Homework 4 COMS4733

dy2462
Dongxiao Yang

Problem 2(a):

The reason why we use gaussian maps instead of one-hot images is because gaussian maps have non-zero information across the image, while one-hot images only contain information on the keypoints. Therefore, gaussian maps are much more dense representations than one-hot images, and the model can learn faster and better on gaussian maps.

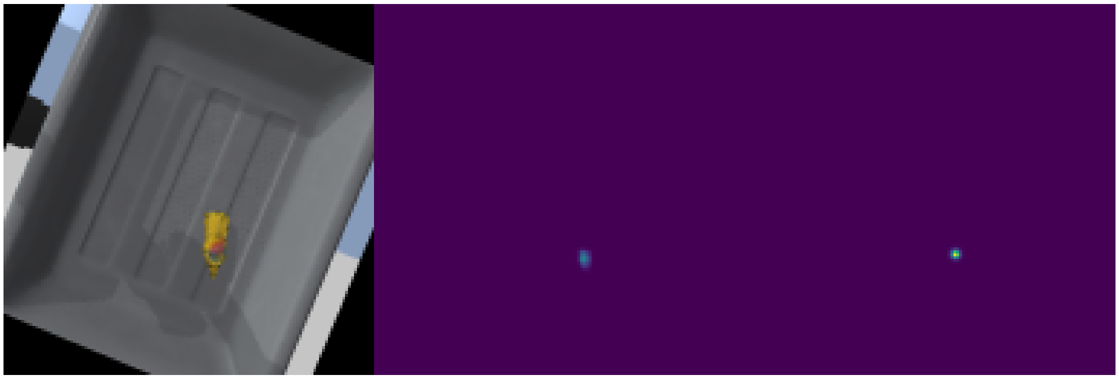
Problem 2(b):

The pipeline performs transformations on images as follows with a 70% chance:

1. Translate images by -20 to +20% on x- and y-axis independently.
2. Rotate images by -angle_delta/2 to angle_delta/2 degrees.

Problem 2(d) (Training):

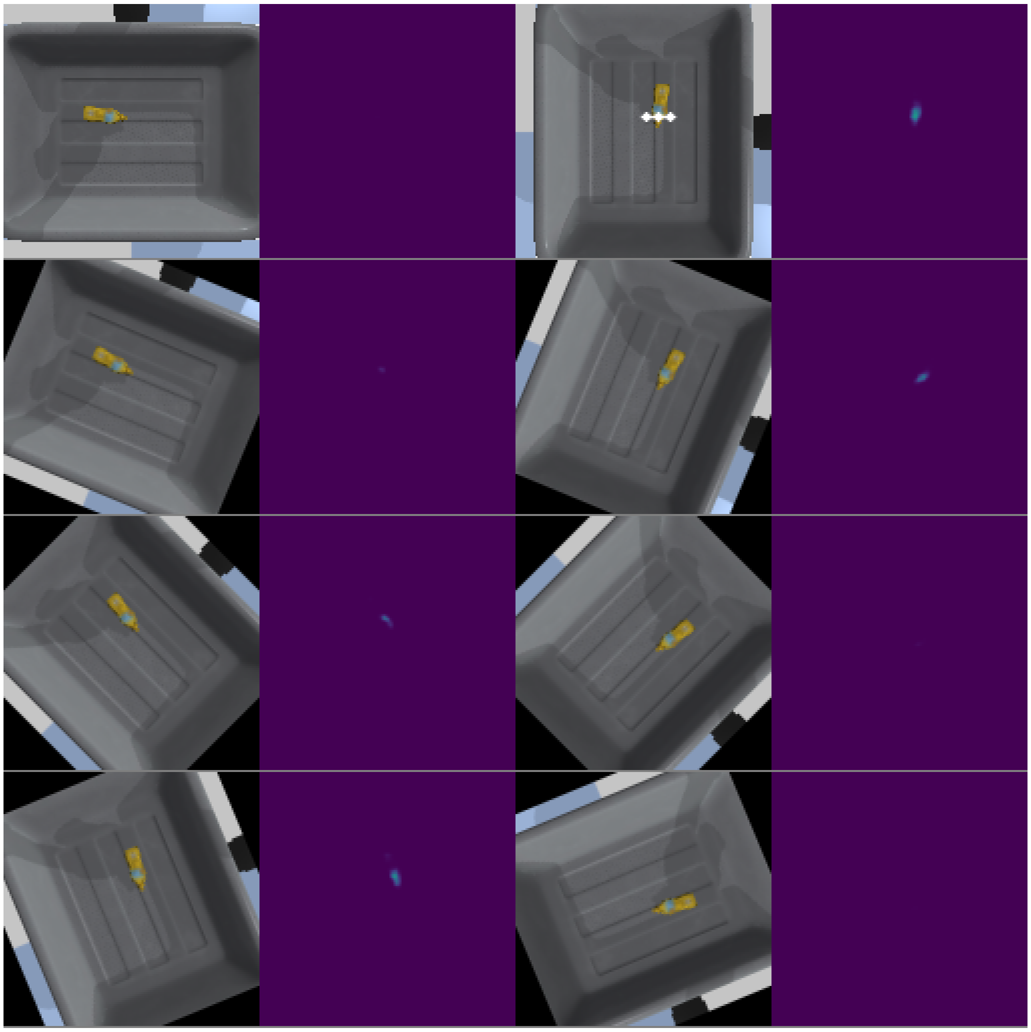
After training, my loss for the training set was 0.0011, and validation loss was 0.0010.



Problem 2(f) (Evaluation on the training set):

Video link: https://drive.google.com/file/d/1r7YNUCk5RJ7fwEXL2oUAWEqBpw__0-kW/view?usp=share_link

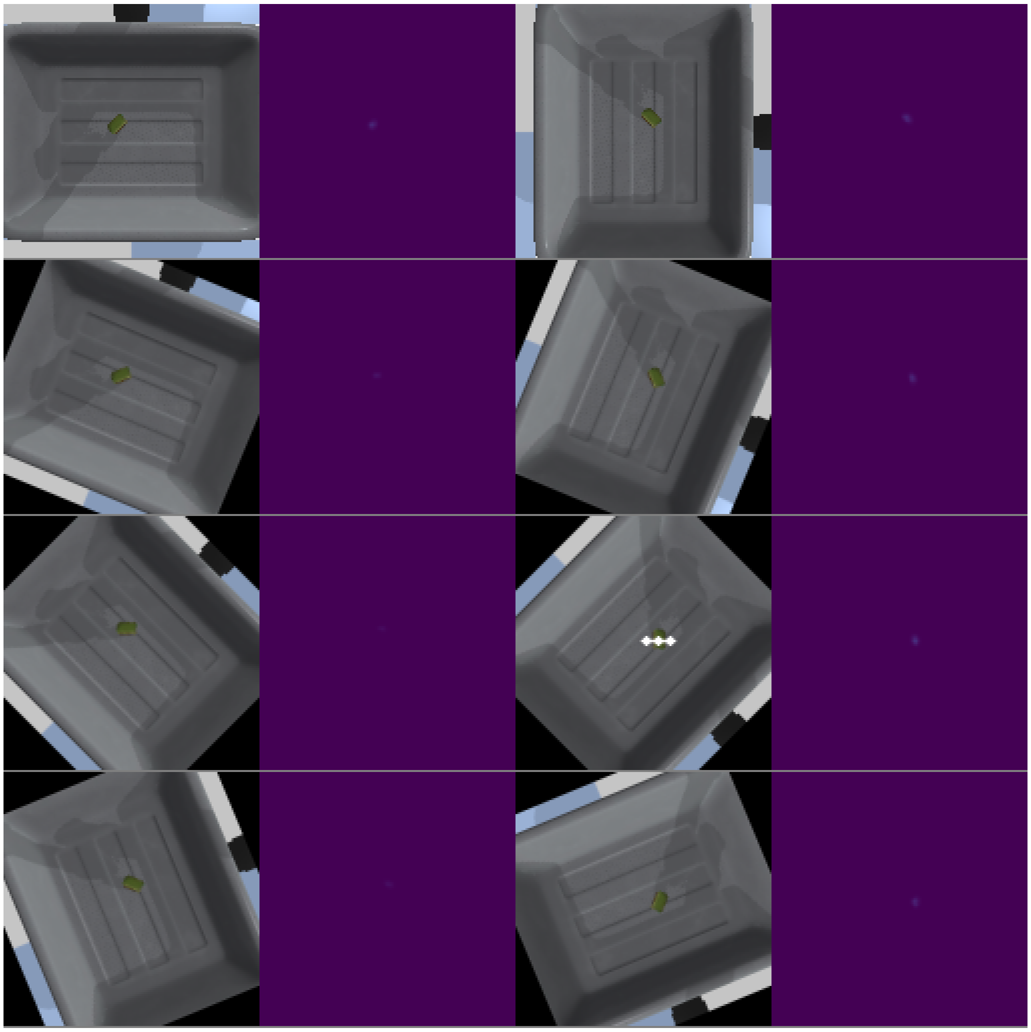
The success rate was **86.7%**. See an example below:



Problem 2(g) (Evaluation on the heldout objects):

Video link:
https://drive.google.com/file/d/1rOZgEf1C2arhgMEOgu_0jltUSNCm3_Sk/view?usp=share_link

The success rate was **70.0%**. See an example below:



The images would be similar to the visualization from the previous part. The model will predict a gaussian heat map centered at the grasping position. Heat maps corresponding to good grasping angles have higher peaks. The difference is that, for unseen objects, sometimes the gaussian score is much lower than objects that it has seen during training. For example, in the figure above, even the best rotation angle can only give a very dim heat map. Furthermore, the blue can, which is one of the unseen objects, does not trigger any attention from the model, no matter where it is located.

Problem 2(h) (Evaluation on mixed objects):

Video link:
https://drive.google.com/file/d/12mqpPVSlj6RlPt4FnG8_5pzSxJlr9rOQ/view?usp=share_link

In this test, only 1 object was left (though one of them was a model bug).

Problem 2(i):

The reason for sample efficiency is that the affordance map is a highly dense representation of the action. Instead of directly predicting the 3-dimensional actions, predicting the affordance map allows the model to learn much faster with limited amounts of data.

Problem 3(a):

Video link:
https://drive.google.com/file/d/1otroMVug5yTvEfhVPMIaYSx35NHrNVnW/view?usp=share_link

Problem 3(b):

Video link:
https://drive.google.com/file/d/1hkYwXdD-5UBjJchYCSDCXpDpVcvbWiLU/view?usp=share_link

Problem 3(c):

Only 1 object was left.

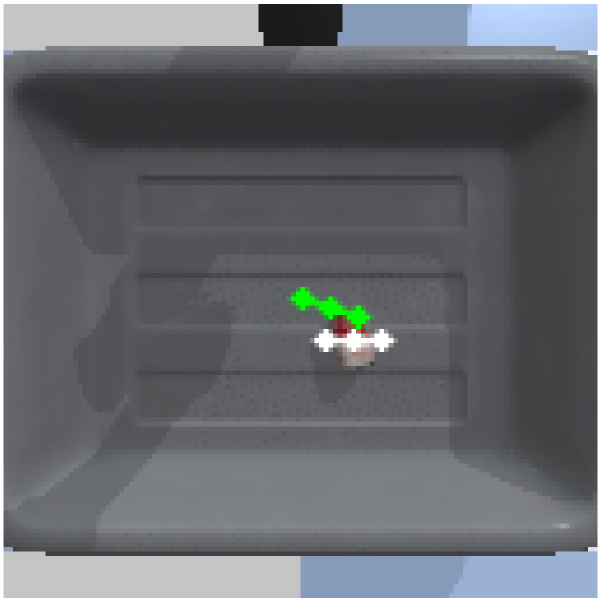
Video link:
https://drive.google.com/file/d/1cUYZbq3vnLuiQLIqmzh8KaNy8IGXX8rg/view?usp=share_link

Problem 3(d):

For 3(a), the performance is the same, since eval.py does not utilize n_past_actions in this part.
For 3(b), the performance is much better (**from 70% to 86.7%**) since the robot is able to execute multiple attempts and filter out previous failures. However, for some unseen objects, since the model is somehow not able to recognize the shapes, the robot always tries to grasp the edge of the bin, which leads to failures.
For 3(c), The performance is improved but not obviously, since the performance in Problem 2 is already very good (only 1 left). This might be due to the piling of objects that makes “failure” not straightforward. The gripper has a large chance to collide with other objects so that the configuration of objects is changed, and in the next attempt the robot may try to pick another object. Therefore, it might be tricky to evaluate the improvement on this part.

Problem 4(a) Training:

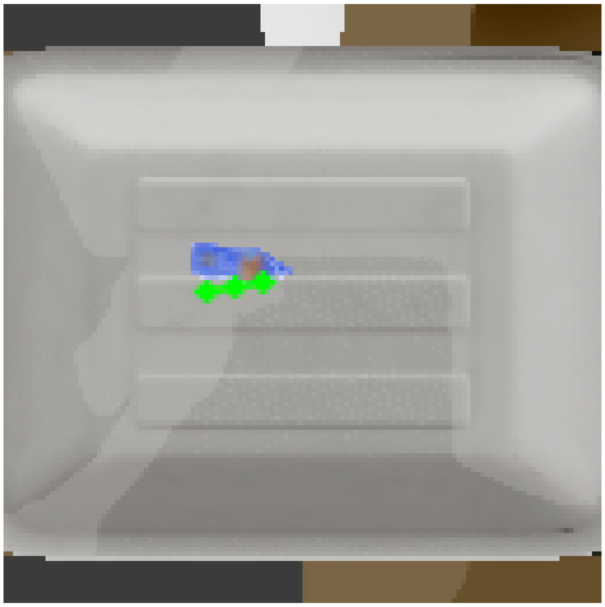
The saved checkpoint has a training loss of 0.0160 and a validation loss of 0.0097. See an example below:



Problem 4(b):

Video link:
https://drive.google.com/file/d/1oND2VAOUmWyL1y6Q-sD5t_CI35SZ70OB/view?usp=share_link

The success rate was **26.7%**
See an example below:



Problem 4(c):

Video link:
https://drive.google.com/file/d/1u1IIldfOCrMH-RdzMjYVarGNoqDuIztg/view?usp=share_link

In this test, 14 out of 15 objects were left. Only 1 object was picked out from the bin.

This is much worse than the affordance model. The reasons are:

1. As can be seen from the visualizations, the action regression model is not able to make precise predictions.
2. It does not have a failed action buffer. Once an action fails and the gripper does not touch anything, the robot will get stuck in that failed action and can never be recovered.
3. The dataset for the action regression model to train on is too small. The model can barely generalize to unseen configurations.
4. Since the dataset is too small, the training results fluctuate each time. Sometimes the final loss can be high as 0.02, which means that the model cannot learn well.
5. Compared to the affordance model, the representation of the action of this model is much more naive and cannot encode the dense information from the observation images.