In our experimenting process two cameras at the top and at the front side respectively are used. To be specific, a separate camera at one side of the table provides a side view with useful inclination information of two sides of the object, and for some of the items one of these sides (left side in our experiment) is labeled as the position to be gripped. The image from each camera is converted to 100*100 grayscale by MATLAB codes before using. As a result, each image comes as a 1*100*100 integer matrix with each value between 0 and 256 representing the brightness level of each pixel. Figure 1 shows the setup and Figure 2 shows the top and side views of some example tableware. The robotic arm is not shown but at this stage it is somewhere not obstructing the two cameras, such as retracted in a corner. 6 example pairs of images are shown in Figure 3.

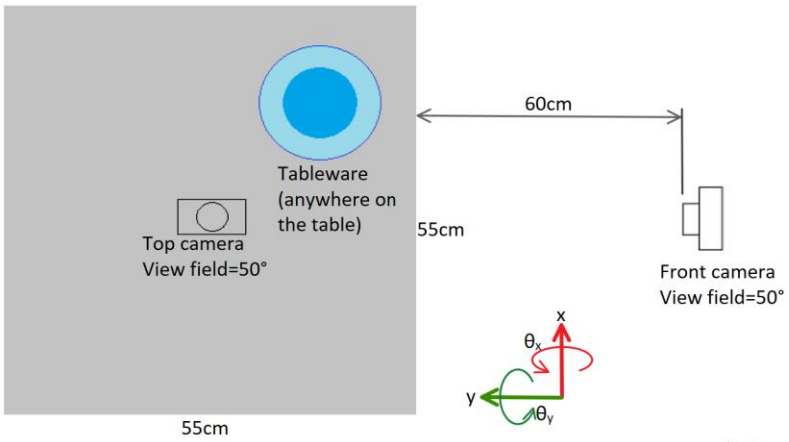

Figure 1 Top view of experimental setup

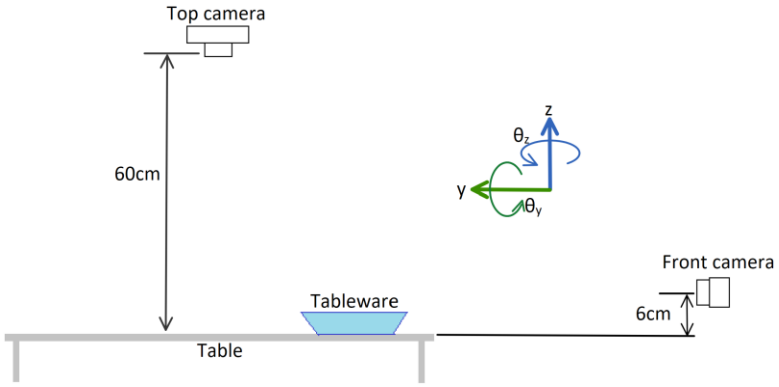

Figure 2 Side view of experimental setup

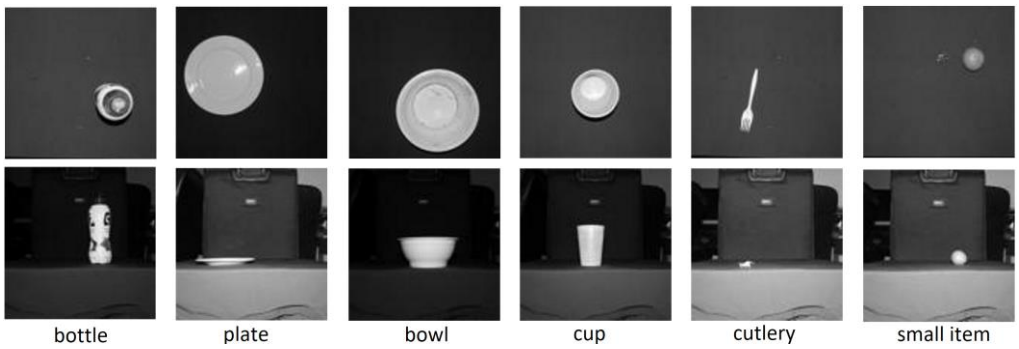

Figure 3 Example pairs of images

The items to be gripped are firstly observed by the top camera with the image sent into a mask R-CNN network, which gives a bounding box of the items with for values in the field of the top camera. The four bounding box values include the two coordinates of the center of the box and the box width and height. We did not test mask R-CNN's performance with our dataset since the dataset is too small and mask R-CNN is too deep for such a small dataset to give good results. But when there are more samples with more diversity mask R-CNN is proven to give very accurate bounding boxes. Meanwhile, the side view image and the top view image is sent into another neural network which classifies the tableware into 6 categories including plate, bowl, cup, bottle, cutlery and small items. The reason to use another network for classification is because mask R-CNN and other industrialized algorithms only accept one image input, whereas we have two images from two angles. Therefore we developed a specific CNN model for doing this, as shown in Figure 4, with image-wise split test accuracy of 91.7%. The bound box output in Figure 4 is not used anywhere else because we already employed the mask R-CNN to do this. However, adding this layer is found to help boost classification accuracy somehow. The model building and training job was all done in Python with Pytorch and other supporting packages including pandas, numpy, matplotlib, os and skimage. The optimizer is an SGD optimizer with MSEloss criterion, learning rate of 5E-6 and momentum of 0.9. While training, the batch size is 20.

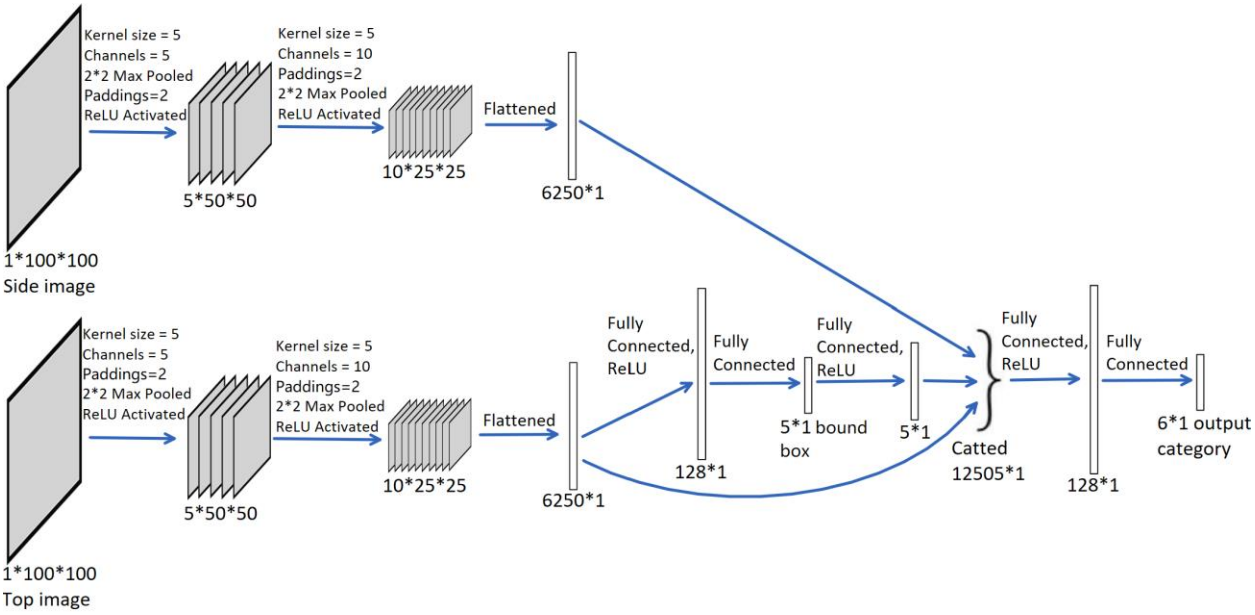

Figure 4 Network for classification of kitchenware

Then the bounding box predicted by mask R-CNN, the top view image, as well as the side view image, are fed to another hybrid (CNN and ANN combined) neural network altogether, from which the 3-dimentional grasping point and the 3-dimensional gripper angles are predicted, giving a complete grasping pose (x, y, z, θ_x, θ_y, θ_z) of the gripper.

To make a firm grip, different categories of tableware are labelled to be gripped differently. Plates, bowls and cups are labeled to be griped at the left edge, bottles and small items are labeled to be griped from the top, and cutleries from the handles or anywhere applicable for a firm grip.

For the network to be trained properly, the tableware category information is transformed into one-hot format before feeding into the network. Hence there are two 1*100*100 grayscale image inputs and 10 individual inputs (6 categories plus 4 bounding box indications). The labels (3 coordinates and 3 angles) for training the grasping point locating model are all determined by human eyes. There are in total 123 samples, created from 2 plates, 3 bowls, 3 cups, 5 bottles, 4 cutleries and 2 small items. These samples are split with a train-validation-test ratio of 0.72:0.18:0.1 image-wise. The detailed construction of the model is shown in Figure 5:

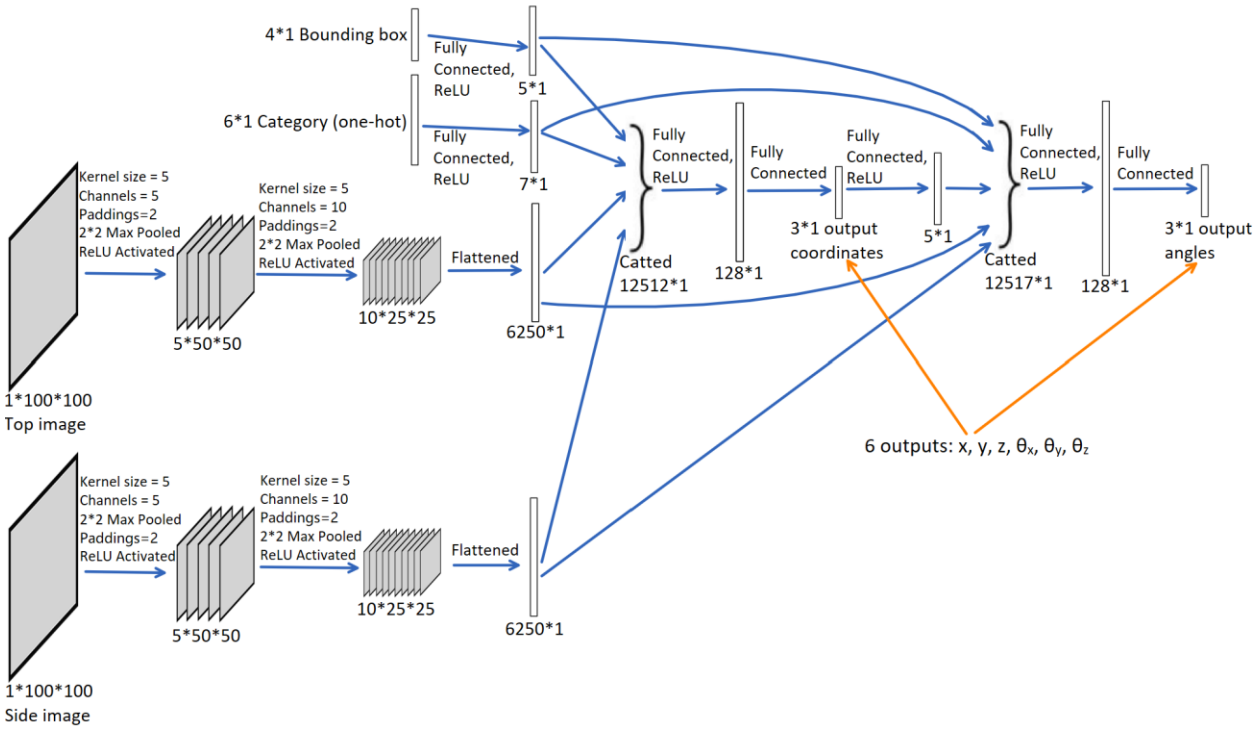

Figure 5 Grasp pose detection model

As shown in Figure 5, the six outputs come at two different stages. The coordinates are predicted first, then they are used as an assistance for predicting the angles. The model building and training job is all done in Python with Pytorch and other supporting packages. The optimizer is an SGD optimizer with MSEloss criterion, learning rate of 5E-6 and momentum of 0.9. While training, the batch size is 20. After 50 epochs, the learning curve is as shown in Figure 6:

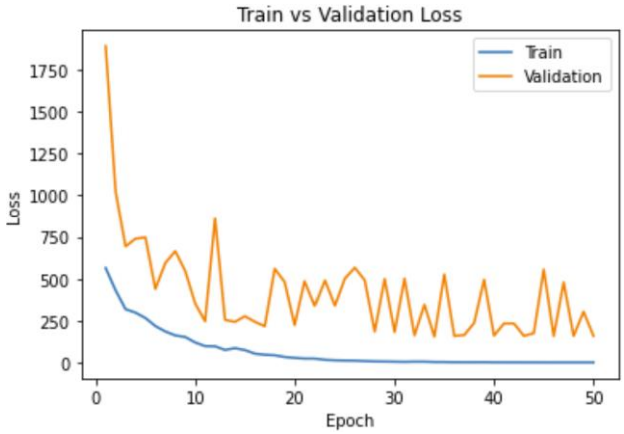

Figure 6 Learning curve of GPD model

From Figure 6, the training curve is smooth and approaches 0 while the validation curve is noisy. The major cause for this is a very small dataset lacking diversity. Since the training loss reaches close to 0, this model should have the ability to perform much better with less noisy validation curve and small validation error when the dataset broadens.

By training on the training dataset and testing on the separated test dataset that had never been used for training, the results from two runs are shown in Figure 7.

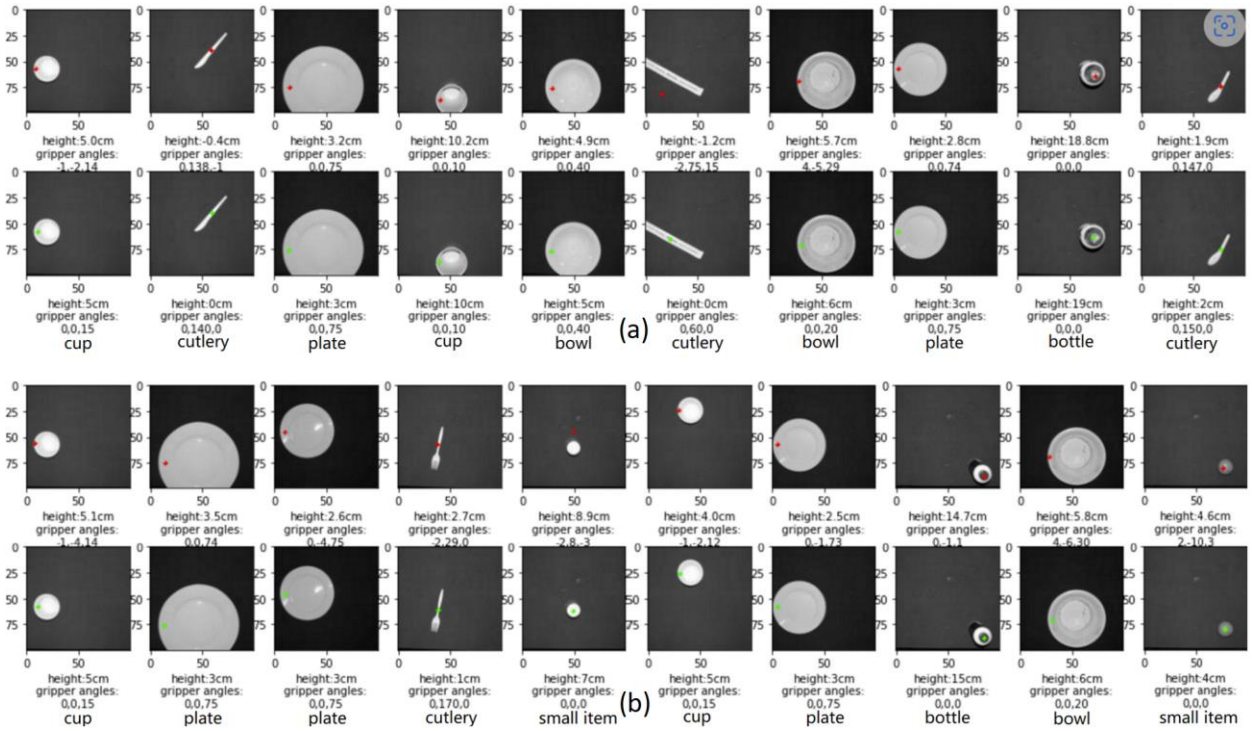

Figure 7 GPD results

In Figure 7, the labels on axes are pixels and the 100 pixels transforms into 55cm on the table. The upper row are the predictions of the trained model in Figure 5 and the red dot indicates the predicted grasping position. Below each image the predicted heights and angles are indicated. The lower row are the labels

and the green dot indicates the labeled grasping position. Below each image the labelled heights and angles are indicated. On the bottom of each pair of predicted and labeled image the category is indicated. By comparing the predictions and labels, the model performance seems excellent given that there were merely 89 low resolution images used for training. It showed the ability to decide whether grasping from right above the item (as for bottles and some small items) or grasping the edge (as for bowls, plates and cups). Also, the predicted gripper angles are generally close to labeled and usable. The major errors are on cutleries and small items.

Given the speciality of this task, the accuracy of the model cannot be simply judged by numerical standards like RMSE. Instead, it is more essential to justify whether the grasp can be successfully made. A firm grip needs to be a force-closure grasp, when the line segment connecting the two contact points of the two fingers on the object points out of and into the two friction cones respectively at the two contact points. To make a force-closure grip, adequate force needs to be applied with high friction constant fingers to make the combined forces pointing into the friction cones with the presence of gravity. However, in our task, even if a force-closure grip cannot be made at once (i.e. the object slips and rotates while the gripper closes), in many cases the object will rotate into a pose where a force-closure grip is resulted. For instance, in the first example of Figure 7, the predicted gripper pose angles are (-1°, -2°, 14°), whereas the ground truth (labeled) pose angles are (0°, 0°, 15°). The -2° error of the gripper about the y-axis will force the cup to rotate for -2° about the y-axis at the grasping point, and the -1° error of the gripper about the z-axis will force the cup to rotate for -1° about the z-axis at the grasping point, then finally the normal forces between the cup wall and the gripper fingers reach the nominal value as the gripper closes to the minimum possible, making the combined forces at the gripping points into the friction cones and leading to a firm force closure grip. By observing Figure 7, there are only two cases the predicted grasping point (red dots) did not fall near (within roughly 2 pixels) the labeled locations (green dots), being the sixth example in Figure 7(a) and the fifth example in Figure 7(b). Assume an error of 1cm in height and 10° about each axis is acceptable, then one more failed grasp could be led by the fourth example in Figure 7(b) due to a wrong angle about y-axis. Overall the theoretical success rate is about 80 to 85% based on this test set.