Sketch recognition library applied for an image replication with a humanoid robot in a simulated environment

1st Steven Pacheco-Portuguez
Escuela de Computación
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
stpacheco@ic-itcr.ac.cr

2nd Martín Naya-Varela
Integrated Group for Engineering Research
Industrial Department, Universidade da Coruña
A Coruña, Spain
martin.naya@udc.es

3rd Francisco Bellas-Bouza
Integrated Group for Engineering Research
Industrial Department, Universidade da Coruña
A Coruña, Spain
francisco.bellas@udc.es

4th Esteban Arias-Méndez
Escuela de Computación
Instituto Tecnológico de Costa Rica
Cartago, Costa Rica
esteban.arias@tec.ac.cr

***Abstract*—Sketches have been one of the most ancient techniques used by humans to portray their ideas and thoughts. Replicating this ability would help us to better understand the way in which human beings obtain their capabilities. In this work, we implemented an architecture using convolutional neural networks capable of transforming an image to a sequence of strokes to be replicated by a Poppy humanoid robot using inverse kinematic to reproduce the sketches.**

***Index Terms*—Inverse kinematic, neural network, stroke, sketch, convolution, humanoid.**

I. Introduction

The action of drawing is one of the nonverbal and visual forms that humans have to create representations of a wide variety of things. This behavior requires basic cognitive concepts and skills to process the visual information to later execute the reproduction of the corresponding lines [1]. Knowing the process by which a robot can recognize and subsequently perform replication of objects in a drawing, can help to understand how these movements are executed and what are the similarities the making of drawings with a robot and that of the human being.

Understanding these skills represents a challenge due to multiple factors that affect and generate different styles of drawings among people, associated with both the development of characteristics and cultural processes in their growth [2]. Recent research has used generative models such as adversarial neural networks or convolutional LSTM [3], to create images simulating this replication.

Nevertheless, these approximations are based on the treatment of pixels as stated in [4], human beings do not see the environment as a pixel matrix, therefore these alternatives are considered less real compared to the human being's perception to be transformed into movements.

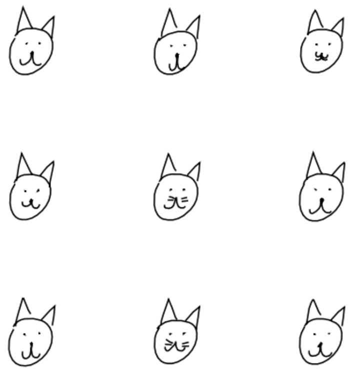


Figure 1. Multiple examples of outputs corresponding to the sketch-pix2seq architecture where the drawing is based on the strokes of the cat class

An alternative is to represent the images as a sequence of strokes as presented by the sketch-seq2seq architecture [4], however, this is based on memorizing the strokes to be drawn and not on understanding the sketches.

Starting from this problem it is necessary to use another type of architecture, which allows us to better emulate the ability of human beings to capture a drawing through the sense of sight and then redo it. For this, the sketch-pix2seq model [5] presents a feasible way to do it, as shown in Figure 1.

Following the investigation line of humanoid robots within the Integrated Engineering Group at Universidade Da Coruña in research projects as the influence of morphology on learning ability and locomotion, this work focuses on the application of a library recognition system for image replication with a robotic device in a simulated environment.

The robot selected for this research is the Poppy humanoid robot, created by the Flowers team at INRIA laboratories in France [6]. Poppy is inspired by basic aspects of human morphology, focusing mainly on the structure to simulate human movements in a more fluid and natural way [7].

In Section II, several works used to solve the cognitive problem of drawing are presented, as well as applications of these techniques in robotic systems. In Section III, the proposed architecture is described for image replication by the Poppy robot and which technologies were used in this work. Furthermore, the dataset description that was used for model training.

In Section IV, the experiment execution is described as well as the training process of the sketch-pix2seq model employed and the processing of the output of the model to obtain absolute points to draw. In the Section V, the results obtained are described and a code repository for this project can be found at https://github.com/Stevenpach10/Practica2019-UDC, videos and images of examples obtained from this project. Finally, there are conclusions and acknowledgments for this project.

II. Drawing replication models

A. Learning to draw

Drawing is a behavior that allows human beings to express different feelings, ideas, and knowledge, however, it requires complex skills to capture the information from the environment that allows them to recognize objects and later be replicated. A fundamental component of the drawing is the relationship between the vision input mechanisms and motor orders [8].

A proposed model for the treatment of sequential images is based on the Convolutional LSTM architecture [3], however, from one image to another, it can present more than one figure, trace, or line for the next prediction and it approaches the problem as pixel treatment.

Another alternative suggests a model based on the representation of an image as a sequence of strokes, proposing an alternative to traditional pixel models, nevertheless, its sketch-seq2seq architecture [4] simulates the process of remembering the drawing process and not understanding the figures that compose it unlike the sketch-pix2seq architecture [5].

Another proposal extends the QuickDraw dataset, to serve as real-time guides to freehand sketches, to show suggestions on the next stroke to draw, using Siamese CNN and LSTM networks [9].

B. Development of drawings with robotic systems.

Carrying the capabilities to draw from software systems to hardware such as a robot, represents a challenge because there are other types of variables to consider, although several studies have shown that it is possible to develop these capabilities in robotic devices. For example, a learning model based on catastrophe theory that uses primitive characteristics of figures called critical points [10] and a model that is trained to reproduce lines given a combination of primitive movements of a robot.

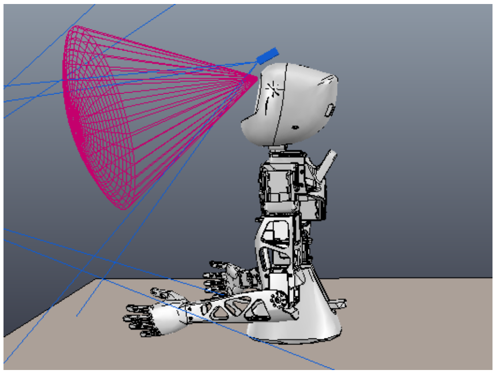


Figure 2. V-REP simulator example with the humanoid Poppy Torso and its vision sensor

Other models of computational art have been designed [11] which apply specific drawing rules for the desired images or using the NAO robot to make drawings defining the points of the image concerning the final effector and provide a solution with inverse kinematics [12].

Another model proposal is to use two deep neural networks Autoencoder and Multiple Timescale Recurrent Neural Network to learn to draw sequences with the humanoid robot NAO [13], but with sequences of a maximum of 15 strokes, which limits the ability to the drawings that can be made, likewise proposed an adaptive drawing model where it is capable of completing images given certain initial strokes [14], or the proposal of the implementation of inverse kinematics for a robotic arm with three degrees of freedom in a two-dimensional plane [15], thus limiting movements.

III. Methodology

In this section, we first define which technologies are used to carry out the project, later describe the dataset used, and next the proposed model is introduced.

A. Technologies used

One of the principal technologies used was Python [16], the programming language in which the Pypot library is written, which aims to control aspects of the robot such as sensors and motors. The version used for this work was the modification proposed by the Integrated Group of Engineering of the Universidade Da Coruña [17], attaching a vision sensor for the robot allowing the visualization of the environment among other characteristics. Furthermore, the OpenCV library was then used for image processing of the vision sensor.

Another software tool used was the V-REP [18] simulator, which is a simulation environment that allows the creation of scenes in which it contains robot models and their respective sensors, allowing interaction with a remote client from the use of APIs. Figure 2 shows a screenshot taken from the V-REP simulator that corresponds to the Poppy Torso robot with the vision sensor installed and the image capture range it has.

B. Data set

The dataset used was developed by Google researchers. This dataset is a vector of detected drawings from an online game called *Quickdraw!*[4], where players are asked to draw a particular kind of drawing in less than 20 seconds. It is available at the following link: https://github.com/googlecreativelab/quickdraw-dataset.

The dataset consists of a collection of 50 million drawings from 345 categories, contributed by Quick Draw! Game players. These drawings were captured as vectors and labeled due to the sketches in the drawings are represented as a set of strokes, corresponding to the displacement relative to its last position, except for the first element that is located in its absolute position. It was necessary to pre-process the images to convert those image traces into .png format to obtain 48x48 size images that can be useful in the training process.

C. Model

As Chen et al. [5] mentions in their work, the structure used by the sketch-pix2seq model is similar to that proposed in sketch-rnn by making some modifications. One of them, which is useful for our work is to change the recurring bidirectional neural network to a convolutional layer so that it allows us obtain an image as input data.

Applying the previous model we obtain a list of points from the sketch where each point is shown in a vector of 5 elements (Δx, Δy, $p1$, $p2$, $p3$) where x and y represent the displacement on the respective axes $p1$ indicates if the pencil is touching the surface and should connect to the next point, the state $p2$ indicates if the pencil should stop touching the surface and not connect to the next point and $p3$ indicates found at the end of the drawing.

This allows us to obtain resulting images following a sequential and logical process similar to how a human being performs it.

For the image replication on the Poppy robotic device, you need to tell it the set of values α, β, γ to reach a certain position in space. Where α represents the value of the shoulder joint, β the value of the arm joint and γ represents the value of the elbow.

In order to find these values, inverse kinematics has been applied where, given a certain point in space, the value of its joints can be obtained. In this way, the robot arm effector can be placed in a specific position. By performing this process for the entire network output sequence it is possible to successfully obtain an image as a consequence of its movements.

Figure 3 describes data flow of the proposed design, where the image is first obtained with the vision sensor of the Poppy robot and this serves as input to the sketch-pix2seq model which results in set relative points that are subsequently transformed to obtain a set of absolute points, which will be achieved according to the movements made by the robot applying inverse kinematics.

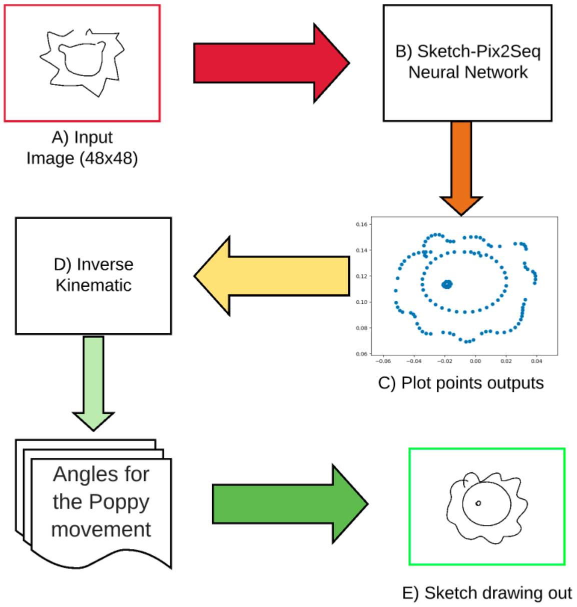


Figure 3. The pruposed design flow for replicating images with hte Poppy robot. A) Input image for the Neural Network. B) Sketch-Pix2Seq will process the image and will transform into strokes. C) Represent all points for drawing the image input. D) To calculate all angles of the Poppy's joints for drawing with inverse kinematic. E) Sketch result

IV. Experiment

The proposed experiment is based on the replication of a sketch of an image which is observed by the Poppy robot and is obtained through its vision sensor available in the modifications made in the Pypot library. The image is captured at a size of 512 x 512 pixels and rescaled to a size of 48 x 48 using OpenCV, to serve as input for the sketch-pix2seq network.

The experiment was carried out in two ways, the first is to train the model with a single class to determine the learning ability of the Poppy humanoid. The second experiment consisted of observing the behavior of the model with multiple classes, to observe the motor capacity of the learning process of different abstract concepts.

Next, the training process of the neural network is presented and how the transformation of the traces to absolute points was applied.

A. Model training

The training process follows the steps described by [5]. For this specific project, we used the data set belonging to the cat class available in QuickDraw! for the case of a single class, and for the multiclass example the bear, bicycle, and lion classes were used, this selection was made randomly.

Since the development of network optimization requires a high availability of resources both in terms of calculation and time, it was necessary to use a large amount of resources. The calculations were performed using the high-performance computing supercomputer "Finis Terrae II" at the Supercomputing Center of Galicia (CESGA, http://www.cesga.es). The

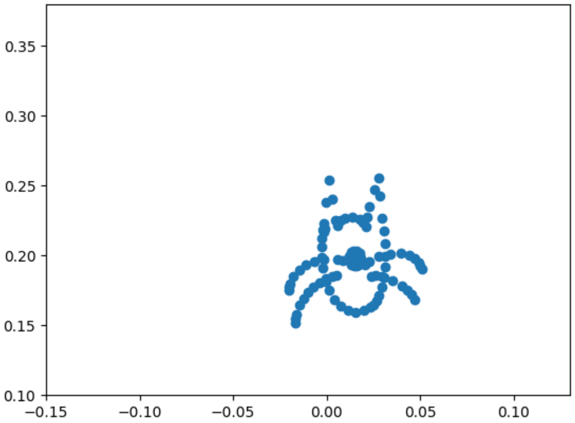


Figure 4. All absolute points for drawing a cat, this is the result of applying the stroke transformation to the output of sketch-pix2seq.

approximate duration of training was 5 days per experiment using a K2 GPU, reaching 60,000 iterations.

B. Stroke transformation

The sketch-pix2seq output consists of a list of points corresponding to the drawing, represented by the offset from the previous point with the exception of the starting point. Due to the shape of the output, it is necessary to apply a transformation to the absolute position of each of the points.

The Poppy robot must execute the corresponding movements to the axis X and the axis Z to draw the shape of the drawing, the axis Y represents when it is or not touching the canvas, while the robot is performing a stroke, while the robot is performing a stroke this axis should remain stable over the canvas on which it is drawing. Figure 4 represents the absolute points corresponding to an example of the cat class.

V. Results

A. Single class

For the experiment with a single class, we have the capture of the image by the robot's vision sensor in Figure 5(a) and the image produced as a result in Figure 5(b), once applied the transformations of the model's output points, it can be seen that they have similar main characteristics, but it is not an exact duplication of the drawing.

Figure 5(c) shows the result obtained by the traces taken as correct within a threshold, with this it is necessary to highlight that not all the movements executed by the robot to reach a position were correct, that is, more than one movement were required in certain cases in order to reach a good position of the effector within an acceptance parameters.

An error of 0.02 meters has been taken as a threshold as shown in the Pseudocode 1. Where the function *mov_poppy_arm* receives what will be the acceptance threshold to be used in a vector form corresponding to a threshold for each axes, the point to be reached by the effector, the Poppy chain corresponding to the arm to move, and a number of tries that the algorithm must be executed before the end

The *calc_difference* function receives as parameter the point which should reach and the end effector of the chain, in this case, corresponding to the Poppy's left arm, this function calculates the absolute difference between each of the axes of the effector and the point to be reached.

```
def mov_poppy_arm(umbral, point,poppy_chain
,limit):
    i = 0;
    poppy_chain.goto(point)
    effector = poppy_chain.end_effector
    diff = calc_difference(point, effector)
    while((diff[0] >= umbral[0]
    or diff[1] >= umbral[1]
    or diff[2] >= umbral[2])
    and iteration <= limit):
        poppy_chain.goto(point)
        effector = poppy_chain.end_effector
        diff = calc_difference(point,
        effector)
```

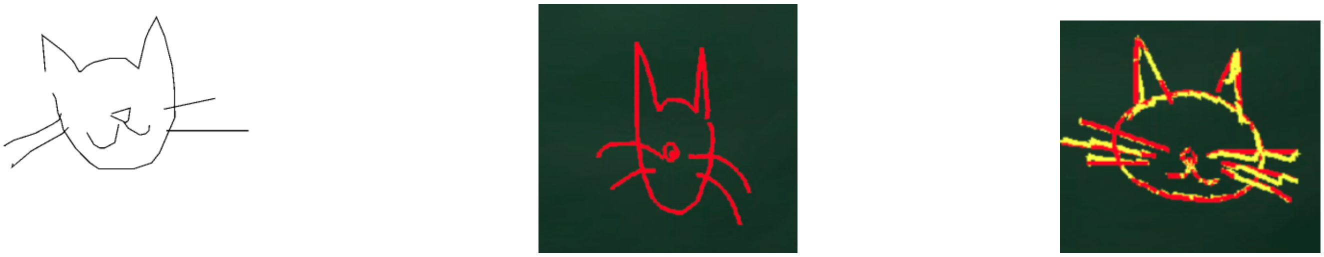
Algorithm 1. Algorithm used to achieve proper movement of the Poppy robot's effector.

Figure 5(c) shows the line corresponding to the union of each point, from the starting point to the end in red, while, all the movements made by the Poppy robot effector are traced in yellow, including all corrections of the effector that has to perform when a movement that is not within the specified threshold is executed. It can be seen that on certain occasions it takes more than one movement to reach a point causing more strokes.

To justify these results, it is necessary to observe the tracing for each of the axes of the effector. The positions of the axes effector X and Z correspond to the sketch tracing as shown in Figures 6 and 7 while the axis Y belongs to the profundity. This axis is determined when the pencil is touching the canvas on which it is being drawn or not. This produces a total of 120 correct movements that must have been made to complete the drawing in Figure 5(c)

As discussed in subsection IV-B the Y axis corresponds to if the Poppy robot is touching the canvas or not and always be maintained at the same value.

For the execution of the movements, inverse kinematics are used to calculate the angles of the motors of the chain of the arm of the Poppy robot. It can be seen in Figure 8 that the scale is negative, which means that the value 0 on the Y axis corresponds to the position where the robot is situated. The most negative values represent when the canvas is being touched while the minor values represent when the canvas should not be touched to finish a line and start a new one,



(a) Image corresponding to the QuickDraw cat class

(b) Image resulting from applying the point transformation to obtain the strokes

(c) Image resulting from applying the point transformation to obtain the strokes

Figure 5. (a) The image was loaded in a board from Quickdraw cat class dataset and later the image was taken with the Poppy's vision sensor. (b) It is the result of applying the transformation of output points of the sketch-pix2seq model in absolute points and following the path of each one from the starting point to the end. (c) The red lines show the suitable strokes that the robot must follow when is drawing, while the yellow lines show the strokes performed by the robot, including all variations and recalculations of movements. Note that images (b) and (c) correspond to results of individual examples with the same input images

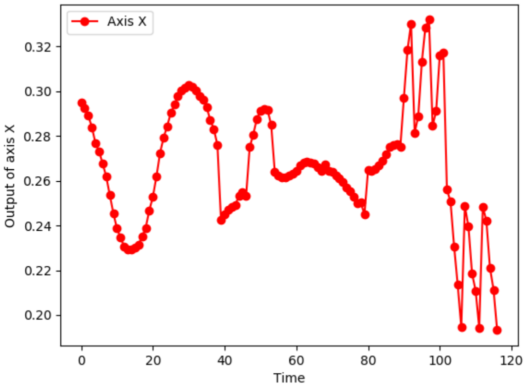


Figure 6. Effector tracking on the X axis in 120 movements, drawing a cat. This axis represents the execution of horizontal movements, from right to left and vice versa.

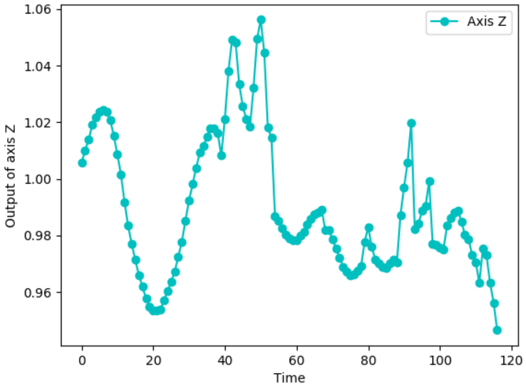


Figure 7. Effector tracking on the Z axis in 120 movements, drawing a cat. This axis represents the execution of vertical movements, from top to bottom and vice versa.

this allows the Poppy robot to execute the strokes similarly as humans do.

It can be noticed that there are variations in the Y axis when drawing, that is when the pencil is on the canvas. These variations can hinder the trace causing contact with the canvas at times that should not or the opposite effect, not make contact at times that should, remember that this value should be constant and not variable.

B. Multiple class

In the case of multiple classes, it can be observed that the sketch-pix2seq architecture achieves good results when classifying the input image, for our case it successfully classifies when the image is a bicycle or an animal, in the same way, that with the single class experiment, exact replication is not executed but it maintains the main characteristics of the drawing.

The distinction of classes is not so precise with the division of classes between the lion and the bear, this can be affected by the classification of the drawings in the data set and the concept that people have about a certain abstract class and how people who draw on the platform do it. It is important to remember that the dataset is obtained from sketches made by people within the QuickDraw! platform.

Figure 9 shows Poppy robot finishing the drawing path, after capturing the image with the vision sensor from the TV in the V-REP simulator. You can inspect the result of these experiments as well as all the work was done including the source code, images of results of the experiments with the cat, bicycle and lion class, and demonstration videos in the following repository https://github.com/Stevenpach10/Practica2019-UDC

VI. Conclusions

The knowledge of human beings is determined by the environment in which they are involved and how we perceive abstract elements. Similarly, leaning these concepts to a robotic device depends on the elements that are used to teach it.

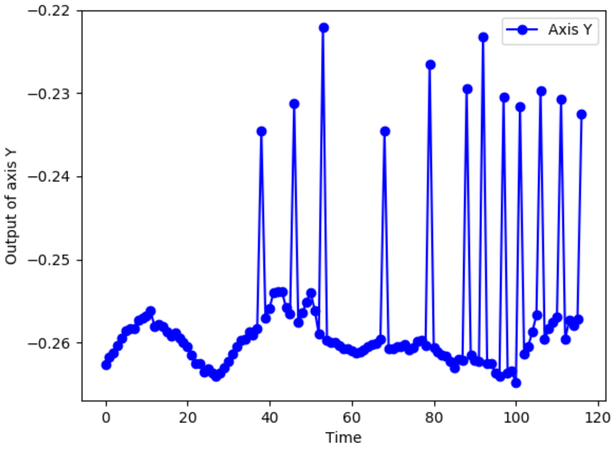
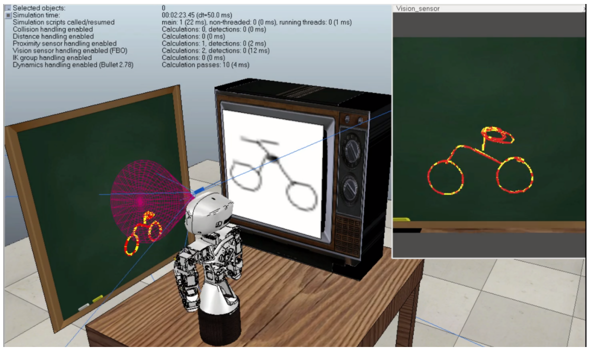


Figure 8. Effector tracking on the Y axis in 120 movements, drawing a cat. This axis represents the execution of movement in depth, when the canvas is touched and when it is not



```
Selected objects:                        0
Simulation time:                         00:02:23.45 (dt=50.0 ms)
Simulation scripts called/resumed        main: 1 (22 ms), non-threaded: 0 (0 ms), running threads: 0 (1 ms)
Collision handling enabled               Calculations: 0, detections: 0 (0 ms)
Distance handling enabled                Calculations: 0 (0 ms)
Proximity sensor handling enabled        Calculations: 1, detections: 0 (2 ms)
Vision sensor handling enabled (FBO)     Calculations: 2, detections: 0 (12 ms)
IK group handling enabled                Calculations: 0 (0 ms)
Dynamics handling enabled (Bullet 2.78)  Calculation passes: 10 (4 ms)
```

Figure 9. The Poppy robot completing a drawing that is captured from its camera and later drawn on a blackboard in the V-REP simulator.

For this reason, it is necessary to have a well-classified data set and quality images to ensure a better result. Performing a thorough selection of the classified images can improve results such as class classification in the multi-class experiment, demonstrating the importance of associating concepts such as the concept of a cat with the corresponding visual perception of what the concept means according to certain knowledge.

The present work shows the capacity of a robot to determine which are the necessary strokes to make a drawing, and then transform them into movements in a similar way as humans make a replication of a drawing.

However, this capacity to draw requires that the movements executed by the robot are extremely precise so that it does not require more than one movement to reach the appropriate points of a drawing. Therefore, it is concluded that it is necessary to improve the movement mechanisms in robotic systems to achieve that these fine motor capabilities can be exercised adequately.

We also performed tests on the physical robot, obtaining similar results, even the movements tended to be more clumsy due to the friction on the surface, which could be interesting future work.

VII. Acknowledgements

Thanks for all support of the PARMA group members, the International Cooperation Office, Computer School at Technological Institute of Costa Rica for the internship scholarship provided for the realization of this work. Moreover, the support provided by all people at the Integrated Group of Engineering Research at Universidade da Coruña for all knowledge contributed to this work.

References

[1] P. v. Sommers, “A system for drawing and drawing-related neuropsychology,” *Cognitive Neuropsychology*, vol. 6, no. 2, pp. 117–164, 1989.

[2] M. Amenomori, A. Kono, J. S. Fournier, and G. A. Winer, “A cross-cultural developmental study of directional asymmetries in circle drawing,” *Journal of Cross-Cultural Psychology*, vol. 28, no. 6, pp. 730–742, 1997.

[3] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” in *Advances in neural information processing systems*, 2015, pp. 802–810.

[4] D. Ha and D. Eck, “A neural representation of sketch drawings,” *arXiv preprint arXiv:1704.03477*, 2017.

[5] Y. Chen, S. Tu, Y. Yi, and L. Xu, “Sketch-pix2seq: a model to generate sketches of multiple categories,” *arXiv preprint arXiv:1709.04121*, 2017.

[6] M. Lapeyre, P. Rouanet, and P.-Y. Oudeyer, “Poppy: A new bio-inspired humanoid robot platform for biped locomotion and physical human-robot interaction,” 2013.

[7] M. Lapeyre, “Poppy: open-source, 3d printed and fully-modular robotic platform for science, art and education,” Ph.D. dissertation, 2014.

[8] S. McCrea, “A neuropsychological model of free-drawing from memory in constructional apraxia: A theoretical review,” *American Journal of Psychiatry and Neuroscience*, vol. 2, no. 5, pp. 60–75, 2014.

[9] J. Choi, H. Cho, J. Song, and S. M. Yoon, “Sketchhelper: Real-time stroke guidance for freehand sketch retrieval,” *IEEE Transactions on Multimedia*, vol. 21, no. 8, pp. 2083–2092, 2019.

[10] V. Mohan, P. Morasso, J. Zenzeri, G. Metta, V. S. Chakravarthy, and G. Sandini, “Teaching a humanoid robot to draw ‘shapes’,” *Autonomous Robots*, vol. 31, no. 1, pp. 21–53, 2011.

[11] P. Tresset and F. F. Leymarie, “Portrait drawing by paul the robot,” *Computers & Graphics*, vol. 37, no. 5, pp. 348–363, 2013.

[12] A. K. Singh, P. Chakraborty, and G. C. Nandi, “Sketch drawing by nao humanoid robot,” in *TENCON 2015 - 2015 IEEE Region 10 Conference*, 2015, pp. 1–6.

[13] K. Sasaki, H. Tjandra, K. Noda, K. Takahashi, and T. Ogata, “Neural network based model for visual-motor integration learning of robot’s drawing behavior: Association of a drawing motion from a drawn image,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2736–2741.

[14] K. Sasaki and T. Ogata, “Adaptive drawing behavior by visuomotor learning using recurrent neural networks,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 1, pp. 119–128, 2018.

[15] R. Putra, S. Kautsar, R. Adhitya, M. Syai’in, N. Rinanto, I. Munadhif, S. Sarena, J. Endrasmono, and A. Soeprijanto, “Neural network implementation for invers kinematic model of arm drawing robot,” in *2016 International Symposium on Electronics and Smart Devices (ISESD)*. IEEE, 2016, pp. 153–157.

[16] M. F. Sanner *et al.*, “Python: a programming language for software integration and development,” *J Mol Graph Model*, vol. 17, no. 1, pp. 57–61, 1999.

[17] J.-A. Ruiz-Jara, M. Naya-Varela, F. Bellas-Bouza, and E. Arias-Méndez, “Control library modification to improve robot interaction with its environment,” in *2019 International Conference in Engineering Applications (ICEA)*. IEEE, 2019, pp. 1–6.

[18] E. Rohmer, S. P. Singh, and M. Freese, “V-rep: A versatile and scalable robot simulation framework,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1321–1326.