SAPIENZA
UNIVERSITÀ DI ROMA

Human Robot Interaction Project

Lorenzo Vitali

Università degli Studi di Roma "La Sapienza"

DIAG “A. Ruberti“

May 14, 2018

1 Idea

Nowadays there is an increasing trend regarding the implication of machines and robots, before it was more concentrated on autonomous machine performing iterative works or tasks, in the industries, but now it is changing, involving also robots in domains inside the society and around people.

The developing of this field has given a new aspect to different applications such that the entertainment, the elderly care, education, or communication. This activities are enhanced thanks to the ability of the robots to have a mobility inside an environment instead of only being a static screen attached to the wall showing information.

The level of technologies and innovation also permits to receive the help of robots for simple interaction tasks, in general, perceiving the requests of a human being through gestures, audio recognition or through the vision, and being able to answers to the requested needs. To perform this activities the robot needs to perceive the presence of a person, to trigger the event of recognizing a person and to prepare to interact with him.

The idea of this project came from this problem. It would like to propose a solution to build a system capable of discriminate different situations of a human robot interaction. Consequently permit a reaction depending on the resulting case.

1.1 Cases and dataset

The cases studied are three, the basic case when in the environment **no one** is present, and the other two regarding the persons. One to detect if the person is **close**, and the other if the person is **distant**.

So for the different cases the robot can start an interaction with the person or can come closer to perform the interaction at an acceptable user distance.

Due to the fact the problem is too specific, there is the lack of dataset targeted for this purpose.

In order to achieve this result, a proper dataset was built, capturing images for the train and for the validation phase, meanwhile the tests are done on real-time situation. To build the data, and to perform the project analysis the camera sensor used was used the **Asus xtion pro Live**.

Meanwhile in the third case, pieces of photos of empty room were used.

Figure 1: Example of train and test sample of the close class

Figure 2: Example of train and test sample of the far person class

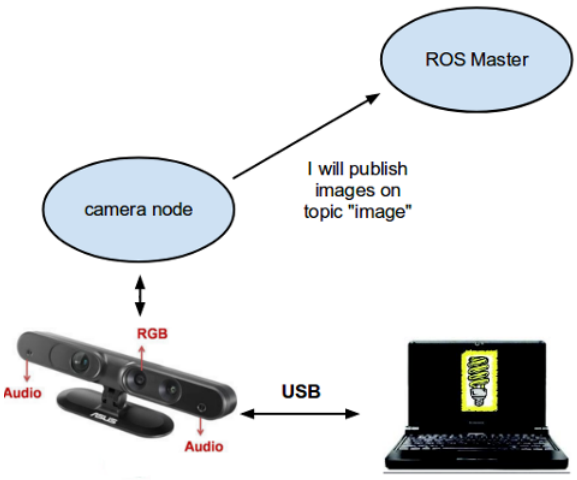

Figure 3: Asus xtion pro Live ROS connection

2 Sensor

2.1 Asus xtion pro Live

The module used to build the dataset and to test it in real time, is the camera shown in figure 3.

This device uses two different viewpoint to obtain depth information, is equipped with RGB can acquire images with the size 480x640, and Audio sensors. Furthermore it does not require a power supply, so only the USB connection is needed. It is supported by various OS, as Windows, Linux, and Android, and it can be programmed with Java, C#, C++. Thanks to all these features, and the prices contained with respect to other hardware as the Microsoft technologies, it is considered one of the best solution with respect to the developer point of view.

2.2 ROS

The camera is connected through Robot Operative System (ROS)[2], a set of libraries targeted to develop robots applications. It helps the developers providing operating system services, as for instance, driver connection, hardware abstraction, processes communication.

The basic elements of ROS are the nodes, they communicate between themselves sending and retrieving information, the mean where nodes can pass and read every kind of data are the topics.

In the case of this project the camera creates the topic **/camera/rgb/image_raw** to send the images. After that a software module implemented in python using the **openCv** libraries to take the image and convert it from the ROS format to an executable format.

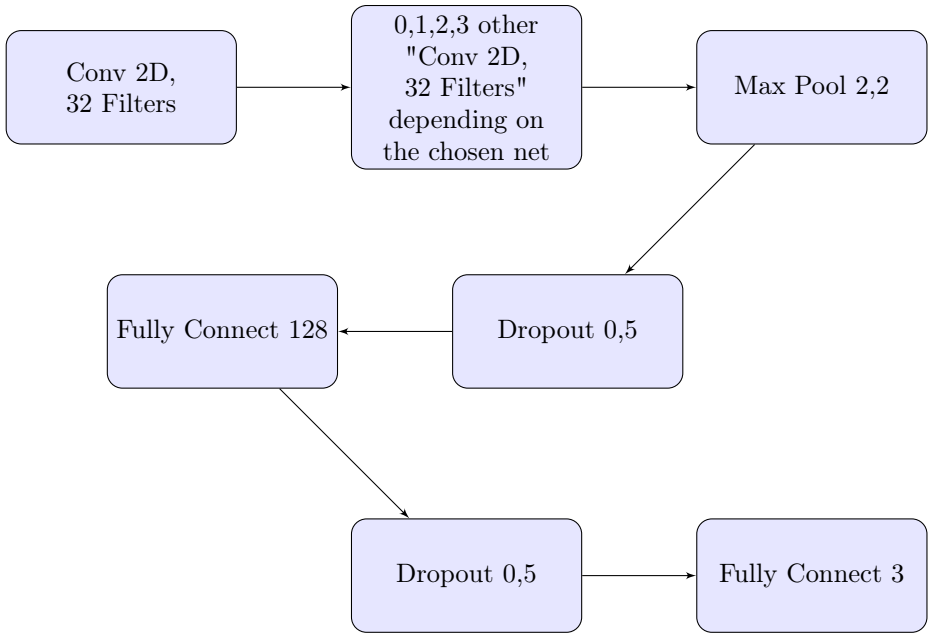

Figure 4: Structure of the Net

3 Method

The methodologies developed in this software module are neural networks. In the specific case these convolutional nets, were developed by the layers described in the figure 4, there are four nets, each one has different number of convolutional layer, in order to evaluate how the performances change with respect to the depth.

These nets take in input an image of size 240x120, the main reason of this shape is the object to classify that is, usually, with a long thin shape, as legs and human figure.

The **convolution** layers is needed to extract the features, each filter applied creates a **feature map**, that extrapolate local dependencies of the figures, relative to the orientations and the inclinations in the processed picture.

The main function of the max-**pooling** layer is to reduce the size of the data of each obtained map with all the advantages of a lower dimension input, as having less computation preserving important information. It captures a statistic of a certain area (usually the maximum has the best results), replacing that value instead of the all area.

The **fully connected** layers pass the features to a low dimensional space, trying to learn non linear relation from the higher dimension spaces and the new space. The last layer is a fully connected, its output has as many values as many classes have to be classified, this output is the classification.

The non linear relations that distinguish this method from a simple matrix multiplication, can be achieved through several functions like the sigmoid, the tanh and the ReLU. In this net the **ReLU** was used for all the nonlinear activations from the convolutional layers, to the fully connected, it maps as zero all the negative values, and let the positive values as they are.

3.1 Tests

The test are made using the camera in different environments and different situations.

To understand the possible issues of the method it is tested against different orientation of the camera, different persons, and different background. From these data it is possible to notice some issues and some limitations of this system.

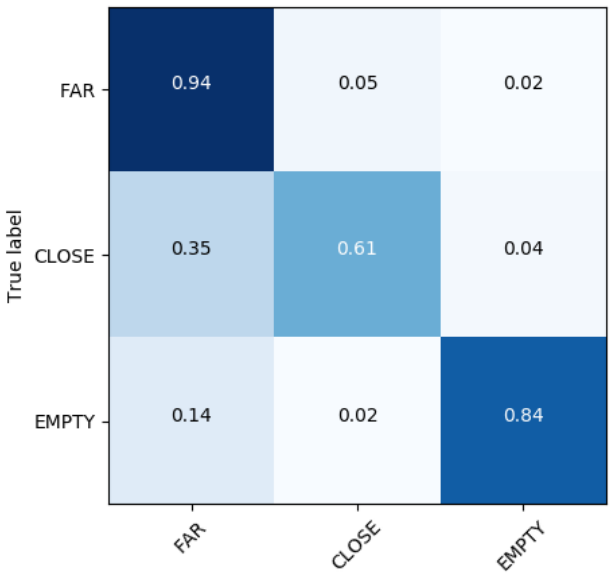

Figure 5: Confusion Matrix ResNet

One limitation regards the similar shapes, if there is a reflecting surface or a lucid floor, showing a shape very similar to a leg, or with a thin part and another part resembling a shoe, in most of the cases the built strategy will output the recognition of a close person.

Often happens the non-detection of the far person, probably it was because of the training images are obtained from the same camera inclination, this fact highlight important consequences.

The accuracy depends on the camera inclination, so it is a more difficult task if the position of the camera

changes, this can be expressed as **overfitting** of the training dataset.

The confusion matrix [3] shows how the system classifies, his bad and right predictions and how much a class tends to be misclassified for another specific one.

In the person cases we have a tendency towards the empty class, as the system was not completely able to capture properly the persons features. As said before, the shapes similar to the legs can be misunderstood by the classifier.

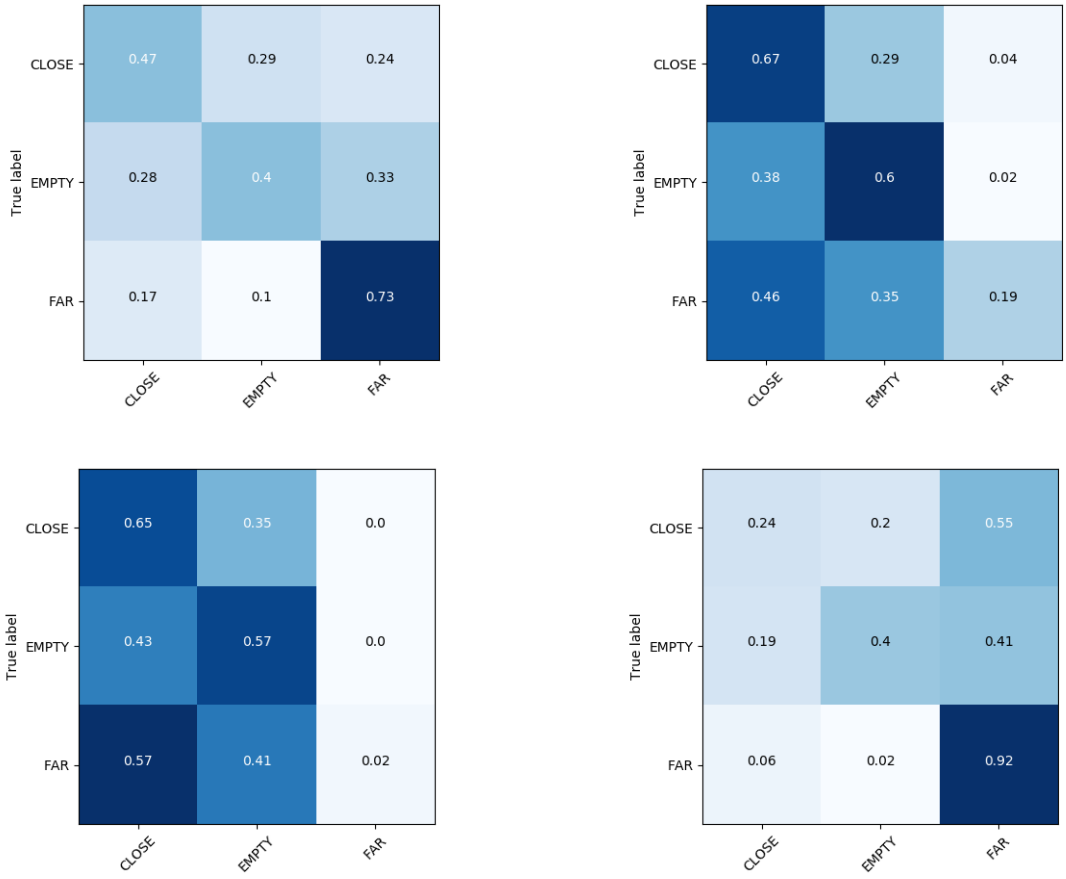

Figure 6: Confusion Matrices relatives to four different nets, starting from the first in the top left having only one convolutional layer, following till the fourth, that has four convolutional layers.

The nets created from scratch are compared to a pretrained ResNet fine tuned ad hoc for this case study. The **Residual Network** [1] has a different approach, instead of learning through the extraction of the features it learns from the residuals, the subtraction of the input of the layer and the features extracted. The aim of this different methodology is to not loose accuracy increasing the depth of the net, that can happen

due to the vanishing gradient issue.

The pretrained compared to the other examples, has the best performances, it outperforms the sensivity with a relatively large difference (around 0.3), and the difference of false positives of the empty class with a minimum gap. It shows how the latter case is better to properly recognize the classes, (rate of positive cases recognized as that) , furthermore it less often obtain a person class where no one is, having the less false positives for the empty class.

	Sensitivity	**False Positives empty**
conv-1	0.53	0.61
conv-2	0.49	0.40
conv-3	0.41	0.43
conv-4	0.52	0.60
ResNet	**0.79**	**0.39**

Table 1: Result comparison between networks

4 Robot Interaction

In this chapter there is the description of the test executed with the robot, more precisely with **Pepper**, a humanoid robot developed by Aldebaran Robotics.

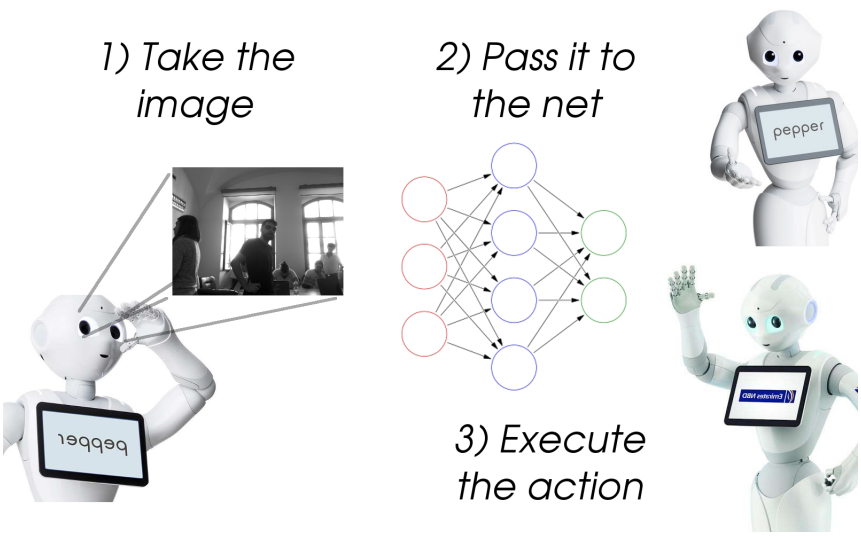

Figure 7: Workflow of the experiments

Figure 8: Image seen from the robot during the tests, classified as close person in the first picture and as a far person the second one

This robot is mostly used to interact and collaborate with people, thanks to its aspect and to its ability to express emotions through the gestures it fits perfectly roles as receptionist or assistant in conferences, events, meetings. Moreover it has the possibility to communicate with spoken language, and through the tablet positioned on its chest, where messages can be displayed.

Fig. 7 shows the workflow of the experiments conducted with Pepper. First step, an image is acquired from the robot front cameras, it is processed and passed as input to the pretrained ResNet. Once there is an output from the net, the action relative to the output is selected and it is sent to the robot. The last step is the execution of the command.

In these experiments the cases of interaction are two different, and the robot behaviour can be distinguished noting the what is displayed on the tablet (images and messages), what is spoken, and how the pepper moves its joints ([4] taken from a set of predefined animation).

In the fig 8 there are some examples of the images obtained from the cameras of the robot. Furthermore the video attached in the folder of this project show the robot behaviours in the different presented cases.

References

[1] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778., doi: 10.1109/CVPR.2016.90,

[2] ROS Docs,

http://wiki.ros.org/

[3] Confusion Matrix,

http://www.ritchieng.com/machine-learning-evaluate-classification-model/

[4] Pepper Gestures,

http://doc.aldebaran.com/2-5/naoqi/motion/alanimationplayer-advanced.html,