Team Members : 陳靖雯、蔡詩旭、王子駿

Instructor : Prof. Ren C. Luo

June 27, 2019

Advanced Robot Sensing and Control

Security Patrol Robot

1. Introduction :

In order to make the security guard robot, we need to add the face recognition and voice control function. Firstly, we need to make the robot recognize the people we know and have interactions with guests using voice. Finally, we need to control the robot to move to our desired position using voice. In addition, we will introduce the tracking skill we once added in the robot. Although we cannot show in demo due to some package incompatible, we will still try to illustrate the thing we have done and succeeded before.

2. Face recognition :

We use the existing package provided in https://github.com/ageitgey/face_recognition, and the basic tool is dlib(the package import in python code). In fact, it contains the CNN (convolutional neural network structure in there) the significant merit is that the speed of recognizing the face is fast, while the drawback is that we need a lot of rams.

Following is how the package work :

(1).Finding all the Faces(HOG algorithm)

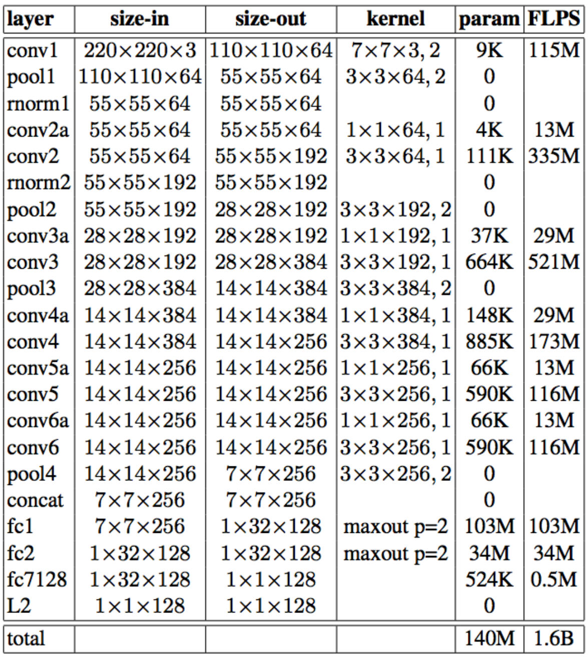
break up the image into small squares of 16x16 pixels each => making our image black and white => figure out how dark the current pixel is compared to the pixels directly surrounding it => repeat that process => find most similar to a known HOG pattern. That's so called Histogram of Oriented Gradients.

(2).Posing and Projecting Faces (face landmark estimation)

In order to recognize the same person with different angle, here the author use the face landmark estimation. The machine learning algorithm can help this part, such as SVM or XGBoost. The author choose 64 points for each image.

(3).Encoding Faces(CNN)

In order to deal with the problem of long time recognizing if we directly use the data provided in step (2), we have to use CNN with a specific loss function called triplet loss function which we have to load two image with the same person and another image with another person. This will help us not to retrain a model each time when we add another new face. The model is invented in 2015 by researchers at Google. Following is the structure provided by https://www.cv-foundation.org/openaccess/content_cvpr_2015/app/1A_089.pdf



layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

(4).Finding the person’s name from the encoding

From the above step, we can find a vector with size of (128, 1). In this step, the author use a simple linear SVM classifier and then output is the name of the person.

3. Tracking :

In this part, our goal is to use the package named “openni_tracker” to detect the skeleton of the detected person. After detected, we take the depth of one of detected point and make it move forward and turn with the movement of the person and stop under certain interval of the distance. The package we use to connect the kinect is "freenet"

4.Voice Control :

(1).Model:

Hidden Markov Model(HMM) is the model of speech. Other models correlated to speech recognition are acoustic model, phonetic dictionary and language model. The latter is to restrict word search.

(2).What is optimized :

Firstly, the so called Word error rate is defined as below,

WER = (I + D + S) / N

where I is the number of inserted words, D is the number of deleted words and S represent the number of substituted words.

Secondly, "Accuracy" is similar to previous one, but didn't take the intersection into account.

Accuracy = (N - D - S) / N

5.Conclusion :

I think the most difficult part of this project is to choose the proper package(once choosing the package, we need to check whether the vision is the same as the one in our ros software). In the beginning, we think it would have enough time to train our YOLO model used in face recognition, but it took us lot of time to figure out how to catkin_make those packages successfully. Besides, we found that it should be very careful while installing the additional package. For example, the first package we successfully install and integrate on the robot is the tracking process; however, while we install the face recognition, the whole previous effort cannot work anymore. Therefore, we are forced to give up this part.

Though our work seems quite easy compared with other group, we're still very grateful to have this precious experience to practically integrate the software and those algorithms on the real world robot, and it also let me learn a lot(including knowing about the structure of ros, how to check the topic and finally how to combine the python code with ros). It's still has a lot needed to learn, such as how to establish the launch file and how to build our own gazebo model. Therefore, thanks for giving our this chance to finish such a interesting project, and thanks those who helped us while we stuck somewhere.

6.Division of Work :

蔡詩旭：install all the package, debug the system incompatible problem while installing. Find the tracking and the voice control package for us to use.(He has the experience of using ros, without him, I can’t even know how to open the ironfish gazebo on my computer! He really helps us a lot)

陳靖雯：Prepare checkpoint ppt. Prepare final report. After 詩旭 catkin_make all the package successfully, I’m responsible for modifying the python code in the face recognition python file to integrate the face recognition with the robot(kinect), the voice control and the action of robot. Tracking python code modification.

王子駿：Prepare the final ppt. Test and study how to use face recognition pyhon code on the computer before checkpoint. Since he is the only one who has the right to enter the lab after the 6:00pm, without him, we would have difficulty in entering the lab to do some experiment.

6.Reference :

(1).https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

(2).http://wiki.ros.org/cv_bridge/Tutorials/ConvertingBetweenROSImagesAndOpenCVImagesPython?fbclid=IwAR2ShfEz-wLUTx6LvBnUbH5exyukB-AkOG8htSvDD8P9Comv2pshOewsC_0

(3).http://wiki.ros.org/cob_people_detection?fbclid=IwAR0wHPsArTkXjQXSqZMsDWyfo9zUtQAJ4GqPoVirpjLdBMHeqczHYgbIk_4

(4).https://blog.csdn.net/weewqrer/article/details/48183969?fbclid=IwAR2mKMHqsOwn6mXOUsxuFe7haiPSi9hQfsXSc9iVFsFvRMkFp2DQSgV6MyE

(5).https://github.com/mikeferguson/pocketsphinx/?fbclid=IwAR0Xb1UZZM6dbyvboxhaO0O6Db1AByIBgFzAEHPs2e1ZfijToup71Zrt-VI

(6).http://www.theconstructsim.com/construct-learn-develop-robots-using-ros/robotigniteacademy_learnros-2/