CS231N Project: Encoding Visual Modality for Robotic Manipulation Task of Peg-in-Hole-Insertion

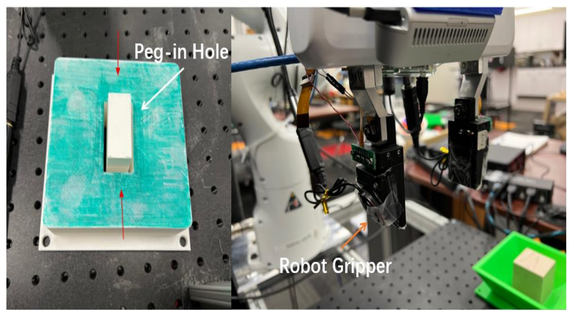
Hao Li,[1] Kaixin Chen[1]

Stanford
Computer Science

Abstract

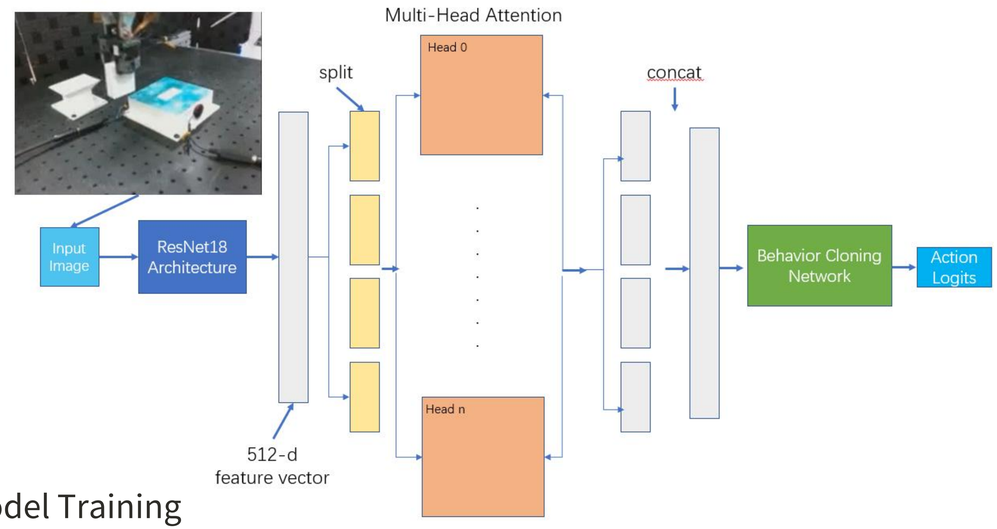
This project focuses on the application of vision encoders in the robot manipulation task of peg insertion. We used a ResNet-18 network pretrained on ImageNet as our baseline method and proposed a new algorithm. This new algorithm is based on ResNet-18 but with an additional multi-head self-attention layer. Our experiments showed that this additional multi-head self-attention provides better performance in our setup.

Task Setup & Data set



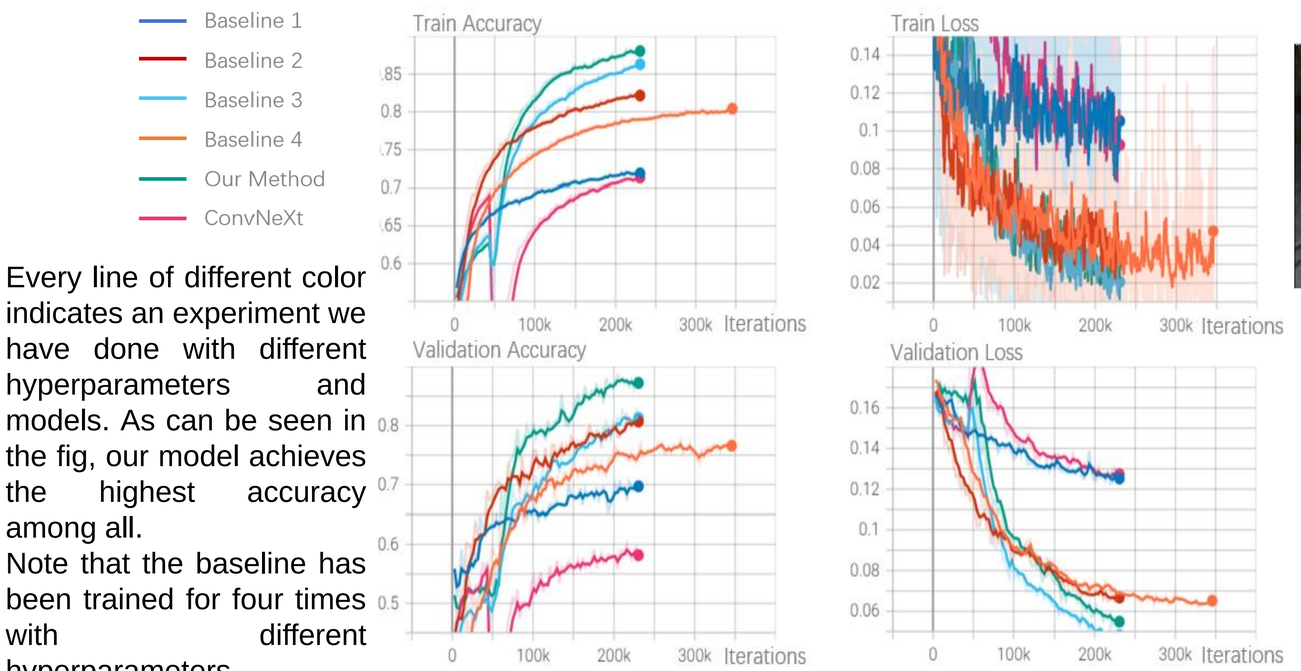
The visual encoder built in this project should guide a given controller to learn and perform peg insertion shown above.
The data set includes a series of images of the robot's workspace and corresponding ground truth actions (human demonstration).
Since the action space of the robot is discrete, the evaluation metrics are cross entropy loss and accuracy.

Model Architecture/ Method



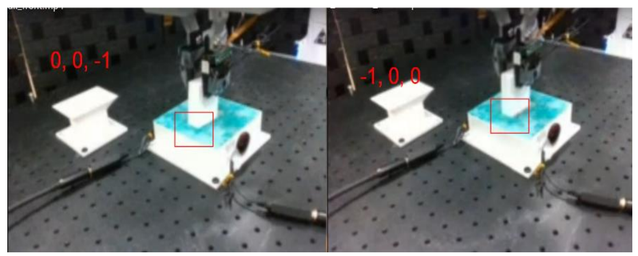
To better address the features of the input images. We design the model as shown on the left. The model architecture starts with a standard Resnet18 structure without the last classification layer. Therefore, the output will be a 512 dimension feature vector. Then, we implemented a multi-head attention structure. The output feature is split input 4 different pieces, of which has 128 dimensions. The split feature vectors will be input into the multi-head attention to get a more fine-grained feature of the input images. Finally, the features will be input into the Behavior Cloning Network (controller) to get the action logits.

Experiments for Model Training



Every line of different color indicates an experiment we have done with different hyperparameters and models. As can be seen in the fig, our model achieves the highest accuracy among all.
Note that the baseline has been trained for four times with different hyperparameters .

Real Robot Test Results



We performed 50 real-robot test with each trail starting at a random initial position. If the peg is inserted in the hole in the end, we will count it as a successful case. Finally, our model achieves a success rate of 84%. In the failure cases. the model predicts the wrong action at the edge of the hole base. This is because there is some vague representation in the vision

Stanford