Backbone Neural Network Design of Single Shot Detector from RGB-D Images for Object Detection

Purvesh Sharma
Ingram School of Engineering
Texas State University
San Marcos, USA
p_s218@txstate.edu

Damian Valles
Ingram School of Engineering
Texas State University
San Marcos, USA
dvalles@txstate.edu

***Abstract*—Recognition technology has gained state of art performance with the dawn of deep convolutional neural network and with these achievements in the field of computer vision, machine learning and 3D sensor, industries are near to start new era of the automation. However, object detection for robotic grasping in varying environment, low illumination, occlusion and partial images gives poor accuracy and speed to detect object. In this research, a multimodal architecture is designed to be used as a base network/ backbone network of Single Shot Detector (SSD). This architecture uses RGB and Depth images as an input and gives single output. Most of the researchers used VGG16/19, ResNet and MobileNet for detection purposes. In this paper, a new architecture is designed to perform a specific task of grasping. For classification using RGB-D architecture, it achieved an average accuracy of 95% with the learning rate of 0.0001 and outperforms the other architectures in accuracy for limited objects.**

Keywords—Deep Convolutional Neural Network, Machine Learning, 3D object recognition, detection, SSD.

I. INTRODUCTION

Robots, cobots (collaborative robots) have been instrumental in performing the tasks that are unfeasible for humans to attain. Robots potentially raise efficiency and safety levels due to their intelligence, dexterity, adaptability and personalisation. Situation like, COVID-19 pandemic which led to a partial or complete shutdown of plants and factories, and resulted in economic downturn. To overcome the situation companies may need to reconfigure their workplace, warehouses and factories as replacing humans with intelligent machines has now become the necessity. The practice of social distancing may also hasten the process of automation. People have already begun to see the progress in this regard.

After the commencement of Amazon Picking Challenge (APC) in 2015, the idea of object recognition and localization attracted attention. It also granted individual the right to create their own robotic system so as to aid with mass production and management. APC challenged the robotics research community that involved integrating the state of the art in object recognition and perception, path planning, motion planning and grasp planning. They comprised a streamlined model of the task to pick items from the shelves and put them into receptacles which is generally faced by humans in warehouses [1]. Object detection comprises mainly two parts: 1) Object Recognition, and 2) Object Localization. Object Recognition is a classification of different objects and localization is finding out the location and bounding box generation in the image or real-time application.

Broadly speaking, researchers have found two methods of object detection, one is based on Regional Proposal Network which works in two stage also known as Two Stage Method; RCNN, Fast RCNN, Faster RCNN are the types of this method. This method is designed with the consideration of accuracy; therefore, it takes more time to detect any object, hence these are slow methods and not suitable for real-time application. The other one is based on Regression, these methods are also known as One Stage Method, You-Only-Look-Once (YOLO) and Single Shot Detector (SSD) comes under this method. This Methods are very fast and accurate as compared to two stage method and designed for real-time application. For the designing of an object detection method based on deep convolutional neural network, a feature extraction network is needed. A large number of feature extractor networks has emerged in the field of deep learning such as VGG, ResNet, DenseNet, and many more. Zhu Dongtao et. al. [2] used AlexNet for the design of improved SSD and achieved mAP of 75.28% which 10% higher than classical SSD. Xin Lu et. al. [3] replaced VGG-based SSD to SSD-ResNet101 to detect dangerous goods and achieved mAP of 72.40 % which is outperformed by 17.4% as compared to original VGG-based SSD.

In this research, industrial robot arm and its controller from the ASEA Brown Boveri (ABB) company manufactures will be used, more technical details of robot arm and controller discussed in [1]. To grasp the objects, a gripper is mounted on robot arm with the Structure Core camera to detect the given object. An implementation of a vision system for detection and recognition of 3D-objects in Red-Green-Blue-Depth (RGB-D) image is captured by the Structure Core camera. Experimentation work will be performed under different illumination and challenging conditions such as cluttered, occluded, and obstructed partial images. The problem of detecting an object and recognizing it with the given RGB-D images includes low-illumination, cluttered background, occlusion and partial images. Most of the current state-of-art in this problem are using low-resolution camera, and basic deep learning CNN architecture such as VGG, ResNet, pretrained on ImageNet which provides a lower accuracy.

The Deep Convolutional Neural Network (DCNN) proposed is a Multi-Modal approach which will be used as a base or backbone network for object detection network. In this base

978-1-7281-9656-5/20/$31.00 ©2020 IEEE

network two inputs are given, RGB image and Depth image. In this approach, an architecture is designed with two streams, one for RGB input and other one for Depth image which are concatenated after extracting essential features from both the streams to get a single output. Image processing is an indispensable part of this process as the motive of the research is to detect an object in the low illumination, cluttered background and partial images. Base network is used for feature extraction from the input images using convolutional layers, max-pooling layers which will be fed to extra feature extractions layer of Single Shot Detector (SSD) network for detection purpose. The intention to use SSD network is to get the bounding box for the object to be detected for grasping purpose which will be passed to robot controller. Finally, results of these methods will be compared for better and accurate performance. Contribution to the robotic vision community are:

- Research contributes to the new multi-modal DCNN architecture for 3D object recognition and detection.
- It may give state of art accuracy for robotic vision with low cost and better-quality camera.

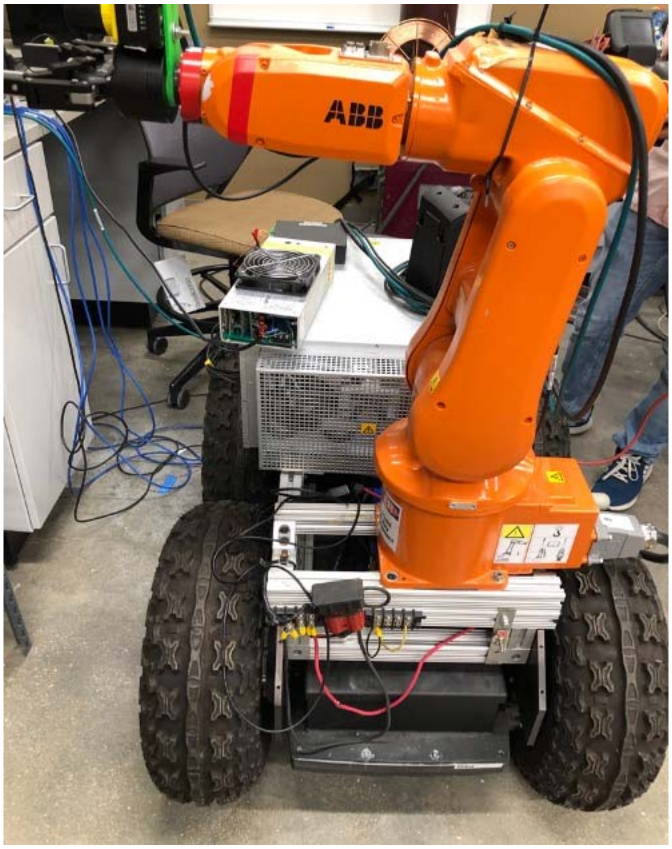


Fig. 1 ABB Robot Arm to be used for hardware implementation.

The method proposed here is chiefly for warehouse and industrial applications. Where humans need to tackle several objects and situation which are at times difficult and lethargic for humans and effortless for robots. However, notwithstanding remarkable performances, when real scenes are considered the problem for robots to detect things accurately and efficiently still challenges scientifically. To overcome the problem of warehouses, where there are amorphous objects, different shapes, and structure with blur, noisy images and complex shapes which makes it difficult to identify and grasp the object, some design development and optimization tools like TensorFlow Inference will be used in Python programming language. This project propounds a design approach on 3D-image in the cluttered and occluded image with poor illumination.

II. Detection Techniques

As it is discussed earlier, there are two methods for object detection: two stage detector and single stage detector. In this section, some more details of types of detectors are discussed.

A. *Two Stage Detector*

1) RCNN: Region-based Convolutional Neural Network (RCNN), it is a two-stage object detection method that uses selective search method and generates around 2,000 regions to detect the object. It takes 40-50 seconds to detect an object using VGG16 on a GPU [4]. This method is computationally heavy and expensive and takes long time to predict, it cannot be implemented in real-time.

*2) Fast RCNN:*Instead of using three different model like RCNN, it uses one combined convolutional neural network model for training with selective search method. This method is faster than RCNN and takes about two second with better accuracy than RCNN. This method is also not good for real time as it takes more time and computationally expensive [4].

3) Faster RCNN: Faster RCNN is improved version of Fast RCNN, it uses Regional Proposal Network (RPN) and much faster than previous two types. It shares image convolutional features with the detection network and takes around 0.2 second to detect an object. The main drawback of Faster RCNN is, its performance depends on the previous operation [5].

B. *One Stage Detector*

Initially, one stage detector was designed to gain better speed than two stage detectors, but with the improvement and advancement of deep neural network architecture, it outperformed the accuracy as well. Until now, two well-known one stage detectors are available; Single Shot Detector and You Only Look Once (YOLO).

1) YOLO: You look only once as name suggest, it is a single stage detector, which looks only once to detect an object. Different versions of YOLO have been designed to improve the performance like v1, v2, v3, Tiny YOLO. The basic idea behind YOLO architecture is, it generates (S x S) cells grid in the image, in each cells n number of bounding boxes are generated. It generates ($S\ x\ S\ x\ (n\ x\ 5 + P)$) where, five represents different attributes height, weight, confidence score, and center co-ordinates (x, y) of detected objects. P represents probability of class. In version-2 version of YOLO, batch normalization is added and in version-3 version, Logistic regression is added to get better results. There are some drawbacks of YOLO, it has hard time to find tiny objects and accuracy is also compromised to improve speed [6].

2) SSD: Single Shot Detector, it is the best choice if you don't want to compromise with accuracy for speed. In SSD, base network is used for feature extraction. Feature extraction can be done by known architectures like VGG16/19, Resnet, Inception or own designed model. It uses multiple feature extraction map using anchor boxes of different ratios similar to Faster RCNN. Researchers observed a better accuracy with different datasets and trying to optimize and improving further. SSD outperformed all of the detectors with mAP of 76.9% [7].

III. Proposed Design Approach

In this paper, the SSD model was designed with RGB-D classification deep neural network architecture as a base network. The foremost ambition of the proposed technique is to design a model to successfully recognize and localize the object in the warehouse under the poor lighting conditions, partial view, low perceptibility, and challenging circumstances. Furthermore, this model will be integrated with the ABB robot which can grasp the object from the warehouse shelves. In robotic vision, most of the researchers implement different variations of DCNNs for 3D-object recognition and detection to get the optimized results for different objects in diverse challenging conditions [8]. In this section, methods to be used in the research work is explained in the brief and shown in Fig. 2.

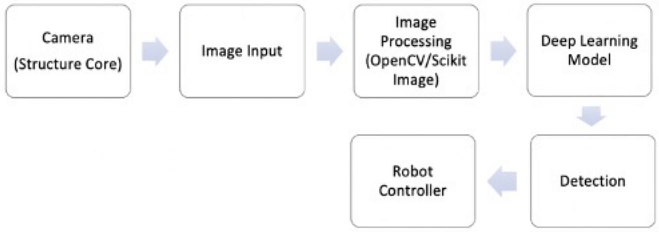


Fig. 2 Process Block Diagram [1].

A. Dataset Acquisition

The main goal of this research is to identify objects in a semi-structured environment that includes cluttered objects, poor illumination, amorphous body, occluded objects by other objects, reflective surface, and partial views of an object. For the model training, the dataset with the large number of samples are needed containing several objects in warehouse environments. Some datasets used for the APC competition and other purposes include blurry images, low lights, high brightness, reflective images, partial, and RGB-D images of multiple diverse objects were reviewed in literature review. In the proposed design, some public repository are used. The available datasets are particularly designed for 3D-grasping purpose and for pick-and-place robot such as the MIT Princeton dataset [9][10], and the RGB-D Washington dataset [11].

The dataset used for this research is Princeton and Washington RGB-D dataset. These datasets are developed by Team MIT, Princeton University and Washington University which consists of 136,575 and 41,877 RGB-D images. The images contain common household objects which have been classified into 39 classes for Princeton dataset and 51 classes for Washington dataset. The images are captured using a RealSense F200 and Kinect style sensor to generate 640x480 RGB-D frames [12]. These datasets contain most of the challenging conditions which is mentioned in the proposal. For the proposed method, it is planned to use the mentioned datasets RGB-D images. OpenNI 2 tool will be used to capture the images from the Structure Core camera, more technical details have been discussed on OpenNI (Open Natural Interaction) and Structure Core in [1]. The machine learning model is be trained on both public dataset and will be trained on the captured images later, because in the dataset available to us does not have specific and enough partial images to train the proposed model.

B. Pre-processing

To meet the environmental and surrounding challenges for object recognition and localization, pre-processing is required, as available dataset is with a limited variation on the images, to make the model robust and applicable to all the illumination changes, different lighting conditions, and to get better accuracy at prediction. In this research, objective is to find the solution for cluttered, occluded and the object in the low light. However, pre-processing increases the complexity and computation cost to the research work, but without pre-processing section testing accuracy rate will vary and less accurate for challenging condition. In this research, OpenCV is used for pre-processing, DCNN is used for feature extraction, these methods are compatible with Python and both of them supports machine learning libraries and deep learning frameworks as previously discussed in [1].

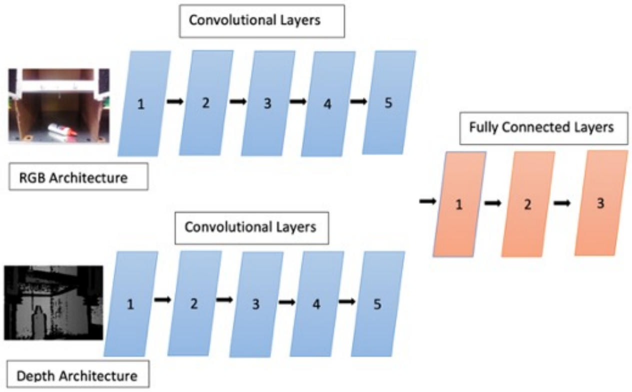


Fig.3 RGB-D Proposed DCNN Architecture [1].

C. Machine Learning Model and Base Classification Network

In the machine learning model, Deep Convolutional Neural Network is designed in the Python programming using Keras, TensorFlow open library. After the pre-processing of the images by using an appropriate tool, it is served to the different layers of the DCNN model. In the CNN model, there are numbers of hidden layers which is made up of neurons to extract the features. DCNN model is the best suited for image data classification and detection purpose. A simple block diagram of neural network for the proposed methods is shown in Fig. 3 and Fig. 4.

For the object image classification, DCNN is the state-of-art technology, there are number of well-known DCNN architectures available to recognize and detect the object. For this research design, own architecture is made to get better results for object detection and warehousing automation for specific objects. It is desired to experiment with two different DCNN models to obtain the best accuracy in detecting objects. After capturing the RGB-D images from the camera and pre-processing, each image is fed to the two parallel DCNN architecture for convolution and feature extraction shown in figure[3]. Before going to the output layer, it is required to merge the output of both DCNN. Afterwards, it is connected to fully-connected layers, which is be used for final classification.

In the model, multiple convolutional layers are used, which are followed by other layers such as activation layer, batch normalization, max pooling layer in a pattern for both RGB and Depth architecture. To combine this RGB and Depth network concatenation layer is used, at last fully connected layers are used to get classified output using SoftMax layer.

D. SSD Model using Base Network

So far, object classification network or base network was discussed to extract the feature of an input image. To grasp an object using robotic arm, location of object is needed, which is called localization. Combined model of section classification and localization is called detection. In earlier section, some techniques of detection are discussed, over careful consideration of speed and accuracy, SSD model to be implemented using RGB-D architecture to achieve better accuracy. In this method, a smaller version of SSD will be used [7] to train model from scratch using defined dataset. In the figure shown SSD architecture is designed. In this proposed model, five sections are mentioned, input, base network, extra feature extraction layer, detection and non-maximum suppression. Single Shot Detector completes the task in one sweep of the network, thus results in faster detection with small and lighter architecture.

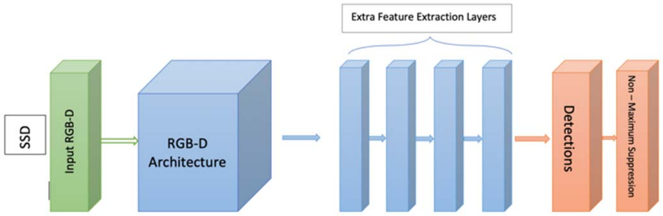


Fig.4 SSD Representation using RGB-D Architecture.

To train the deep layers for the DCNN architecture with large image datasets of RGB-D images, it requires significant amount of computational time and memory. Computational time can be reduced by using GPUs (Graphic processing unit). Furthermore, the proposed model can also be optimized using the Tensor Inference model to increase the computation efficiency using GPU's. Finally, the recognized and detected object position will be fed to the robot arm controller which will be in coordinate (*x, y, z*) form to help the robot arm to grasp the object from correct position using gripper.

IV. Experimental and Training Procedure

In this paper, purpose of object detection is to grasp an object in warehouse and industry in the low illumination and partial view. In this experiment two different datasets are considered which resembles the given ambience condition with some image processing performed on it. Both the datasets have number of objects in it, but for the grasping purpose small objects are considered which can fit into two finger grippers, such as water bottle, coffee mug, school glue, bowl, soap bar. For making the images more realistic and relevant to the problem given, brightness and contrast of the images are varied to get better results at the time of prediction. Initial experiments are trained on LEAP cluster and HiPE3 server provided by Texas State University and HiPE research group [13]. The HiPE3 server utilizes Dual Intel Xeon Gold with a 2.3GHz with 18 cores processor with dual Nvidia Tesla V100 GPUs, and LEAP cluster comprises 28 CPU cores on each compute-node.

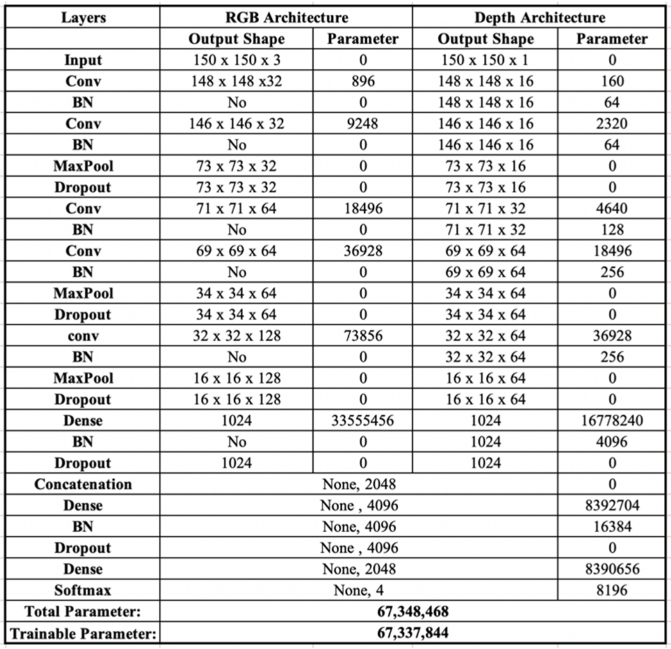
In the RGB-D architecture designed, four architectures are designed with two input and one output, separate and similar DCNN architecture for RGB and Depth image are designed in all four architecture. To get the best base model for feature extraction for object detection model from the initial design of different architecture, experiments executed on the leap cluster with the different learning rate of 0.001 and 0.0001, and momentum of 0.9, batch size to 32, 64, 128 with input size of 150x150 and 200x200. In the performed experiments, brightness and contrast are randomly generated to train the proposed model for different illumination condition, for object detection SSD training, more image processing techniques will be used to make system robust and accurate in any give challenging environmental conditions.

In the multi-input architecture, RGB and depth images are used to train the model. Out of four models, Architecture-I and II models have similar architecture for RGB and depth path with five convolutional layers, and Architecture-IV has six convolutional layers and Architecture-III with the different pattern of combination of five convolutional layers to get the suitable learning curves.

V. Initial Results

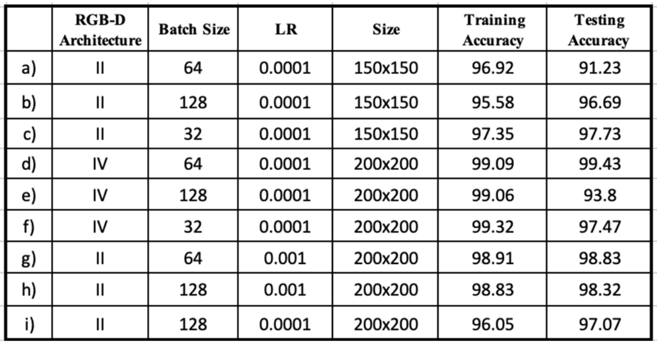
In this section, some initial results of proposed base RGB-D architecture are shown with different batch size and learning rate. In the Fig. 5 accuracy curves with batch size of 32, 64, and 128 is shown with learning rate of 0.0001 and 0.001. As it is seen, training curves are smooth and progressive with each epoch, total 80 epochs were run for 150x150 size image and 70, 50 epochs for 200x200. Architecture II performed better than others in speed and accuracy for RGB-D images. The F1-scores obtained from this architecture for the *bowl* and *soda can* objects reached a 0.94 and 0.93. The *dove soap bar* and *school glue* objects is 1.00 with the batch size of 128. Its learning curves also gave better and smooth performance, more details of RGB-D Architecture II with parameters and shape is mentioned in Table I. Some initial results and learning curves are shown in Fig. 5 with the details on Table II. Some more experiments are under progress and consideration for future work.

TABLE I. DETAILS OF RGB-D ARCHITECTURE II



Layers	RGB Architecture		Depth Architecture	
	Output Shape	Parameter	Output Shape	Parameter
Input	150 x 150 x 3	0	150 x 150 x 1	0
Conv	148 x 148 x32	896	148 x 148 x 16	160
BN	No	0	148 x 148 x 16	64
Conv	146 x 146 x 32	9248	146 x 146 x 16	2320
BN	No	0	146 x 146 x 16	64
MaxPool	73 x 73 x 32	0	73 x 73 x 16	0
Dropout	73 x 73 x 32	0	73 x 73 x 16	0
Conv	71 x 71 x 64	18496	71 x 71 x 32	4640
BN	No	0	71 x 71 x 32	128
Conv	69 x 69 x 64	36928	69 x 69 x 64	18496
BN	No	0	69 x 69 x 64	256
MaxPool	34 x 34 x 64	0	34 x 34 x 64	0
Dropout	34 x 34 x 64	0	34 x 34 x 64	0
conv	32 x 32 x 128	73856	32 x 32 x 64	36928
BN	No	0	32 x 32 x 64	256
MaxPool	16 x 16 x 128	0	16 x 16 x 64	0
Dropout	16 x 16 x 128	0	16 x 16 x 64	0
Dense	1024	33555456	1024	16778240
BN	No	0	1024	4096
Dropout	1024	0	1024	0
Concatenation	None, 2048			0
Dense	None , 4096			8392704
BN	None, 4096			16384
Dropout	None , 4096			0
Dense	None, 2048			8390656
Softmax	None, 4			8196
Total Parameter:	67,348,468			
Trainable Parameter:	67,337,844			

TABLE II. INITIAL RESULTS FOR THE CURVES SHOWN IN FIG. 5



	RGB-D Architecture	Batch Size	LR	Size	Training Accuracy	Testing Accuracy
a)	II	64	0.0001	150x150	96.92	91.23
b)	II	128	0.0001	150x150	95.58	96.69
c)	II	32	0.0001	150x150	97.35	97.73
d)	IV	64	0.0001	200x200	99.09	99.43
e)	IV	128	0.0001	200x200	99.06	93.8
f)	IV	32	0.0001	200x200	99.32	97.47
g)	II	64	0.001	200x200	98.91	98.83
h)	II	128	0.001	200x200	98.83	98.32
i)	II	128	0.0001	200x200	96.05	97.07

CONCLUSION AND FUTURE WORK

Deep learning and CNN for object recognition and localization are arriving at another fields, and when it is joined with a robot arm, it can carry profitable changes to the industries as working labor is diminishing and robot can replace human. In this research work, the Deep neural system model is designed for 3D object recognition and localization utilizing RGB-D pictures for pick and place robot in various testing conditions. The base network for SSD gives an average accuracy of 95% with learning rate of 0.0001 and batch size of 128 and 64 on Architecture-II. Architecture-II outperformed other architectures on computation speed and classification. After further analysis, it will be implemented to detection network to detect the object under challenging conditions such as illumination variations, partial view, cluttered image and send the controlling commands to the robot arm. Method discussed, which is planned to implement on robot arm with structure RGB-D camera mounted on it. The output of detection network needed in coordinate (*x, y, z*) form to give the grasping command to the robot arm.

For future work, robot and AI will change the manner in which people work affecting industries, production with more with robots than engineers. Just to make a robot solid and innocuous, more exact vision is required with better exactness. Besides, this base model will be added to SSD detection network and will be actualized to the mechanical robot after thorough testing in a modern situation, which can get higher exactness for incomplete pictures and more objects can be added to the quality preparing with bigger dataset.

REFERENCES

[1]. P. Sharma and D. Valles, "Deep Convolutional Neural Network Design Approach for 3D Object Detection for Robotic Grasping," *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2020, pp. 0311-0316, doi: 10.1109/CCWC47524.2020.9031186.

[2]. Z. Dongtao, C. Jie, Y. Xing, S. Hui and S. Liangliang, "Traffic Sign Detection Method of Improved SSD Based on Deep Learning," *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, Chengdu, China, 2018, pp. 1516-1520, doi: 10.1109/CompComm.2018.8780999.

[3]. X. Lu, X. Kang, S. Nishide and F. Ren, "Object detection based on SSD-ResNet," *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, Singapore, 2019, pp. 89-92, doi: 10.1109/CCIS48116.2019.9073753.

[4]. R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.

[5]. S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, June 2017, doi: 10.1109/TPAMI.2016.2577031.

[6]. P. Adarsh, P. Rathi and M. Kumar, "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model," *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2020, pp. 687-694, doi: 10.1109/ICACCS48705.2020.9074315.

[7]. W. Liu, D. Anguelov, D. Erhan, and C. Szegedy, "SSD: Single Shot MultiBox Detector," ECCV, vol. 1, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0.

[8]. S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, 2017, pp. 769-776, doi: 10.1109/IROS.2017.8202237.

[9]. SUN3D-database, http://sun3d.cs.princeton.edu [Accessed: Oct. 27, 2019] SUN RGB-D: An RGB-D Scene Understanding Benchmark Suite.

[10]. SUN RGB-D: An RGB-D Scene Understanding Benchmark Suite, http://rgbd.cs.princeton.edu, [Accessed: on Oct. 27, 2019].

[11]. RGB-D Object Dataset, https://rgbd-dataset.cs.washington.edu [Accessed: Oct. 27, 2019].

[12]. Griffiths, David, and Jan Boehm. "A Review on Deep Learning Techniques for 3D Sensed Data Classification." *Remote Sensing* 11.12 (2019): 1499. Crossref. Web.

[13]. High-Performance Engineering (HiPE) Research Group, https://hipe.wp.txstate.edu/technology/ [Accessed: Aug. 28, 2020].

[14]. A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, 2015, pp. 681-687, doi: 10.1109/IROS.2015.7353446.

[15]. S. Zhai, D. Shang, S. Wang and S. Dong, "DF-SSD: An Improved SSD Object Detection Algorithm Based on DenseNet and Feature Fusion," in *IEEE Access*, vol. 8, pp. 24344-24357, 2020, doi: 10.1109/ACCESS.2020.2971026.

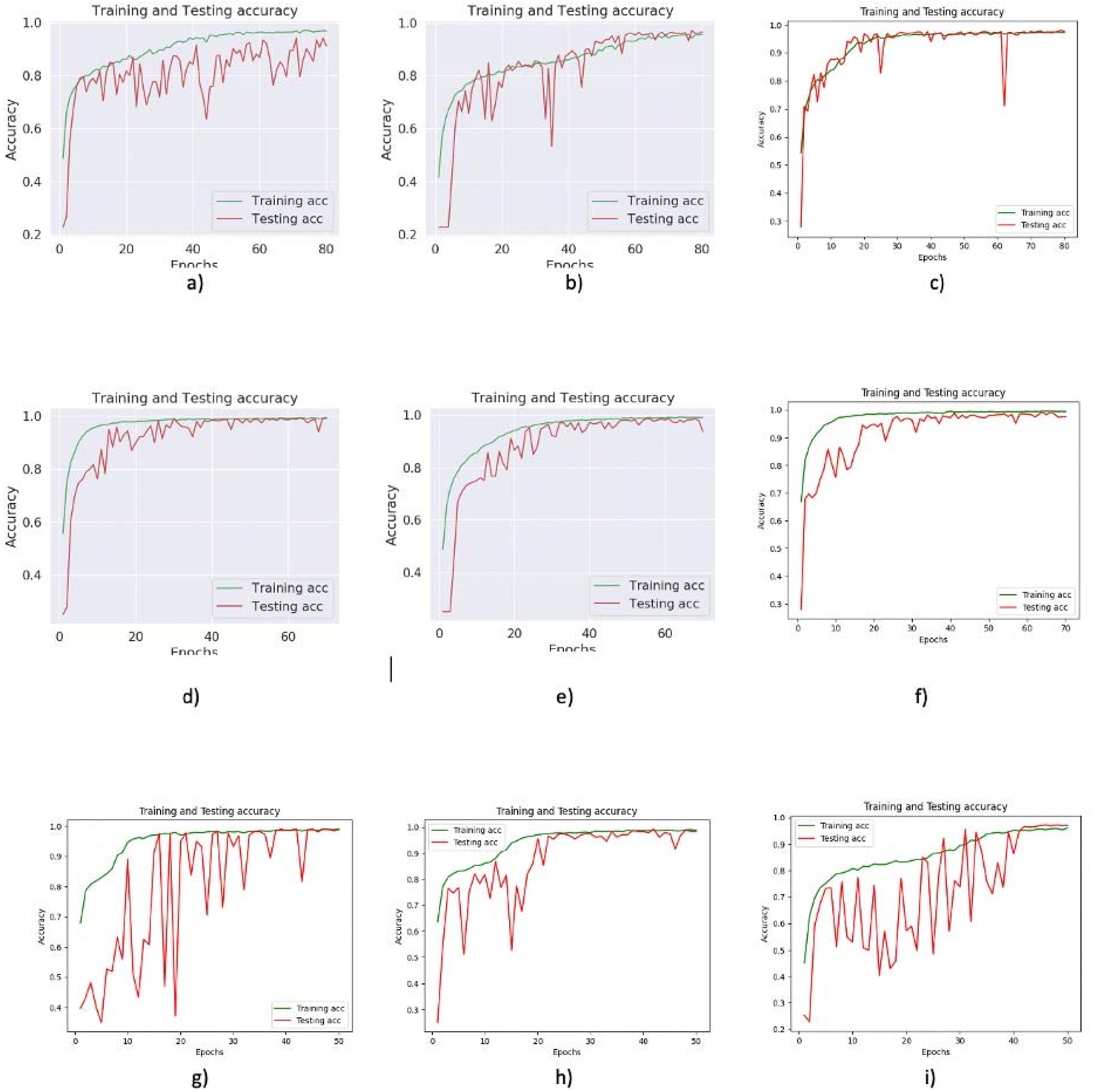


Fig. 5 Accuracy curves for Architecture II and IV. a) RGB-D Architecture-II with Learning rate of 0.0001 and Batch size 64. b) RGB-D Architecture-II with Learning rate of 0.0001 and Batch size 128. c) RGB-D Architecture-II with Learning rate of 0.0001 and Batch size 32. d) RGB-D Architecture-IV with Learning rate of 0.0001 and Batch size 64. e) RGB-D Architecture-IV with Learning rate of 0.0001 and Batch size 128. f) RGB-D Architecture-IV with Learning rate of 0.0001 and Batch size 32. g) RGB-D Architecture-II with Learning rate of 0.001 and Batch size 64. h) RGB-D Architecture-II with Learning rate of 0.001 and Batch size 128. i) RGB-D Architecture-II with Learning rate of 0.0001 and Batch size 128.