

Design and development of DrawBot using image processing

Krithika Vaidyanathan¹, Nandhini Murugan¹, Subramani Chinnamuthu², Sivashanmugam Shivasubramanian¹, Surya Raghavendran¹, Vimala Chinnaiyan³

¹Department of Mechatronics Engineering, SRM Institute of Science and Technology, Kattankulathur, India

²Department of Electrical and Electronics Engineering, SRM Institute of Science and Technology, Kattankulathur, India

³Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, Kattankulathur, India

Article Info

Article history:

Received Jun 24, 2020

Revised Jul 15, 2021

Accepted Jul 29, 2021

Keywords:

DrawBot

Image processing

MATLAB/Simulink

SolidWorks

Text extraction

ABSTRACT

Extracting text from an image and reproducing them can often be a laborious task. We took it upon ourselves to solve the problem. Our work is aimed at designing a robot which can perceive an image shown to it and reproduce it on any given area as directed. It does so by first taking an input image and performing image processing operations on the image to improve its readability. Then the text in the image is recognized by the program. Points for each letter are taken, then inverse kinematics is done for each point with MATLAB/Simulink and the angles in which the servo motors should be moved are found out and stored in the Arduino. Using these angles, the control algorithm is generated in the Arduino and the letters are drawn.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Subramani Chinnamuthu

Department of Electrical and Electronics Engineering, SRM Institute of Science and Technology

Kattankulathur, 603203, India

Email: csmsrm@gmail.com

1. INTRODUCTION

Earlier, a lot of researches were undergone for robots in recreational uses. Now, the researchers concentrate and dedicated consider able attempt on emergent robots which is capable to counterpart human behavior on high level tasks. To achieve these robots, it involves integration of various elements (namely computer vision, physical motion, and intelligence). By these integrations in robot, it makes the behavior to be more like human. One such type of robot is the drawing robot. In recent days, quite a lot of exhibitions of drawing robots have taken place. A lot of researchers have been working on the DrawBot to perfect it and each of them has come up with their own unique ideologies. A notable few are listed in the paragraphs that follow [1]-[5].

The robotic installation named, Robot Paul can reproduce facial features of people by capturing an image and processing it. Paul cannot reproduce these as good as an artist. However, Paul did deploy several techniques to imitate drawing skill. BARC, Mumbai made a demo to perform portrait drawing on a complex surface [6]-[9]. They did so by employing sensors to detect forces acting on the drawing tool and identifying the tool's orientation with respect to the drawing surface.

Kumar and Kumar [10] proposed a morphological method of segmenting the image. This method had various stages for image segmentation. They were preprocessing the input image, then conversion of color space followed by adjusting the threshold values, after which the image has to go through feature extraction, segmentation of the image and at last valuation. A new approach to text detection was proposed by Li and Lu [11], which was based on the width of stroke. Initially, to extract the character candidates, distinctive contrast improved maximally stable extremal regions (MSER) algorithm is designed. Next to eliminate the non-text

regions, effortless geometric limits are applied. Secondly, set of rules for the geometry of text were introduced to remove non-text regions. Also including the generated stroke width ensured to eliminate all remaining irrelevant areas. Then the remaining areas are clustered to form text regions [12]-[16]. Their algorithms turned out more favorable compared to other sophisticated solutions. Neumann and Matas [17] elaborated about a real time text detection and recognition was ventured in this paper. A set of extremal regions were posed to make selection in sequence which enabled real time detection of text. This method proved efficient and reliable against blur, varying light intensities, and made it possible to handle very low contrast text [18]-[20]. Alvaro *et al.* [21] proposed to localize text more accurately and reliably. Proper character candidates were to be found in the first stage. Later, an analysis based on connected components to discard non text regions and accept text candidates. At the final step, with gradient features a text line classifier to support vector machines. The paper described by Huizhong *et al.* [22] focused on detecting text from natural image where it introduced edge enhanced MSER approach to detect text regions for more suitable candidates. Geometry and Stroke width data enabled to eliminate non text regions. Text regions were grouped to form as words [23]-[25].

The first step in this work is aimed at designing a robot which can perceive a design shown to it and accurately reproduce it on any given area as directed. It can be used for decoding of unreadable scriptures and reproducing them, can also come in handy in situations where markings in parking areas or fields have to be made in short periods of time, during interior design painting of houses, and in Sail making and intricate detailing of vehicles. Even printing huge billboards using printers are cumbersome as the printing process has to be fragmented and if automated one can save time and improve efficiency. This DrawBot can be used to replicate text in an image on a drawing board. This can come in handy in situations where the human eye finds it hard to read and recognize letters in an image.

2. DESIGN AND FABRICATION OF PROPOSED ROBOT

The software for modeling of the drawing robot used in this work was done using SolidWorks 2016b software with representation of 3-D geometries and analyzing various proximities associated to modeling and action. Two cylindrical links with lengths 17.5 cm and 21.5 cm were modelled. A base was designed to hold the robotic arm. There were 3 Servo motors out of which the first one was attached to the base, the second one was attached at the inter-section of the first and the second link mounted on a servo bracket and the final motor was attached to the end effector to help positioning the pen on the drawing board. A metal horn was attached to the shaft of the motor seated on the base. Over the metal horn a hub-link connector was attached to be able to attach the robotic arm to the base servo motor. The two links were mounted on the base as shown in the Figure 1. The material chosen was Aluminum 6061.

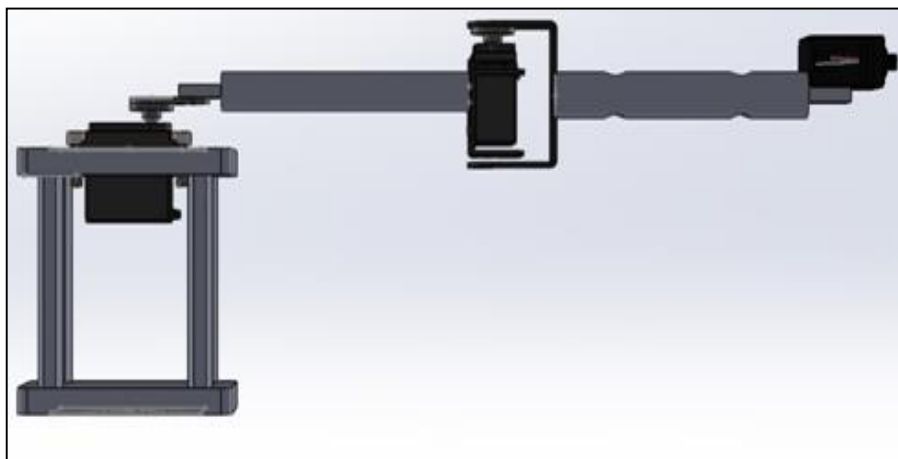


Figure 1. SolidWorks design of the DrawBot

2.1. Stress analysis

In order to determine the stress and strain in materials, the Stress-strain analysis (or stress analysis) uses various methods. To check the stress bearing capacity of the design, stress analysis was performed using SolidWorks software. With the calculation of maximum possible load on the body, the maximum deformation on each part of the design was tested. The yield strength was noted and cross verified with the stress in each

part to make sure that the stress is much below the yield strength. Stress analysis in SolidWorks contains the following:

- Creating the part/assembly that is to be used.
- New study is started from the simulation Add-In. Material is chosen from the SolidWorks database.
- Fixtures are chosen for instance sliders, elastic supports, or rollers.
- External loads are chosen (for example: pressures, torques and forces), other options that can be added are distributed mass, gravity, or centrifugal forces.
- Run the simulation model created.

The displacement and stress analysis results obtained for HubLink connector and Link are as shown in Figures 2 and 3. Similarly, the displacement and stress analysis results obtained for multipurpose Bracket and U bracket are as shown in Figures 4 and 5 (see in appendix).

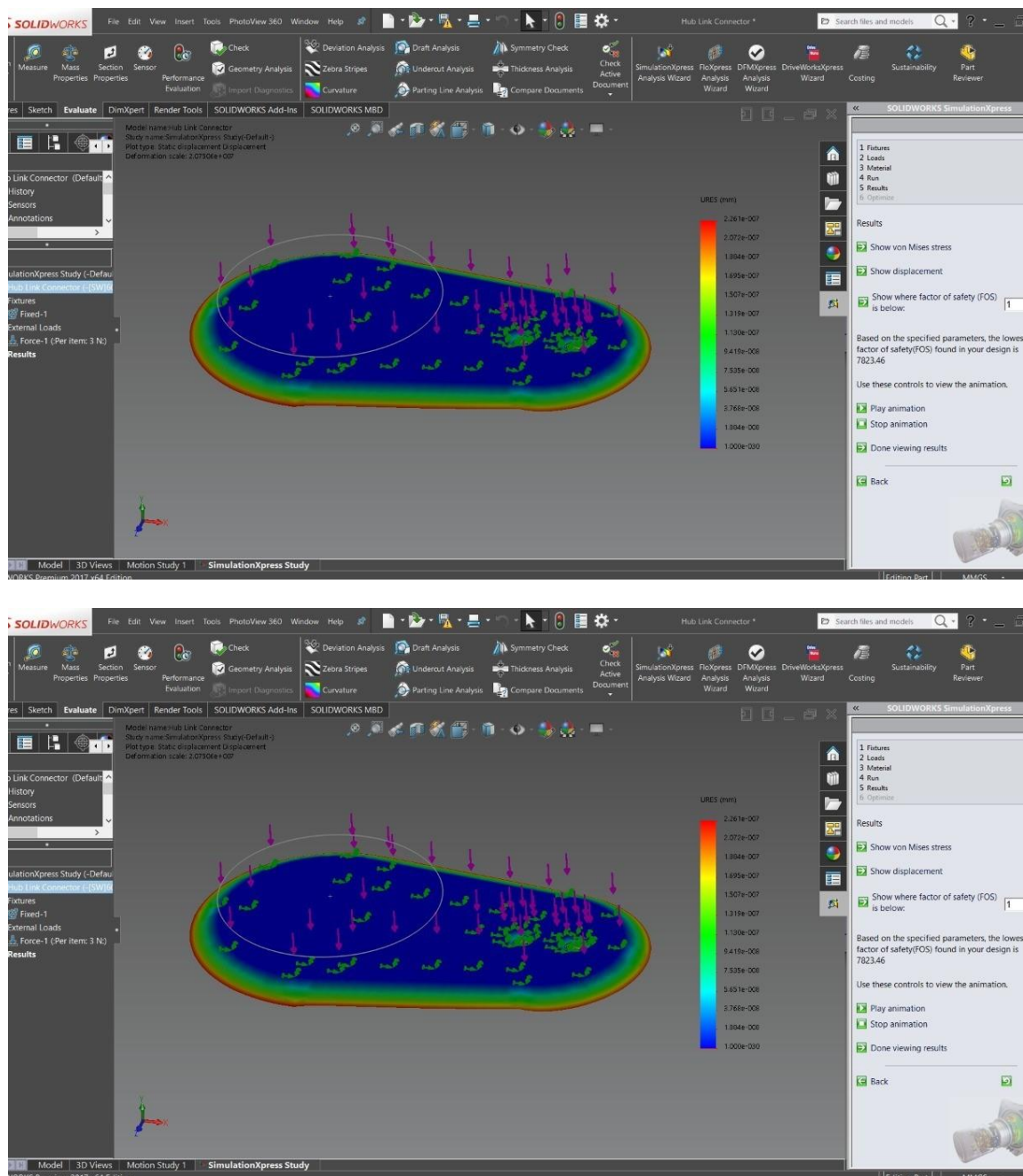


Figure 2. Displacement and stress analysis of HubLink connector

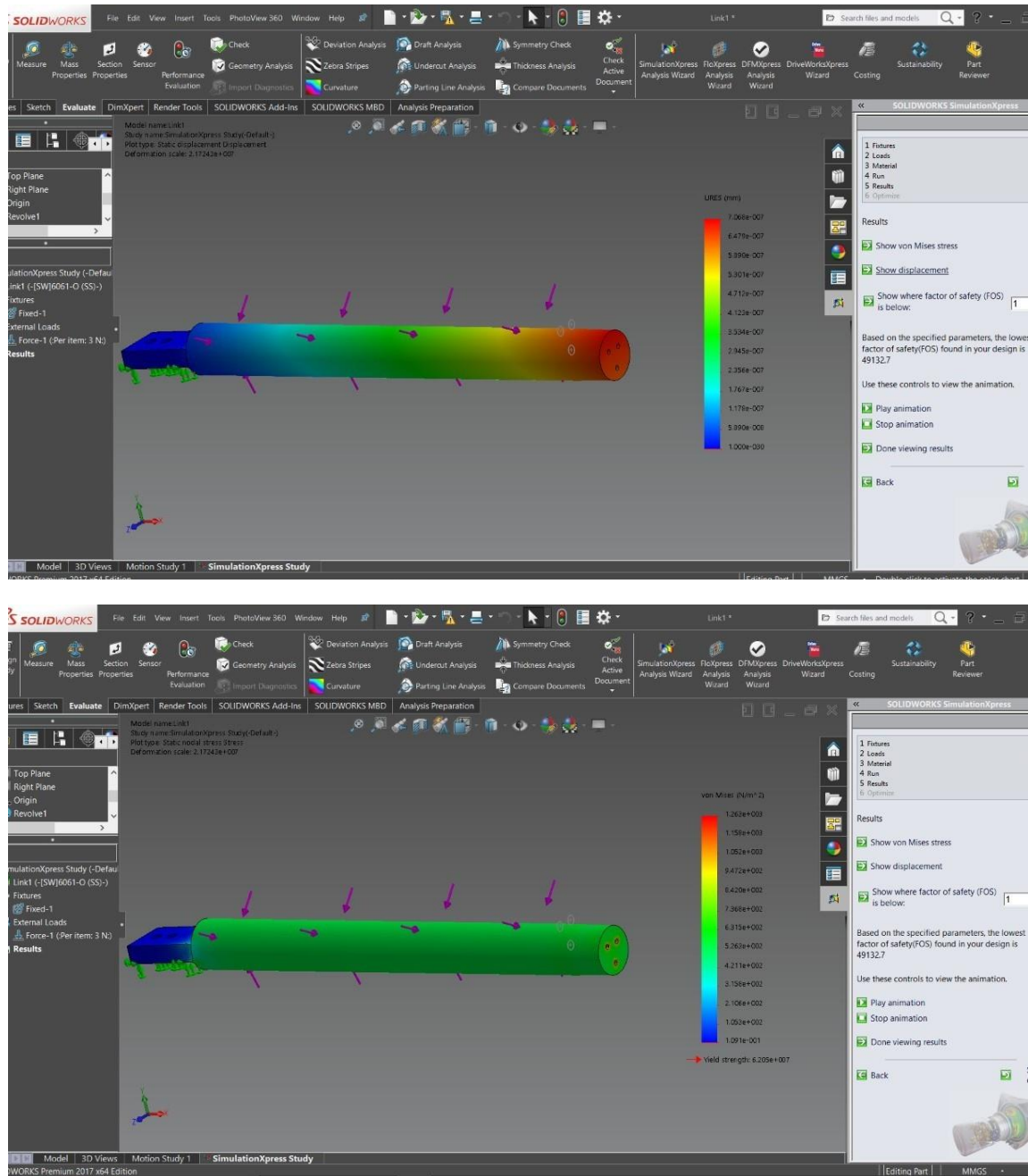


Figure 3. Displacement and stress analysis of link

3. METHOD

3.1. Input image

Considering that the first step of the work was to identify letters from an image, an image with letters was first clicked. An input image of any format (JPG, PNG) is then uploaded to the system. Then the image is added to the path of the image processing code. The flow of the proposed method is explained schematically in the Figure 6.

3.2. Extraction of letters from image

The following steps are performed in the image processing code where the letters in an image are identified. As mentioned in the flow chart in Figure 7, initially by using MSER the candidate text regions are detected. The MSER is used as mentioned before is best to find the text regions. Then the non-text regions are removed which is based on geometric features. Even though the MSER is capable of selecting the most text

available, it also identifies other regions which are not a text in the image available. The program utilizes a normal rule-based method. This is done basically to extract the non-text regions in the image. Next the final detected result is obtained by merging all the text regions. At last, the OCR is utilized to detect the recognized text.

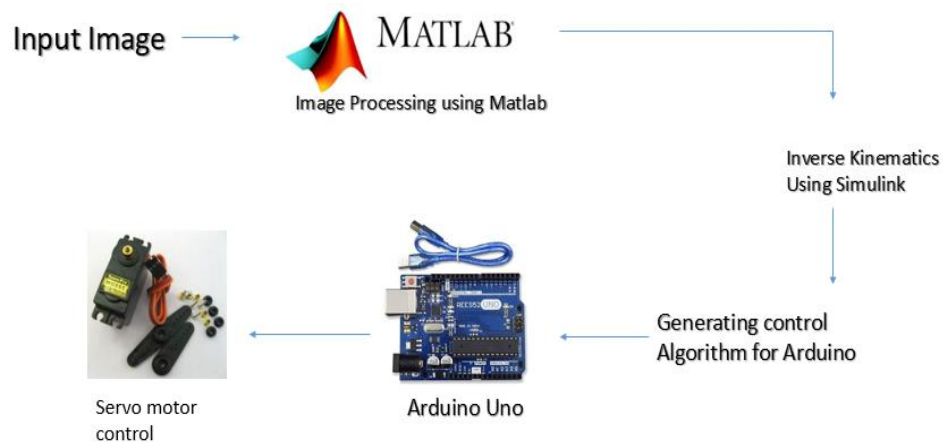


Figure 6. Proposed methodology

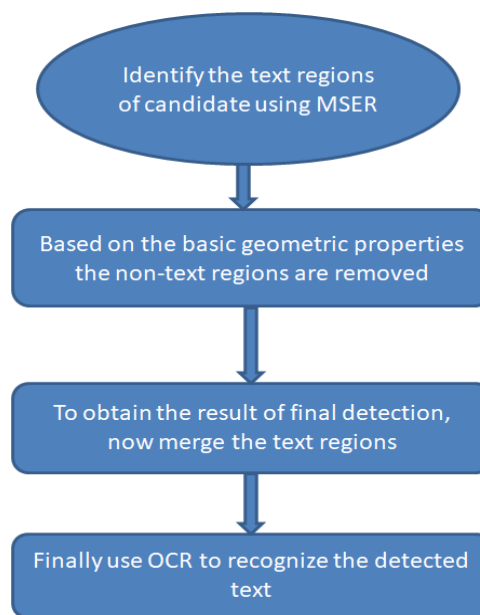


Figure 7. Flow of the image processing work

4. RESULTS AND DISCUSSION

4.1. Simulation results

The Figure 8 depicts the MATLAB Simulink block diagram to do inverse kinematics and find out the angles θ_1 and θ_2 in which the robotic arms should be moved. The left side of the image is the part where the formula for finding the angles using inverse kinematics is solved. The images right side contains the code to display, angles θ_1 and θ_2 and also shows a simulation output graph in Figure 9, that gives an idea of what the end drawing result will be when the two arm links move in the angles they get from inverse kinematics as depicted in Figure 10.

The X, Y points are then substituted manually and for each point θ_1 and θ_2 are found out and noted. These are tabulated as presented for each and every letter. For instance, to make the DrawBot to draw the letter U, the X, Y point values and its corresponding θ_1 and θ_2 values are shown in Table 1. Similarly for letters V, P, T and X the corresponding values are tabulated as shown in Tables 2, 3, 4 and 5.

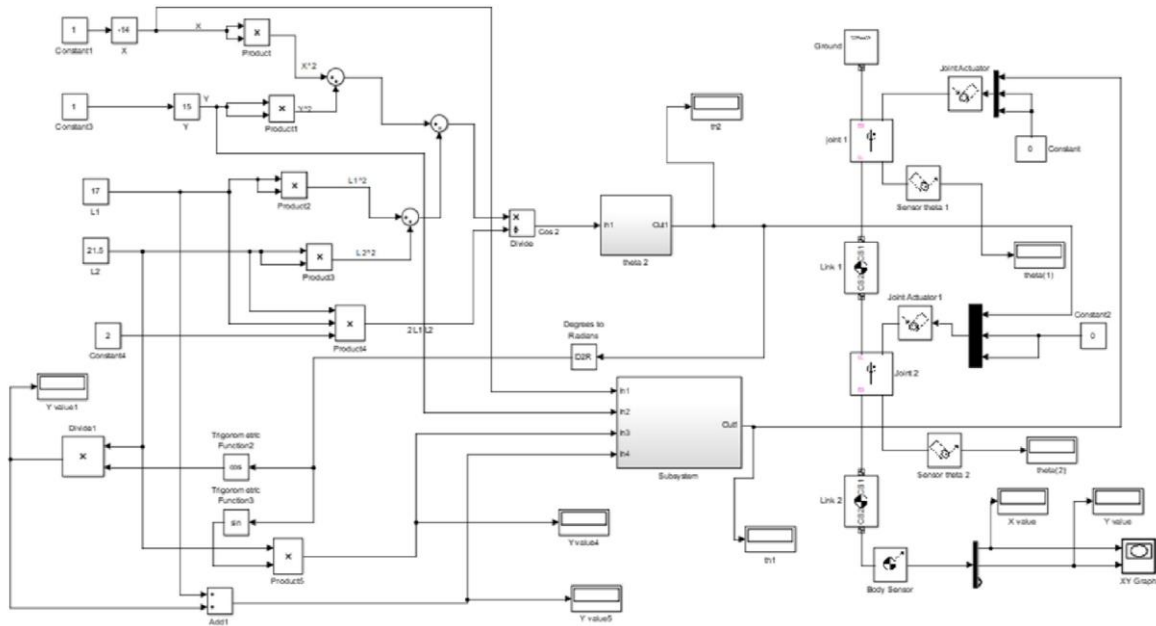


Figure 8. Simulink block diagram

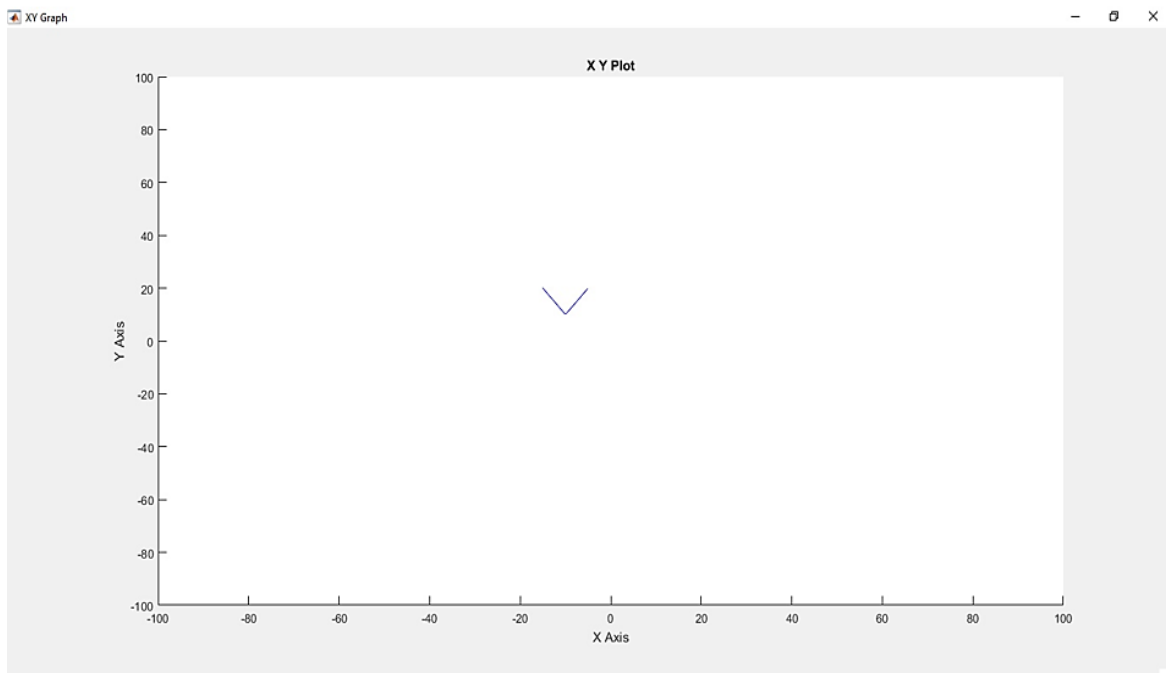


Figure 9. Simulation plot for letter 'v'

4.2. Control algorithm

Initially the image processing code in MATLAB sends the letters that it has recognized to the Arduino via serial communication. Arduino program accepts each and every word and stores it as string. Then each letter from the word is singled out. Once the angles are listed for each point in which the arm has to move to draw an alphabet, they are fed to the Servo motor through Arduino coding. For each alphabet and for each stroke the angles are fed in the form of an array. Then as each stroke progresses each section of the array is executed. The hardware setup for the proposed DrawBot is shown in Figure 11.

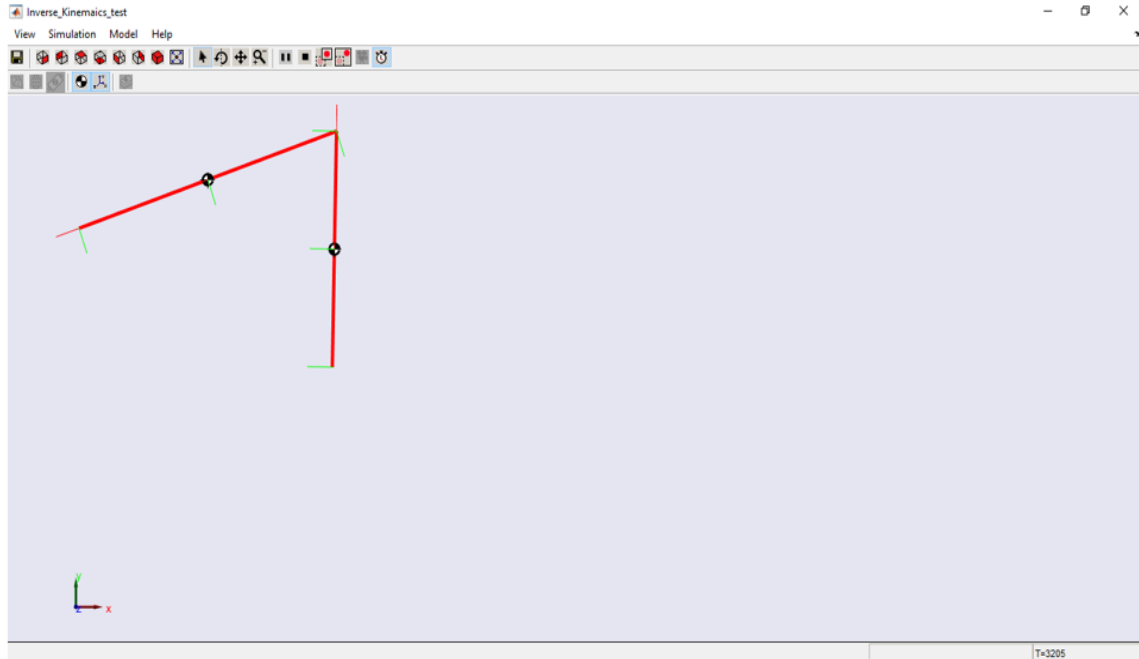


Figure 10. Simulation Arm diagram

Table 1. Table for Letter 'U'

LETTER 'U'			
STROKE1			
S.NO	(X, Y)	Theta1	Theta2
1	(-20, 25)	90	68
2	(-20, 22.5)	87	78
3	(-20, 20)	86	86
4	(-20, 17.5)	85	94
5	(-20, 15)	85	100
STROKE2			
S.NO	(X, Y)	Theta1	Theta2
1	(-15, 25)	74	82
2	(-15, 22.5)	71	92
3	(-15, 20)	69	100
4	(-15, 17.5)	68	108
5	(-15, 15)	68	114

Table 2. Table for Letter 'V'

LETTER 'V'			
STROKE1			
S.NO	(X, Y)	Theta1	Theta2
1	(-20, 20)	86	86
2	(-20, 22.5)	85	94
3	(-20, 25)	85	100
STROKE2			
S.NO	(X, Y)	Theta1	Theta2
1	(-20, 10)	89	110
2	(-18, 14)	79	108
3	(-17.5, 15)	77	108
4	(-17, 16)	75	106
5	(-16, 18)	71	104
6	(-15, 20)	69	100

Table 3. Table for Letter 'P'

LETTER 'P'			
STROKE1			
S.NO	(X, Y)	Theta1	Theta2
1	(-20, 22.5)	90	68
2	(-20, 22.5)	87	78
3	(-20, 20)	86	86
4	(-20, 17.5)	85	94
5	(-20, 15)	85	100
STROKE2			
S.NO	(X, Y)	Theta1	Theta2
1	(-22.5, 25)	99	59
2	(-15, 25)	74	82
STROKE3			
S.NO	(X, Y)	Theta1	Theta2
1	(-15, 25)	74	82
2	(-15, 22.5)	71	92
3	(-15, 20)	69	100
STROKE4			
S.NO	(X, Y)	Theta1	Theta2
1	(-15, 20)	69	100
2	(-20, 20)	86	86

Table 4. Table for Letter 'T'

LETTER 'T'			
STROKE1			
S.NO	(X, Y)	Theta1	Theta2
1	(-10, 25)	59	92
2	(-12.5, 25)	66	88
3	(-15, 25)	74	82
4	(-17.5, 25)	82	76
5	(-20, 25)	90	68
6	(-22.5, 25)	99	59
7	(-25, 25)	109	47
STROKE2			
S.NO	(X, Y)	Theta1	Theta2
1	(-15, 27.5)	78	72
2	(-15, 25)	74	82
3	(-15, 22.5)	71	92
4	(-15, 20)	69	100
5	(-15, 17.5)	68	108
6	(-15, 15)	68	114
7	(-15, 12.5)	69	120
8	(-15, 10)	71	126

Table 5. Table for Letter 'X'

LETTER 'X'			
STROKE1			
S.NO	(X, Y)	THETA1	THETA2
1	(-20, 25)	90	68
2	(-19, 23.5)	85	77
3	(-18, 22)	80	86
4	(-17, 20.5)	76	93
5	(-16, 19)	72	101
6	(-15, 17.5)	68	108
7	(-14, 16)	64	114
8	(-13, 14.5)	60	121
9	(-12, 13)	56	127
10	(-11, 11.5)	52	133
11	(-10, 10)	48	139
STROKE2			
S.NO	(X, Y)	THETA1	THETA2
1	(-10, 25)	59	92
2	(-11, 23.5)	60	96
3	(-12, 22)	61	100
4	(-13, 20.5)	63	103
5	(-14, 19)	65	105
6	(-15, 17.5)	68	108
7	(-16, 16)	71	109
8	(-17, 14.5)	75	110
9	(-18, 13)	79	111
10	(-19, 11.5)	84	111
11	(-20, 10)	89	110

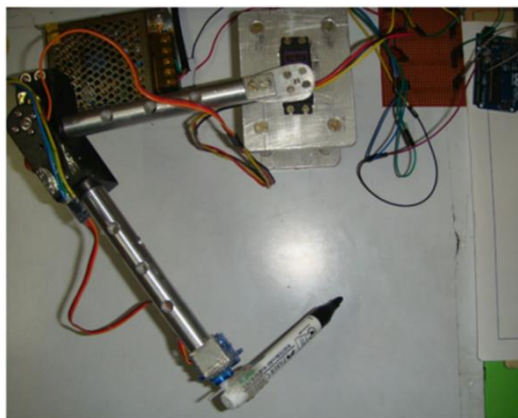


Figure 11. Hardware setup for the proposed DrawBot

5. CONCLUSION

The work was started with the sole objective of designing and fabricating a robotic arm which can perceive a design shown to it and reproduce it as a 2-D drawing on an even surface as directed. However, in the process of doing the work, it was discovered that this work can be used for a superior application such as reading and reproducing texts available in old historic documents and scriptures that have deteriorated in condition over time and are hard to read. The knowledge in image processing in MATLAB along with working extensively with it to do inverse kinematics was incorporated to obtain servo angles. In future the scope of our draw-robot can be increased by improving its ability to not just replicate letters but also shapes and extremely complex designs. We would also like to enhance it such that it can read and reproduce contents of other major languages in the world as well. As of now our work uses servo motors to control the movement of the arm links however in the future, we would like to replace it with stepper motors as they would be easier to work with for precision applications such as ours.

APPENDIX

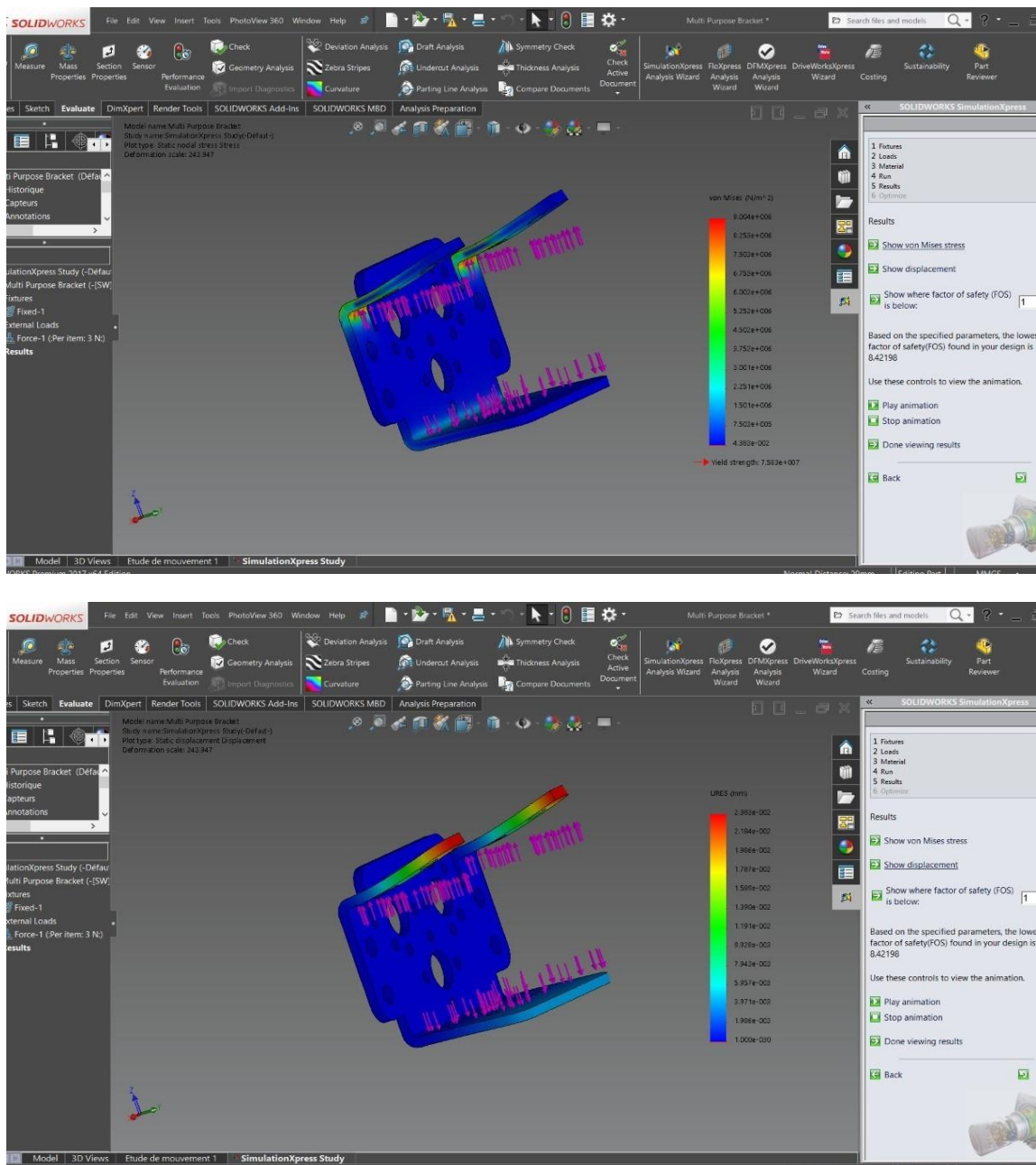


Figure 4. Displacement and stress analysis of multipurpose bracket

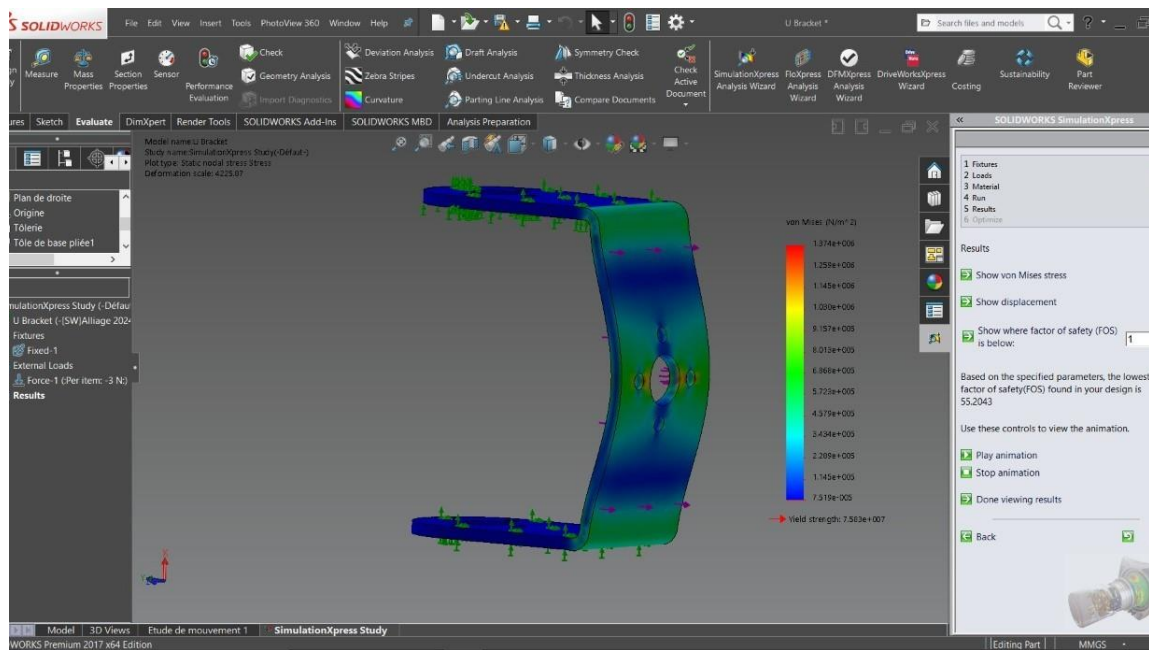
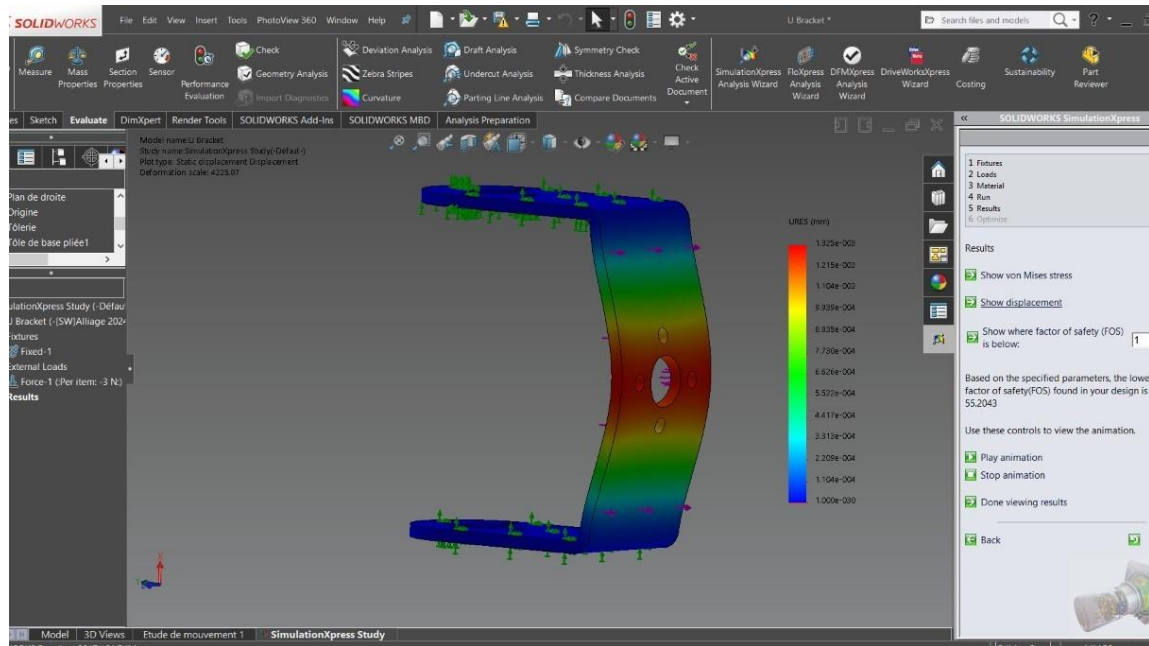


Figure 5. Displacement and stress analysis of U bracket

REFERENCES

- [1] A. Billard, "Robota: clever toy and educational tool," *Robotics and Autonomous Systems*, vol. 42, no. 3-4, pp. 259-269, 2003, doi: 10.1016/S0921-8890(02)00380-9.
- [2] R. Kannan *et al.*, "Design, implementation and analysis of a low-cost drawing bot for educational purpose," *International Journal of Pure and Applied Mathematics*, vol. 118, no. 16, pp. 213-230, 2018.
- [3] U. J. Pai, N. P. Sarath, R. Sidharth, A. P. Kumar, S. Pramod, and G. Udupa, "Design and manufacture of 3D printed myoelectric multi-fingered hand for prosthetic application," *2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA)*, Kerala, India, 2016, pp. 1-6, doi: 10.1109/RAHA.2016.7931904.
- [4] F. Ghedini and M. Bergamasco, "Robotic creatures: anthropomorphism and interaction in contemporary art," *19th International Symposium in Robot and Human Interactive Communication*, 2010, pp. 731-736, doi: 10.1109/ROMAN.2010.5598720.
- [5] J. Saha, T. Niphadkar, and A. Majumdar, "Drawbot: A mobile robot for image scanning and scaled printing," *International Journal of Mechanical Engineering and Robotics Research*, vol. 5, no. 2, pp. 168-172, 2016, doi: 10.18178/ijmerr.5.2.124-128.
- [6] C. P. Shinde and Kumbhar, "Design of myoelectric prosthetic arm," *International Journal of Management, IT and Engineering*, vol. 3, pp. 325-333, 2013.

- [7] V. S. Padilla, R. A. Ponguillo, A. A. Abad, and L. E. Salas, "Cyber-physical system based on image recognition to improve traffic flow: A case study," *International Journal of Electrical and Communications Engineering (IJECE)*, vol. 10, no. 5, pp. 5217-5226, 2020, doi: 10.11591/ijece.v10i5.pp5217-5226.
- [8] J. D. Escobar and V. Kober, "Natural scene text detection and segmentation using phase-based regions and character retrieval," *Mathematical Problems in Engineering, Hindwai*, vol. 2020, 2020, Art. no. 7067251, doi: 10.1155/2020/7067251.
- [9] X. Wang, L. Huang, and C. Liu, "A new block partitioned text feature for text verification," *2009 10th International Conference on Document Analysis and Recognition*, 2009, pp. 366-370, doi: 10.1109/ICDAR.2009.61.
- [10] K. Kumar and R. Kumar, "Enhancement of image segmentation using morphological operation," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 2, pp. 108-111, 2013, doi: 10.1109/CVPR.2012.6248097.
- [11] Y. Li and H. Lu, "Scene text detection via stroke width," *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012, pp. 681-684.
- [12] Z. Liu and S. Sarkar, "Robust outdoor text detection using text intensity and shape features," *2008 19th International Conference on Pattern Recognition*, Tampa, FL, 2008, pp. 1-4, doi: 10.1109/ICPR.2008.4761432.
- [13] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: recent advances and future trends," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 19-36, 2016, doi: 10.1007/s11704-015-4488-0.
- [14] Q. Ye and D. Doermann, "Text detection and recognition in imagery: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 7, pp. 1480-1500, 2015, doi: 10.1109/TPAMI.2014.2366765.
- [15] H. Zhang, K. Zhao, Y.-Z. Song, and J. Guo, "Text extraction from natural scene image: a survey," *Neurocomputing*, vol. 122, pp. 310-323, 2013, doi: 10.1016/j.neucom.2013.05.037.
- [16] P. Shivakumara, T. Q. Phan, and C. L. Tan, "New fourier-statistical features in RGB space for video text detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp. 1520-1532, 2010, doi: 10.1109/TCSVT.2010.2077772.
- [17] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," *Computer Vision-ACCV 2010*, vol. 6494, pp. 770-783, 2011, doi: 10.1007/978-3-642-19318-7_60.
- [18] L. Neumann and J. Matas, "Real-time scene text localization and recognition," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, 2012, pp. 3538-3545, doi: 10.1109/CVPR.2012.6248097.
- [19] P. Y. Feng, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 800-813, 2011, doi: 10.1109/TIP.2010.2070803.
- [20] S. Kucuk and Z. Bingul, "Robot kinematics: forward and inverse kinematics," *INTECH Open Access Publisher*, 2006, doi: 10.5772/5015.
- [21] G. Alvaro, L. M. Bergasa, J. J. Yebes and S. Bronte, "Text location in complex images," *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012, pp. 617-620.
- [22] C. Huizhong, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable extremal regions," *2011 18th IEEE International Conference on Image Processing*, 2011, pp. 2609-2612, doi: 10.1109/ICIP.2011.6116200.
- [23] P. Tresset and F. F. Leymarie, "Portrait drawing by paul the robot," *Computers and Graphics*, vol. 37, no. 5, pp. 348-363, 2013, doi: 10.1016/j.cag.2013.01.012.
- [24] M. S. Munna, B. K. Tarafder, R. Md Golam and M. T. Chandra, "Design and implementation of a drawbot using MATLAB and arduino mega," *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2017, pp. 769-773, doi: 10.1109/ECACE.2017.7913006.
- [25] P. Sahare and S. B. Dhok, "Review of text extraction algorithms for scene-text and document images," *IETE Technical Review*, vol. 34, no. 2, pp. 144-164, 2017, doi: 10.1080/02564602.2016.1160805.