ISTANBUL TECHNICAL UNIVERSITY
FACULTY OF COMPUTER AND INFORMATICS

IMPLEMENTING DEEP LEARNING METHODS FOR NOVEL OBJECT GRASPING

Graduation Project Final Report

Burak Mete
150140131

Department: Computer Engineering
Division: Computer Engineering

Advisor: Assoc. Prof. Dr. Sanem Sarıel

May 2019

Statement of Authenticity

We hereby declare that in this study

1. all the content influenced from external references are cited clearly and in detail,
2. and all the remaining sections, especially the theoretical studies and implemented software/hardware that constitute the fundamental essence of this study is originated by my/our individual authenticity.

Istanbul, 24.05.2019

Burak Mete

IMPLEMENTING DEEP LEARNING METHODS FOR NOVEL OBJECT GRASPING

(SUMMARY)

Grasping is a very essential utility to attain for a general purpose robot to be part of daily tasks that can aid humans. Moreover, along with the recent developments in both hardware and software in the domains of robotic and machine learning, building robotic grasping subsystems has become one of the important open research areas. As in most of the computer science topics, researches try to implement deep learning solutions for various tasks in robotics, mainly for activity recognition, object detection and tracking, and also for humanoid action planning, such as grasping. Along with the physical constraints, lack of feasible training data has been one of the most crucial constraints for a system to learn accurately. Therefore gathering or synthesizing meaningful data has become an important aspect to build intelligent humanoid robots as well.

In our work, the main focus is to create an intelligent system for robotic grasps. Grasping robotic systems have been performing accurately in the industrial field, due to working within a more static environment where external factors that affect robots visual sensors, like illumination conditions or surrounding objects, are mostly stationary and does not change drastically. However, this conditions do not apply for the daily life tasks all the time, where external factors are constantly changing. Another obstacle to build a well performing grasp subsystem arises due to over fitting, which is caused by the lack of training data that generalizes large range of objects. In order to overcome this issues, a better learning model is needed along with a large dataset. Thus, with this work, we are aiming to build a more robust system that performs better with novel objects, in different environmental settings.

The problem of robotic grasp can be divided into three fundamental sub-problems, perception(detection), planning and control. Our focus in this project is in the perception part, mainly how the robot understands its surroundings, detecting the object and a grasping position on the object. In order to overcome this problem, a broad dataset is used to train the model that robot can learn through. Since creating data for this purpose is cumbersome, synthetically created dataset is used with large variety of object so that model do not over fit to some limited types of objects.

There are several different approaches to create a solution for these problems presented in the earlier work, however they are significantly different from each other such same data set cannot be feed into the model since they accept different features and evaluate different metrics. In this work, Convolutional Neural Network (CNN) is used for regression to predict the optimal grasping region from scratch. Data provided to the network is consisted of multimodal RGB-D images which are provided by the Jacquard Dataset[1], and model gives an output with a five-element representation of grasping area (x, y, h, w, θ) with elements being center coordinates, height, width and orientation.

Since there are multiple ground truth values for each simulated object in the training data, the loss function determines the closest positive annotation to the model output, and tries to optimize itself to different possible valid grasping regions at each iteration.

This approach avoids the model to over fit to a single grasping positioning and works with a better accuracy in overall.The architecture of the final model will be quite similar to ResNet[2]., since the model itself proven to be efficient and has a very high capability to extract useful information from images.

To sum up, our deep learning based grasping prediction subsystem, provides the regression outputs to the grasping robot, while being trained by a very generalized synthetic dataset which includes multiple labels. Moreover the model is trained by a IoU loss function which is not very common to use in deep learning systems, and we are proposing that it would be more feasible to use it as a loss function for the grasp planning subsystem in this project.

ÖĞRENİLMEMİŞ NESNE KAVRAMA PROBLEMİNDE DERİN ÖĞRENME YÖNTEMLERİ

(ÖZET)

Kavrama yetisi, günlük hayatta insan görevlerine yardım edecek robotlar için edinilecek çok temel bir görevdir. Makine öğrenme ve robotik alanları, yazılım ve donanım alanındaki hızlı gelişimler ile birlikte popülerlik kazandıkça, günlük hayatımızda kullanılabilecek, insan görevlerine yardımcı olan robotların kavrama sistemlerini tasarlamak ve programlamak açık bir araştırma alanı haline gelmiştir. Günümüzde, bilgisayar biliminin konu edindiği birçok farklı görevde olduğu gibi, araştırmacılar robotik konu başlığı altında, obje tanıma ve takip etme ve robotik kavrama planlayıcısı gibi insansı görevleri başarmak amacıyla derin öğrenme metotları kullanarak daha gelişmiş sonuçlar almaya çalışmaktalar. Fiziksel kısıtların yanısıra, öğrenmenin en temel unsurlarından biri olan iyi eğitim verisi eksikliği, geçmişte robotik kavrama görevinin öğrenilmesinde engel teşkil etmiştir, bu sebeple, daha genel kapsamlı eğitim verisi üretmek ve toplamak da önemli bir konu haline gelmiştir.

Bu çalışmanın temel fikri, gündelik objelerin nasıl kavranacağını öğrenen akıllı bir sistem inşa etmektir. Daha durağan ve aynı koşullara bağlı ortamda çalışa geldiği endüstriyel robotlarda robotik kavrama konusu bugüne kadar daha çok gerçekleştirilmiş ve bu görev için robotlar kullanılmıştır. Ancak, günlük hayatta ise robotun görüş alanındaki nesnelerin değişimi ve ışıklandırmanın devamlı değişmesi sebebiyle oluşan dinamik ortam, günlük hayattaki robotlarda bu konunun ürünleştirilmesinde önemli bir engel teşkil etmektedir. Bu konudaki bir diğer engel ise, eğitim verisinin yeterince genel olmamasından kaynaklanmakta ve bu sebeple sistem yalnızca eğitilirken beslendiği nesneleri öğrenmeye yakınsamaktadır. Bu problem yüzünden, robot daha önce hiç görmediği objeleri kavrayacağı noktayı planlamakta güçlük çekebilir. Bu sebeple, bu projenin diğer amaçlarından biri de robotun daha önce görmediği objeleri bile tutuş pozisyonlarını iyi tahminleyebileceği genel bir kavrayıcı sistem inşa etmektir.

Robotik kavrama problemi algılama, planlama ve kontrol etme olarak üç temel alt problem bölünebilir. Bu projenin odaklandığı kısım ise algılanma kısmıdır. Temel olarak robotun çevre faktörlerini anlaması, objeyi tanıması ve bu obje üzerinde başarılı olacak bir kavrama pozisyonlaması seçmesi üzerinedir. Bu problemi aşmak için eğiteceğimiz modelin başarılı olabilmesi için yeterince geniş bir veri kümesine ihtiyaç duyulmaktadır; fakat tekil bir amaç için görsel veri kümesi üretmek birçok açıdan zor bir iştir. Aynı zamanda hazırlanan bu veri kümesinin, birbirinden oldukça farklı objeleri içeriyor oluşu, modelin yalnızca belirli tipteki nesnelere eğilimli olarak çalışmasının önüne geçmekte önemli bir adımdır.

Daha önceki çalışmalarda bu problemin önüne geçmek için sunulan birkaç farklı yöntem olsa dahi, aynı veri kümesini kabul etmemeleri, veriden farklı özellikler çıkarımlamaları ve farklı hesaplama metrikleri kullanma bakımından farklılık teşkil ederler. Bu çalışmada, başlangıçta model üzerinde yapılacak denemeler için RGB ve derinlik bilgisine sahip resimleri içeren Jacquard Veri Seti [1] kullanılacaktır. Aynı zamanda model, kavrama merkezi, en, boy ve yönelme bilgilerini içeren 5 parametreli (x, y, h, w, θ) tutuş gösterimini çıktı katmanında tahmin olarak verecek şekilde tasarlanacaktır.

Veri setinde, birden fazla doğru olarak etiketlenmiş tutuş pozisyonu olduğu için, gradyant hesabı yapılırken tahmine en yakın etiket seçilmekte, bu işlem ise IoU metriği ile hesaplanmaktadır. Bu sayede her iterasyonda model kendini daha farklı doğru tutuş pozisyonlarına uyacak şekilde eğitebilmekte, ve sistemin yalnızca bir doğruya eğilimlenmesinin önüne geçilmektedir. Kullanılan model, ResNet [2] temel alınarak yapılmış, kendi veri setimize uyacak şekilde son katmanları düzenlenmiştir. Gradyant hesabı yapılırken regresyon kullanılmaktadır.

Sonuç olarak, derin öğrenme tabanlı kavrama pozisyonu tahminleyici sistemimiz, regresyon parametrelerini robota sağlamakta, bunu yaparken geniş ve birden fazla doğru etiket içeren bir veri seti kullanmakta, öğrenme işlemini ise gradyant hesabı yapılırken kullanımı pek yaygın olmayan, fakat bu çalışmada kullanılması önerilen metrik olan IoU metriği ile yapmaktadır.

Table of Contents

List of Figures

List of Tables

1 Introduction and Problem Definition

1.1 Purpose and Context of the Project

As mentioned before, robotic grasping is a challenging and a trending topic in research field, due to recent improvements. Recently researches achieved somewhat better performances for grasp learning methods with different approaches. Nevertheless, most of the projects are restricted in a very limited scope in terms of actions and objects involving in the task, and do not give high accuracies for novel objects or different environmental conditions. Thus, the purpose of the project is to implement an intelligent grasping subsystem that performs with a better accuracy. The implementation of the system includes the architecture of the modal, gathering and creating a synthetic dataset that will increase the learning performance for various objects and implementing new loss function that will enhance the overall performance of the learning procedure of grasp prediction system. Hence the project includes the network model, creating a dataset with automatized generation having large number of positive ground truth labels, and presenting the novel loss function for grasping networks. The purpose of this report is to present the results of our model with different evaluation criteria, and share more insights about the problems to be overcame for grasping subsystem.

1.2 Motivation

As we will be mentioning in further detail, there are few fundamental approaches to build a grasp prediction system. However, there are still some problems to those approaches. One of the most important problem is the optimization of the model with respect to the ground truth labels in the training data. While providing the training data, since there could be more than one optimal grasping region, there are numerous positively annotated labels. One important decision in the model optimization is whether to train the model with all possible outcomes, or selecting the closest ground truth with respect to the model output. Since first method results in prediction to be converged to the centre of all possible labels that wrongly predict, especially for the circular shaped objects, second method regresses to only one positive label which may cause over fitting for that specific type of objects with the use of L2 norm for the loss function of the model. Our motivation in this work, is to build a loss function that will outperform the previous ones that can overcome the mentioned issues while predicting grasps. Thereby, in this project we are presenting a new loss function for our network that can generalize a grasping position better when there is more than one optimal grasping position for an object, which is nearly the case all the time. Furthermore, another motivation behind our project include creating a synthetic dataset with objects related to the actions that are implemented for Baxter in our Artificial Intelligence and Robotics Laboratory. Those actions contain mostly picking, placing, pouring and placing for cooking related actions in general, that robot can help perform. Moreover, As Caldera et. al. mentioned, most of the approaches regarding to grasp detection still require a human expertise both for creating the modal and the dataset to get efficient results[3]. With both of these ideas behind our motivation for this work, what we are

trying to overcome is to eliminate the analytic thinking procedure and human intervention to the process as much as possible while developing an autonomously working robot, to create a more general purpose grasp subsystem.

1.3 Project Summary

With this report, we are first telling a reader about the context, purpose and the motivation behind of our project. Then a brief background knowledge on Convolutional Neural Networks and the transfer learning, followed by a comparative literature survey to mention about important methods used in past for dataset creation and implementing the model distinctively. For the next section, methods and experiment details will be mentioned in great detail. Furthermore, detailed approached will be covered while presenting an in-depth explanation about the system model in overall. Then, a comparative evaluation and discussion will be presented for the results obtained for our model with different evaluation criteria. Then, the report will be concluded with final comments, conclusions and the research challenges about future work.

2 Comparative Literature Survey

Since the important aspects for this project are the modal, methods and the dataset, researches are examined through three main topics, background information about important concepts, modal and method research and dataset research in the grasping prediction topic.

2.1 Background Knowledge

2.1.1 Convolutional Neural Network

Convolutional Neural Network (CNN), is a specific type of an Artificial Neural Networks, which stems its name from the most fundamental operation used within the network, which is the convolution operation. They are mostly used in tasks related to image recognition tasks due to the nature of the convolution operation. Convolution is the operation that takes two signals and outputs the third function that represents how one input gets affected from the other. In images, which are basically signals in spatial domain, convolution operation can represent the relationship among the pixels within the image, which is called local connectivity. Therefore, those information about the relationships can be used to extract the details about the whole image. Since mostly the local relationships within the image are necessary to extract information about the objects in the scene, convolution layers are better to use than fully connected layers while working with visual input data, since fully connected layers include the relationship with all of the possible pixel pairs. An exemplary illustration of convolutional neural networks can be seen in the Figure 2.1

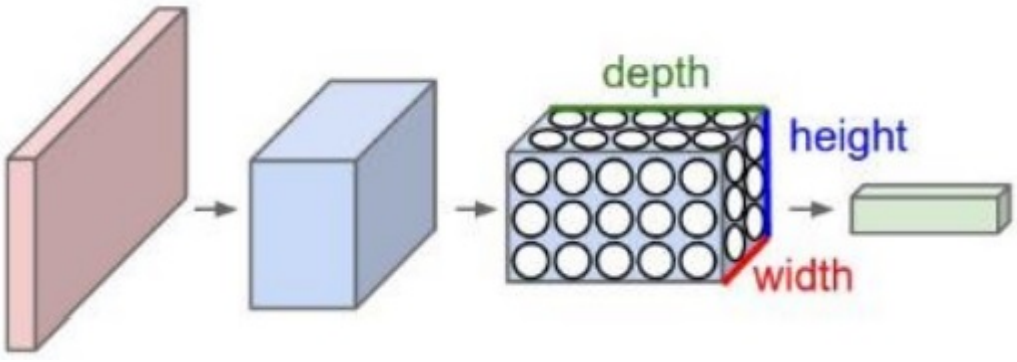

Figure 2.1: Overview of Convolutional Neural Network Architecture [4]

One of the main differences of CNN's from the other artificial neural networks is having a 3D volumes of neurons, since the input is a 3D signal, which is just an image with height, weight and depth values. Originally, depth can refer to number of channels in the input image. As the network goes into higher levels, the size of the image shrinks while the depth grows. At the end of the network, the data would be reduced to a single vector of scalar values, namely the class prediction values or regression parameters that can be used for computing the loss.

The most important and frequently used elements in CNN layers are mainly, Convolutional Layer, Pooling Layer, Activation Layer and a Fully Connected Layer, that can be seen in the Figure 2.2. In the convolutional layer, the hyper parameter to be optimized is the receptive field which affects the spatial extent of connectivity in the data. Pooling layer is the down sampling layer to reduce the number of parameters to be optimized in the network which is also used to control over fitting. Activation layer consists of mostly a ReLu function that can bring non-linearity to the model, and lastly fully connected layer is a layer that is just a matrix multiplication with a bias of the previous layer's neurons.

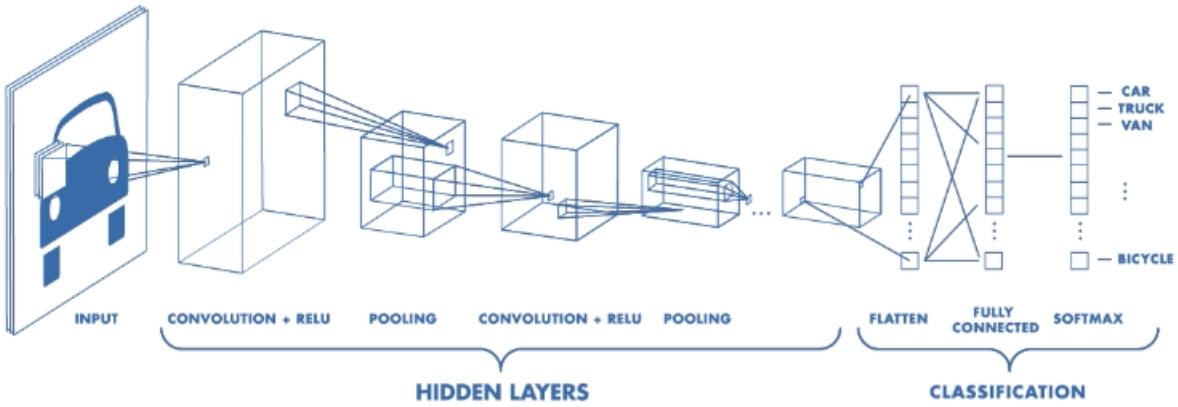

Figure 2.2: CNN Layers [5]

2.1.2 Transfer Learning

While building a CNN modal, whether the problem is to predict the object within the image or find grasping region for a robot, there needs to be several information to be extracted in common in both of these tasks regarding images. Moreover, several common architectures are proven to be extracting those information quite well, regardless of the problem. In deep learning, those information are named as "features". Thus, while solving in a large variety of problems, it might be wiser to use the pre trained weights of those models. This approach is called transfer learning. After those features are extracted, the final layers are formed for a model to have an understanding about its specific task, therefore the weights of this final layers should be calculated from scratch.

There are two main approaches to transfer learning, which are feature extraction and finetuning. Even though, pre trained values have meaningful features in its hidden layers for a general problems related to image recognition, as the scope of the problem changes, the features needed to be extracted to network to understand the problem as a whole changes as well. In that case, all of the parameters of the network should be calculated again with respect to the gradients, starting from the pre trained values, this approach is called fine-tuning. Whereas in the feature extraction, only the output

layer parameters are updated, hence as the name implies, the features that the hidden layer extracts stays the same throughout the training phase. While building the model, it is important to select which approach applies better for the regarding problem.

2.2 Dataset Research

As we discussed earlier, selecting the training dataset is one of the crucial factors of the modals overall accuracy. The modal should be generic enough such that the modal does not over fit to objects. Therefore, it is wise to train the modal with the dataset that are proven to be give high training accuracies, however, as the scope of the problem changes, the modal may need to be trained with its own specific data related with its task. Hence, in some problems it is very crucial to gather and provide task related data by generating a new dataset.

The most challenging part of generating a dataset is annotating, meaning if a selected grasp is successful or not. As Depierre et al. [6] suggests here are three main types of data creation in terms of creation phase, human labelling which relies on physical trials with the robot, analytic computation and physics simulation. Human labelling is a process, where for each individual visual data, which might be a real image or a simulated image, an expert is responsible to annotate the image, by manually selecting the grasping points and applying physical trials whether the selected grasp is successful or not. One problem may arise with human labelling is that it is very time consuming, since an expert needs to select the candidate positions and wait for the robot to execute grasps.

The ones that are created via a simulating environment are called as synthetic dataset and they are less time consuming then human labelling approach. One important aspect of creating a synthetic dataset is to avoid teaching the model only the grasps that are constructed with human comprehension. Since parallel grippers functioning different than the human hand, there may be a better grasping positions for the robot than what a human perceives as a good grasping position. Furthermore, since the creation would be much more efficient, more objects with different poses and background textures can be included to the dataset, which will yield a more generalized robotic grasping system.

Dataset	Number of objects	Modality	Number of images	Multiple gripper sizes	Multiple grasps per image	Grasp location	Number of grasps	Automatized generation
Levine et al. [4]	-	RGB-D	800k	No	No	Yes	800k	No
Mahler et al. [2]	1500	Depth	6.7M	No	No	No	6.7M	Yes
Cornell	240	RGB-D	1035	Yes	Yes	Yes	8019	No
Jacquard (ours)	11k	RGB-D	54k	Yes	Yes	Yes	1.1M	Yes

Figure 2.3: Comparision Among Different Datasets Prior to Different Aspects [7]

Most of the datasets which gives high precision for grasping assignment, distinguish among each according to certain type of features. One important feature is the input type that is used, most of the modals use multimodal RGB-D, whereas some modals used only the depth map. Automatized generation is also a crucial factor, due to the reasons listed above regarding to the drawbacks of human labelling. A comprehensive

comparison of the most common datasets used can be seen in the Figure 2.3

Another feature that distinguish the common dataset that highly affects the learning procedure is the representation of the grasp annotation. There are several methods to provide annotations, which are going to be the parameters that will be provided to the robot to execute the grasping action, once the model is ready to be used. While some of the methods using point clouds as input, creates just a point to represent a graph, such as the representation proposed by Saxena et.al [8], with 2 degrees of freedom by only having x and y coordinates, most of the recent work uses representations with higher degrees of freedom which formulates the output as a grasping rectangle. An exemplary grasping rectangle provided for both RGB and depth image data can be seen at Figure 2.4

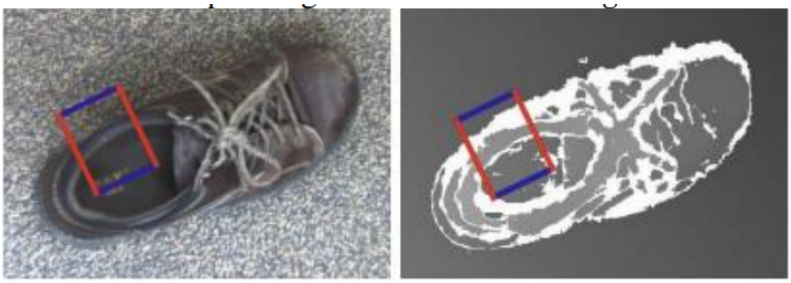

Figure 2.4: Grasp Representation for RGB and Depth Image Data [9]

While some methods uses 3 degrees of freedom to represent a grasping rectangle, by having x, y and , namely the centre coordinates and the orientation of the rectangle around the x axis, this approach fails to work on robots with different gripper jaw sizes and different sizes of objects to be grasped, since the jaw opening for a gripper should be arranged as well. For this reason, Jiang et.al, proposes a new approach, by proposing a representation with 7 degrees of freedom, with 3D location and orientation information, gripper size and the jaw opening length [9]. However, for the methods that are using only RGB values, the depth location and the depth orientation is not very useful, since, 5 degrees is enough to be used in those models.

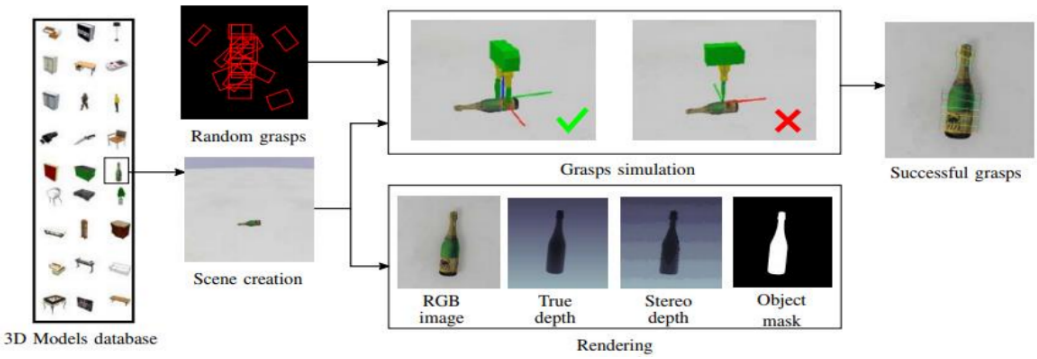

Figure 2.5: Creation Process of Jacquard Dataset [1]

To sum up, for the project, the Jacquard Dataset, [10] whose creation process can be seen in the Figure 2.5, is chosen, for the training of our model and also performance comparison at the conclusion of the work as a whole with respect to our dataset which is used for testing and validation. Input of this model is constructed with only the

ground truth 2D image data, which is then applies to a pipelined architecture for one part to create grasps and annotates the positive grasp locations via simulator, and the other part creates a depth image mapping that can be used later for object detection. One important aspect of this dataset is the making use of important heuristics for several important sub-tasks, such as parallel edge finding, or cancelling the candidates which are quite similar to each other. Furthermore, since the dataset is quite vast, with more than 10k objects and 50k images, which includes 1 million unique successful annotations at total.

2.3 Method Research

2.3.1 Modal Research

As Depierre et al. stated [11] improvement of the general Neural Network architectures and particularly in the Convolutional Neural Network (CNN), since it is very efficient in the visual data computation, empowers the researchers to train models with ground-truth visual data rather than computer aided graphics. Some models in earlier work, tries to find a grasping parameters for a given input image by doing regression at the output layer, whereas in other work creates possible candidates for a good grasps and tries to find a probabilistic measure for a selected candidate, as their comparative overall architecture can be seen in the Figure 2.6.

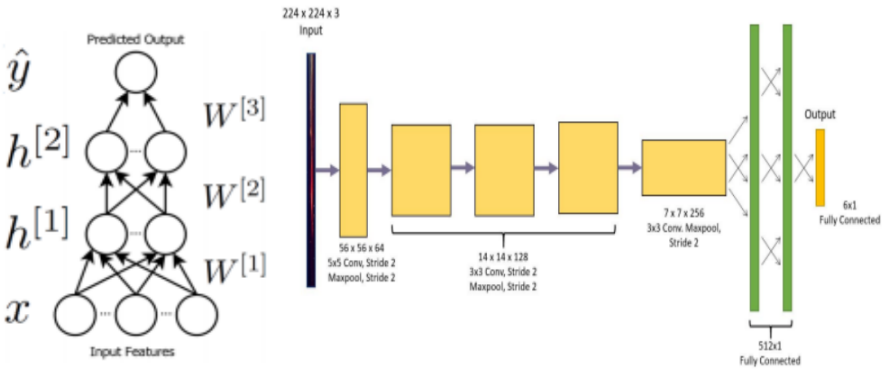

Figure 2.6: Differences Between Prediction and Regression Networks [7]

Since a probability measure necessitates an earlier candidate grasp region to be created and fed into the network, there should be another network that generate meaningful and intelligent grasps. However, this modal does not need to be very complex and learns all important aspects for a given data. Therefore less detailed modal with less computations can be used for that purpose. In one of the landmark work that deals with this approach shown in the Figure 2.7., Lenz et al. proposes [7] to use a cascaded system, the first one creates a random sliding windows with different height, width and orientations, which is going to be used to create favourable candidates, via a Sparse Auto-Encoder.

For transfer learning, most of the recent work AlexNet architecture created by Krizhevsky [12] is used as a reference model, and most of them are built on top of the AlexNet

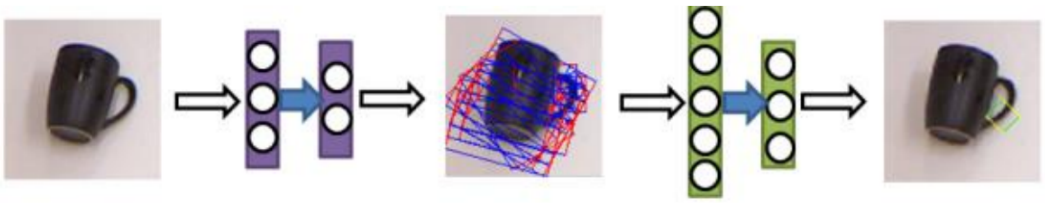

Figure 2.7: Cascaded System for Candidate Creation and Selection[7]

Network, by applying small differences in convolution parameters, such as padding, stride, along with the final layers of the network according to the output of the model Furthermore Pinto [13] suggests another architectural style. In that work, AlexNet is used as a base point as the model structure can be seen in the Figure 2.8., and the modal is changed such that it has 18 classifiers at the final layer. 18 classifiers are equal to the division of the orientation space where each classifier is separated from the subsequent classes by 10 degrees. Therefore, the modal gives a probabilistic measure for the orientations, along with the coordinates of randomly created patches.

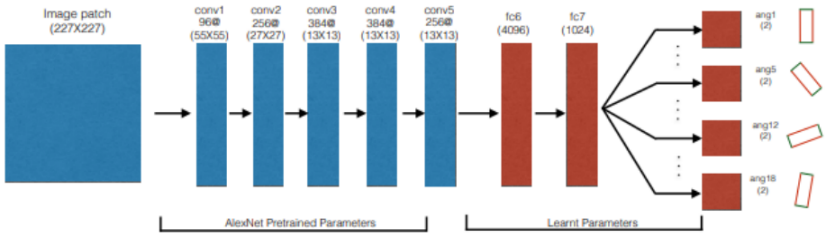

Figure 2.8: CNN with Angle Estimation for Grasping[13]

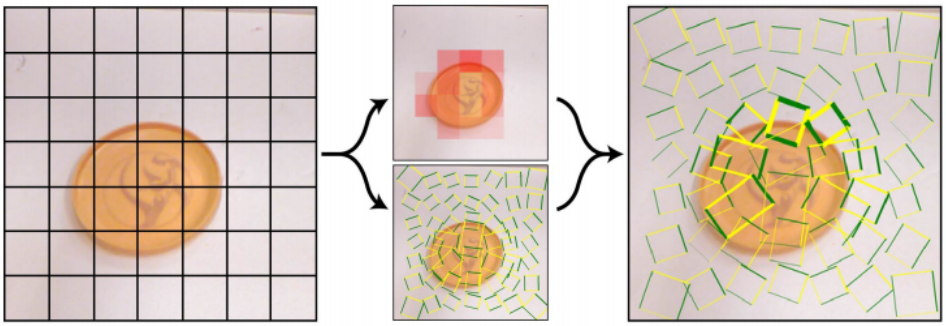

Figure 2.9: Candidate Creation Within Each Patch[13]

There is also an interesting approach by creating candidate grasps, suggested by Redmon [14], which is simply dividing the ground-truth image into several grids, as shown in Figure 2.9, and with a sub-model finding the best looking grasp within those each individual small grids. This process helps the evaluating model to work with a better precision. In this work, an object heat map is also found by using specific heuristics, to eliminate non-object area patches

2.3.2 Evaluation Criteria

Even though, until recently the Rectangle Metric was being used for the testing phase, which relies on several calculations. Basically, rectangle metric method uses the distance between the prediction and the ground truth label. When there are more than one ground-truth labels, the prediction is measured according to the closest label. The comparison might be done with several evaluation metric, such as L2 norm, or Intersection over Union. However, from a recent work Depierre [15] suggests another evaluation metric which is called Simulated Grasp Trials (SGT). With this approach, the prediction is always going to be simulated or applied with a real robot trials and the results are calculated accordingly to the outcome of the grasping action, whether object is successfully grabbed or not.

3 Developed Approach and System Model

Our experiment environment divides into two sections. Physical trials with real grasping robots are done through the industrial robot Baxter, in ITU Artificial Intelligence and Robotics Laboratory, as the experimental environment with the grasped objects can be seen at Figure 3.1

Figure 3.1: Experiment Environment with Baxter

There are various parameters to be selected for the dataset, and also for the modal. Those parameters are, the dataset type, data modality, grasp representation, the base modal, finetuned layers, loss function, optimizer, learning rate, number of epochs and evaluation criteria.

- Dataset Type: Jacquard Dataset [1] , with 10k different objects, 50k images and 1M positive annotations
- Data Modality: RGB
- Grasp Representation: Rectangle representation with 5-DoF with parameters (x, y, h, w, θ), namely as center coordinates, height, width and the orientation of the rectangle respectively

- Model: Pre-trained Resnet [2]
- Fine-tuned Layers: Final fully-connected layers are trained to optimize the result
- Loss Function: Intersection over Union metric are used to calculate the gradients, which is a novel aspect of this project.
- Optimizer: Stochastic Gradient Descent (SGD)
- Number of epochs: 200
- Batch Size : 4
- Learning Rate: 0.001
- Evaluation Criteria: Intersection over Union

4 Experimentation Environment and Experiment Design

The project is developed in Python, and the essential libraries used are pyTorch [16] , numPy [17] , shapely [18] and matplotlib [19]. The network modal is implemented in pyTorch with pretrained ResNet architecture. Moreover, pyTorch's autograd feature has helped us to build our custom functions, since the gradients would be calculated by the library itself through the autograd structure. Numpy is used for numerical operations, and some basic operations in matrices. Shapely is used to calculate intersection over union metric for evaluating the models accuracy. Matplotlib is used for visualizing the grasp representation and also for plotting confusion matrices and accuracy graphs.

4.1 Dataset

Since we are trying to optimize the grasping accuracy in different environmental conditions and large variety of objects, it would be very beneficial to use a large dataset with various objects, and also multiple grasping points. Another aspect that can be important for our modal is a dataset to be created in automatized fashion. As mentioned in the Section 2.2, automatized creation for a synthetic dataset, can improve the learning procedure since the modal do not rely only on human comprehension to execute successful grasps. Considering all of the facts and conditions listed above, we have decided to use Jacquard Dataset [1] , due to its vastness, automatized creation, and proven to give high accuracies for grasping tasks. Jacquard Dataset is created over 10k simulated objects with nearly 5 different poses for each object. There are multiple positive annotations that represent the grasping position, which are varying between 10 and 100 for each pose. The images are created via physics simulator pyBullet[20] , from the images taken from ShapeNet [21]. Each pose is generated by simply dropping an object from a fixed height, which yields a random pose. Then, random grasps are attempted with different gripper jaw sizes and opening lengths, whereas only successful grasps are selected as labels for that corresponding pose. Depth maps for each pose are extracted from the environment and added to the dataset, hence, it is up to the implementer to select the modality as RGB or RGB-D for a specified modal. In order to increase the accuracies of random grasps trials, object detection heuristics are used, with the help of depth maps.

Furthermore, our objects at ITU Artificial Intelligence and Robotics Laboratory will be used with physical trials with Baxter for the test set. Those objects, which are gathered such that they can be used in kitchen related actions like cooking and cleaning, can be considered as novel objects, since they are not included in the training dataset. Moreover they are real objects, as contrary to the train dataset which include only simulated images. An exemplary setup of grasping with real objects via our robot Baxter, can be seen in Figure 4.1. The accuracies are calculated in the training and validation sets with respect to the Intersection over Union metric, which is evaluated by comparing the output with the closest annotation. Tests are conducted with real robot trials with Baxter, and the accuracies of them are calculated whether the robot

can successfully execute the grasping action or not.

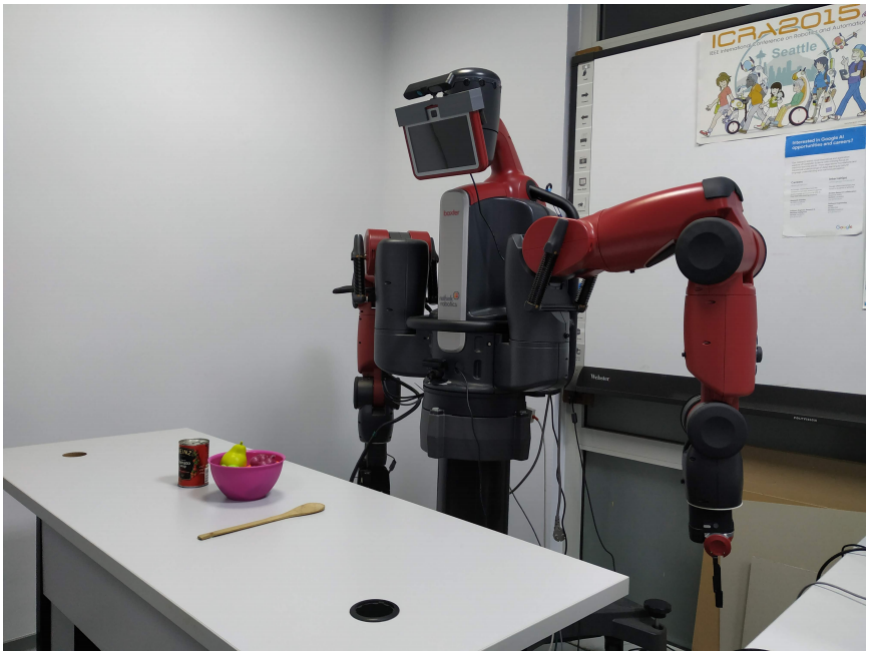

Figure 4.1: Baxter Grasping Setup for Novel Objects

4.2 Methods and Model

The most crucial aspects of this work along with the dataset being used is the methods used in implementation, namely as the network modal and the loss function used to calculate the gradients.

4.2.1 Network Model

Understanding and inferring information from images is quite a trivial task within the computer vision domain. With the recent developments in deep learning, better results have been gathered, since neural networks can extract relevant information from the spatial connectivity among the pixels of the image. However, as the task gets more complicated, such as the optimal grasping point detection for a novel object, it becomes harder to generate relation that represents a good solution to the problem, with the same number of operations. Therefore, as a rule of thumb principle, a convolutional neural network can said to be learning better as the number of hidden layers increase. However there are multiple problems to be solved, as the number of layers increases in a network. One important problem is the performance degradation, since number of operations increase as well as number of parameters to be updated. The other crucial problem arises from the increase in number of layers, as pointed out by Hochreiter, is vanishing gradient problem [22]. When updating the parameters in back propagation according to the layer, the effect of error gets decreased whenever the gradients are propagated towards the initial layers. Hence, while the final layers get updated rapidly, the initial layers changes very slowly, caused them to learn very slowly as

contrary to the further layers. Since the starting layers are quite important as they are responsible to extract the initial information from the image, vanishing gradient problem is a huge problem that slows down the learning phase quite drastically. Most of the recent architectures are aware of this problem, and uses ReLu function as a non-linearity operation rather than sigmoid or softmax operations, which causes the vanishing gradient problem for the convolutional neural networks. Hence, while we are selecting the candidate architectures for our modal, we have given importance to the two criteria, the learning performance and the overall network performance in training phase. With this purpose, we have selected some candidate modals that are proven to be working well for the image recognition tasks, since by intuition, features that have to be extracted for the object recognition tasks should be very close to the grasping point detection problems The modals that have been tried are AlexNet [12], ResNet18[2] and ResNet34[2] respectively. AlexNet is one of the most fundamental architectures for convolutional neural networks, which is composed by 5 sequential convolutional layers followed by 3 fully connected layers, the modal uses ReLu activation function rather than tanh or sigmoid, to solve the vanishing gradient problem that is discussed previously in this section. However the depth of the modal was not enough and the modal stop learning at some point while the accuracy stays nearly the same. To extract more relevant information, ResNet architecture is decided to be used later on. ResNet, where gets its name from Residual Networks, is a network proposed by Kaiming et.al. [2], that solves an important problem that is caused by the increasing number of layers just like in the vanishing gradient problem.

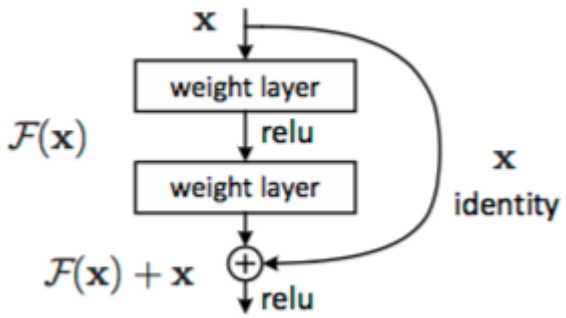

Figure 4.2: A Learning Block in Residual Network [2]

When there are more layers in the network, to optimize the overall learning process, higher training errors may be achieved while trying to optimize the parameters space, due to increasing number of operations. Thus, counter-intuitively, accuracy can decrease drastically when the number of layers increase. This problem is called degradation [23] and Residual networks are proposed to solve this problem. As it can be seen in the Figure 4.2, an identity mapping used in each residual block, to optimize the outcome when there are more layers stacked to the topology. The whole network is constructed by with this residual blocks, which each block of the network is consisted of. An exemplary block that is the initial layer of the whole network can be seen at Figure 4.3, which includes convolutional layer, batch normalization layer and an activation layer with a ReLu, except the final block that does not have a ReLu. Thus, the network has the same components as the AlexNet with 5 fundamental convolution layers, however, each layer is consisted of several residual blocks, as contrary to the non-residual architectures.

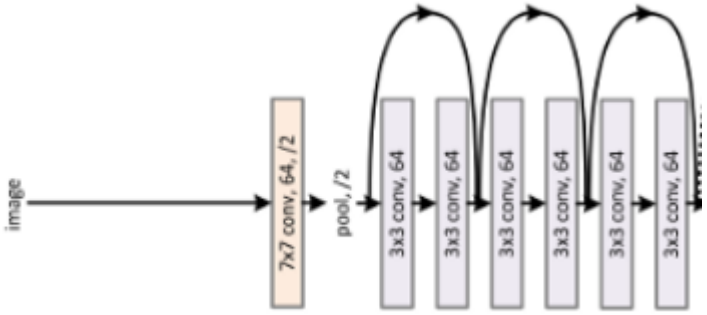

Figure 4.3: Initial Layer in ResNet Architecture [2]

Justified by those notions mentioned above, we have decided to use ResNet architecture for our model. The pyTorch implementation of ResNet [24] includes several different topologies that distinguish among each other according to the number of blocks used. The pertained modals are used as a reference, with only the last fully connected layers are changed such that the final block will produce 5 parameters, which will correspond to the 5 regression parameters for a grasping rectangle representation. The final topology of our network can be seen in the Figure 4.4.

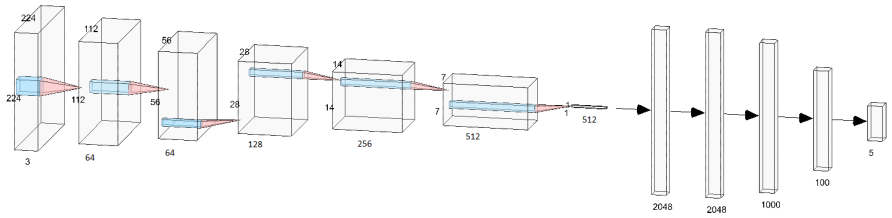

Figure 4.4: Architectural Overview of the Network Used

4.2.2 Loss Function

As Rezatofighi stated, recent work for computer vision related tasks such as robotic grasp predicting systems, tried to improve accuracy by enhancing the architectural style of the network, simply by adding layers or changing the approach for extracting features, but most of those work overshadowed the idea that changing the regression loss might cause a bigger leap in the overall performance. [25]

The most important aspect of this work, is the usage of the Intersection over Union (IoU) metric, and its comparison with widely used Mean Squared Error(MSE), which is a novelty aspect of this work whereas this metric has not been implemented for the loss function evaluation in the convolutional neural networks for the robotic grasping predictor systems. The intuition behind the idea why IoU Loss would be more efficient to use in grasp predictor systems is the mean squared error with the 5 Degrees of Freedom grasping representation(x, y, h, w, θ) does not generalize how the prediction is close to the annotation quite well. There might be several different approaches to use the MSE with 5-DoF grasping rectangle prediction, such as calculating the L2 norm for all of the corners of the grasping rectangle, or simply calculating the mean

squared error for each individual parameters, which are the centre coordinates, height, width and the orientation of the grasping region. However, there are several reasons of why using MSE does not give the best error calculation, which will be explained and justified in this section of this work.

Each different set of objects with particular shapes have their own individual grasping types. For instance, a circular shaped object have numerous symmetrical grasping positions, where the centre coordinates, height and width stays the same and only the orientation changes. If network predicts a grasp that has the same centre coordinates, height and width with different orientation, the MSE loss for all parameters would be quite small due to having the identical 4 predicted parameters. If the MSE loss is calculated according to the L2 norm of the corners of the grasping region, an unnecessary error can be calculated in several occasions. For instance when the predicted output have different scale than the annotation, where there are similar centre coordinate and orientation values and a different width and height values, even though, executing the predicted grasp would cause the same action with the annotation, the MSE metric gives high errors. An exemplary scenario for this case can be seen in the Figure 4.5, where yellow rectangle refers to the ground-truth value and the right rectangle refers to the prediction from the modal output when MSE is used.

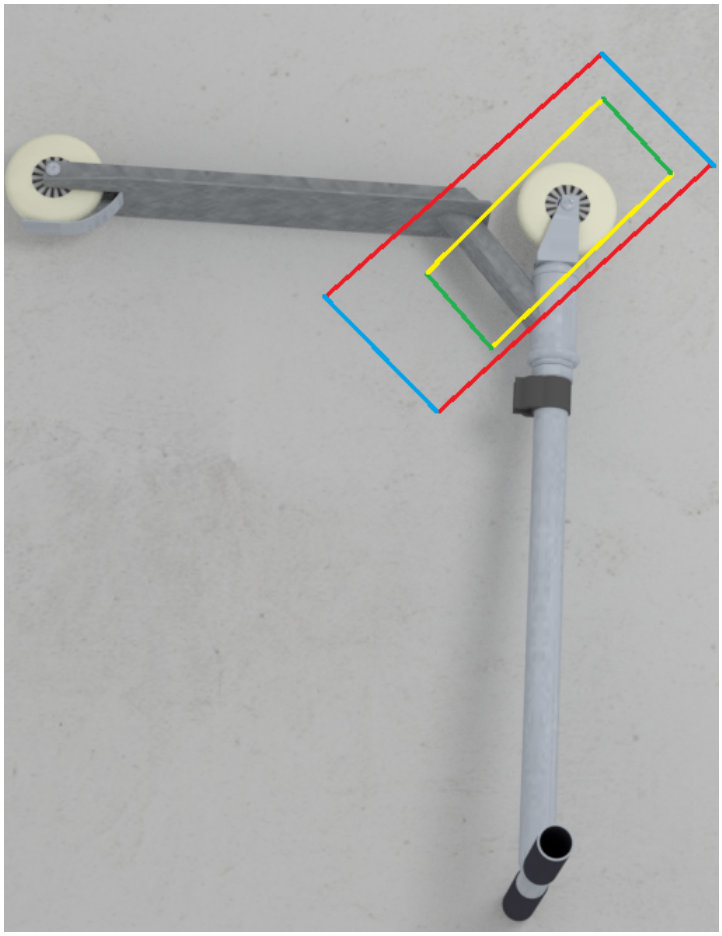

Figure 4.5: Erroneous Case When MSE is Used

s

In order to have a better generalization to the loss calculation to solve the mentioned

problems, we are proposing to use the Intersection over Union (Jaccard Loss). IoU is a common metric that is used in accuracy evaluations for object detection and several computer vision related problems, however, since the function itself is non-differentiable, it cannot be implemented to a neural network system to calculate the gradients.

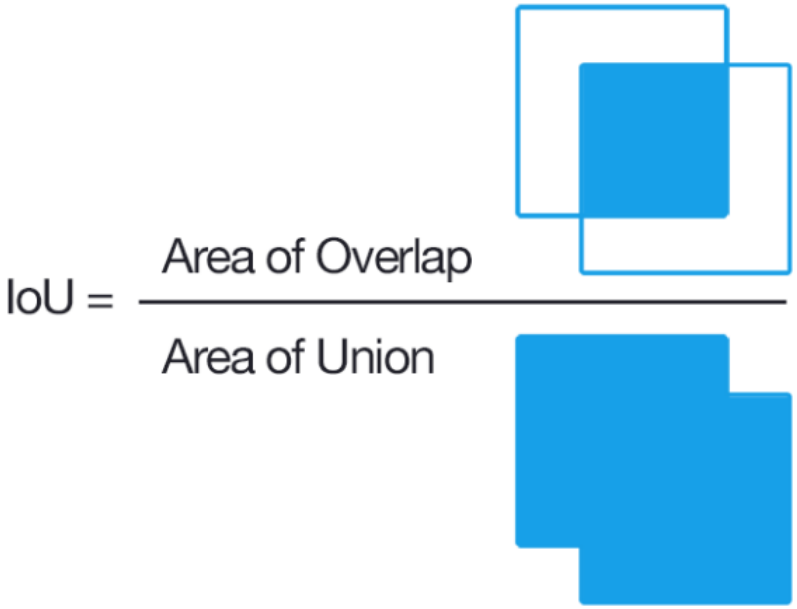

Figure 4.6: IoU Calculation [26]

Suppose there are two bounding boxes, one is the prediction whereas the other is the groundtruth label. The IoU metric can be calculated as the intersection area of those two bounding boxes, divided by the union area of them, as it can be seen in the Figure 4.6. Since the metric would be in the range of 0 and 1, while it yields 0 when the boxes are completely distinct and 1 when the ground truth label is a subset of the predicted bounding box. However, since we are trying to calculate a loss rather than an evaluation metric, the IoU should give the maximum value when the boxes are distinct and the minimum value when the prediction and the ground truth are completely identical. Thus, to represent the loss, the metric should be subtracted from 1, as it can be seen in the Equation 1

$$1 - \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

As Berman et.al, proposed, there are several different approaches to differentiate the IoU loss [27] , under the condition of those bounding boxes have class probabilities, however, it is a trivial task to differentiate the metric when the results are regression parameters, rather than class probabilities.

In order to solve this issue, we are proposing a two staged approach to come up with a better loss calculation that might be a better generalization for grasp prediction subsystem. Firstly, since most of the training data include multiple positive ground truth grasping positions, we have to decide which ground truth data should be used to calculate the loss. Redmon mentioned a problem arising from implementing direct

regression for a grasp prediction, to a multiple positively annotated training dataset in their work, in which the loss is calculated according to the average coordinates of all positive ground truth data that results in failed convergence for the predictions for circular shapes [28].

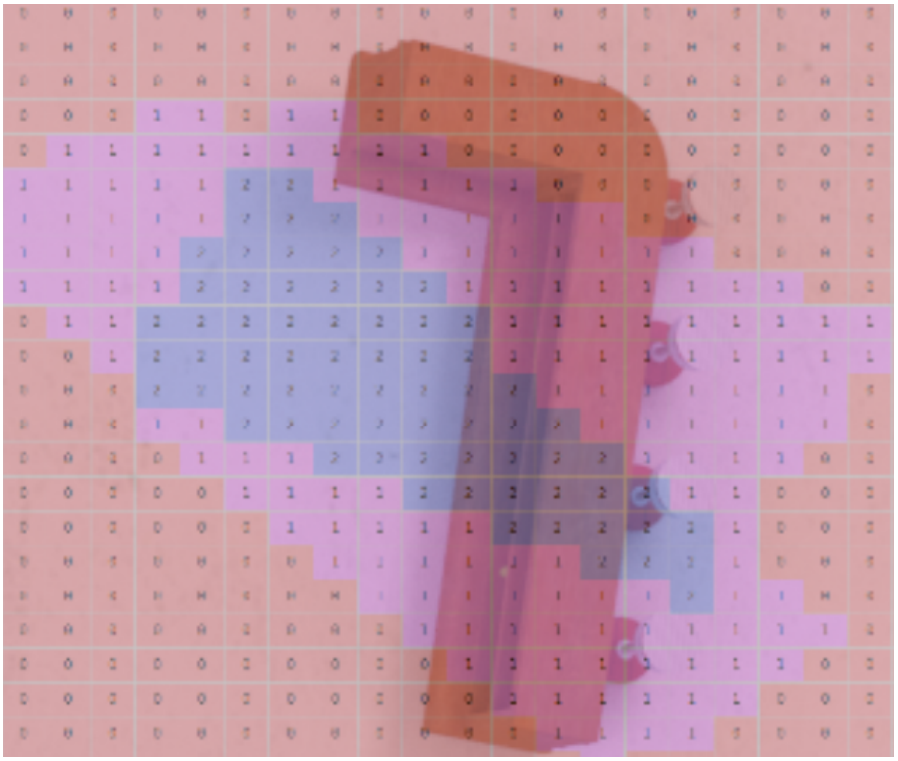

Figure 4.7: Cumulative Mask Image

Therefore we have tried another approach, to select the ground truth value which is the closest to the system prediction with respect to the IoU metric. We are implementing this phase outside of the network calculations such that it does not affect the gradient calculations. In order to create four vertexes from the 5 parameters, we are firstly creating a mesh, which is between 0 and 224 which are the minimum and the maximum indexes for the images in our dataset. Then, those mesh is multiplied with the image transformation matrix, created by the 5 parameters. Those parameters can be considered as affine parameters, such that x, y coordinates would be the amount of translation, height and weight would be the scale parameters whereas θ is the rotation parameter. According to the number of elements in our mesh, as many points as desired can be found, which will be calculated as the matrix multiplication between the image transformation matrix and the mesh coordinates. This procedure would be applied for both ground truth and the prediction to extract the vertex coordinates of the grasping rectangles. After the extraction of the vertex coordinates, a mask image is created, in which the values equal to 1 for the pixels within the rectangle, and 0 for others. We can create this matrix, since we have sufficient amount of coordinate values that are a part of the grasping rectangle, as we increase the number of elements in the mesh. Then, we can add the two mask images, to get a cumulative mask image, where the pixels with the value 2 indicate the intersection area whereas the pixels not having the value 0 indicates the union area. The cumulative mask image displayed on top of a grasping object, can be seen in the Figure 4.7, whereas blue points indicate the pixels

having the value of 2, meaning the intersection area, pink pixels having the value of 1, stating the are of union, except the intersection area.

If we subtract the cumulative mask image by one at every pixel, the sum of the resulting matrix would give us the number of units in the intersection area, and the sum of the original masked image, subtracted by the number of intersection, would give us the number of units in union area.

When this evaluation is applied for all of the ground truth labels in the training dataset, the label yielding the least loss in terms of intersection over union is selected, and the gradients are calculated according to that specific label. After that, the loss is calculated with respect to the selected ground truth candidate. The loss calculation process is similar to the label selection phase, where the transformed image is formed with respect to the grasp parameters, which is used as affine parameters. The intuition behind making the IoU metric differentiable, is very similar to calculating the intersection area within two polygons. Measuring this metric is quite feasible and easy for bounding boxes, which are rectangles without orientations, as the calculation of the intersection area can be seen in the Equation 2

$$[\min(X_{mx,p}, X_{max,g}) - \max(X_{mn,p}, X_{mn,g}) + 1] * [\min(Y_{mx,p}, Y_{mx,g}) - \max(Y_{mn,p}, Y_{mn,g}) + 1] \tag{2}$$

However, this metric does not apply to our case, since we have an extra parameter, orientation. Calculation of the intersection are of the convex shapes is called as Polygon clipping, and the calculation of it depends on several conditions, which makes the overall function non-differentiable. Thus, as a work-around, we are calculating the loss by the Euclidean distance for the all corresponding edges points, gathered from the transformation matrix, and normalize the calculated loss by the number of vertexes in the transformed mesh. Although it is not the desired approach, selecting the candidate ground truth label with respect to the IoU loss among more than 100 annotations for each pose, still gives results that converge to the grasp prediction which gives high intersection with the ground truth grasping rectangle.

5 Comparative Evaluation and Discussion

After the modal is complete, the evaluation is done with respect to several different parameters, such as different optimizers, different modals, loss functions and learning rates. The accuracies are calculated through IoU metric, which is explained in detail in Section 4.2.2

There are several different loss function evaluation methods are used, namely as the IoU and the MSE methods. Moreover, MSE is calculated with different number of regression parameters. The MSE calculation with only the x, y parameters are used is called MSE-2, whereas the calculation with x, y, θ grasp parameters refer to MSE-3 and the calculation with all the grasp parameters referred as MSE. The train accuracies and the test accuracies are calculated with, how much the predicted grasp rectangle close to one of the ground truth labels according to the IoU metric.

Please note that, the accuracies do not represent whether the grasps are successful or not, it only represents how the grasping prediction is closed to one of the annotations for all optimal ground-truth values.

In 5.1, we are presenting our results with respect to the various parameters mentioned above

Network Modal	Finetuned layers	Loss Function	Optimizer	Learning Rate	Epochs	Train Accucary	Validation Accuracy
AlexNet	Last Layers*	MSE-2	SGD	0.001	200	0.085	0.0068
AlexNet	Last Layers*	MSE-5	SGD	0.001	200	0.123	0.099
AlexNet	Last Layers*	IoU	SGD	0.001	200	0.137	0.128
ResNet18	Last Layers**	MSE-2	SGD	0.001	200	0.133	0.107
ResNet18	Last Layers**	MSE-5	SGD	0.001	200	0.138	0.104
ResNet18	Last Layers**	IoU	SGD	0.001	200	0.178	0.183
ResNet34	Last Layers**	MSE-2	SGD	0.001	200	0.206	0.197
ResNet34	Last Layers**	MSE-2	Adam	0.0001	200	0.097	0.091
ResNet34	Last Layers**	MSE-2	SGD	0.001	400	0.284	0.167
ResNet34	Last Layers**	MSE-3	SGD	0.001	200	0.348	0.288
ResNet34	Last Layers**	MSE-5	SGD	0.001	200	0.233	0.218
ResNet34	Last Layers**	IoU	SGD	0.001	200	0.461	0.312
ResNet34	Last Layers**	IoU	Adam	0.001	200	0.410	0.297

Table 5.1: Training and Validation accuracies with varying parameters.
(* Only the last fully connected layer is reshaped to 5.)
(**The last layer is replaced with a sequential block, that includes 2 fully-connected layers, such that the output gives 5 parameters.)

The training and test sets are created according to the k-fold cross validation where k=5. The whole dataset includes 1000 images, 200 objects, where different number of poses feed into the network depending on the object.

As we changed the learning parameters of the modal, we tried to come up with a more generalized experiments that can give insights about the differences between IoU and the mean squared error losses. In most of the experiments MSE-3 loss, that uses 3 different regression parameters which are x, y, is the most accurate loss among all MSE measures. However, as it can be inferred from the results, whose accuracy graphics are presented at Figure 5.1 and 5.2 , the method we proposed, that is calculating the closest grasping rectangle with respect to IoU loss, then continue with the loss evaluation with

respect to Euclidean distance for all vertexes in the candidate ground truth value and the prediction, performed slightly better in terms of representing a good grasp. To sum up, we can say that selecting the closest ground truth label with respect to IoU metric gives us a better generalization to grasp learning process

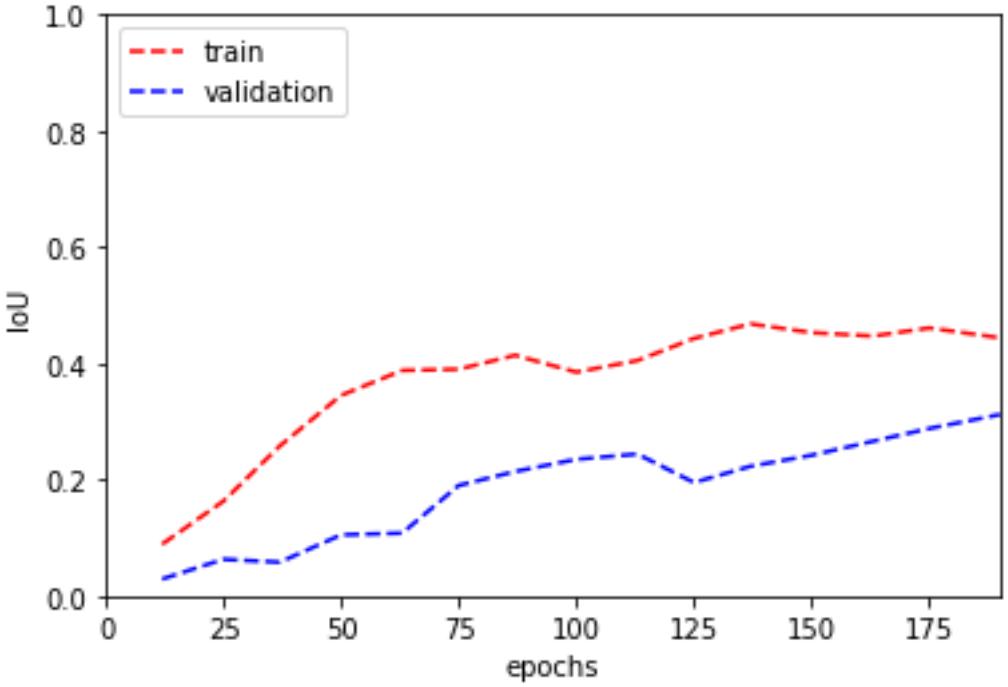

Figure 5.1: Training and Validation Accuracies for a Modal Trained with IoU Loss

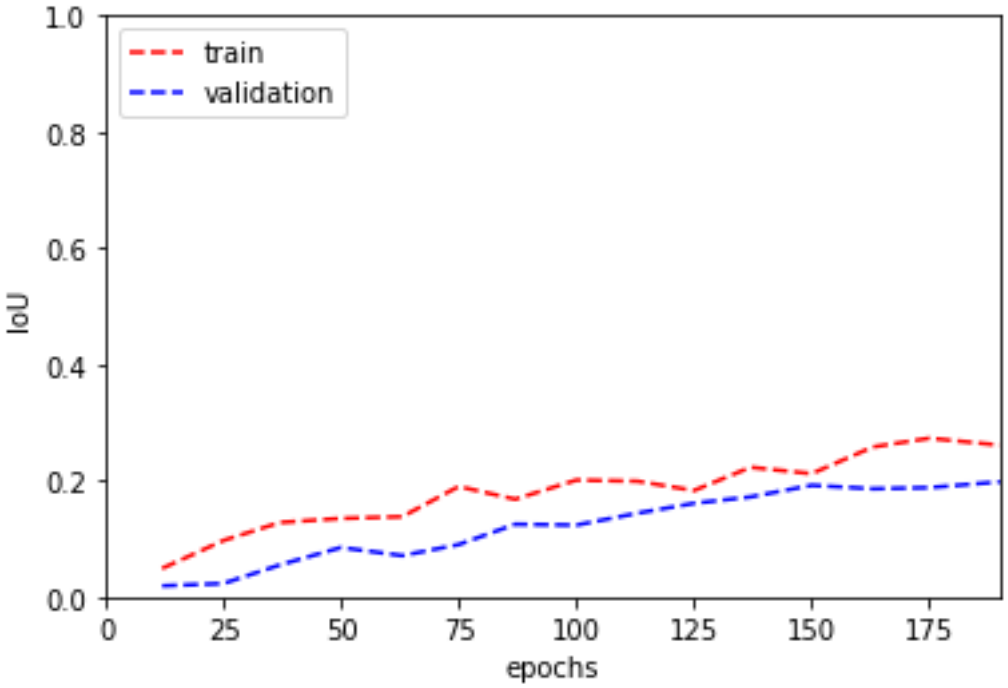

Figure 5.2: Training and Validation Accuracies for a Modal Trained with IoU Loss

6 Conclusion and Future Work

As a result of this project, it is expected to implement a fully-functional neural network that regresses grasping region parameters, which will be fed into the robot to execute the grasping action on objects in real life. Moreover, we have gathered results to prove that our proposed approach, which includes selecting the closest ground truth grasping region with respect to IoU, then calculating the loss according to the generated vertexes of the grasping rectangle from the output parameters taken from the input, performs better than the most frequently used error metric in recent grasp prediction systems, which is mean squared error.

In order to have a more generalized result, we need to come up with a system that can differentiate the Intersection over Union loss, to train the whole system according to this measure. We could not implement this behaviour for our modal, since PyTorch's autograd mechanism needs every operation that is used on the output tensors to be differentiable. One method proposed by Rahman and Wang, to make the intersection over union metric differentiable [29], can be achieved while the parameters to be differentiated are the bounding box parameters which does not have any orientation, and have class probability parameters that signifies whether the bounding box is a background or a foreground object. However, it is not very meaningful to satisfy these constraints in the grasping rectangle parameter estimation task, one approach we proposed for this work is to create a mesh with 2500 pixel values, that will refer to the points within the grasping rectangle. We have also implemented this approach, that calculates this for both closest ground truth candidate and the output, with the idea of calculating the number of points that have the same coordinates for output and the label within all 2500 pixels for the two meshes. However, since the outputs are float, we have seen that it is not very likely to the transformed pixel maps have a matching coordinates in any of those 2500 pixels. We tried to solve this problem by converting the pixel maps to integer, which made the whole operation non-differentiable.

Therefore, one aspect to be improved for a later work, is to come up with a IoU metric calculation for ground truth annotations and the predictions, which are differentiable. With this approach, it would be more meaningful to learn the advantages of IoU loss with respect to other L2 Norm based losses, such as MSE.

7 References

[1] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 3511–3516.

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

[3] S. Caldera, A. Rassau, and D. Chai, "Review of deep learning methods in robotic grasp detection," *Multimodal Technologies and Interaction*, vol. 2, no. 3, p. 57, 2018.

[4] Cs231n. (). Cs231n convolutional neural networks for visual recognition, [Online]. Available: http://cs231n.github.io/convolutional-networks/ (visited on 05/12/2019).

[5] Mathworks. (). Introduction to deep learning: What are convolutional neural networks? [Online]. Available: https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html (visited on 05/12/2019).

[6] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, p. 5.

[7] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[8] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[9] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 3304–3311.

[10] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, p. 3.

[11] A. Depierre, E. Dellandrea, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, p. 2.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *2016 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2016, pp. 3406–3413.

[14] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 1316–1322.

[15] A. Depierre, E. Dellandrea, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 5–6.

[16] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[17] T.Oliphant. (). Numpy: A guide to numpy, 2006, [Online]. Available: https://www.numpy.org/ (visited on 01/11/2019).

[18] Shapely. (). Shapely library for geometric object and operations, [Online]. Available: https://pypi.org/project/Shapely/ (visited on 02/19/2019).

[19] Matplotlib. (). Matplotlib, 2d plotting library, [Online]. Available: https://matplotlib.org/ (visited on 04/22/2019).

[20] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," *GitHub repository*, 2016.

[21] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[22] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, p. 5.

[24] Pytorch. (). Pytorch, torchvision module for pretrained network models., [Online]. Available: https://pytorch.org/docs/stable/torchvision/models.html (visited on 03/17/2019).

[25] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized ßntersection over union: A metric and a loss for bounding box regression," *arXiv preprint arXiv:1902.09630*, 2019.

[26] pyimagesearch. (). Iou for object detection., [Online]. Available: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/ (visited on 04/28/2019).

[27] M. Berman, A. Rannen Triki, and M. B. Blaschko, "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4413–4421.

[28] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, p. 5.

[29] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *International symposium on visual computing*, Springer, 2016, pp. 234–244.