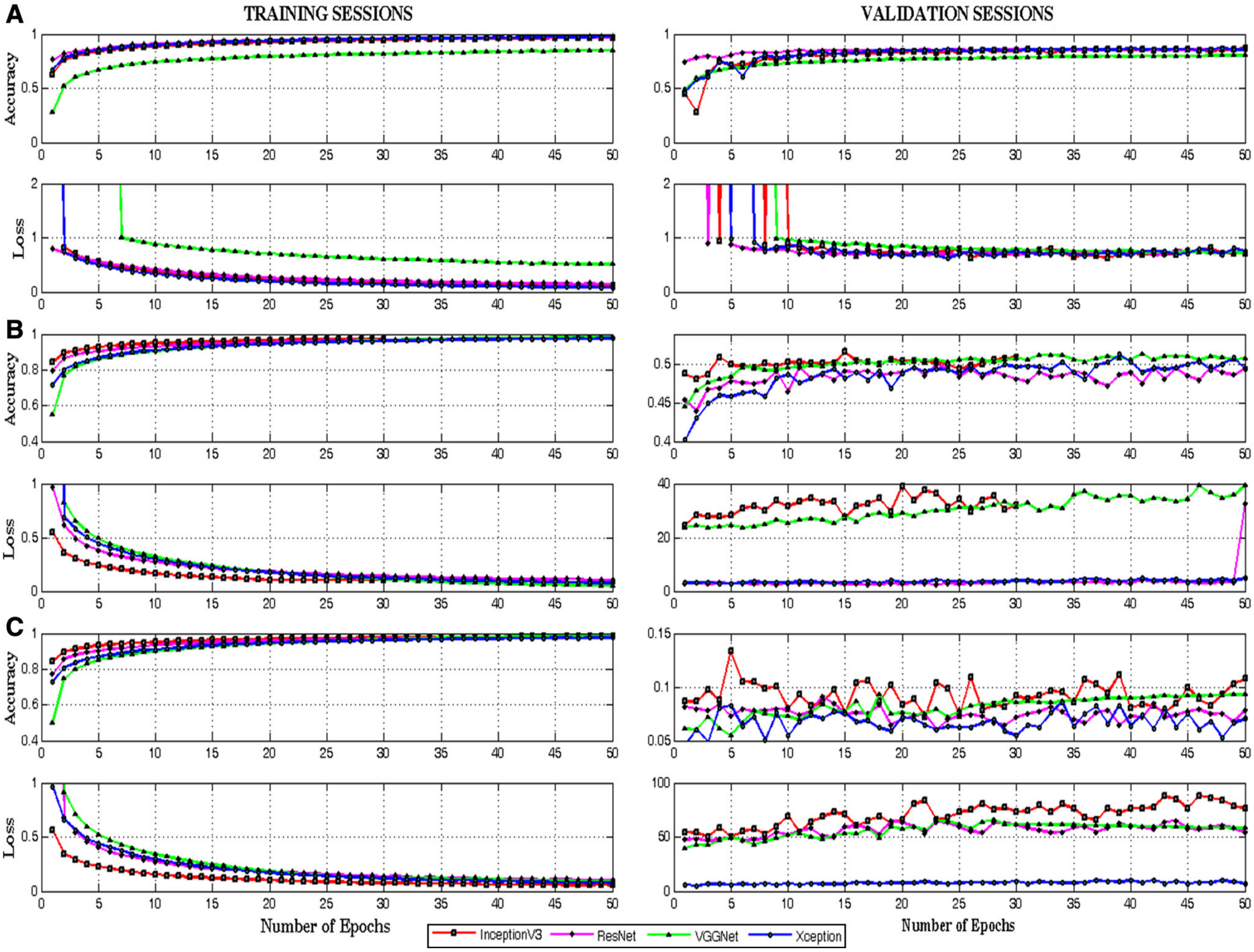

Fig. 9 Progress of performance parameters during training and validation sessions. Training data are composed of only real images. **a** Real images for validation, **b** real and simulation images for validation, **c** simulation images for validation

the validation accuracy values. It can be observed that the similar training and validation data types result in high accuracy rates for all models, as seen in Fig. 9a. Nevertheless, the usage of incompatible data pairs yields unsatisfactory validation accuracy values. A poor performance using mixed type of data as validation set is presented in Fig. 9b. When the training data consist of only real images, but the validation set has mixed data type, the recognition rate is approximately 50%. Moreover, the worst case is observed when the training images were completely from real world and the validation set was drawn from purely synthetic images. The validation accuracy rate of all models fluctuates around 10% in the worst case, as seen in Fig. 9c.

5.2 Experiments with synthetic images as training data

In this set of experiments, only the synthetic images generated in GSE are fed into the networks as training data while the validation data are varied as real-world, hybrid, and synthetic images. The resulting performance parameters are displayed in Table 6. The progress during the training and validation sessions is given in Fig. 10. Similar to the previous results, the selected data types for training and validation greatly affect the performance metrics. The batch size values for all cases are set to 32. The validation accuracy values for the case of having the same data types in training and validation sessions are the highest throughout all cases. The decrease at validation accuracy rates is distinct when the real images are fed into the model as validation data. One might easily say that the data type incompatibility is explicit in the resulting low accuracy rates, when the data type configuration is set to utilize synthetic images as training data and real images as validation, as seen in Table 6 and Fig. 10. In other words, variations in the synthetically generated images were not adequate to resemble the variations available in the real images; therefore, the validation results in poor accuracy values. The deep learning algorithms were not able to cope

Table 6 Performance results if training data consist of only simulation images

Model	Data type and amount		Train acc (%)	Val. acc (%)	Time per epoch (s)	Batch size
	Train	Validation				
VGGNet	S 750	R 375	98.87	5.01	185.92	32
	S 600	R 150 + S 150	98.11	53.80	175.56	32
	S 500	S 250	97.03	95.92	164.44	32
InceptionV3	S 750	R 375	89.93	5.71	492.53	32
	S 600	R 150 + S 150	97.61	51.63	484.94	32
	S 500	S 250	98.78	97.58	475.64	32
ResNet	S 750	R 375	93.49	7.85	299.37	32
	S 600	R 150 + S 150	96.13	49.41	284.41	32
	S 500	S 250	98.49	95.53	275.82	32
Xception	S 750	R 375	97.37	5.05	745.85	32
	S 600	R 150 + S 150	96.91	47.00	687.17	32
	S 500	S 250	97.67	95.27	666.00	32

R stands for real images, and S stands for simulation images. The numbers near R and S denote the number of images

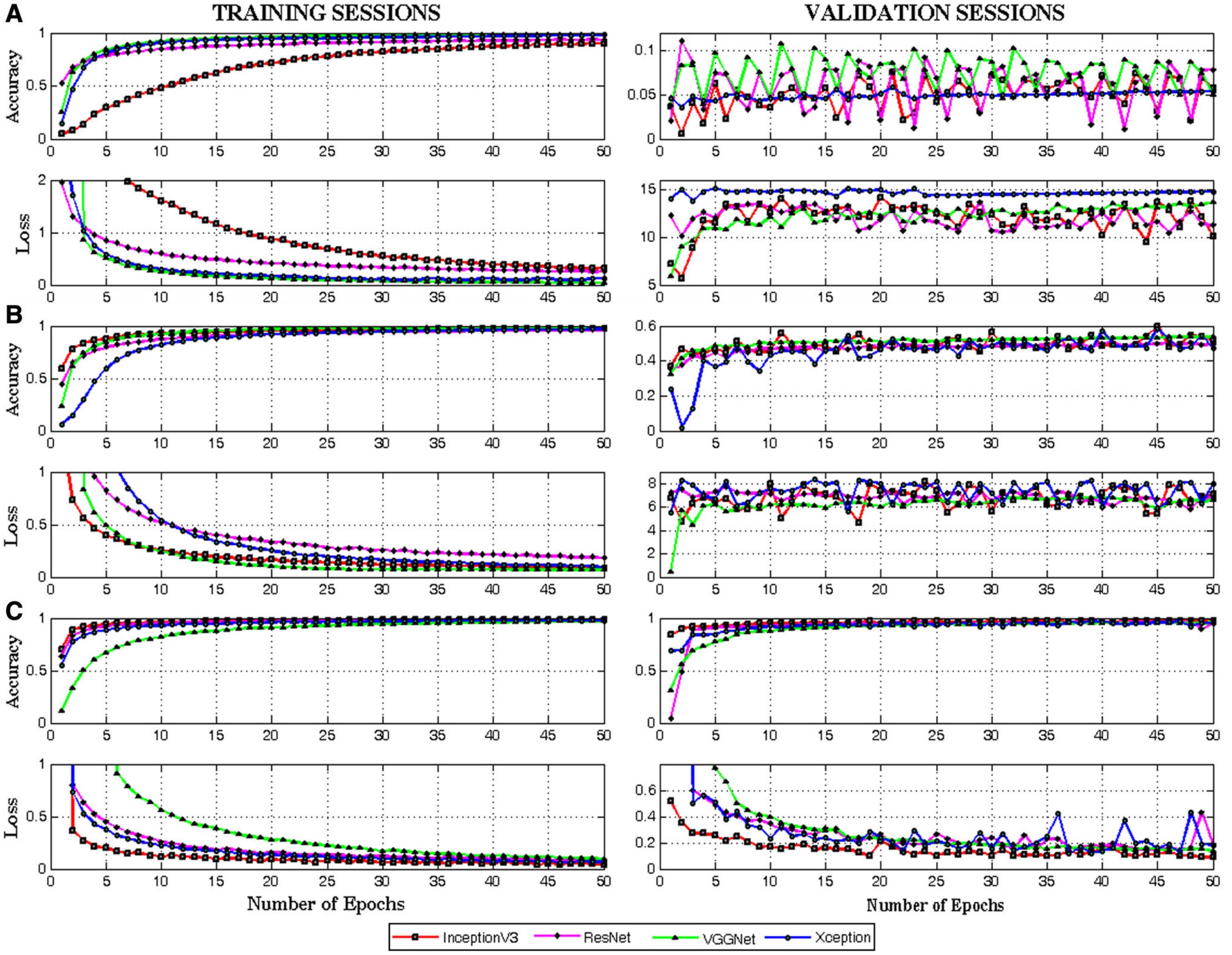

Fig. 10 Progress of performance parameters during training and validation sessions. Training data are composed of only real images. **a** Real images for validation, **b** real and simulation images for validation, **c** simulation images for validation

Table 7 Performance results if training data consist of both real and simulation images

Model	Data type and amount		Train acc (%)	Val. acc (%)	Time per epoch (s)	Batch size
	Train	Validation				
VGGNet	R 750 + S 750	R 750	95.49	85.37	427.96	32
	R 1775 + S 500	R 925 + S 250	98.08	90.50	717.18	32
	R 375 + S 375	S 375	96.48	93.06	194.2	32
InceptionV3	R 750 + S 750	R 750	96.54	86.03	495.85	32
	R 1775 + S 500	R 925 + S 250	98.15	89.97	1685.51	32
	R 375 + S 375	S 375	95.76	93.54	432.26	32
ResNet	R 750 + S 750	R 750	95.88	86.70	427.64	32
	R 1775 + S 500	R 925 + S 250	97.02	87.54	609.44	32
	R 375 + S 375	S 375	95.05	91.60	212.72	32
Xception	R 750 + S 750	R 750	99.54	90.41	497.44	16
	R 1775 + S 500	R 925 + S 250	97.74	89.00	2408.61	16
	R 375 + S 375	S 375	98.01	96.27	645.00	32

R stands for real images, and S stands for simulation images. The numbers near R and S denote the number of images

with the variations in the real images because they were not adequately trained to infer the intrinsic information.

5.3 Experiments with hybrid images as training data

In the experiments so far, only one type of images is used as the training data that was either real or synthetic. In this experiment, various numbers of hybrid data depending on the validation data type are fed into the models as the training data. Additionally, the available total number of images for training and validation images is the highest in this experiment configuration. As a result of larger data size, the time spent during the training and validation operations is the highest as can be seen from Table 7. All fine-tuned models succeed in outperforming the results of the base models by using both real and synthetic images as shown in Fig. 11. The batch size for all models is adjusted to 32 other than the cases of real and real-synthetic images as validation data combinations for Xception model, which are fixed to 16. Thus, the memory requirement of Xception is higher than other models that depends on the number of layers updated during fine-tuning and the natural structure of model itself. The performance evaluations show that the hybrid format of ADORESet is able to give highest validation accuracies independent of the validation data type selected.

6 Conclusion and future work

Object detection and recognition for robotics research in the context of dexterous manipulation, grasping, tracking are still challenging research topics. Even though the classical computer vision approaches provided some progress, the deep learning-based methods usually outperform them supported by the recent hardware developments and available large dataset. In current technology, it has become feasible to run deep learning algorithms within acceptable time spans and use the resulting net in real-time recognition tasks.

As an important part of the development in deep learning algorithms, the datasets have become the focus and enabler of the relevant robotic research involving object recognition, localization, and segmentation. The quality and the properties of such datasets determine how successful the learning algorithms can be trained to operate in implementations. Whether labeled or unlabeled, several image datasets with millions of images for thousands of categories exist. However, not all of them consider the parameters defining their quality such as number of images per category, image types and formats, object classes, dimensions. From this point of view, ADORESet considers these parameters and provides a dependable data source for computer vision and robotics community. Because of its hybrid structure, it allows researchers to implement their algorithms in both real-world and simulation environment conditions, enabling the transitions in between. The auxiliary tools provided with ADORESet contain ITUrk GUI and make it possible to label, eliminate, and resize the large number of images. Furthermore, the relationships between object categories are identified with the annotations of the successor objects. Thus, giving this type of semantic information between object categories depending on their existence puts ADORESet one step ahead among other image datasets that only give images and annotations. To the best of our knowledge, our study provides one of the most comprehensive detailed experimental performance results for state-of-the-art CNNs, besides a new densely labeled hybrid dataset. Despite the fact that the

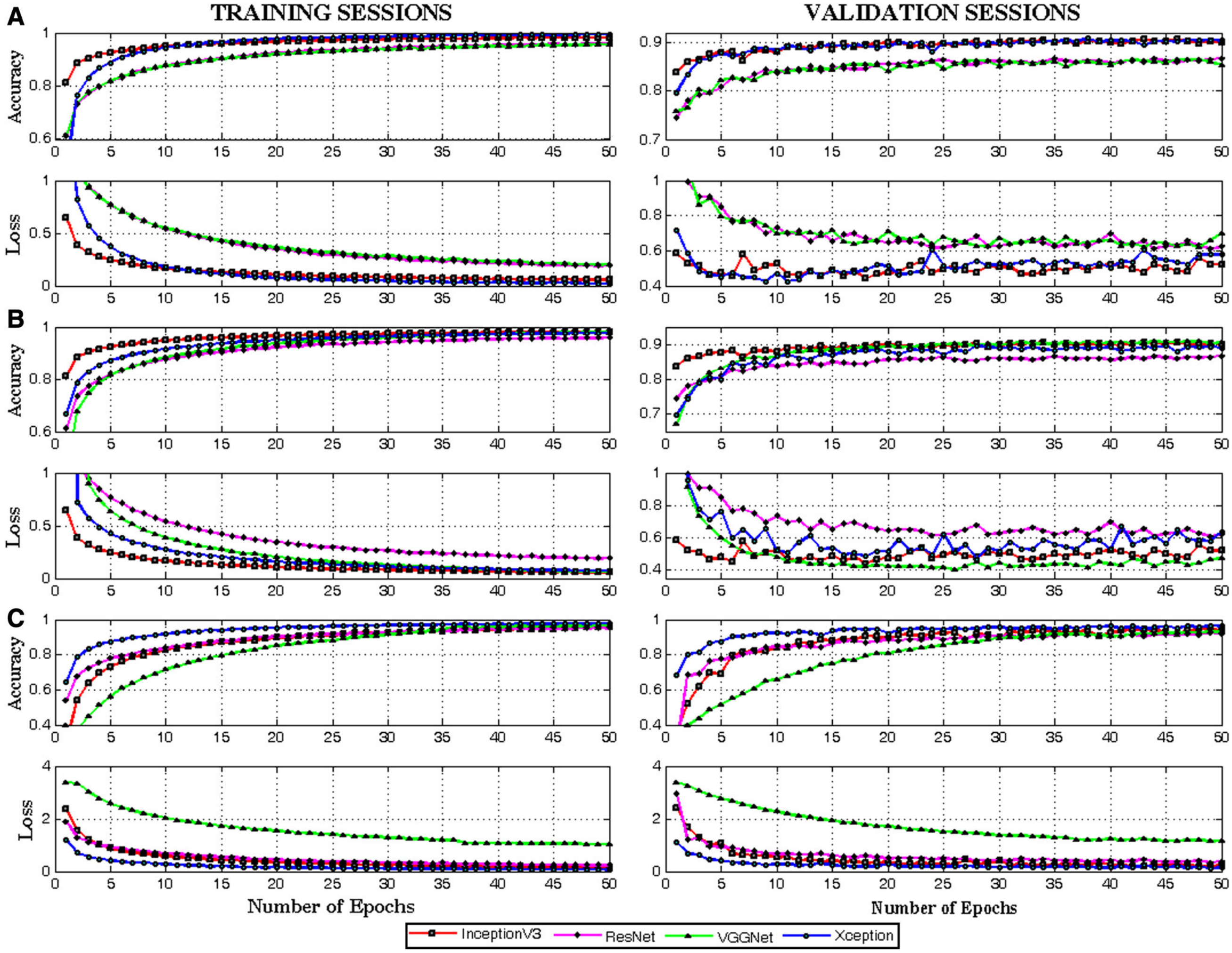

Fig. 11 Progress of performance parameters during training and validation sessions. Training data are composed of only real images. **a** Real images for validation, **b** real and simulation images for validation, **c** simulation images for validation

incompatible data pairs result in deep CNN weights that cannot be further used, the performance results clearly reveal that usage of real and synthetic images together as training data gives satisfactory validation accuracy rates independent of the selected validation data. It has to be emphasized that our reproducible results indicate the significant power of training–validation data types. We carefully divided the whole data into training and test sets for satisfactory results to avoid overfitting. (Approximately 67% of the images are employed for training, and 33% of the images are used for cross-validation.) Since all the models are trained using dropout and are tested with the sufficient number of images (ADORESet consists of more labeled images per category than most of the existing relevant datasets as explained earlier in this study and all of our experiments are conducted with enough data compared to the similar studies), our results are not due to overfitting. Furthermore, the progresses of accuracy and loss values during training and validation sessions for all scenarios illustrate the prevention of overfitting. On the other hand, the unsuccessful results are due to underfitting as expected because of the inconsistency between training and testing images.

In essence, once a CNN model is obtained using a hybrid dataset such as ADORESet, it can be applied to real and simulation images together or separately. ADORESet is suitable for developing novel algorithms, which can be CNNs or classical methods, intended to detect and/or recognize objects. Moreover, combining fine-tuned object recognition CNN models with additional inputs such as tactile information and depth of the object may allow development of better grasping and manipulation in robots. As future work, the real-time robotics experiments will be conducted using this object recognition algorithms in real implementation on a robotic arm.

References

1. Buhrmester, M., Kwang, T., Gosling, S.D.: Amazon's mechanical turk: a new source of inexpensive, yet high-quality, data? Perspect. Psychol. Sci. **6**(1), 3–5 (2011)
2. Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.M.: The ycb object and model set: towards common benchmarks for manipulation research. In: IEEE International Conference on Advanced Robotics, pp. 510–517. IEEE (2015)
3. Carlucci, F.M., Russo, P., Caputo, B.: A deep representation for depth images from synthetic data. In: IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 1362–1369. IEEE (2017)
4. Chollet, F.: Xception: deep learning with depthwise separable convolutions. arXiv preprint arXiv:1610.02357 (2016)
5. Chung, K.L.: On a stochastic approximation method. Ann. Math. Stat. **25**(3), 463–483 (1954)
6. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2758–2766 (2015)
7. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
8. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. Comput. Vis. Image Underst. **106**(1), 59–70 (2007)
9. Fischer, P., Dosovitskiy, A., Brox, T.: Descriptor matching with convolutional neural networks: a comparison to sift. arXiv preprint arXiv:1405.5769 (2014)
10. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4340–4349 (2016)
11. Georgakis, G., Mousavian, A., Berg, A.C., Kosecka, J.: Synthesizing training data for object detection in indoor scenes. arXiv preprint arXiv:1702.07836 (2017)
12. Giusti, A., Guzzi, J., Cireşan, D.C., He, F.L., Rodríguez, J.P., Fontana, F., Faessler, M., Forster, C., Schmidhuber, J., Di Caro, G., et al.: A machine learning approach to visual perception of forest trails for mobile robots. IEEE Robot. Autom. Lett. **1**(2), 661–667 (2016)
13. Griffin, G., Holub, A., Perona, P.: Caltech-256 Object Category Dataset. Technical Report 7694, California Institute of Technology, Pasadena (2007)
14. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2315–2324 (2016)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
16. Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Rosaen, K., Vasudevan, R.: Driving in the matrix: can virtual worlds replace human-generated annotations for real world tasks? In: IEEE International Conference on Robotics and Automation, pp. 1–8. IEEE (2017)
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR) (2014)
18. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images, vol. 1, No. 4. Technical report, University of Toronto, p. 7 (2009)
19. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
20. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
21. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. J. Mach. Learn. Res. **17**(1), 1334–1373 (2016)
22. Levine, S., Pastor, P., Krizhevsky, A., Quillen, D.: Learning hand-eye coordination for robotic grasping with large-scale data collection. In: International Symposium on Experimental Robotics, pp. 173–184. Springer (2016)
23. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: common objects in context. In: European Conference on Computer Vision, pp. 740–755. Springer (2014)
24. Milford, M., Shen, C., Lowry, S., Suenderhauf, N., Shirazi, S., Lin, G., Liu, F., Pepperell, E., Lerma, C., Upcroft, B., et al.: Sequence searching with deep-learnt depth for condition-and viewpoint-invariant route-based place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 18–25 (2015)
25. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM **38**(11), 39–41 (1995)
26. Ødegaard, N., Knapskog, A.O., Cochin, C., Louvigne, J.C.: Classification of ships using real and simulated data in a convolutional neural network. In: Radar Conference (RadarConf), 2016 IEEE, pp. 1–6. IEEE (2016)
27. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1717–1724. IEEE (2014)
28. Peng, X., Sun, B., Ali, K., Saenko, K.: Learning deep object detectors from 3D models. In: IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1278–1286. IEEE (2015)
29. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The synthia dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
30. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
31. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: a database and web-based tool for image annotation. Int. J. Comput. Vis. **77**(1), 157–173 (2008)
32. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 3, p. 6 (2017)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR (2014). **(abs/1409.1556)**
34. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
35. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: a large data set for nonparametric object and scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. **30**(11), 1958–1970 (2008)
36. Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., Birchfield, S.: Training deep networks with synthetic data: bridging the reality gap by domain randomization. arXiv preprint arXiv:1804.06516 (2018)

37. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3D shapenets: a deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912–1920 (2015)
38. Yan, K., Wang, Y., Liang, D., Huang, T., Tian, Y.: Cnn vs. sift for image retrieval: alternative or complementary? In: Proceedings of the 2016 ACM on Multimedia Conference, pp. 407–411. ACM (2016)
39. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems (NIPS), pp. 3320–3328 (2014)
40. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833. Springer, Cham (2014)
41. Zheng, L., Yang, Y., Tian, Q.: SIFT meets CNN: A decade survey of instance retrieval. IEEE. Trans. Pattern. Anal. Mach. Intell. **40**(5), 1224–1244 (2018)
42. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. **40**(6), 1252–1264 (2017)
43. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Advances in Neural Information Processing Systems, pp. 487–495 (2014)
44. Zhuo, L., Jiang, L., Zhu, Z., Li, J., Zhang, J., Long, H.: Vehicle classification for large-scale traffic surveillance videos using convolutional neural networks. Mach. Vis. Appl. **28**(7), 793–802 (2017). https://doi.org/10.1007/s00138-017-0846-2
45. Zuo, H., Lang, H., Blasch, E., Ling, H.: Covert photo classification by deep convolutional neural networks. Mach. Vis. Appl. **28**(5), 623–634 (2017). https://doi.org/10.1007/s00138-017-0859-x

Ertugrul Bayraktar completed Mechanical Engineering program at the Yildiz Technical University in 2009, and masters programs in Finance Engineering and Mechatronics Engineering from Kadir Has University and Istanbul Technical University in 2011 and 2013, respectively. In 2018, he obtained his Ph.D. in Mechatronics Engineering from the Istanbul Technical University. During his graduate studies, he worked as a research assistant at Istanbul Technical University.

Cihat Bora Yigit completed Mechanical Engineering in Istanbul Technical University in 2010 and Masters program in Mechatronics Engineering of the same university in 2012. He received his Ph.D. in Mechanical Engineering from Istanbul Technical University in 2018. Between 2012 and 2018, he worked as a research assistant in the Department of Mechanical Engineering, Systems Engineering and Control of Istanbul Technical University.

Pinar Boyraz obtained her Ph.D. degree in Mechatronics from Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, UK, in July 2008. She worked as a Post-doctoral RA in the Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, USA, from 2008 to 2010. She was an Assistant Professor from 2010 to 2014 and an Associate Professor from 2014 to 2018 in Mechanical Engineering Department of Istanbul Technical University, TR.