



Generative model based robotic grasp pose prediction with limited dataset

Priya Shukla¹ · Nilotpal Pramanik¹ · Deepesh Mehta² · G. C. Nandi³

Accepted: 13 November 2021 / Published online: 10 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

In the present investigation, we propose an architecture which we name as Generative Inception Neural Network (GI-NNet), capable of predicting antipodal robotic grasps intelligently, on seen as well as unseen objects. It is trained on Cornell Grasping Dataset (CGD) and attains a 98.87% grasp pose accuracy for detecting both regular/irregular shaped objects from RGB-Depth images while requiring only one-third of the network trainable parameters as compared to the existing approaches. However, to attain this level of performance the model requires the entire 90% of the available labelled data of CGD keeping only 10% labelled data for testing which makes it vulnerable to poor generalization. Furthermore, getting a sufficient and quality labelled dataset for robot grasping is extremely difficult. To address these issues, we subsequently propose another architecture where our proposed GI-NNet model is attached as a decoder of a Vector Quantized Variational Auto-Encoder (VQ-VAE), which works more efficiently when trained both with the available labelled and unlabelled data. The proposed model, which we name as Representation based GI-NNet (RGI-NNet) has been trained utilizing the various split of available CGD dataset to test the learning ability of our architecture starting from only 10% label data with the latent embedding of VQ-VAE to 90% label data with the latent embedding. However, being trained with only 50% label data of CGD with latent embedding, the proposed architecture produces the best results which, we believe, is a remarkable accomplishment. The logical reasoning of this together with the other relevant technological details have been elaborated in this paper. The performance level, in terms of grasp pose accuracy of RGI-NNet, varies between 92.1348% to 97.7528% which is far better than several existing models trained with only labelled dataset. For the performance verification of both the proposed models, GI-NNet and RGI-NNet, we have performed rigorous experiments on Anukul (Baxter) hardware cobot.

Keywords Intelligent robot grasping · Generative inception neural network · Vector quantized variational auto-encoder · Representation based generative inception neural network

✉ Priya Shukla
priyashuklalko@gmail.com

G C Nandi
gcnandi@iiita.ac.in

¹ Student at Center of Intelligent Robotics, Indian Institute of Information Technology Allahabad, Prayagraj, 211015, U.P., India

² Winter intern at Center of Intelligent Robotics, Indian Institute of Information Technology Allahabad, Prayagraj, 211015, U.P., India

³ Professor and Head of the Center of Intelligent Robotics, Indian Institute of Information Technology Allahabad, Prayagraj, 211015, U.P., India

1 Introduction

Recently, with the advancement of Machine Learning and Deep Learning technologies, the capabilities of robots are increasing day by day, starting from pedestrian classification for real-time autonomous driving [2], prosthetic hand control [24], manipulating the very rudimentary type of tasks such as palletizing, picking/placing [6, 27] to the complicated tasks like trying to grasp and manipulate previously seen as well as unseen objects intelligently, the way we grasp and manipulate objects. Such capability enhancement of the robots, may qualify them to perform many routine household tasks and share the social as well as working space with us as collaborative robots, known as cobots, in the near future. However, to make it happen, we need to solve many

challenging problems, one of them, say for example, is to make it learn the tasks, such as grasping and manipulating objects intelligently, the way a human kid learns to grasp and manipulate various objects [26, 32]. A child normally has poor grasping skill and that's the reason we are reluctant to allow them to grasp sophisticated items for fear of damaging them. But when the same child grows up, develops enough grasping skills based on learning through experience, the detailed mechanism of such learning by our brain is hitherto unknown to us. Also, we can not afford to provide such a long childhood to the robots for learning and hence the other alternatives are being explored using machine learning and deep learning based architectures so that cobots can learn quickly to manipulate tasks more adaptively in a realistic household and industrial environment.

Thus in the present research, we intend to address the issues of correctly predicting antipodal robotic grasps, in the form of grasping rectangles, by developing appropriate deep learning based architectures which perhaps, is one of the most important problems associated with intelligent robot grasping. More specifically, we have contributed in designing a lightweight and object-independent model, GI-NNet which predicts grasps from the trained model and gives output as three sets of images (quality, angle, and width). Here, optimal grasps are inferred from these images only at the pixel level. This model is designed based on the concept of directly generating grasps which is different as proposed in GGCNN [22]. We have used Inception-Blocks [34] so as to keep the trainable parameters comparatively lower along with the ReLU activation function [1], to avoid vanishing gradient problem with a learning rate of 0.001 in order to generate appropriate information from a variety of kernel sizes. These actions improve the accuracy (up to 98.87%) substantially compared to the state-of-the-art (SOTA) models [17].

However, somehow the performance of all the above-mentioned models, in terms of tuning the large number of learning parameters, highly depends on the availability of labelled data. Owing to the scarcity of labelled data in the robot grasping domain, we subsequently propose to create a new architecture by attaching our model as a decoder with unsupervised learning based architecture known as Vector Quantized Variational Auto-Encoder (VQ-VAE), which we design to work efficiently when we train it with available labelled dataset as well as unlabelled data [20]. Our proposed GI-NNet integrates VQ-VAE model, which we name as Representation based GI-NNet (RGI-NNet), has been trained with various splits of label data on CGD with as minimum as 10% labelled dataset together with latent embedding generated from VQ-VAE up to 90% labelled data with latent embedding of VQ-VAE. The

performance level, in terms of grasp pose accuracy of RGI-NNet, varies between 92.13% to 97.75% which is far better than many other existing SOTA models trained with only labelled dataset.

Figure 1 illustrates the overview of our proposed model. We have trained our proposed models, GI-NNet and RGI-NNet over RGB-D and RGB images respectively, to obtain corresponding grasps from generated quality, angle, and width images, which are then used to infer an optimal grasp at the pixel level. Following are the major contributions of the present research:

- In this research, two novel grasp prediction models, GI-NNet and RGI-NNet, have been proposed to predict an optimal grasp at the pixel level. Subsequently, both the models have been trained and evaluated on CGD.
- Evaluation of GI-NNet on CGD provides a promising increase in grasp accuracy of 98.87% whereas GGCNN [22] and GR-ConvNet model [17] reports SOTA success rate of 73.0% and 97.7% respectively.
- GI-NNet shows an improved success rate incorporating a lesser number of total trainable model parameters (5,92,300) as compared to GR-ConvNet [17] (19,00,900).
- The performance of proposed RGI-NNet architecture is analysed on CGD for split ratios of 0.1, 0.3, 0.5, 0.7, and 0.9 respectively. For minimal (10%) labelled training data, it attains an accuracy of 92.13% which shows a significant performance improvement on the limited available grasping dataset.
- In the final output of the convolutional layers we have experimented with two transfer functions, Sigmoid and Tanh which provide quality output and angle output ($\sin 2\Psi$ and $\cos 2\Psi$) respectively.

The rest of this paper is arranged in the following format: Section 2 discusses previous related research works with their limitations. In Section 3 the problem definition has been elaborated. Section 4 depicts the grasp pose preliminaries with the concept of generative grasp approach, inception module, Variational Auto Encoder (VAE), Vector Quantized Variational Auto-Encoder (VQ-VAE) architectures, training dataset details and grasping metric. Section 5 describes detailed methodologies of our proposed model architecture along with the logical reasons for designing this model on rectifying the limitation of the previously proposed approaches. It also comprises the training method, incorporating loss and activation functions. Section 6 illustrates the details about robotic grasp pose generation and execution. Section 7 presents experimental setup design, models' evaluations, and their comparative analyses with existing SOTA models. Conclusions and recommendations for future research have been presented in the Section 8.

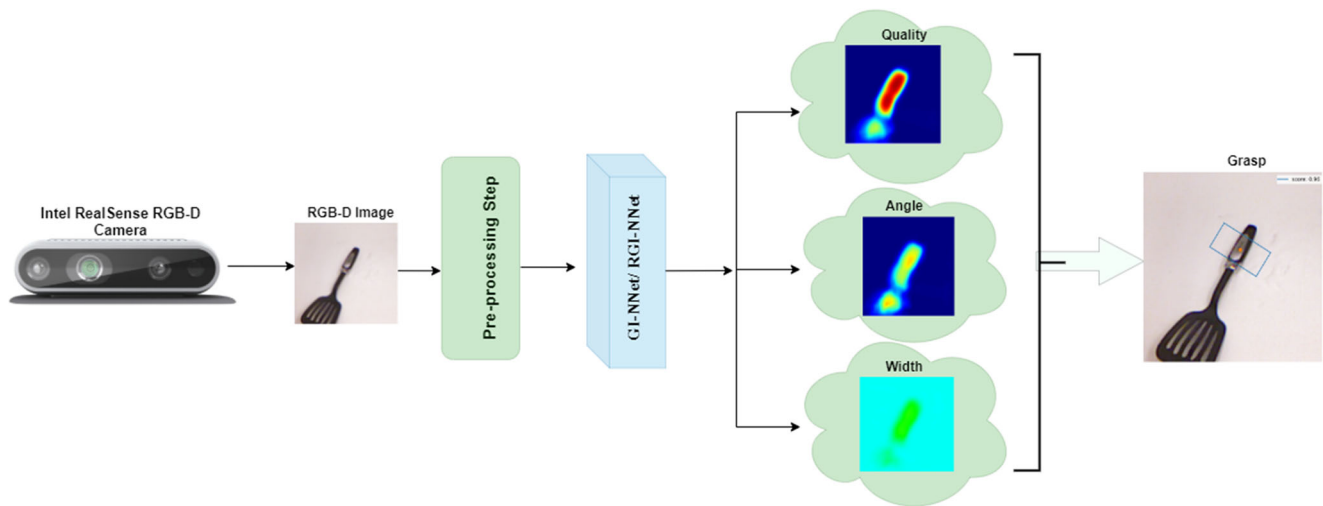


Fig. 1 Structural overview of our approach to predict an optimal grasp for an object

2 Related work

In the field of robotics, intelligent robotic grasping has been one of the very compelling areas to the researchers. In spite of the fact that the issue appears to simply have the option to locate an appropriate grasp on any object, the objective includes multifaceted components like different shapes and physical properties for the particular object. Previously, the robotic grasping research works are based on hand engineering [16, 21], which is essentially time consuming, lacks agility (slight changes in the grasping environment results grasp failure) but provided a headway in grasping utilizing multiple fingers [6, 15]. However, to acquire a steady grasp, the mechanics and contact kinematics of the end effector in touch with the objects are examined from the study [5, 31]. Before advances in machine learning, the grasping is performed by utilizing the supervised learning and the models used to learn from synthetic data [29] with a restriction over the environments such as office, kitchen, and dishwasher. Overcoming these limitations requires huge data and bigger models having much better learning/training abilities which, these days, can be accomplished using deep learning architectures for detecting the skilled/optimal grasp. By the way, in this paper, like any other papers in the Machine Learning and Deep Learning paradigm, Learning in a conventional way means when we have labelled data, the network parameters are getting adjusted/fixed through several iterations run through Back-propagation Algorithm. We call this as supervised learning and the models which get trained by this process are known as discriminative models. On the other hand, when we have unlabelled data and we want our models to adjust/fix its parameters using again the same back-propagation algorithm is called unsupervised learning [7] and such models are called generative models.

VQ-VAE which we have used in our approach is one such generative model.

In recent work, Fully Convolutional Grasp Quality Convolutional Neural Network (FC-GQ-CNN) predicts an optimal grasp quality utilizing a data collection policy and synthetic training environment which has poor depth information derived from Reinforcement Learning based on policy [28]. Though it generates stable grasps, recent research works are increasingly focusing on RGB-Depth data, which has more precise and ready-made depth information obtained directly from the images captured by stereo cameras, to produce grasp poses. Various research shows that deep neural network models are more useful for generating an efficient grasp pose for seen as well as unseen objects [30, 39].

With the introduction of the grasping rectangle concept in [13], RGB-D images are utilized to detect a grasp pose with a two-step learning process which are as follows:

1. Primarily, the search space is narrowed down by sampling candidate grasp-rectangles.
2. Subsequently, an optimal grasping-rectangle is determined from the candidate grasp-rectangles.

A similar two-step method is used in [19, 23] but the model performance decreases due to their large execution time. Whereas, AlexNet like architecture has also performed better on new objects by increasing the size of the data [23]. Due to large grasp inference time Morrison et al. introduces a generative approach, GGCNN, that produces a grasp pose from depth images [22]. Existing limitations of Computational complexity and discrete sampling are rectified through GGCNN architecture. In a very recent research work, Kumra et al. [17] has introduced a Generative Residual Convolutional Neural Network (GR-ConvNet), based on

GGCNN architecture, which predicts grasp more efficiently with a large number of trainable parameters. However, the performances of all the models are highly dependent on the availability of labelled data which are not available in sufficient quantity in the robot grasping domain. Therefore, we subsequently propose to develop a new architecture by attaching our model as decoder with a semi-supervised learning based architecture known as Vector Quantized Variational Auto-Encoder (VQ-VAE), which we make to work efficiently when we train it with available labelled dataset as well as unlabelled data [20]. Our proposed GI-NNet integrated VQ-VAE model, which we name as Representation based GI-NNet (RGI-NNet), has been trained with various splits of label data on CGD with as minimum as 10% labelled dataset together with latent embedding generated from VQ-VAE up to 50% labelled data with latent embedding obtained from VQ-VAE. The performance level, in terms of grasp pose accuracy of RGI-NNet, varies between 92.13% to 97.75% which is far better than many other existing SOTA models trained with only labelled dataset. The detailed architecture of our proposed models, GI-NNet and RGI-NNet, have been elaborated in the subsequent sections.

3 Problem formulation

Our main objective is to design an architecture that can be trained even with limited labelled data, since getting labelled data for robot grasping in a sufficient quantity is extremely difficult and also we wanted to reduce the network parameters so that our training time can be minimized. Here, optimization means minimizing trainable parameters (we have reduced trainable parameters from 19,00,900 to 5,92,300) and thereby minimizing training time. The novel architecture we built putting the GI-NNet as a decoder in the VQ-VAE network is our major innovative idea which made it possible to achieve these optimization criteria without compromising the accuracy much in getting optimal grasping rectangles. The conventional wisdom of optimization, although philosophically remains the same, in terms of formulating the problem mathematically with a concrete objective function with constrained equations/inequations is not possible for deep neural network architecture. An excellent paper [40], discusses details about why the conventional analyses approaches are not suitable for analyzing deep neural network architectures. In this paper, the authors argue that we are yet to discover a precise formal measure under which these enormous models can be checked. They have also shown through experiments that optimization continues to be empirically easy even if the resulting model does not generalize, which shows why the optimization is

empirically easy and must be different from the true cause of generalization.

More specifically our research is all about how to predict an optimal grasp pose by designing appropriate deep neural network architectures and after obtaining appropriate grasp poses how to execute such grasps in a real-time with a physical robot. In this work, robotic grasp pose is represented by (1) where G_R is the robot gripper grasp pose, $p_R = (x, y, z)$ is the center point position of the end-effector, Ψ_R is gripper's orientation measured about z -axis, for table top grasping, w_R is the gripper's opening width and q_R represents the effectiveness of the estimated grasp.

$$G_R = (p_R, \Psi_R, w_R, q_R) \quad (1)$$

Let us consider an n -channel input image of size, $I = R^{n \times h \times w}$. On this image frame, the grasp can be defined by (2) where, s_I is the (x, y) coordinate of the center point in pixels, Ψ_I is the angle of rotation in camera frame, w_I represents the width of grasp in pixel coordinates of image and q_I is the effectiveness score of the grasp.

$$G_I = (s_I, \Psi_I, w_I, q_I) \quad (2)$$

The effectiveness score, q_I is the nature of the grasp at each point in the input image and is shown as score esteem somewhere in the range of 0 and 1 where closeness to 1 denotes a more prominent possibility of a successful grasp irrespective of objects. Ψ_I shows the antipodal estimation of angular rotation needed at each point to grasp the object of interest which has been addressed here in the range of $[-90^\circ, 90^\circ]$. w_I represents the required width which has been kept in between $[0, w_{max}]$ pixels, where w_{max} denotes the maximum width of the grasping rectangle.

For robotic grasp execution, the predicted grasp pose in an image frame is used to infer the robotic grasp pose in the robot's reference frame by applying a known set of transformations. The grasp pose relation between image frame and robot frame has been shown in (3) where ${}^C T_I$, first transforms predicted grasp in an image frame to 3-D frame of a camera by means of its intrinsic parameters, then ${}^R T_C$ is used to transform camera frame coordinates to robot frame coordinates.

$$G_R = {}^R T_C {}^C T_I (G_I) \quad (3)$$

The set of grasps in image frame are then collectively represented by (4) where Ψ , W and Q are each of $R^{h \times w}$ dimension and contain the values of angle, width, and grasp quality respectively for each pixel in an image.

$$G = (\Psi, W, Q) \in R^{3 \times h \times w} \quad (4)$$

To predict a grasp pose in an image frame, we have designed two models- GI-NNet and RGI-NNet, the architectural

details of which together with the associated preliminaries have been discussed in the following section.

4 Preliminaries

Primarily, in this research two models have been proposed—one a Convolutional Neural Network (CNN) based inception model which we named as GI-NNet, and another one using this GI-NNet as a decoder of VQ-VAE which is primarily a generative model which we named as Representation based GI-NNet, i.e. RGI-NNet. The idea behind GI-NNet is to deploy Inception Blocks to utilize the best kernel size for feature extraction [35], so that more meaningful features can be extracted with lesser computation, whereas semi-supervised model RGI-NNet ensures that the latent embedding compensates for the scarcity of labelled data which ensures prediction of optimal grasping rectangles with limited labelled data only. Hence, it can be inferred that our proposed models also belong to the family of Generative grasp approaches [22]. All the required concepts have been discussed in the following sections.

4.1 Generative grasp approach

The approach of generative grasping is to generate a grasp pose for every pixel in the input image obtained from a camera. The input image thus obtained is passed through the generative convolutional neural network, which results in an output of four images. These four outputs are the grasp quality score, width of grasp, sine component of angle, and the cosine component of the angle for every pixel of the input image. The two angle components are then processed to form a single angle output to infer the orientation of a grasp rectangle. The advantage of this approach is to reduce the computational time compared to other SOTA approaches as discussed in [22]. Our proposed approach for simultaneously generating a grasp for an object with a lesser number of total trainable parameters shows promising results and hence has the potential to be used as a de-facto architecture for robot grasping rectangle prediction in the future.

4.2 Inception block

In this work, we have designed the inception block as a part of our proposed GI-NNet model in order to avoid any bias on the choice of kernel size. The detailed architecture of this inception module block is illustrated in Fig. 2. The structure of the inception block is inspired from [34] and [10]. We have used a minimal number of pools in order to minimize dependency on high value features. It has been observed

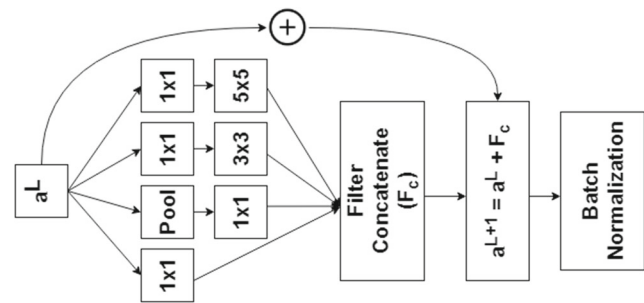


Fig. 2 Design of the Inception-Block architecture

that our decision to include this block in the architecture produces significantly better results with comparatively lesser parameters. The inception block after concatenating the output of different filters adds to the activation of the previous layer to the concatenated output. This helps our architecture to incorporate properties of both inception and residual networks. The output of each inception block is then passed to a subsequent Batch Normalization layer to speed up the learning process. Our second proposed architecture uses Variational Auto Encoder (VAE) which is discussed below.

4.3 Variational Auto Encoder (VAE)

For grasp detection, supervised learning approaches may be suitable in case of having a distribution of unlabelled training dataset with explicitly models latent representations [20]. The probability density function for training samples over the latent variables (say, r) is defined as -

$$\mathbf{p}_{\theta}(\mathbf{x}) = \int_r \mathbf{p}_{\theta}(\mathbf{r})\mathbf{p}_{\theta}(\mathbf{x} | \mathbf{r})\mathbf{d}\mathbf{r} \quad (5)$$

A training sample is produced from the unexplored latent r which is sampled over true priors of $p(r)$. Subsequently, the sample x is generated with the help of true priors and conditional Gaussian Distribution over the latent r . $p_{\theta}(x|r)$ is designed to learn the optimal parameters which are obtained by maximizing the likelihood of the learning samples. Since the likelihood of samples is highly intractable, the posterior distribution of the model appears to be also intractable. To get rid of, true posterior approximation $q_{\phi}(r|x)$ has been used which provides the lower bound of the training sample likelihood. Owing to the VAE architecture, the encoder is able to infer whereas the decoder helps in generation. Both the encoder and decoder networks help in determining the mean and the diagonal covariance for the probabilistic density function. The Evidence Lower Bound (ELBO) of the training samples likelihood can be determined as -

$$ELBO = \mathbf{E}_{\mathbf{r}} [\log \mathbf{p}_{\theta}(\mathbf{x} | \mathbf{r})] - \mathbf{D}_{\text{KL}}(\mathbf{q}_{\phi}(\mathbf{r} | \mathbf{x}) || \mathbf{p}_{\theta}(\mathbf{r})) \quad (6)$$

The encoder generated value enhances the likelihood of the learning samples and subsequently, the decoder produced value estimates the posterior which tends to the true prior. Hence, the VAE's encoder network empowers the inference of $q_\phi(r|x)$ which can be further utilized during representation learning. However, it is being observed that when VAE architectures are designed with more efficient decoders which enable the ignorance in learning of latent vectors. This ignorance problem is termed as the posterior collapse. VAE architecture based model VQ-VAE solves this issue by keeping the discrete latents only which has been elaborated in the next sub-section.

4.4 Vector quantized variational auto encoder (VQ-VAE)

In their research work, Vinyals et al. [37] claims that the image can be modelled efficiently only utilizing discrete symbols. Along with this, latent embedding space is integrated to the base architecture of VAE. A latent embedding can be represented as $e \in R^{N \times D}$. Here, N denotes the number of embeddings comprising latent embedding space and D denotes the embedding dimension. Encoder network generates $R_e(x)$ over the embedding vector space and following nearest neighbour lookup which outputs a continuous vector that is being further quantized $R_e(x)$. The described process is termed as vector quantization. Later on, decoder performs reconstruction tasks with the quantized vectors. The posterior distribution over the latent embedding is formulated as -

$$q(R = k|x) = \begin{cases} 1 & \text{for } k = \operatorname{argmin}_j \|R_e(x) - e_j\|_2, \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

The VQ-VAE network has been trained to obtain the discrete latent embedding by using the CGD data set which has been described below.

4.5 Dataset

For training purposes, we have used the Cornell Grasping Dataset [19] which consists of 885 RGB-D images of real objects comprising 5,110 positive and 2,909 negative grasps. The dataset represents antipodal grasps in the form of rectangular bounding boxes with their coordinates being the pixel itself. We have augmented the data by means of random cropping, zooming, and rotating resulting in nearly 51,000 grasps, since the used dataset is less in number. We have experimented with the model training procedure with and without augmentation. After examining the model performance in both the cases, it has been determined that training with data augmentation helps in the learning process. Subsequent to the training process, we have used a standard grasp rectangle metric [13] for evaluating the

performance of an optimal grasping rectangle as mentioned in the next sub-section.

4.6 Grasping metric

For the declaration of an optimal grasp, we make use of the rectangle metric, proposed in [13]. This metric defines a generated grasp as a successful grasp if it satisfies the following two conditions:

1. Intersection Over Union [IOU] score of the generated grasp rectangle and the ground truth grasp rectangle should be greater than 25%.
2. Offset in orientation of the grasp between generated and ground truth grasping rectangles should be less than 30° .

5 Methodology

Initially, in the proposed approach to make an effective use of Inception Blocks to improve the efficiency, we have tried incorporating ideas from Inception-V2 [35], but there is no significant improvement from it, as those ideas are relevant to very deep networks involving huge trainable parameters, not suitable for our robotic grasp applications due to vulnerability of over-fitting problem due to the availability of limited dataset. Hence, we have experimented with dropout layers within different layers of the network to improve the generalization capability of our proposed GI-NNet model, and we have found that incorporating three dropout layers, one before Inception Blocks, one within these blocks and one before feeding the network into the transposed convolutional layers, produce the best results. Further, we have designed a cascaded model by integrating VQ-VAE and GI-NNet architectures to implement the semi-supervised learning approach, where the GI-NNet has been augmented as a decoder. Detailed architecture of our proposed models, GI-NNet and RGI-NNet with training details have been elaborated in the following sections.

5.1 GI-NNet architecture

Our proposed GI-NNet takes RGB-D images as inputs and feeds them to a set of three 2-dimensional convolutional layers for extracting appropriate feature map, then to five inception blocks for parallelly selecting different filter sizes to reduce the computation cost, followed by a set of transpose convolution layers and finally through a convolution operation, generating the desired output images, as illustrated in Fig. 3.

Output of the network consists of four images representing grasp quality, angle ($\sin 2\psi$ and $\cos 2\psi$) and grasp

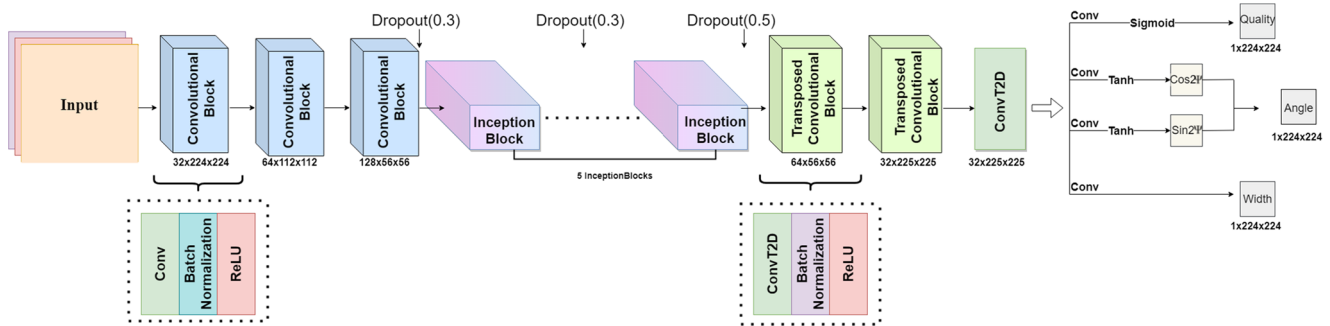


Fig. 3 Architecture of GI-NNet model

width, for each pixel of an image. These are generated as network output from the transposed convolutional layers. We use such filter layers and padding to obtain the final output images of the same dimension as an input image. We make use of *He* initialization (also known as Kaiming initialization) as proposed in [9] which is suitable for usage when employing ReLU activation function, in order to avoid exploding and vanishing gradient problems. This initialization sets the initialized weights with a zero mean and standard deviation of $\sqrt{2/n}$, where n is the number of inputs to the node.

5.2 RGI-NNet architecture

The proposed cascaded model takes RGB images as input and passes the input to the pre-trained VQ-VAE encoder which acts as a feature extractor and then a simple 2-D convolution operation is performed to match the dimensions of the latent space, the output is then passed to the Vector Quantization layers which translate the extracted features to discrete latent embeddings. The reinitialized decoder then takes latent embeddings and begins to reconstruct the images which are then fed to our proposed GI-NNet architecture. The re-initialization of the decoder helps the model to learn the reconstruction of the input image in a manner to get the best (in terms of intensity value) pixels for optimal grasp prediction. The detailed architecture of RGI-NNet is depicted in Fig. 4.

5.3 GI-NNet training details

For training GI-NNet architecture, we have used RGB-D images, whereas for training RGI-NNet only RGB images have been used with different train-test splits of CGD data. The training performances have been measured as shown in Fig. 5 which shows that 90-10 (train-test) split is the best in terms of test accuracy, since for such split the difference between training loss and validation loss gets minimized in and around 15 epochs. Before and after this

underfitting and overfitting takes place respectively. During training, a batch-size of 8 and the Adam optimizer have been used with a learning rate of 0.001 with 5,92,300 trainable parameters which is only one-third of the existing GR-ConvNet. The significant reduction in trainable parameters makes our model especially attractive and inexpensive in terms of training time. Therefore, this lightweight feature suggests that our proposed GI-NNet could be a suitable training model for closed-loop control applications in real-time robot grasp executions.

5.4 RGI-NNet training details

The training approach employed for the RGI-NNet model is primarily to train the VQ-VAE model to learn meaningful latent embedding for the input images. The entire CGD RGB images are fed in the unlabelled form to train the VQ-VAE model. Once the VQ-VAE is trained, we use its Encoder and Quantization layers to initialize our supervised learning based model, GI-NNet. Subsequently, Decoder parameters are re-initialized for training the grasp prediction model where GI-NNet is trained with various labelled data split ratios such as 0.1, 0.3, 0.5, 0.7, and 0.9, (here split represents a fraction of labelled data used out of the total labelled and unlabelled data for training) and the training performance of the network has been presented with Fig. 6 which shows RGI-NNet performs better with split ratio 0.5 onwards, since both the training and validation losses start stabilizing between 20-25 epochs from this split ratio. The input is fed in the batches of 8 to the respective layers of VQ-VAE and the Decoder of VQ-VAE then passes the output to GI-NNet which finally produces an optimal grasp rectangle. This model enables the grasp prediction with limited labelled training data. However, as shown in Table 2, the split ratio 0.5 gives the maximum accuracy and hence we have selected this model for real time experiments. In the subsequent sections we discuss the loss function and the activation functions we have used for training our networks.

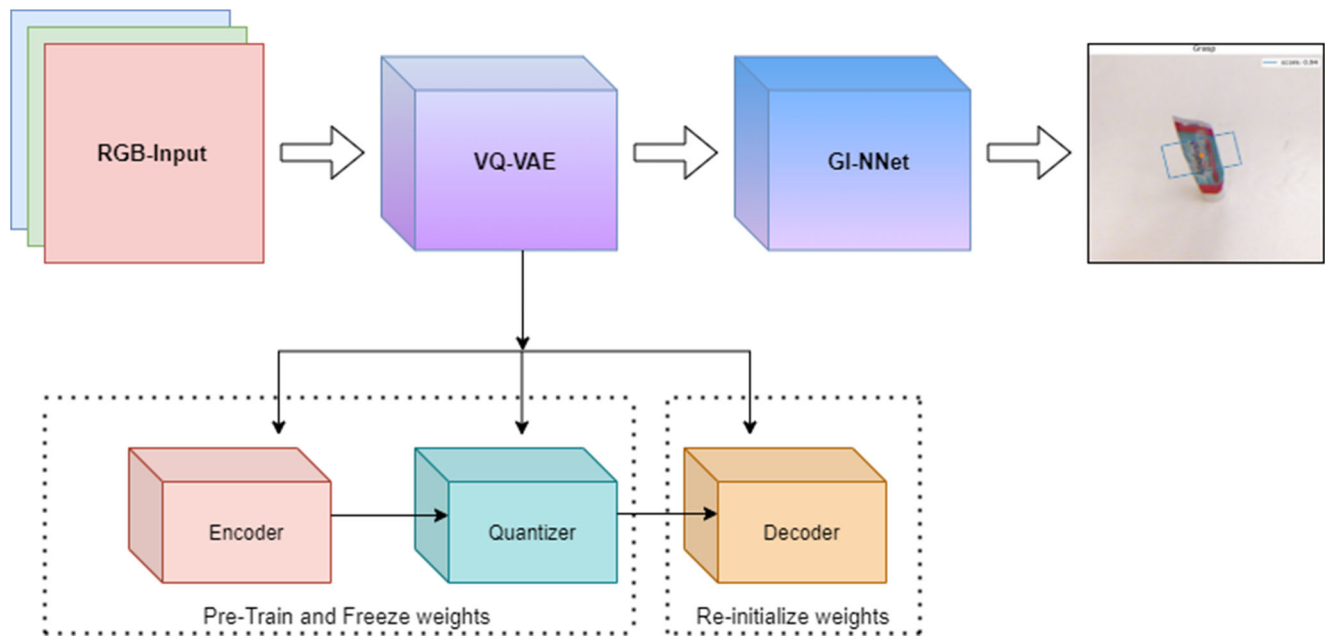


Fig. 4 Architecture of RGI-NNet model

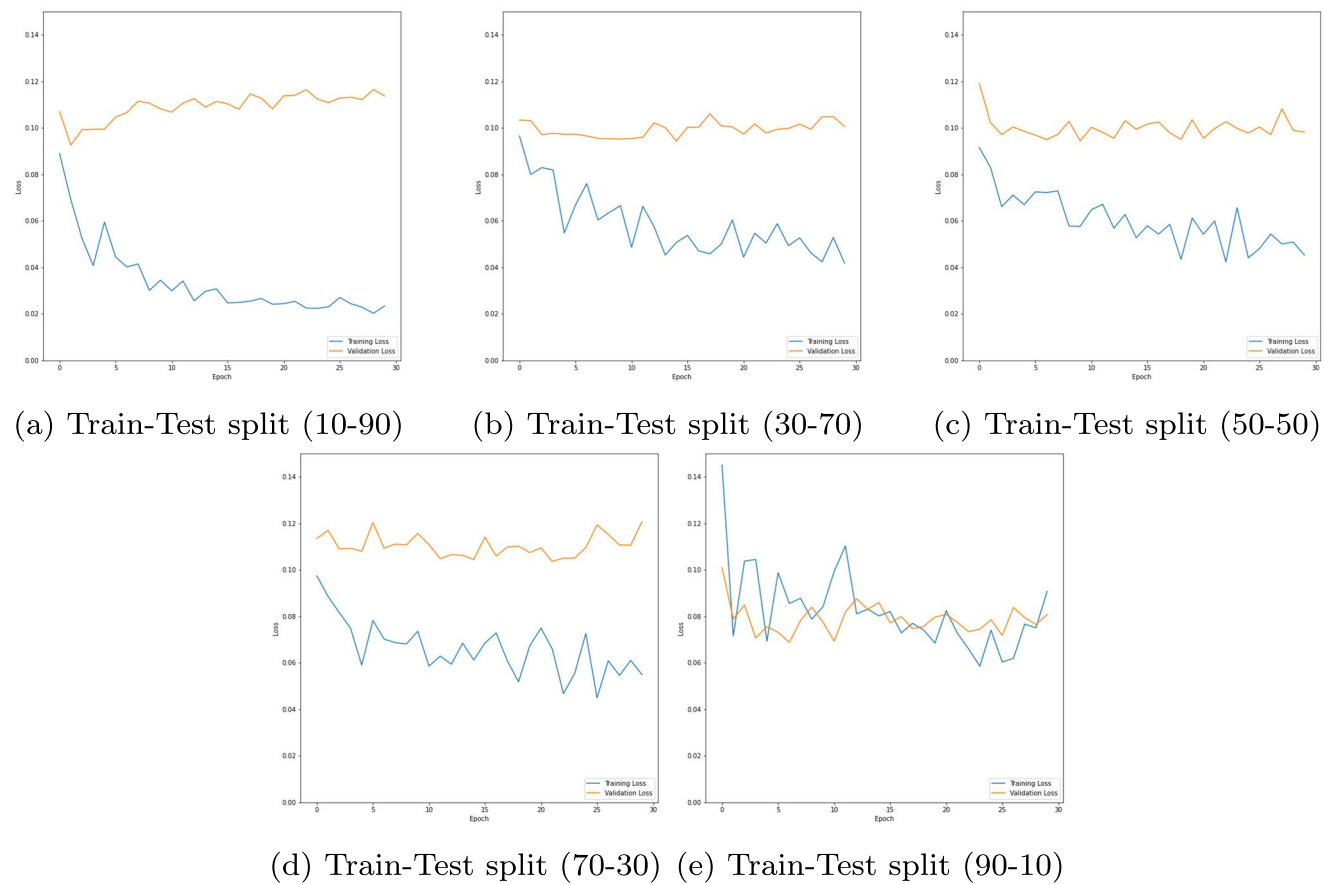


Fig. 5 GI-NNet performance validation with various Train-Test splits

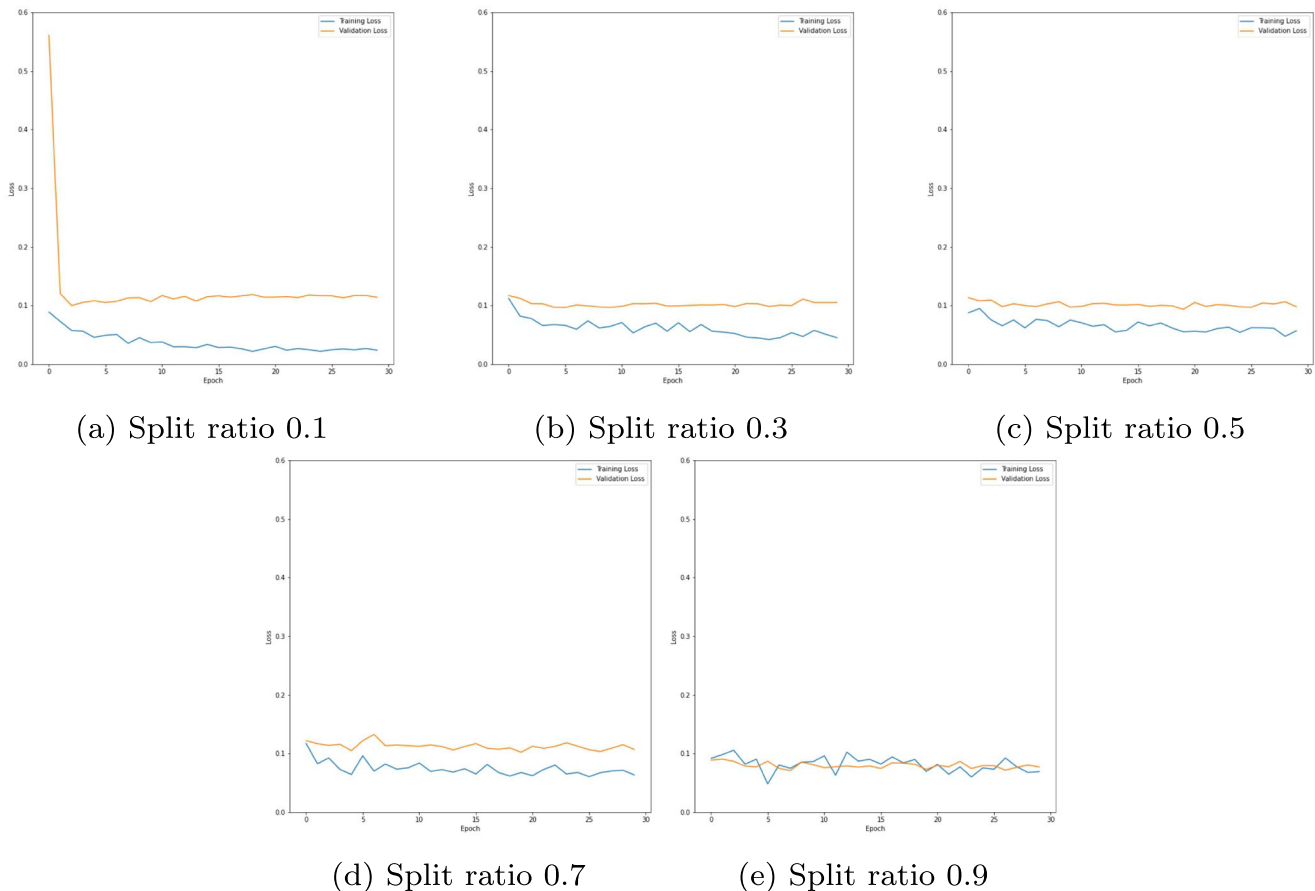


Fig. 6 RGI-NNet performance validation with various split ratios

5.5 Loss function

During experiments, Huber loss [11] i.e the smooth L1 loss has been used which shows the best results, since it negotiates between absolute and square loss function. The expression for loss function is given by (8) and (9) where \widehat{G} represents the predicted grasp and G represents the ground truth grasp.

$$L(G_i, \widehat{G}_i) = \frac{1}{n} \sum_k^k z_k \quad (8)$$

$$z_k = \begin{cases} 0.5 (G_{i_k} - \widehat{G}_{i_k})^2, & \text{if } |G_{i_k} - \widehat{G}_{i_k}| < 1 \\ |G_{i_k} - \widehat{G}_{i_k}| - 0.5 & \text{otherwise} \end{cases} \quad (9)$$

5.6 Activation function

Initially, GI-NNet is tested with two activation functions-Leaky ReLU and Parametric ReLU with gradient values of 0.1 and 0.001 respectively. It has been found that ReLU activation with a learning rate of 0.001 provides the most stable results and hence we have used this one only for our network training purpose. However, in the final output

layers three different activation functions-Sigmoid, Tanh and linear have been used, where Sigmoid with its two asymptotes at 0 and 1 represents quality of the output image with ground truth pixel sets to 1 and other pixels set to 0, Tanh with its two asymptotes at +1 and -1 represents the angle image with colour gradient values ranging from -1.5 to +1.5 measured in radian and for generating image output representing width, we have tried several activation functions but the linear activation function turns out to be the most suitable one. Finally, the obtained pose in image configuration space needs to be converted into the robot configuration space for real time grasp execution which is equally challenging as has been discussed in the subsequent sections.

6 Robotic grasp pose generation and execution

The grasp generation performance of our proposed models, GI-NNet and RGI-NNet, are also being evaluated practically by executing them in real time on Anukul research robot by designing an experimental setup as discussed in

Section 7.1. To execute a grasp on a physical robot, we need to map the generated grasp pose into the robot's configuration space. Subsequently, invoking the inverse kinematics and trajectory planning modules, robot gripper is able to reach the desired pose (position and orientation) of the grasping rectangles to grasp the object physically as detailed below.

Initially, generated grasp rectangle in an image coordinate frame is obtained by using model GI-NNet or RGI-NNet. To execute it on a physical robot, the generated grasp pose is transformed from 2-D image frame to 3-D coordinates frame of camera and then followed by camera frame to robot frame transformation as shown in (3). Primarily, external camera device has been calibrated and camera calibration matrix (K) has been obtained [36] which has been shown in (10) where f_a , f_b , c_a , c_b are intrinsic parameters representing focal length and the offset of the principal point respectively. Subsequently, obtained grasp pose in an image coordinates (x , y) frame has been transformed to the camera coordinates (u , v , w) frame by using (11), (12) and (13) respectively.

$$K = \begin{bmatrix} f_a & 0 & c_a \\ 0 & f_b & c_b \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$u = ((x - c_a)/(f_a)) \times depth[x][y] \quad (11)$$

$$v = ((y - c_b)/(f_b)) \times depth[x][y] \quad (12)$$

$$w = depth[x][y] \quad (13)$$

The obtained grasp pose in camera coordinates is further mapped to robot configuration space by using ${}^R T_C$ which is obtained by a camera to robot mapping using [33] and which finally calculates the joint angles for the generated robotic grasp pose (G_R) to execute the grasp. All the details related to experimental setup and models' performance evaluation have been discussed in the subsequent sections.

7 Experimental results and evaluation

7.1 Experimental setup design

Our experimental setup consists of a Baxter (Anukul) research cobot (collaborative robot), an externally mounted stereo camera (Intel realsense D435), with high resolution, required for getting quality images that will be compatible with the training images used for training our architectures. The designated Robot hardware has been used for

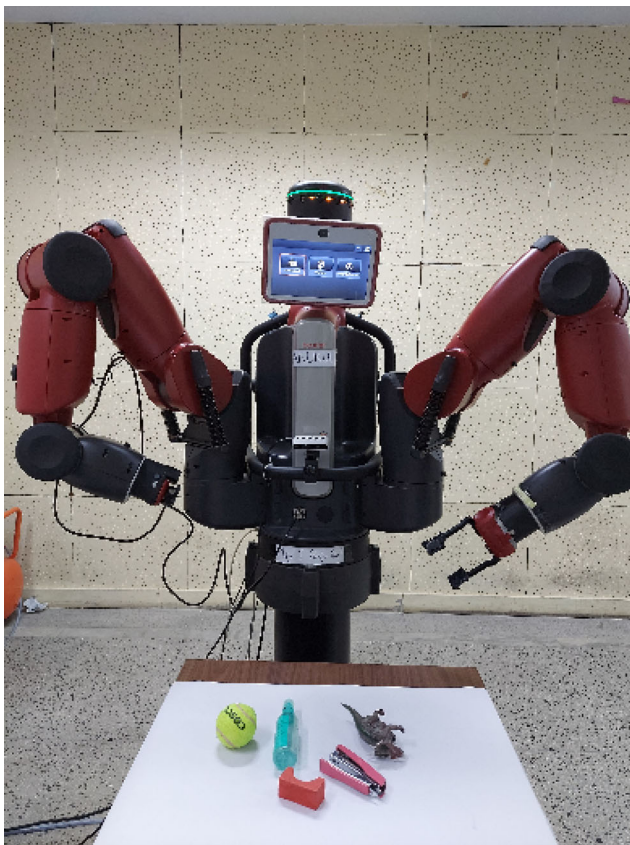


Fig. 7 Experimental setup



Fig. 8 Tested object's sample

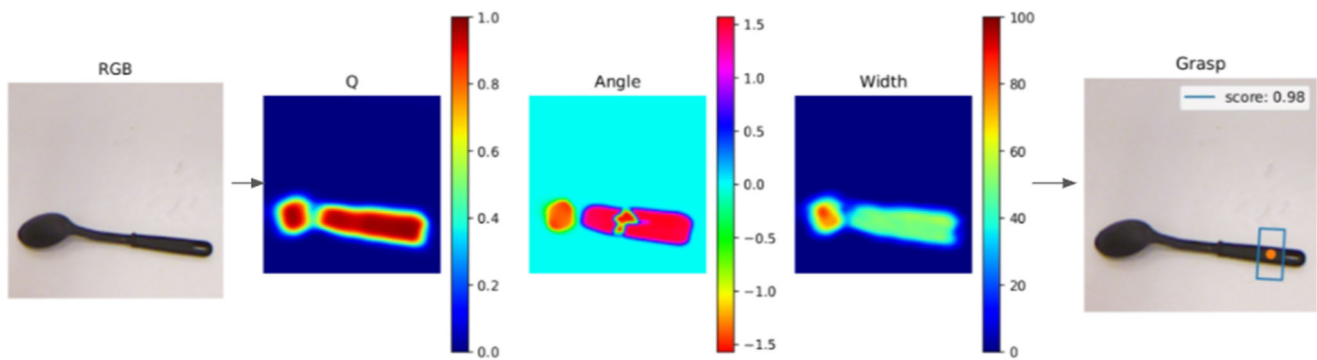


Fig. 9 Pose prediction with GI-NNet model

verification of the grasp pose prediction with real-time grasp execution. The experimental setup for table-top grasping has been shown in Fig. 7. Anukul has three internal cameras, two on both the arm's EE and one on top of the head with a maximum resolution of 1280×800 pixels incorporating 30 Frame Per Second (FPS) and focal length of 1.2 mm. However, for grasp manipulation, object depth is required and hence we had to mount an additional camera, as mentioned above, on the robot torso, with much higher resolutions [12] with appropriate calibrations so as to get the maximum workspace visibility on the table-top. We have used Baxter Software Development Kit (SDK) which is inbuilt in the Robot Operating System (ROS) environment for computation of inverse kinematics, trajectory planning, with appropriate resolve rate control, the functions for which have been accessed and our program has been composed using Python language. Though the robot model has both electrical and vacuum grippers, for the experiments the left arm with an electrical gripper has been used. However, without loss of generality the other arm and the other grippers could also be used. For real-time grasp execution, regular/irregular shaped medium-sized objects have been used. Some sample objects have been presented in Fig. 8.

7.2 GI-NNet model evaluation

To validate the proposed model we have tested and compared it with the SOTA approaches along with the performance analyses on CGD. The proposed GI-NNet model is found to be robust, applicable for real time environment, and it generalizes well towards various geometrical shapes and sizes of the objects and thus making it suitable for the closed loop control of grasp execution. To test the generalization capability of the proposed model, GI-NNet is evaluated on the CGD which predicts optimal grasp pose on object images with the SOTA success rate of 98.87%. The output of GI-NNet is shown for an object image in Fig. 9. Here, the image frame grasp G_I is generated through the optimal grasp quality q_I image, the grasp angle for the grasping ψ_I image and the grasp width w_I image. The bounding rectangle on the object represents the predicted grasp pose. Apart from testing on CGD images, we evaluated the proposed model on some unseen object images which are further illustrated in Fig. 10. GI-NNet has also been compared with existing approaches for different splits of train-test sets. Table 1 shows that without VQ-VAE the performance for the GI-NNet is poorer for the limited label data. As the availability of labelled data increases, GI-NNet

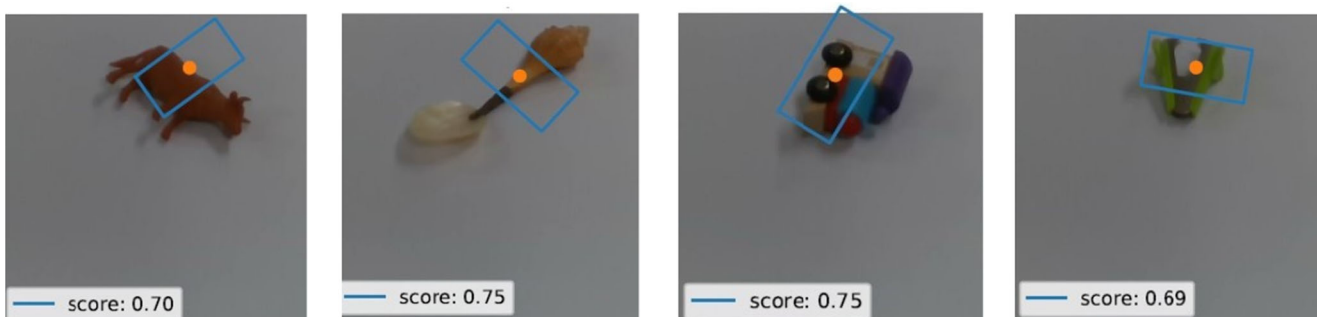


Fig. 10 Performance of GI-NNet on unseen object images

Table 1 GI-NNet comparative study with existing approaches

Train-test split	GGCNN [22]	GR-ConvNet [17]	Ours (GI-NNet)
10-90	76.4040	87.6404	89.8876
30-70	80.8989	89.8876	95.5056
50-50	82.0225	96.6292	96.6292
70-30	87.5025	95.5056	98.8764
90-10	95.5056	97.7528	98.8764

outperforms RGI-NNet (refer Table 2 for RGI-NNet). This implied the significance of our research. If we have an abandoned source of the labelled dataset, then one should use our proposed GI-NNet architecture. Moreover, GI-NNet attains such performance incorporating total trainable model parameters of 5,92,300 which is much lesser in comparison to the other related models which have been discussed in upcoming subsections.

7.3 RGI-NNet model evaluation

RGI-NNet is trained on CGD with split ratio of 0.1, 0.3, 0.5, 0.7, and 0.9, and the performance of each trained model has been presented in the Table 2 which clearly shows that the model provides stable grasping performance though trained on very limited amount of labelled data. Moreover, it attains a significant accuracy of 92.1348% with 10% of labelled data only. Our proposed model, RGI-NNet, is able to achieve such significant improvements in results due to the more powerful decoder training with our proposed, GI-NNet, model. Thus the proposed RGI-NNet is only suitable when we have a scarcity of label data, which is the case for the robot grasping paradigm, and in such cases, the RGI-NNet architecture can be used as a de-facto architecture for intelligent robot grasping. Apart from this, we have also verified its performance for generating an optimal grasp on seen as well as unseen objects.

Table 2 RGI-NNet performance comparison with different split ratios

Split Ratio	Mahajan et al. [20]	Ours (RGI-NNet)
0.1	85.3933	92.1348
0.3	87.6404	96.6292
0.5	89.8876	97.7528
0.7	89.8876	94.3820
0.9	89.8876	95.5056

Table 3 Performance comparison

Model	Accuracy (%)
Fast Search [13]	60.5
GGCNN [22]	73.0
SAE, struct. reg. [19]	73.9
Two-stage closed-loop [38]	85.3
AlexNet, MultiGrasp [25]	88.0
STEM-CaRFs [3]	88.2
GRPN [14]	88.7
ResNet-50x2 [18]	89.2
GraspNet [4]	90.2
ZF-net [8]	93.2
FCGN, ResNet-101 [41]	97.7
GR-ConvNet [17]	97.7
GI-NNet	98.87

7.4 Comparative studies with the existing SOTA approaches

The cross-validation approach as mentioned in various research works [17, 19, 25] and [3], we also have tried to follow the similar approach of image-based data splits. Our model has achieved an accuracy of 98.87% on image-based split as depicted in the Table 3, which compares the success rate of our model with that of the SOTA models. GI-NNet outperforms other enlisted approaches as mentioned in the Table 3. Our inception module based network predicts optimal grasps for various kinds of geometrically distinct objects from the validation set.

Utilizing the data augmentation approach during the training of the network on the CGD, the success rate in generating grasp poses of the GI-NNet model is improved. Moreover, such an improved performance has been observed with much reduced trainable parameters as mentioned in the Table 4.

Such drastic reduction in trainable parameters makes the GI-NNet far more efficient, fast and computationally inexpensive. Thus this model can be deployed in closed loop control on a robot grasp execution in real time applications. We also make a comparative analysis on the performance of GR-ConvNet and GI-NNet models on predicting an

Table 4 Comparison of model parameters

Model	Parameters
GR-ConvNet [17]	1,900,900
GI-NNet	592,300


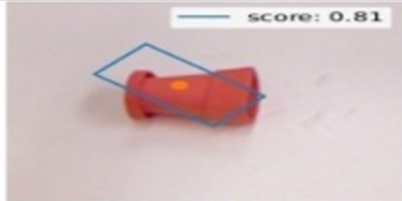
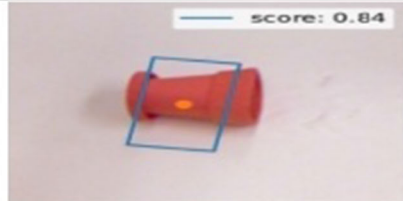

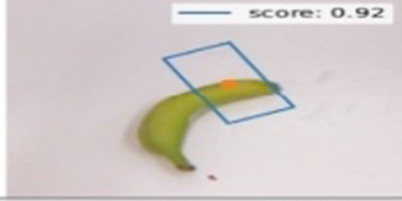
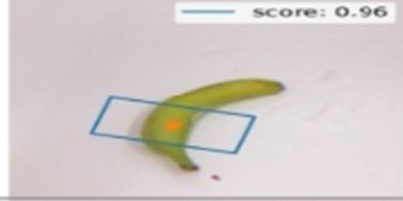



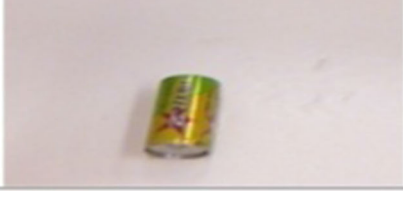


Input Image	Performance of GR-ConvNet	Performance of GI-NNNet
	 score: 0.81	 score: 0.84
	 score: 0.92	 score: 0.96
	 score: 0.92	 score: 0.96
	 score: 0.94	 score: 0.97

Fig. 11 Performance comparison between GR-ConvNet and GI-NNNet

optimal grasp for object images from CGD as shown in the Fig. 11. Subsequently, the performance of RGI-NNNet has been compared with Mahajan et al. [20] and it is being observed that the performance is highly improved with limited labelled data. From the above mentioned studies, we can claim that GI-NNNet is comparatively a much better model to generate optimal grasp pose for the intelligent robot grasping applications even when we have sufficient labelled data for training, whereas RGI-NNNet has the ability to predict optimal grasp pose when we have limited labelled data for training, making it an attractive model for intelligent robot grasping.

8 Conclusions and recommendations for future research

In the present investigation, we have presented two new grasping models, GI-NNNet and RGI-NNNet, where GI-NNNet is capable of generating optimal grasp rectangle in a superior way at the pixel level for the object in a scene with sufficient labelled data and RGI-NNNet model is able to predict an optimal grasp with limited labelled data. Both the models are fast enough to deploy in a closed loop robot

grasp execution control owing to much better performance for the grasp inference. During experiments, we have achieved an improvement in predicting grasp poses with more efficiency for seen as well as unseen objects. With rigorous experimentation we have confirmed that when there is sufficient labelled data for training then the GI-NNNet model may perform better but when there is limited labelled data available for training then the performance of our proposed RGI-NNNet model is much better, perhaps due to the inherent capability of VQ-VAE of generating data suitable for training the network parameters. Currently, models are trained and tested only on CGD images but the same models can be trained on other available labelled datasets also to check for the performance in future. We can also use our proposed models to bring agility by creating sufficient labelled data for the category of novel objects which could make the training more efficient. In the present investigation we have used only table top grasping for experimental verification of grasp efficiency which may be generalised by extending it to 6-D pose estimation in future.

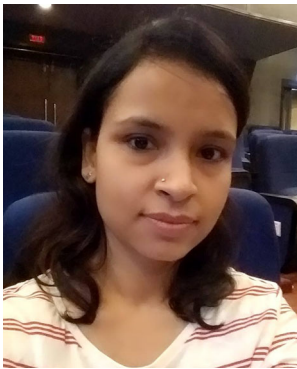
Acknowledgements The present research is partially funded by the I-Hub foundation for Cobotics (Technology Innovation Hub of IIT-Delhi set up by the Department of Science and Technology, Govt. of India).

References

- Agarap AF (2018) Deep learning using rectified linear units (relu). arXiv:1803.08375
- Ahmed MU, Brickman S, Dengg A, Fasth N, Mihajlovic M, Norman J (2020) A machine learning approach to classify pedestrians' event based on imu and gps. *Int J Artif Intell* 16(2). <http://www.es.mdh.se/publications/5255>
- Asif U, Tang J, Harrer S (2018) Ensemblenet: Improving grasp detection using an ensemble of convolutional neural networks. In: *BMVC*. p 10
- Asif U, Tang J, Harrer S (2018) Graspnet: An efficient convolutional neural network for real-time grasp detection for low-powered devices. In: *IJCAI*. vol 7, pp 4875–4882
- Bicchi A, Kumar V (2000) Robotic grasping and contact: A review. In: *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. symposium proceedings (Cat. No. 00CH37065)*, vol 1. IEEE, pp 348–353
- Bohg J, Morales A, Asfour T, Kragic D (2014) Data-driven grasp synthesis—a survey. *IEEE Trans Robot* 30(2):289–309. <https://doi.org/10.1109/TRO.2013.2289018>
- Goodfellow IJ, Bengio Y, Courville A (2016) *Deep Learning*. MIT Press, Cambridge. <http://www.deeplearningbook.org>
- Guo D, Sun F, Liu H, Kong T, Fang B, Xi N (2017) A hybrid deep architecture for robotic grasp detection. In: 2017 IEEE International conference on robotics and automation (ICRA). IEEE, pp 1609–1614
- He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE international conference on computer vision*. pp 1026–1034
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp 770–778
- Huber PJ (2011) *Robust statistics*. Springer, Berlin, pp 1248–1251. https://doi.org/10.1007/978-3-642-04898-2_594
- Intel: Intel realsense - d435. Available online: <https://www.intelrealsense.com/depth-camera-d435/>
- Jiang Y, Moseson S, Saxena A (2011) Efficient grasping from rgbd images: Learning using a new rectangle representation. In: 2011 IEEE International conference on robotics and automation. IEEE, pp 3304–3311
- Karaoguz H, Jensfelt P (2019) Object detection approach for robot grasp detection. In: 2019 International conference on robotics and automation (ICRA). IEEE, pp 4953–4959
- Kopicki M, Detry R, Adjigble M, Stolkin R, Leonardis A, Wyatt JL (2016) One-shot learning and generation of dexterous grasps for novel objects. *Int J Robot Res* 35(8):959–976
- Kragic D, Christensen HI (2003) Robust visual servoing. *Int J Robot Res* 22(10–11):923–939
- Kumra S, Joshi S, Sahin F (2020) Antipodal robotic grasping using generative residual convolutional neural network. In: 2020 IEEE/RSJ International conference on intelligent robots and systems (IROS). IEEE
- Kumra S, Kanan C (2017) Robotic grasp detection using deep convolutional neural networks. In: 2017 IEEE/RSJ International conference on intelligent robots and systems (IROS). IEEE, pp 769–776
- Lenz I, Lee H, Saxena A (2015) Deep learning for detecting robotic grasps. *Int J Robot Res* 34(4–5):705–724
- Mahajan M, Bhattacharjee T, Krishnan A, Shukla P, Nandi GC (2020) Robotic grasp detection by learning representation in a vector quantized manifold. In: 2020 International conference on signal processing and communications (SPCOM). pp 1–5. <https://doi.org/10.1109/SPCOM50965.2020.9179578>
- Maitin-Shepard J, Cusumano-Towner M, Lei J, Abbeel P (2010) Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In: 2010 IEEE International conference on robotics and automation. IEEE, pp 2308–2315
- Morrison D, Corke P, Leitner J (2018) Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. In: *Proc. of robotics: science and systems (RSS)*
- Pinto L, Gupta A (2016) Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In: 2016 IEEE international conference on robotics and automation (ICRA). IEEE, pp 3406–3413
- Precup RE, Teban TA, Albu A, Borlea AB, Zamfirache IA, Petriu EM (2020) Evolving fuzzy models for prosthetic hand myoelectric-based control. *IEEE Trans Instrum Meas* 69(7):4625–4636. <https://doi.org/10.1109/TIM.2020.2983531>
- Redmon J, Angelova A (2015) Real-time grasp detection using convolutional neural networks. In: 2015 IEEE International conference on robotics and automation (ICRA). IEEE, pp 1316–1322
- Ritter H, Haschke R (2015) Hands, dexterity, and the brain. In: Cheng G PhD (ed) *Humanoid robotics and neuroscience: science, engineering and society*. Boca Raton (FL): CRC Press/Taylor & Francis. Chapter 3. <https://www.ncbi.nlm.nih.gov/books/NBK299038/>
- Sahbani A, El-Khoury S, Bidaud P (2012) An overview of 3d object grasp synthesis algorithms. *Robot Auton Syst* 60(3):326–336
- Satish V, Mahler J, Goldberg K (2019) On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks. *IEEE Robot Autom Lett* 4(2):1357–1364
- Saxena A, Driemeyer J, Ng AY (2008) Robotic grasping of novel objects using vision. *Int J Robot Res* 27(2):157–173
- Schmidt P, Vahrenkamp N, Wächter M., Asfour T (2018) Grasping of unknown objects using deep convolutional neural networks based on depth images. In: 2018 IEEE international conference on robotics and automation (ICRA). IEEE, pp 6831–6838
- Shimoga KB (1996) Robot grasp synthesis algorithms: A survey. *Int J Robot Res* 15(3):230–266
- Shukla P, Kumar H, Nandi G (2021) Robotic grasp manipulation using evolutionary computing and deep reinforcement learning. *Intell Serv Robot* 1–17
- Strobl KH, Hirzinger G (2006) Optimal hand-eye calibration. In: 2006 IEEE/RSJ International conference on intelligent robots and systems. pp 4647–4653. <https://doi.org/10.1109/IROS.2006.282250>
- Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp 1–9
- Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*
- Telljohann A (2017) *Introduction to Building a Machine Vision Inspection*, chap. 2. Wiley, New York, pp 31–61. <https://doi.org/10.1002/9783527413409.ch2>. <https://onlinelibrary.wiley.com/doi/abs/10.1002/9783527413409.ch2>
- Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: A neural image caption generator. In: *Proceedings of the*

- IEEE conference on computer vision and pattern recognition. pp 3156–3164
38. Wang Z, Li Z, Wang B, Liu H (2016) Robot grasp detection using multimodal deep convolutional neural networks. *Adv Mech Eng* 8(9):1687814016668077
 39. Zeng A, Song S, Yu KT, Donlon E, Hogan FR, Bauza M, Ma D, Taylor O, Liu M, Romo E et al (2018) Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In: 2018 IEEE international conference on robotics and automation (ICRA). IEEE, pp 3750–3757
 40. Zhang C, Bengio S, Hardt M, Recht B, Vinyals O (2016) Understanding deep learning requires rethinking generalization. [arXiv:1611.03530](https://arxiv.org/abs/1611.03530)
 41. Zhou X, Lan X, Zhang H, Tian Z, Zhang Y, Zheng N (2018) Fully convolutional grasp detection network with oriented anchor box. In: 2018 IEEE/RSJ International conference on intelligent robots and systems (IROS). IEEE, pp 7223–7230

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Priya Shukla is a senior research scholar pursuing an M.Tech-Ph.D dual degree at the Center of Intelligent Robotics, Indian Institute of Information Technology Allahabad. She has completed her three years diploma in Information Technology followed by B.Tech in Computer Science. She has worked as a software developer at MIND, Noida. Her research interests include Computer Vision, Machine Learning, Deep Learning, Reinforcement

Learning, and Robotic Manipulation.



Nilotpal Pramanik received his bachelor's and master's dual degree in information technology (specialization in robotics) from the Indian Institute of Information Technology, Allahabad, India, in 2021. His area of research interests includes machine learning, artificial intelligence, and robotics.



Deepesh Mehta is currently pursuing his Bachelor's in Electrical and Electronics Engineering from Guru Gobind Singh Indraprastha University. He is passionate about research in Computer Vision, Deep Learning, Image Processing, and Model Optimization and Robustness. He has been an intern at the Indian Institute of Information Technology Allahabad.



G. C. Nandi graduated from the Indian Institute of Engineering Science & Technology (Formerly known as Bengal Engineering College, Shibpur) in 1984. He did his post-graduation from Jadavpur University, Calcutta in 1986. He obtained his Ph.D. degree from the Russian Academy of Sciences, Moscow in 1992. He was a visiting research scientist at the Chinese University of Hong Kong during 1997 and visiting faculty at Institute for

Software Research, Carnegie Mellon University (CMU), USA during 2010-2011. Currently, he is working as the senior-most Professor (HAG) and Head of the Center of Intelligent Robotics. He is a Senior Member of the ACM and IEEE. He has published more than 200 papers in referred journals and international conferences. His research interests include Humanoid Robotics, Machine Learning, and Deep Learning.