Teaching the NAO Robot to Play a Human-Robot Interactive Game

Cen Li
Department of Computer Science
Middle Tennessee State University
Murfreesboro, TN, USA
Cen.Li@mtsu.edu

Ebosehon Imeokparia Department of Computer Science Middle Tennessee State University
Murfreesboro, TN, USA
ei2j@mtmail.mtsu.edu

Michael Ketzner Department of Computer Science Middle Tennessee State University
Murfreesboro, TN, USA
mdk4g@mtmail.mtsu.edu

Tsega Tsahai Department of Computer Science Middle Tennessee State University
Murfreesboro, TN, USA
tyt2c@mtmail.mtsu.edu

Abstract— This paper presents the background and methodologies used in programming and teaching the humanoid robot NAO to play the game of "Simon Says" with human players. Choreographe programming was used to provide the overall game logic and incorporate NAO's sensory capabilities. OpenPose pose detection and OpenCV APIs were used to develop the image processing components to convert the raw images captured by NAO for pose classification, and the Keras APIs were used to build the Convolutional Neural Network to classify and recognize the player poses.

Keywords—robotics, computer vision, pose detection, convolutional neural network, image classification

I. Introduction

Recent years have seen a surge of interest in image classification using Artificial Neural Networks (ANN). Convolutional Neural Networks (CNN) and various architectures built based on the CNNs have received much success in many applications ([10][11][12]). There is also great interest in the field of humanoid robots and in developing human-robot interactive games ([5][13]). Humanoid robots typically are equipped with sensors supporting vision, speech, sound, movements, and great flexibility of joint movements. The goal of this project is to experiment with the visual, speech, and computational capabilities of the NAO robot [1] for playing a human-robot interactive game, and to explore the computer vision, image processing and classification methodologies that support real-time vision based games. We decide on making the NAO robot play the game of "Simon Says" where the robot acts as "Simon" telling the human player(s) to make pose(s) per his instruction. Each time, Simon calls out the name of a pose, the player makes the move. Simon determines if the player's pose is correct by comparing it to the pre-defined poses, and if it corresponds to the pose called out for. Once this part of the game was successfully developed, it was extended to work with two human players, playing side by side, and competing against each other. In the rest of the paper, a brief description of the methodology used in the project, as well as the project results are discussed.

This work is funded by National Science Foundation grant for project number 1742518.

II. Background

A. Pose Detection

Pose Estimation is a general problem in Computer Vision that detects the position and orientation of an object. This usually means detecting key point locations that describe the object ([2][3][6][7]). In games like "Simon Says", player pose detection refers to the task of detecting and localizing the major joints of the player body (e.g. shoulders, arm, wrist, knee, etc.). Vikas Gupta's multi-person pose estimation package "OpenPose"[6] has been adapted for this work. OpenPose first processes the raw image with a combination of 10 Pooling and CNNs that create the feature maps for the input image. The feature maps are then sent to the two branch multistage CNN architecture. The first branch predicts the set of 2D confidence maps of body part locations, and the second branch of the architecture predicts the set of 2D vector fields of part affinities which encodes the degree of association between parts ([2],[6]). This method may be used with different image data sets. The Common Objects in Context Dataset (COCO) [9] and the Max Planck Intitut Informatick Human Pose Dataset (MPII) [2] have been experimented for the classification task in this project. The COCO data set, including train, validation, and test sets, containing more than 200,000 images and 250,000 person instances labeled with keypoints. The MPII Human Pose dataset includes around 25K images containing over 40K people with annotated body joints. The images were systematically collected using an established taxonomy of every day human activities. It covers 410 human activities and each image is provided with an activity label [2].

B. Image Classification using CNN

In recent years, Artificial Neural Networks (ANN) have been successfively used for classifying images in various applications. In particular Convolutional Neural Networks have been proven to be effective and are gaining more attention in the field of image classification. Some recent succes stories include applying CNNs for recognizing and classifying remotely sensed images [12], for classifying blood cell images [10], and for wafer map image classification for defect pattern analysis [11], to name a few. Compared to other Neural Network structures, CNN takes advantage of the fact that pixels nearby to each other are often highly correlated for image detection. Each CNN layer looks at an increasingly larger part of the image. Having units

978-1-7281-5584-5/19/$31.00 ©2019 IEEE
DOI 10.1109/CSCI49370.2019.00134

only connected to nearby units also aids in the invariance problem. In this work, CNN is used to classify binary stick figure image data generated based on the locations of key (joint) points found from the player images. The stick figure images are similar to handwritten characters, e.g., the MNIST handwritten digit classification data. Therefore we developed a CNN similar to the one developed for handwritten digits recognition [8] using the Keras APIs [4].

III. METHODOLOGY

The project methodology can be divided into three main parts: (1) Programming NAO behavior in the Choreographe environment; (2) Process the player images received by the visual component of NAO such that the images are converted into a form suitable for learning; and (3) Applying machine learning techniques to learn the classification models corresponding to the poses selected for the game.

A. Programming NAO in Choreographe

Choreographe is a visual programming environment, where programmers code the needed bahaviors as behavior blocks and save them to the library [1]. When developing a program for a specific task, blocks of bahaviors are dragged onto the main programming area and connected with directed lines indicating the flow of data and program control. Fig. 1 shows the Choreographe programming environment with the top level program logic partially displayed in the main window.

The main program logic for the "Simon Say" game consists of the following steps:

1. Give game instruction to the players via NAO Speech module;
2. Repeat the following 4 steps for a preset number of iterations:
   - 2a. Announces a pose selected randomly from the set of game poses;
   - 2b. NAO vision captures the two players' poses, and passes the image to be processed and classified;
   - 2c. Based on the pose classification results obtained, NAO gives feedback to each player;
   - 2d. NAO updates the two players' game points;
3. Announces the final game points to the players and determines the winner.

B. Player Image Processing

B.1 Data Collection

The initial proof of concept was conducted using 3 different poses. Three team members modelled for these poses. To increase the robustness of the learned classification models, the poses were made with deliberate variations. For each pose, a team member would stand at each of the five locations to mimic players standing at closer, farther away, or off center from the robot in real game setting. At each location, the poses were also made slightly differently each time, for example by raising the arms slightly higher or lower than the perfect pose, or make head and body slightly turned or bended. All together 130 image training data were collected. At the end of the project, the number of poses was extended to 8, and the training data size was increased to 405.

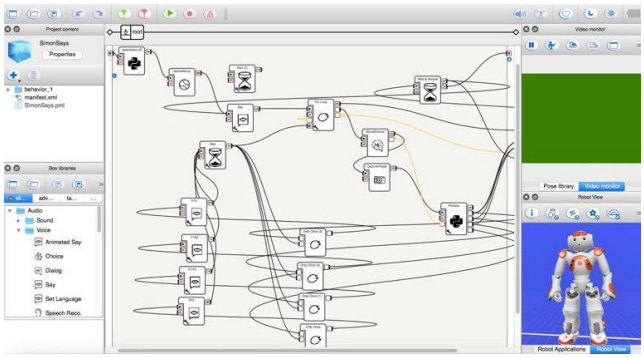

Fig. 1. The main NAO program logic programmed in Choreographe

B.2 Data Preparation with Image Processing

The images received from step 2b of the Choreographe program is processed using the OpenPose package and the OpenCV APIs to convert the raw images into a representation suitable for learning. Each raw image contains one or two players standing in front of a mixed background consists of door, wall, lockers, furniture, etc. Fig. 2 shows a raw image making one game pose.

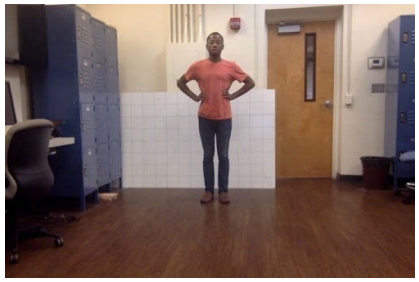

Fig. 2. Raw player images captured by the NAO robot

Two approaches have been experimented to convert the images. Both approaches are based on results generated from Gupta's multi-person pose estimation package "OpenPose". We experimented with two pose estimation data sets [6] : the Common Objects in Context Dataset (COCO) [9] and the Max Planck Intitut Informatick Human Pose Dataset (MPII) [2]. The MPII dataset was simpler to use, yet the COCO dataset led to a higher pose estimation accuracy. Utilizing the COCO model allowed for obtaining a collection of 18-pixel coordinate points based off each raw image. These points would define significant skeletal points on the body (i.e. shoulder, elbow, wrist) encompassing 5 major parts of the body: The head [pts: (neck and nose) 0-1, (ears and eyes) 14-17], the arms [pts: (right arm) 2-4, (left arm) 5-7], and the legs [pts: (right leg) 8-10, (left leg) 11-13]. Fig. 3 presents an image where the 18 key points are clearly identified. These 18 key points are computed in the form of the actual pixel coordinates on the image.

To compensate for the variations of the player standing locations, therefore the variations of the locations and scales of the model appearing in each image, the 18-pixel coordinate points were further post-processed. In the first approach, we experimented with translating and scaling the points about the center axis of the pose identified in the image,

as well as computing their relative locations among wrist, elbow, shoulder, and head. These transformed points were used to form the training data for the pose learning step, i.e., each image is represented with a vector of 18 post-processed location data. Using a typical 3-layer Sequential neural network model implemented with the Keras APIs, this training data lead to an average pose classification accuracy of less than 70%. This prompted us to develop an alternative way to process the key points to increase the classification accuracy.

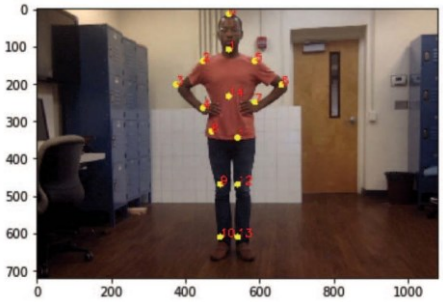

Fig. 3. Poses with 18 coordinate key points

The second approach for data transformation gets inspiration from the fact that CNNs have been successfully applied for character recognition, where the characters were represented in the form of binary images. At an abstract level, the poses themselves are like the alphabetical characters, it is our conjecture that if the poses are converted into character-like binary images, CNN could be used to classify them with high accuracy. In addition, success in human torso and hand location tracking have been reported where color images were first converted into binary images [7]. To track the torso and the hand of a player, the raw input image was converted into a binary image. A torso-shoulder-upper body model was used to locate the position of torso from within the foreground image, and the skin color is used to locate the position of the hand, as shown in Fig. 4 [7].

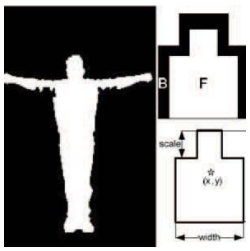

Fig. 4. torso-should –upper body model used on binary segmented player image

We combined these ideas into developing the data transformation approach. Our method takes the key coordinate points identified from OpenPose and form a binary stick figure image using the OpenCV APIs in the following steps:

1. Form line segments based on pairwise coordinate points,
2. Compute and expand the line segments to contours, and
3. Find the width and height of the stick figure, and crop the image so that the stick figure occupies the entire image.

Fig. 5 shows three stick figure binary images representing three different poses. It is quite clear that the stick figures clearly capture the main characteristics among different poses in terms of the placements of the arms and the legs.

C. Pose Classification

The Keras APIs were used to construct the CNN to learn the pose classification model based on the training data. Fig. 6 illustrates the CNN structure with 2 layers of convolution and subsampling, where the convolution is set to the size of 3 by 3 and the subsampling size set to 2 by 2. Rectified linear unit (Relu) has been used as the activation function. The CNN successfully learned the classification model of 8 poses with an accuracy consistently above 98%. A 10-fold-cross validation is performed where the classification accuracy is 99.3%.

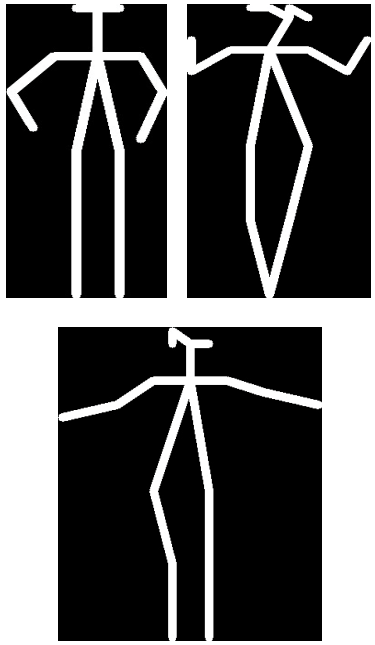

Fig. 5. Binary stick figures representing 3 different poses

D. From One Player to Two Player Game

To allow two players to play side-by-side and compete, additional steps were needed to process the image captured by NAO. First, the image was cropped into two halves. Then, from each half of the image, the key-points were generated, and a stick figure image was formed. After that, each image was fed into the Keras classification model and it predicted a pose and sent the results back to NAO. With this approach, tracking each player was easy. Player 1 is always on the left side and player 2 is on the right side.

IV. Results and Discussion

Separate modules were developed for training data image pre-processing, classification model learning, and the actual game server. This allows for easy updates for game development. Because the player images captured during the game still need processing before classification may be performed, and because the processing power on the NAO robot is limited, a laptop was used for this step.

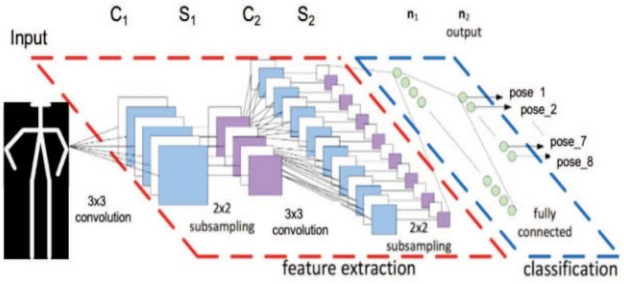

Fig. 6. The CNN model used for classifying and recognizing the pose models

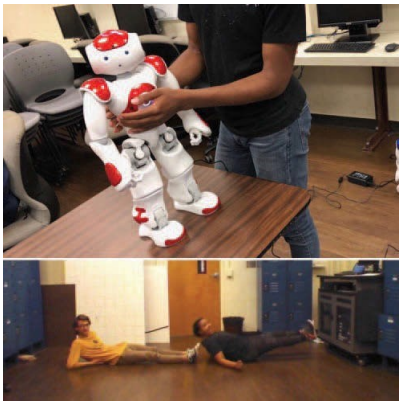

Fig. 7. "Simon" the robot asks the player to lie down

"Simon", the robot, gives out the instruction about which pose to make, the two human players make the pose. Simon takes the picture of the two, send it to the laptop for raw image to stick figure conversion, and for pose classification. The classification result is sent back to Simon, based on which a verdict is announced by Simon concerning whether each of the players made the right move. Initially, during testing, there was a processing time delay of over 3 seconds after Simon issuing each command. The first step taken to reduce the delay was to reduce the dimension of the input image used to generate the key points. Generating the key points was the most processor-intensive task for this game and by reducing the size of the image, it was possible to generate the points faster. The second step taken was multiprocessing. This step allows for the key points for both players to be generated simultaneously. While it was not as significant of a reduction as reducing the image input dimensions, it helped. With those improvements, the delay is far less noticeable than before.

Test run the game with invited players revealed some problems not previously encountered. For example, when Simon says to lie down, the team members all know to lie down sideways, as shown in Fig. 7. But for new players, they all took it as lying down in the front-to-back direction, making the classification process not feasible. Also, some other wording and commands issued by Simon were not readily comprehensible by non-developers, and the Choreographe program was modified accordingly.

A problem we would like to work on in the future is how multi-player image is handled. Given our current method for image processing, NAO won't know if a player switches to the other side. Also, this method only allows two players to play. While it is possible to crop the image into 3rds/4ths/etc., when more players are involved, key point generation would not be as accurate and the delay time would increase accordingly. We would like to address these problems by leveraging more about NAO's built-in facial recognition software.

ACKNOWLEDGMENT

The authors would like to thank the NSF for funding this research work.

REFERENCES

[1] Aldebaran Roboticss, "SoftBank Robotics," SoftBank Robotics, Inc., 2019. [Online]. Available: https://www.softbankrobotics.com/.

[2] M. Andriluka, L. Pishchulin, P. Gehler and B. Schiele, 2D Human Pose Estimation: New Benchmark and State of the Art Analysis, "IEEE Conference on Computer Vision and Pattern Recognition (CVPR)", June 2014.

[3] Z. Cao, T. Simon, S. Wei, Y. Sheikh, Multi-person 2D Pose Estimation using Part Affinity Fields, in Proceedings of Computer Vision and Pattern Recognition, July, 2017.

[4] F. Chollet and others, "Keras," 2015. [Online]. Available: https://keras.io/.

[5] K. Dautenhahn, A. Billard(2002) Games Children with Autism Can Play with Robota, a Humanoid Robotic Doll. In: Keates S., Langdon P., Clarkson P.J., Robinson P. (eds) Universal Access and Assistive Technology. Springer, London

[6] V. Gupta, "Deep Learning based Human Pose Estimation using OpenCV (C++ / Python)," 29 May 2018. [Online]. Available: https://www.learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/.

[7] F. Huo, E. A. Hendriks, A.H.J. Oomes, P. Beek, R. Veltkamp, Detection Tracking and Recognition of Human Poses for a Real Time Spatial Game, GATE (Game Research for Training and Entertainment) Project, pp 43-51.

[8] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based Learning Applied to Document Recognition, in Proceedings of the IEEE 86 (11), 2278-2324.

[9] T.Y. Lin, G. Patterson, M. R. Ronchi, Y. Cui, M. Maire and P. Dollár, "COCO 2018 Keypoint Detection Task," 17 August 2018. [Online]. Available: http://cocodataset.org/#keypoints-2018.

[10] G. Liang, H. Hong, W. Xie, L. Zheng, Combining Convolutional Neural Network with Recursive Neural Network for Blood Cell Image Classification, in IEEE Access (Volume: 6) DOI: 10.1109/ACCESS.2018.2846685

[11] T. Nakazawa, D. V. Kulkarni, Wafer Map Defect Pattern Classification and Image Retrieval Using Convolutional Neural Network, in IEEE Transactions on Semiconductor Manufacturing, Volume 31, Issue 2. May 2018, pp 309-314.

[12] M.E. Paoletti, J.M Haut, J.Plaza, A. Plaza, A new deep convolutional neural network for fast hyperspectral image classification, in ISPRS Journal of Photogrammetry and Remote Sensing, Volume 145, Part A, Nov 2018, pp 120-147.

[13] J. Wainer, B. Robins, F. Amirabdollahian, K. Dautenhah, 2014, "Using the Humanoid Robot KASPAR to Autonomously Play Triadic Games and Facilitate Collaborative Play Among Children With Autism", IEEE Transactions on Autonomous Mental Development, Volumne 6, Issue 3, pp 183-199.