CIS580 Machine Perception

Homework 6
May.3.19

Chun Chang

Step 1: training,

Use the image and given keypoint in the heatmaps to train the model

```
pred_heatmap_list = self.model(batch['image'] );
pred_heatmap = pred_heatmap_list[-1]

loss = self.heatmap_loss(pred_heatmap, batch['keypoint_heatmaps'] )
loss.backward()
```

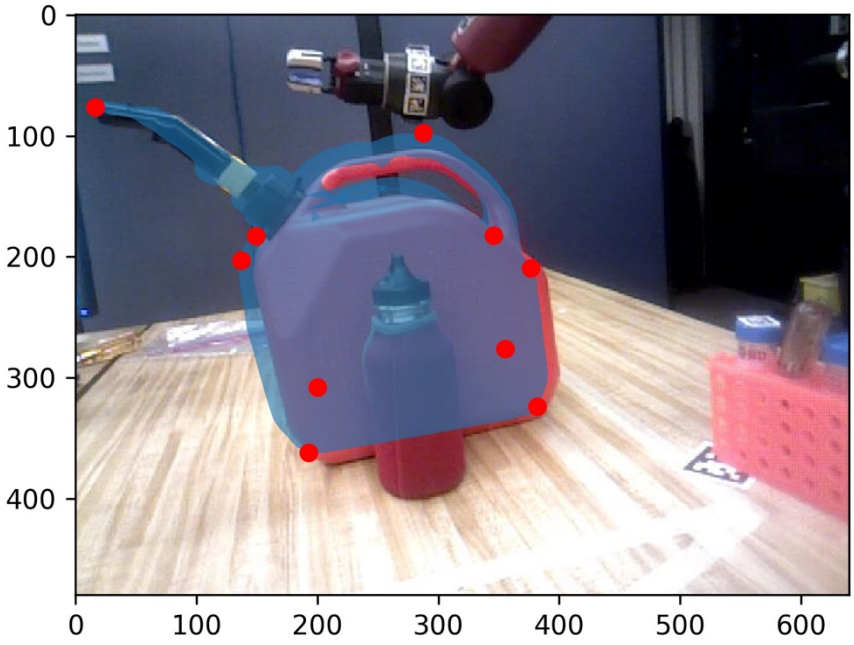
Step 2:,

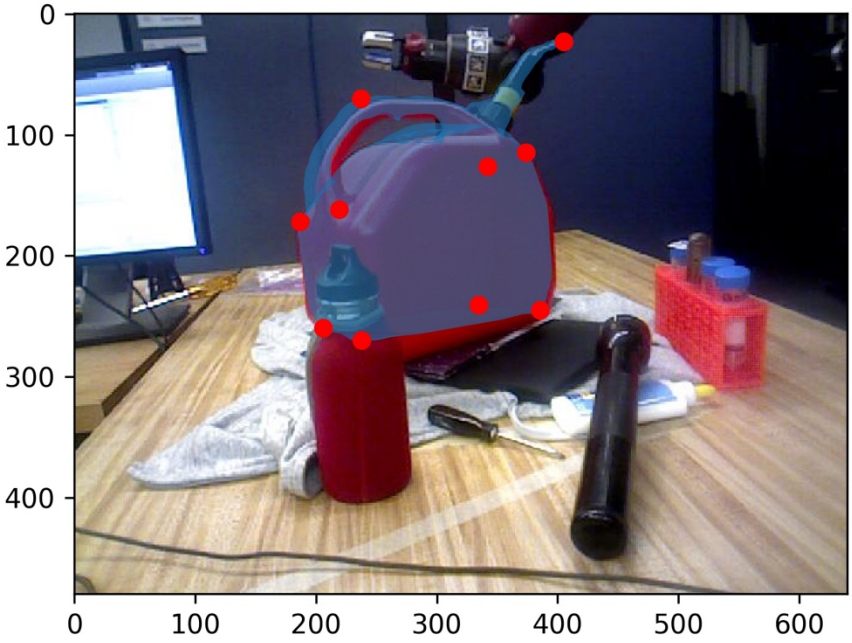
Get the close rotation matrix

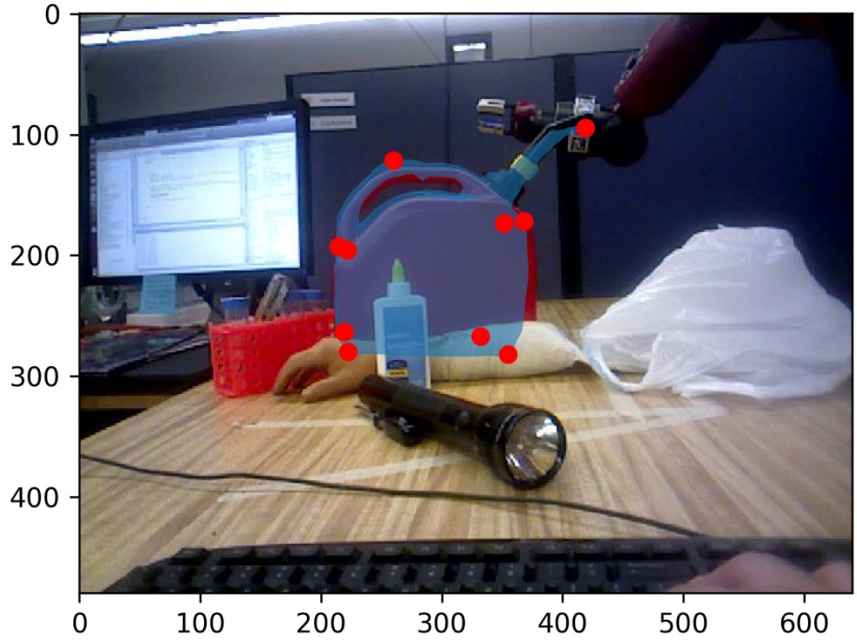
```
# convert axis-angle to rotation matrix
R = rodrigues(r)
# 1) Compute projected keypoints based on current estimate of R and t
im_pts = keypoints_3d.transpose(1, 0)
pt_w = torch.matmul(R, im_pts ) + t[:, None]

pt_w_norm = pt_w / pt_w[2]
projection = pt_w_norm.transpose(1,0)
# 2) Compute error (based on distance between projected keypoints and detected keypoints)

diff_conf = (norm_keypoints_2d - projection[:, :2])*d # function of error
error  = torch.sum((diff_conf)**2) # minimized term

# 3) Update based on error
error.backward()
optimizer.step()
# 4) Check for convergence
if abs(error.detach() - loss_old)/loss_old < rel_tol: # converge
  break
else:
  loss_old = error.detach()   # update loss
```

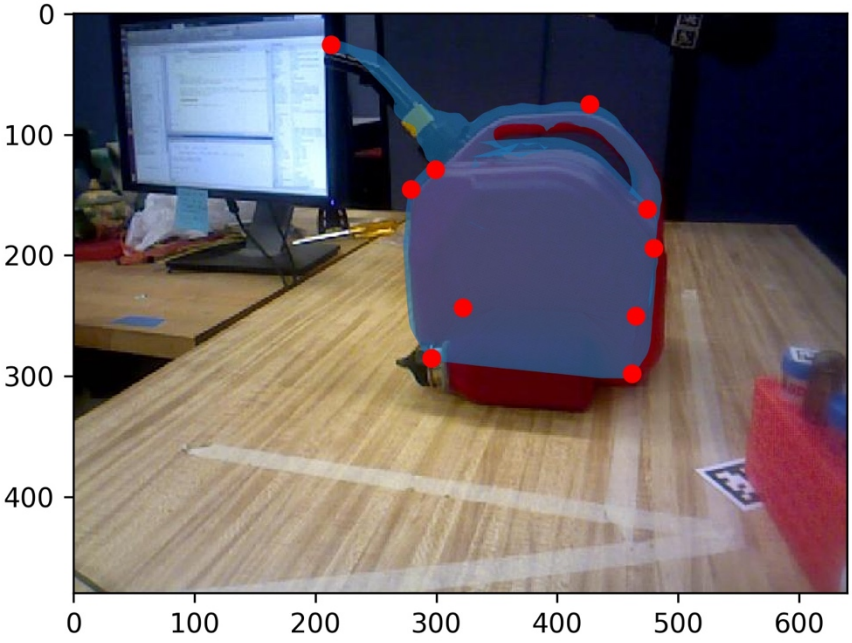
```
# get keypoints
r_max = torch.max(heatmaps,-2)
c_max = torch.max(heatmaps,-1)

# get maxinum pts' locs
r_ind = torch.argmax(c_max[0],-1).unsqueeze(-1)
c_ind = torch.argmax(r_max[0],-1).unsqueeze(-1)
Loc = torch.cat([c_ind,r_ind],dim=-1).float()

# get their confidence
confidence = torch.max(c_max[0],-1)[0].float().unsqueeze(-1)
key_loc = torch.cat([Loc,confidence],dim=-1)

return key_loc
```

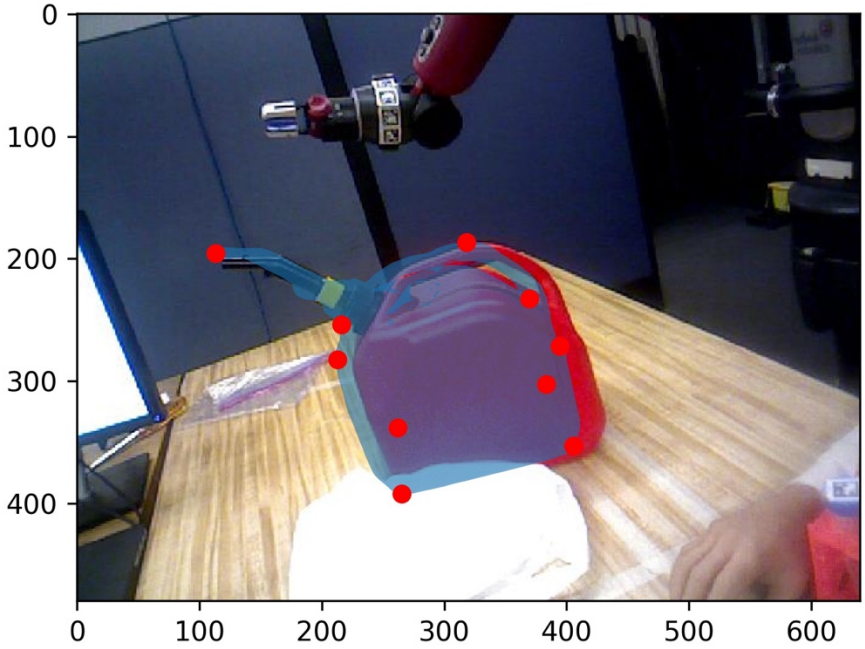
Pose
estimation



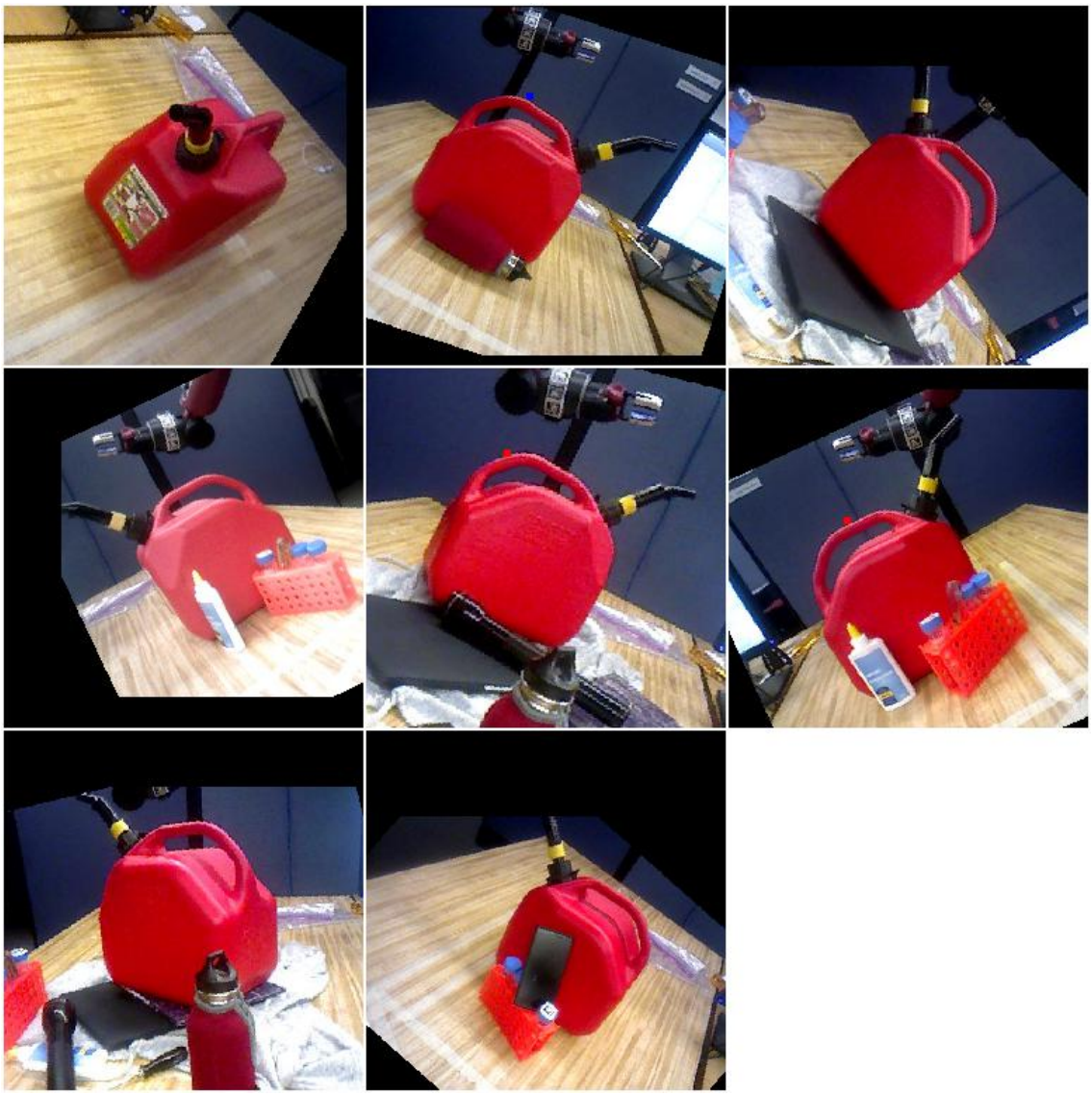


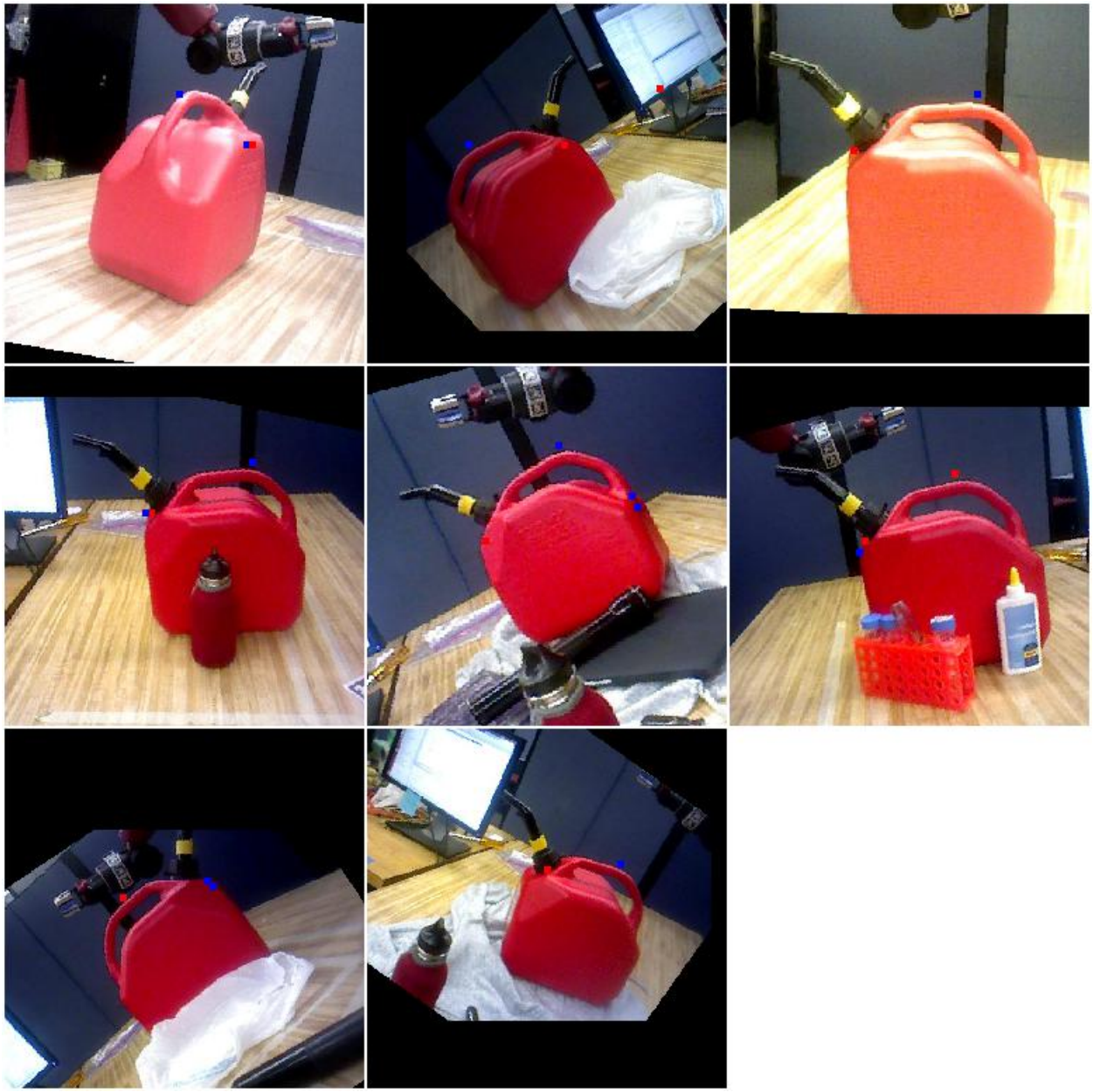

0
100
200
300
400
0
100
200
300
400
500
600

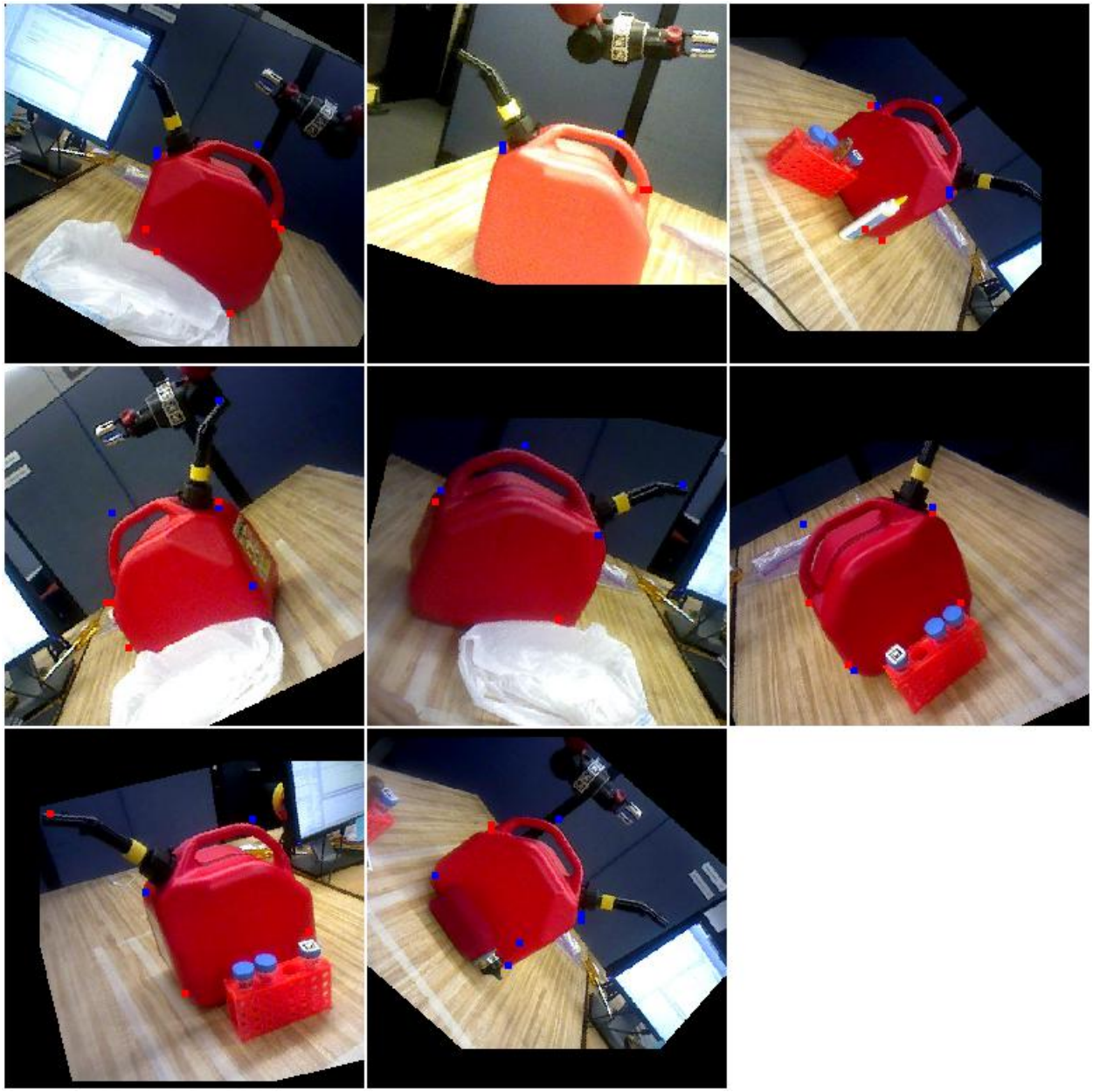

0
100
200
300
400
0
100
200
300
400
500
600

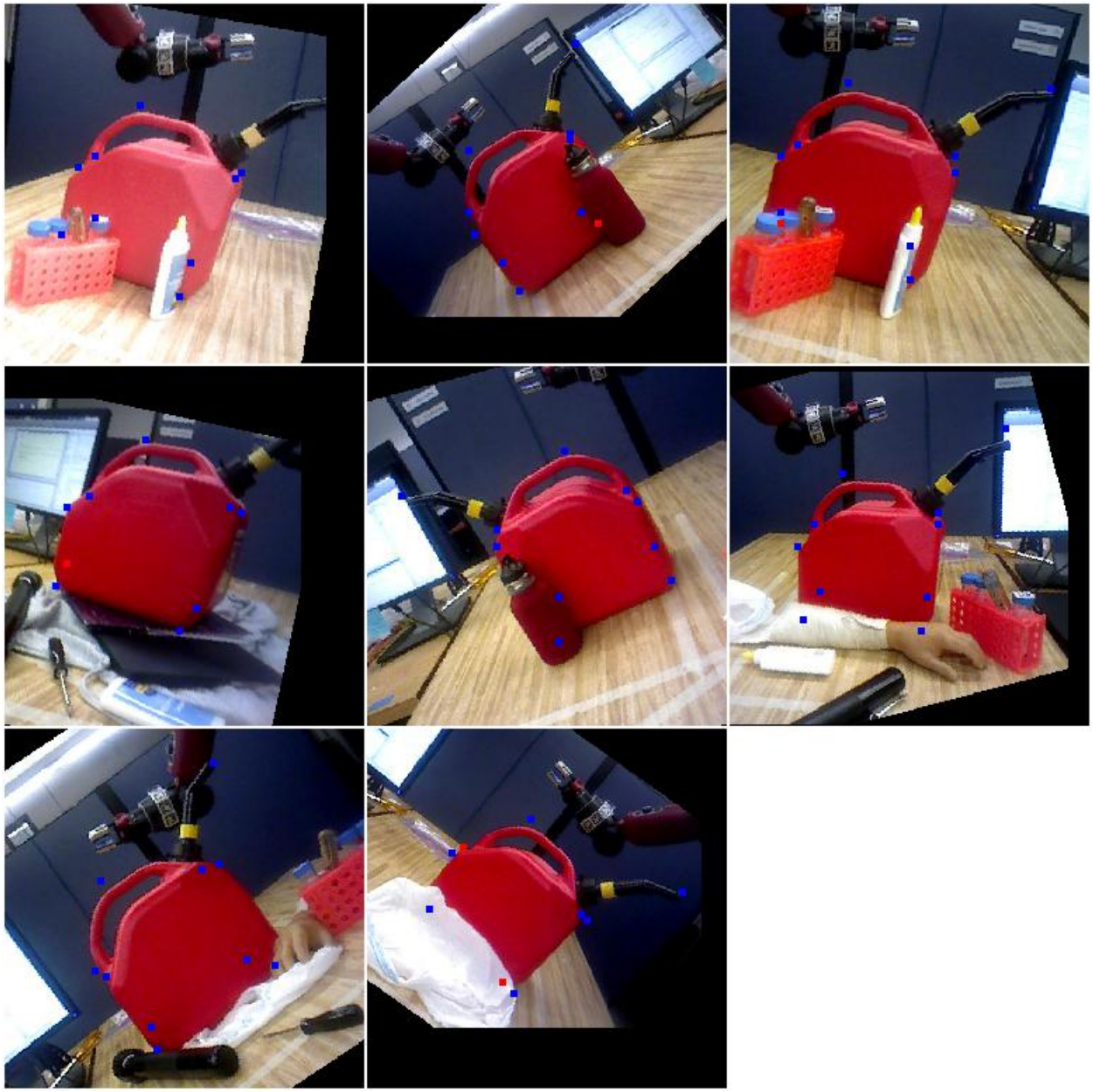


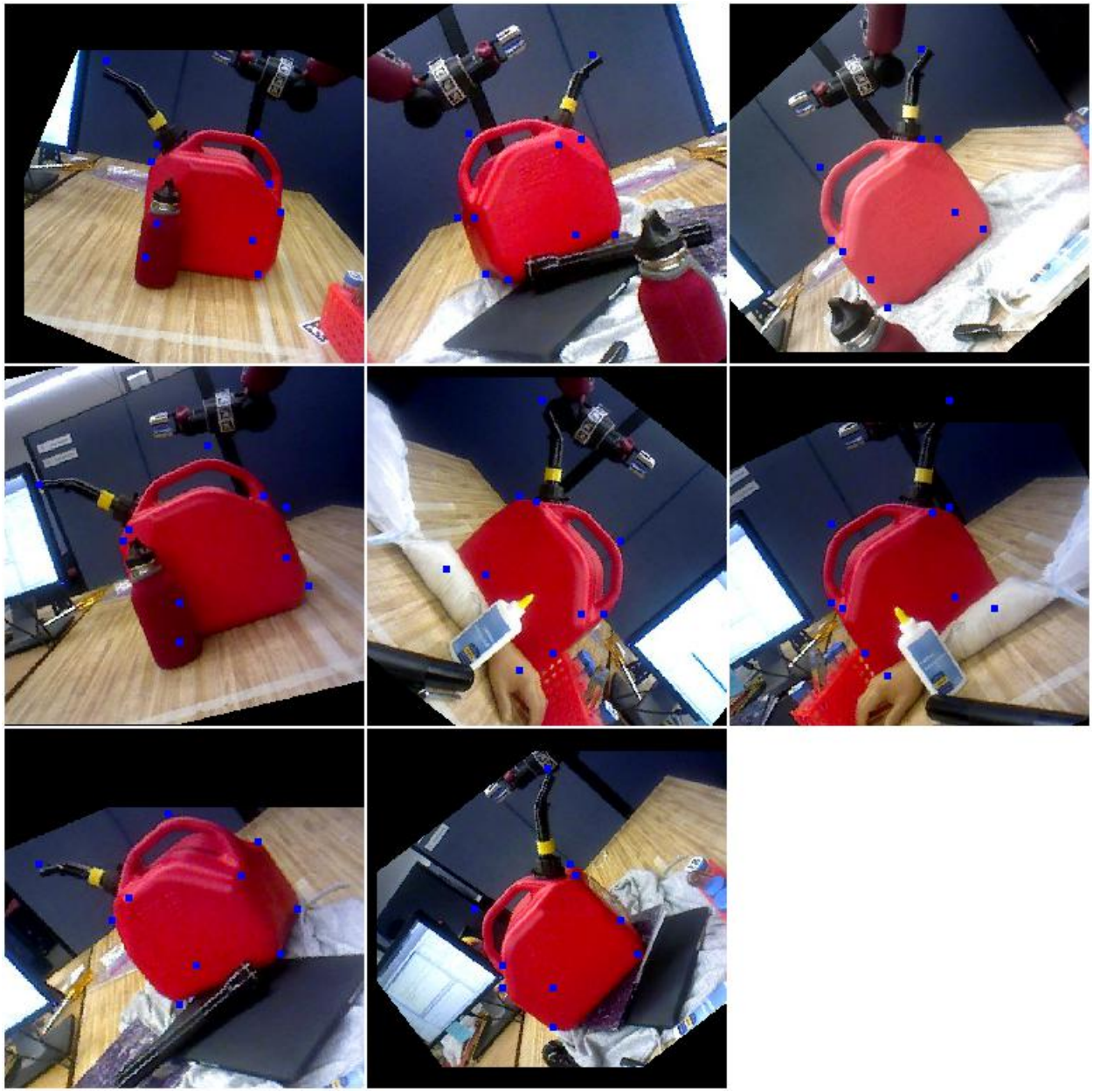
Prediction(order w.r.t number of training iteration)











As iteration increases, the number of uncertain dots decreases. The connection of neural network got more robust.