Towards Autonomous Berry Harvesting using Visual Servoing of Soft Continuum Arm

Shivani Kamtikar,[1] Samhita Marri,[2] Benjamin Walt,[3] Naveen Kumar Uppalapati,[4] Girish Krishnan,[3] Girish Chowdhary[1 4]

[1] Computer Science
[2] Electrical and Computer Engineering
[3] Mechanical Science and Engineering
[4] Coordinated Science Laboratory,
University of Illinois at Urbana Champaign, USA.
(skk7, marri2, walt, uppalap2, gkrishna, girishc)@illinois.edu

Abstract

Autonomous berry harvesting is a challenging problem, especially with hard-to-reach targets inside the plant. Using soft continuum arms is a step towards achieving this task without causing excessive damage to the plant. Visual servoing is a popular control strategy that relies on visual feedback to close the control loop in controlling a soft arm. However, robust visual servoing is challenging as it requires reliable feature extraction from the image, accurate control models and sensors to perceive the shape of the arm, both of which can be hard to implement in a soft robot. This work circumvents these challenges by presenting a deep neural network-based method to perform smooth and robust 3D positioning tasks on a soft arm by visual servoing using a camera mounted at the distal end of the arm. A convolutional neural network is trained to predict the actuations required to achieve the desired pose in a structured environment. An *integrated* approach for estimating the actuations from the image is proposed. In addition, a proportional control law is implemented to reduce the error between the desired and current image as seen by the camera. The model and proportional feedback control make the described approach robust to several variations such as new targets, varying lighting conditions, diminution and uniform load. Furthermore, the model lends itself to be transferred to a new environment with minimal effort.

Introduction

There is an increasing need for autonomous berry harvesting robots due to labor shortage and growing population (Samtani et al. 2019). Traditional industrial robot arms have been difficult to adopt for messy, cluttered, and delicate plants. Soft continuum arms (SCA) (Hughes et al. 2016) have received growing attention due to their superiority in dexterous manipulation and safe interaction with the environment. Their inherent flexibility with high degrees of freedom endows soft robots with good adaptability but raises challenges for accurate position control (Uppalapati et al. 2020). The challenges in SCA control can be attributed mainly to the difficulties in modeling and sensing (Rus and Tolley 2015) its deformed shape. Current modeling methods are either simplistic with a constant curvature assumption that work in 2D plane or valid for SCAs with short lengths (George Thuruthel, Renda, and Iida 2020). On the other hand, exact methods based on Cosserat rod models (Gazzola et al. 2018) are computationally intensive. In addition, even with effective models, there aren't cost-effective sensors (Shih et al. 2020; Thuruthel et al. 2019) to get the spatial position feedback of SCAs.

Recent advances in visual servoing and deep learning in robots can be effectively used to overcome the limitations in both sensing and modeling of SCA. Visual servoing using Neural Networks (NN) in conventional robotic arms has been well studied but not extensively validated on SCA because of its complex behavior. Works like (Xu et al. 2019), (Xu et al. 2021) used a fixed camera (eye-to-hand) to capture the pose and curvature of the soft-arm to perform image-based visual servoing. Additional sensor assistance-based visual servoing was performed in (Wang et al. 2020) in order to track the camera motion but was limited to 2D space. In this work, we focus on eye-in-hand image-based visual servoing in a 3D framework where there are berry-like objects, with a camera at the distal tip of the SCA. We propose the use of NN for visual servoing in SCA using an *integrated* approach to estimate the pose of the soft manipulator, and control it using visual servoing in a structured environment. Our framework takes a single RGB image, *I*, and predicts the control inputs (actuations) required to reach the specific pose of the soft arm (current pose). Then the control policy is implemented using the calculated error between the geometrical features of the current and target images, as well as the error between the actuations of the current and target poses to reach the desired target pose. Fig. 1(f) shows the overall workflow of the proposed approach.

Methods

Experimental setup and Data collection: The experimental set up consists of a BR^2 (Uppalapati and Krishnan 2021) SCA mounted to a planar gantry. This gives the system 5 DOF - Bending (b), Rotation (r), SCA rotation (t), and x and y translation. On the tip of the SCA, a 1200 TVL wireless camera (Caddx Firefly, Micro FPV Camera w/ VTX) and positional sensor (micro sensor 1.8, Patriot SEU, Polhemus) are mounted. See appendix A for more details. The setup of the soft arm is shown in Fig. 1(c).

The data collection process is automated and the actua-

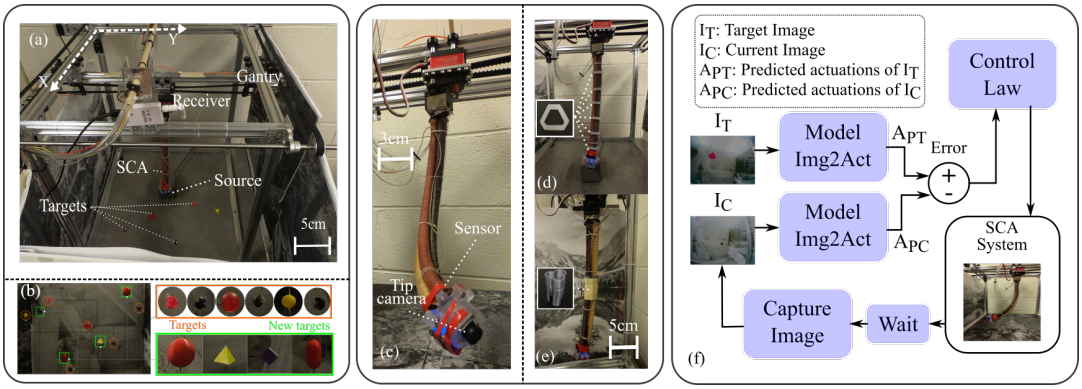

Figure 1: Experimental setup and workflow: (a) BR^2 SCA attached to a rotating servo that can move in X and Y direction in the gantry along with the targets and the wireless receiver to receive the tip camera image. (b) Four new targets (not seen in training) along with the targets used for training. (c) BR^2 SCA with the camera attached to the tip using a 3D printed casing. (d) SCA with uniform loads distributed along its length (inset: silicone cast ring weighing 1.4 grams). (e) SCA with the central region constrained with a rigid 3D printed part. (f) Overall workflow to reach the target image given current image using feedback.

tion inputs to the arm are given in the form of pressures (b, r), x, y and angle (t). Images of the scene are captured at discrete configurations throughout the workspace while corresponding state data (actuations and sensor readings) is simultaneously collected to self-annotate the images. The environment contains berry-like objects as seen in Fig. 1(b) to replicate the berry reaching problem.

Network Architecture: Deep convolutional neural networks (CNNs) are known to effectively extract features from images for various computer vision applications, such as image recognition (Krizhevsky, Sutskever, and Hinton 2012), image segmentation and also have been studied to estimate the pose of a robot manipulator from images (Bateux et al. 2018). Inspired by this, we use VGG16 (Simonyan and Zisserman 2014), to estimate the input actuation values required to reach a specific pose of the soft manipulator arm using image inputs. We use a modified VGG16 for our base network, VSBaseNet, where all the convolutional layers from VGG16 are used and smaller fully connected layers are added (details in Appendix B). Since we use real-world images of scenes captured by the tip camera, we perform transfer learning by using previously trained VGG16 weights on some layers and fine-tune it on our data which effectively helped the network to learn new features pertaining to our task. Our proposed *integrated* approach is implemented and tested in order to see its effectiveness in various scenarios as shown in section . The approach directly outputs the actuations given an input image, I. Since we were dealing with a regression task, the final dense layer consisted of five units is added to VSBaseNet and we call it VSNet, that outputs 5 floats corresponding to the five input actuations in vector form $[b, r, t, x, y]$ where b is bending, r is rotation, t is theta, and x, y are for gantry. **Training:** For training the VSNet, we used a total of 7980 images and corresponding state information. To regress absolute values of actuations, we use the mean-squared error (MSE) loss function which computes the mean of squared errors between the ground truth values and the predictions.

$$loss(I) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (1)$$

Here, Y_i corresponds to the ground truth actuations for the input image. $\hat{Y}_i$ are the predicted actuations for the input image. For other details about training, please refer to the Appendix B. **Control Policy:** There are two possible sources for open loop errors in the system, (i) Non repeatability due to hysteresis could lead to a different end effector position for the same input actuations, (ii) Inaccuracies in the trained model to fit the pose to actuations could also lead to large deviations from the target. To overcome the errors, we integrated the following feedback as shown in the Fig. 1(f):

$$A_{RC}(k+1) = A_{RC}(k) - \lambda(A_{PC}(k) - A_{PT}) \quad (2)$$

where $A_{RC}(k), A_{PC}(k)$ and A_{PT} are the current actuations to the soft arm, predicted actuations for the current image and predicted actuations for the target image at step k. It must be noted that at the end of each step k the arm is made to reach a steady state. As the error between the predicted actuations for the current image and target image reduces to zero the SCA tip reaches its target position (or the tip camera views the target image). λ is the proportional gain (> 0) used for efficient convergence. The overall gain λ used is decoupled to two different gains, $\lambda_r = 0.6$ for the x, y and θ variable and $\lambda_s = 0.7$ for the b, r variables in order for efficient and smooth convergence. These values are empirically obtained (see details in appendix C).

Results and Discussion

In this section, we describe the different scenarios used to validate the proposed approach on the BR^2 SCA.

Integrated approach (base case): Thirty (n = 30) random points in the operating range of the SCA system were collected and their pose $(x, y, z, q0, q1, q2, q3)$ information is recorded with the Polhemus magnetic sensor. VSNet is used for reaching the desired target images. For each test, the

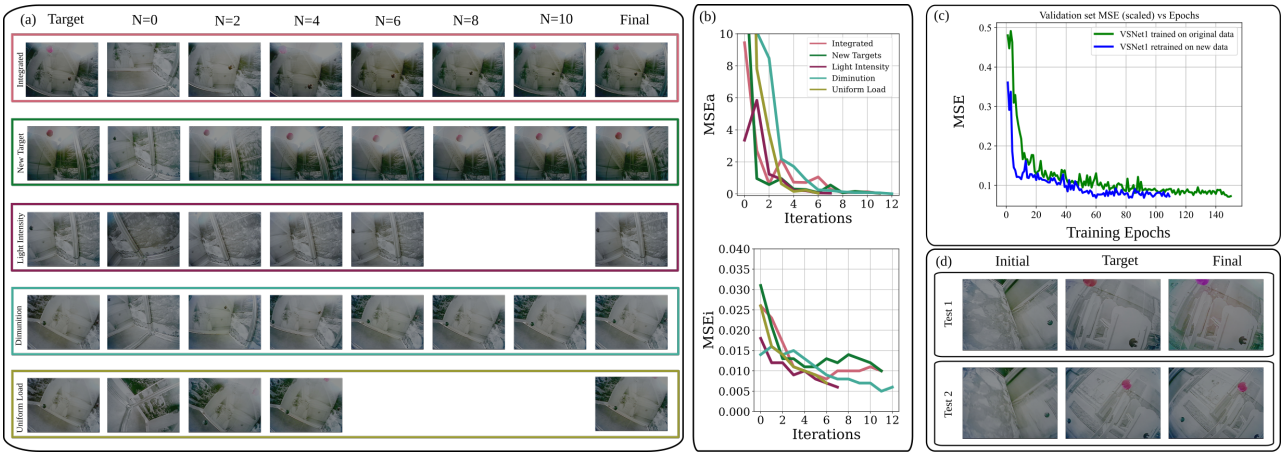

Figure 2: Results (a) The target, current images at different iterations (denoted by N) and the final image when the stopping condition $MSE_a < .05$ was reached and (b) the corresponding MSE_a and MSE_i over iterations for integrated, new targets and light intensity (c) Validation set MSE trend for original data trained on VSNet1, and new data retrained on VSNet1 and (d) The initial, target and the final image when the stopping condition $MSE_a < 0.01$ was reached on new data.

SCA system starts with a random initial configuration. The loop is terminated either when the error between predicted actuations of the target and current image (MSE_a) is less than 0.05 or when the number of iterations (N) reaches 15. The result for one of the test cases is shown in Fig.2(a). From the MSE_a plot in Fig.2(b), it can be observed that the error was reduced to less than one in four iterations. In the remaining iterations, the system has smooth transitions to further reduce the error. The average MSE in actuations, average MSE in image, and average Euclidean distance error between the final and target image for all the 30 tests is given in Table 1. Shown by our experiments, 90% of the data has less than 2 cm translation error (approximately the diameter of the SCA) and less than 0.24 rad for the rotation in 80% of the cases. The only test case with high error occurred as a result of no features in background in two different parts of the workspace causing the model to get confused between them. This can be addressed by having a non-plain background on all sides of the operating region.

New targets: New targets (as shown in Fig. 1(b)) were inserted in the workspace as proof of concept that the arm can reach even if there are new berries. Six target images (n = 6) were randomly collected, out of which three images contained the new target alone, and remaining three images contained both new and old targets (included during training). The integrated approach method was used with the stopping condition of $MSE_a < 0.05$ or when N equals 30. The target image, current images at different iterations, and the final image (when the stopping condition of $MSE_a < 0.05$ was reached) for one of the test cases is shown in Fig.2(a). As seen in the MSE_a plot in Fig. 2(b), the error reduced to less than 1 in two iterations and converges to the new target image in 11 iterations. The average MSE in actuations, average MSE in image, and average Euclidean distance error between the final and target image for all the six tests are given in Table 1 along with average translation and rotation errors.

Robustness to light changes: The robustness of our proposed method against exposure changes that are frequent in real world scenarios of harvesting was tested. Experiments were conducted with an extra light source in the environment, thus making the environment brighter. The integrated approach was used with the stopping condition as $MSE_a < 0.05$ or N = 30. The results for one case are shown in Fig. 2(a)-(b). For this case the target image was reached in six iterations. The errors are reported in Table 1.

Effect of diminution: This experiment is done to see the performance of our method when there are disturbances where the arm comes across an obstacle. This is replicated by restricting the functionality of the SCA by attaching 3D printed clips to its mid section as shown in Fig. 1(e). These clips restrict the bending functionality of the SCA in the sealed section of the arm. The integrated approach method was used with the stopping condition of $MSE_a < 0.05$ or N equal to 30. The approach was tested on 16 different random images. The results of one test case are shown in Fig. 2(a) and (b). As seen in the Fig. 2(b), the SCA reached the target image in 12 iterations. The errors for this case are reported in Table 1.

Uniform load: In a real-world when the arm tries to reach the berries, it can encounter other frequent disturbances like wind. We replicated this and evaluated by adding six uniform rings of 1.4 grams each on to the SCA equidistantly along the length as shown in Fig.1(d). The rings were fabricated with silicon and thus owing to flexibility of silicon, these rings don't affect the functionality of the SCA at the added locations. Ten experiments were conducted keeping the stopping condition as $MSE_a < 0.05$ or when N reaches 30. The integrated method with VSNet1 was used for this experiment. The results of one of the tests with stopping condition $MSE_a < 0.05$ is shown in Fig. 2(a) in which the target was reached accurately in six iterations. The total added weight is around 25% of the total weight of the SCA. The errors for this case are reported in Table 1.

Table 1: Results of experiments

Case and number of tests (n)	Avg. act MSE	Avg. image MSE	Avg. pos error (cm)	Std pos error(cm)	Percentage tests with pos error <2cm (%)	Avg. rot error (rad)	Std rot error (rad)	Percentage tests with rot error <0.24 rad(%)
Base (n=30)	0.055	0.013	1.648	2.046	90	0.233	0.301	80
New targets (n=6)	0.046	0.009	1.111	0.622	80	0.086	0.042	100
Light Intensity (n=10)	0.034	0.009	1.069	0.698	80	0.086	0.041	100
Uniform load (n = 10)	0.033	0.008	1.327	0.512	100	0.098	0.052	100
Diminution (n = 10)	0.049	0.012	1.237	0.557	90	0.077	0.032	100
Adaptability (n=5)	0.045	0.009	1.421	0.554	95	0.125	0.109	80

Adaptability to a new environment: In order to test the transferability and adaptability of the system to new environments, we changed the background of our structured environment. We added previously unseen images in the background of our setup and additionally included images on the ground (bottom of the environment). With the new background, data was recollected as described in Section IID. Our model was retrained on the new background data, with weights initialized as the trained weights from the original VSNet1. Five experiments were conducted using the retrained model in the new environment, keeping the stopping condition as $MSE_a < 0.01$. The results of two cases are shown in Fig. 2 (d), which took 27 and 23 iterations, respectively, to reach the stopping condition. The average number of iterations to reach the stopping condition for all the tests was 23. The mean translation error was 1.4212 cm and the mean rotation error was 0.1252 radians. We also observed that retraining VSNet took fewer steps and converged faster than before (converged in 110 epochs as opposed to 150 epochs from before). This can be seen from the validation set MSE graph in Fig. 2(c). New background data collection was efficient since it is automated. We performed experiments on the new background with a rigid stopping condition than before ($MSE_a < 0.01$) and found that our method is more accurate with a stricter stopping condition (with trade-off of more iterations). We also tested a few points in the new background with the previous model (trained on the original dataset), but it did not converge. This ascertains our claim that retraining the VSNet with new data was required.

Findings Summary To summarize our findings, our approach is able to reach the target positions with errors less than 1.5 cm for more than 80% of tests in all cases. In addition, unlike the previous work on the control of the BR^2 SCA (Satheeshbabu et al. 2020), the image based method also controls the orientation of the SCA. The rotation errors were less than 0.24 rad for 80% of the data. Furthermore, no abrupt changes in actuations were noticed leading to smooth convergence of the end effector to the target. Additionally, we not only control the position of the arm but also the orientation as compared to (Satheeshbabu et al. 2020). The system worked satisfactorily well in a new environment, considering the model was not fine-tuned to the new dataset. We observed that retraining VSNet took fewer steps and converged faster. Since we have a self-supervised system, collecting data and retraining on a new background can be done in a few hours.

Conclusion

To conclude, we demonstrated that visual servoing with deep learning-based architectures leads to a reliable reach-control of soft continuum arms, which are otherwise known to be difficult to control. Our method includes a feedback controller, on top of our modified VGG16-based image-to-actuation predicting model, to accommodate for hysteresis present in the soft-arm as well as the inaccuracies in the actuation predictions. We demonstrated our method in static reach problems in structured non-changing environments, which captures a large operational set for such arms. In these environments, we showed the robustness of our approach by replicating various scenarios in berry reaching problem, ranging from change in environment lighting, new targets in the environment, restricting the functionality of the arm to adding uniform load. Additionally, we not only control the position of the arm but also the orientation as compared to (Satheeshbabu et al. 2020). We also verified the transferablility of our neural network model to a new environment by changing the background images coupled with retraining. As a result, a huge advantage is that the users can easily re-purpose our system for various settings without any need for manual labeling since the data collection for training the prediction model is automated.

While we limited this investigation to the quasistatic response of the SCA, in the future we will explore visual servoing in dynamical environments for which we will leverage the recent advances in spatio-temporal neural networks (Hochreiter and Schmidhuber 1997). Furthermore, our future work will investigate visual servoing in cluttered environments where the soft arm leverages its flexibility and interaction with the obstacles in reaching desired regions that is more close to berry-harvesting settings.

Appendix

A. Experimental Setup Details

The experimental setup consists of five connected systems: Soft Continuum Arm (SCA), gantry, electrical control board, computers, and magnetic sensor. The SCA (Fig. 1(c)) is made of three Fiber Reinforced Elastomeric Enclosures (FREE)(Uppalapati and Krishnan 2018) - one bending, two

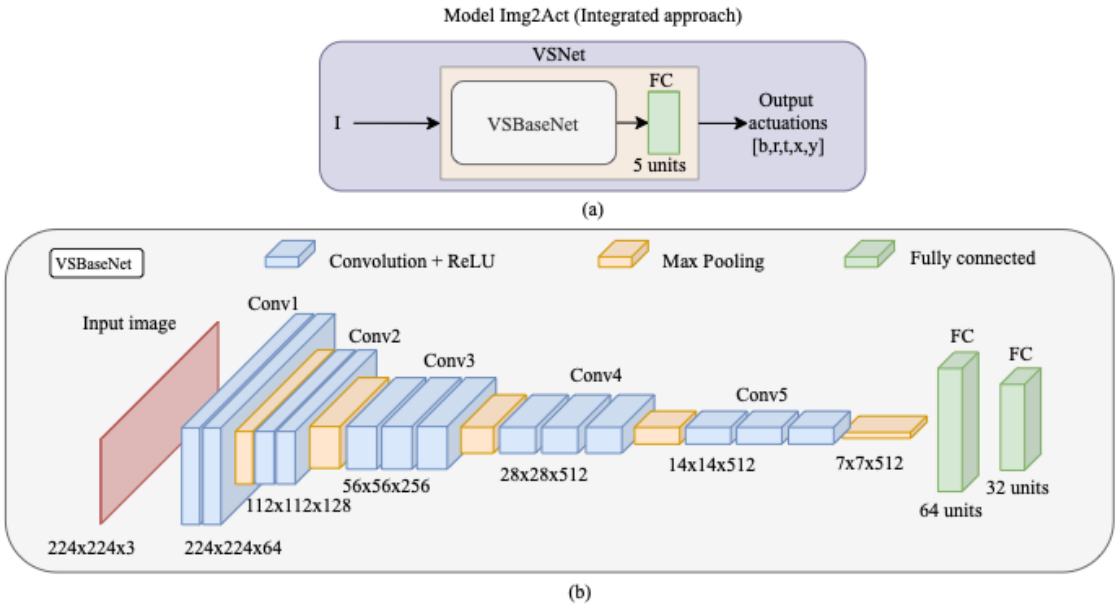

Figure 3: Workflow of our method to reach the target image given current image. (a) Integrated approach for obtaining a mapping from image to actuations (Img2Act) (b) Network architecture of VSBaseNet

rotational (one clockwise(CW) and another counterclockwise (CCW)) and is referred to as a BR^2 (Uppalapati and Krishnan 2021). It has an individually controllable pneumatic actuator for each FREE. The gantry (Fig. 1(a)) adds three degrees of freedom (DOF) to the SCA via an X and Y rail and a rotational mount (θ) for the SCA. The X and Y rails are belt driven by stepper motors (NEMA 17) and have an X travel of 45 cm and a Y of 42 cm with the origin defined by limit switches. Positioning on the gantry is open loop and must be periodically reset to reduce error accumulation. A servo motor (DS3218MG, DSSERVO) joins the SCA to the gantry and controls $\theta(\pm 90°)$. Together the SCA and gantry provide five DOF: bending, rotation, theta, x and y translation. Note that rotation is treated as one DOF as the two rotating FREEs are never actuated simultaneously. The CW and CCW rotations are distinguished by positive or negative value.

The electrical control board contains a pressure regulator (ITV0031-2UBL, SMC) for each FREE in the SCA, a PWM control board (PCA9685, Adafruit) for the servo and two stepper drivers (Big Easy Driver, SparkFun) to control the gantry translation. These devices are operated by a Raspberry Pi 4 (8GB) and an Intel NUC (NUC7i7), both running Ubuntu 18.04 with ROS Melodic. The Raspberry Pi is used to interface with the electrical control board while the NUC is used for the computationally intense control loop. The two computers communicate via ROS multimaster. A magnetic sensor (micro sensor 1.8, Patriot SEU, Polhemus), attached to the SCA, provides pose information about the tip of the SCA relative to a fixed source (TX1, Polhemus) origin that is placed at the center of gantry base.

B. Details about Method

B.1. Network Architecture: For our base network, VSBaseNet, we have used a VGG16-based network. we found that freezing the first 12 layers of the network and retraining the remaining layers gave optimal results in terms of loss and error. In addition to this, we added 2 fully connected layers (with 64, 32) with ReLU non-linearity. To aid regularization, we added batch normalization layers, dropout layers after the dense layers and also applied l_1 and l_2 regularizers to all the dense layers to decrease over-fitting with 0.0001 and 0.0005 as their respective regularization factors.

The workflow of *integrated* approach is given in Fig. 3(a). For this approach, the network used is VSNet which consists of the base network, VSBaseNet, along with a dense output layer with sigmoid activation. The network architecture of VSNet is given in Fig. 3(a), (b).

B.2. Training Dataset: We used electromagnetic tracking (Patriot SEU, Polhemus) with a short-range source (TX1, tracking area 2 to 60 cm) to get the ground truth absolute pose. The sensor has a positional accuracy of less than 1mm. The signal from the sensor provides the real-time spatial coordinates of the soft arm end in the form of $[x, y, z, quaternion]$, while $[theta, r_1, r_2, b]$ come from the requested actuations.
In our integrated approach, we used image data to predict the actuations of the soft arm. Using our self-annotated data collection method, a total of 7980 images corresponding to different poses were collected. The dataset is divided into training, validation and testing sets with 4910, 1676, and 2394 images respectively.

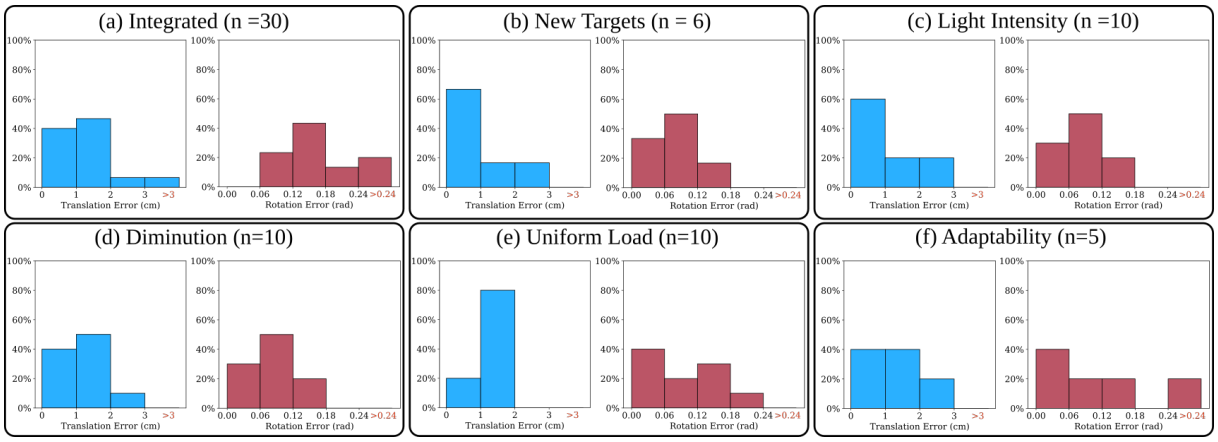

Figure 4: Translation error and rotation errors obtained for the test cases of (a) Integrated (30 points), (b) New Targets (6 points), (c) Change in light intensity (10 points), (d) Diminution of SCA functionality (10 points) (e) Uniform load (n = 10 points), and (f) Adaptability (n = 5 points).

B.3. Loss Function and Optimization: Our network takes in a single image (taken at the current arm pose), I, and outputs the absolute actuation values required to reach that pose. Since this is a regression problem, the last layer of the network outputs floats. The output of the network is in the form of a vector comprising of the 5 actuations (b, r, t, x, y).

We experimented with SGD and Adam optimizer for training and found that Adam optimizer converged faster and with less oscillation. We achieved best results using a time based learning rate scheduler with an initial learning rate of 0.01 and number of epochs as 150. The learning rate at each epoch was calculated as:

$$\eta_n = \eta_{n-1} * \frac{1}{1 + decay * n} \tag{3}$$

where η_{n-1} is the learning rate of the previous epoch, and n is the current epoch number. The value of decay is normally implemented as:

$$decay = \frac{\eta_0}{N} \tag{4}$$

where η_0 is the initial learning rate and N is the total number of epochs. We trained the model for 150 epochs after which the model reached saturation. We used a batch size of 128 to help in generalizing the model better. Using a lower or a higher batch size caused the validation loss to fluctuate.

C. Results

C.1. Estimation of λ_s and λ_r: The different actuations have a disproportionate effect on the SCA tip position. For example, a small change in x or y position will have a larger effect on the SCA tip than a similar change of the pressure in the SCA. The tip position is also dependent on the current shape of the SCA. Therefore λ, the proportional gain, is decoupled to two different gains, λ_r for the x, y and θ variable and λ_s for the b, r variables in order for efficient and smooth convergence. It is empirically obtained that the number of iterations required to reach a test image to obtain the actuation error (MSE_a) less than 0.1 is faster for values of λ_r and λ_s in the range of [0.5, 0.7] and [0.6, 0.8]. Based on this test case, the values of λ for all the following validation tests is set to $[\lambda_r, \lambda_s] = [0.6, 0.7]$.

C.2. Quantitative Evaluation: Figure 4 shows the translation and rotation errors for all the test points in each experiment in histograms. Translation error is calculated using the Euclidean distance between the ground truth (p_x, p_y, p_z) position (obtained from the Polhemus magnetic sensor) of the target image and final image for each test. Rotation error on the other hand is obtained using Euler's Axis-angle representation where R_1, R_2 are rotation matrices at the target and final images respectively. The quaternion pose information obtained by the Polhemus sensor is converted to rotation matrix in order to use the Eq.5.

$$e(R_1, R_2) = cos^{-1}\Big(\frac{trace(R_1 R_2^T) - 1}{2}\Big) \tag{5}$$

References

Bateux, Q.; Marchand, E.; Leitner, J.; Chaumette, F.; and Corke, P. 2018. Training Deep Neural Networks for Visual Servoing. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 3307–3314.

Gazzola, M.; Dudte, L.; McCormick, A.; and Mahadevan, L. 2018. Forward and inverse problems in the mechanics of soft filaments. *Royal Society open science*, 5(6): 171628.

George Thuruthel, T.; Renda, F.; and Iida, F. 2020. First-order dynamic modeling and control of soft robots. *Frontiers in Robotics and AI*, 7: 95.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Hughes, J.; Culha, U.; Giardina, F.; Guenther, F.; Rosendo, A.; and Iida, F. 2016. Soft manipulators and grippers: a review. *Frontiers in Robotics and AI*, 3: 69.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105.

Rus, D.; and Tolley, M. T. 2015. Design, fabrication and control of soft robots. *Nature*, 521(7553): 467–475.

Samtani, J. B.; Rom, C. R.; Friedrich, H.; Fennimore, S. A.; Finn, C. E.; Petran, A.; Wallace, R. W.; Pritts, M. P.; Fernandez, G.; Chase, C. A.; et al. 2019. The status and future of the strawberry industry in the United States. *HortTechnology*, 29(1): 11–24.

Satheeshbabu, S.; Uppalapati, N. K.; Fu, T.; and Krishnan, G. 2020. Continuous control of a soft continuum arm using deep reinforcement learning. In *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, 497–503. IEEE.

Shih, B.; Shah, D.; Li, J.; Thuruthel, T. G.; Park, Y.-L.; Iida, F.; Bao, Z.; Kramer-Bottiglio, R.; and Tolley, M. T. 2020. Electronic skins and machine learning for intelligent soft robots. *Science Robotics*, 5(41).

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Thuruthel, T. G.; Shih, B.; Laschi, C.; and Tolley, M. T. 2019. Soft robot perception using embedded soft sensors and recurrent neural networks. *Science Robotics*, 4(26).

Uppalapati, N. K.; and Krishnan, G. 2018. Towards pneumatic spiral grippers: Modeling and design considerations. *Soft robotics*, 5(6): 695–709.

Uppalapati, N. K.; and Krishnan, G. 2021. Design and modeling of soft continuum manipulators using parallel asymmetric combination of fiber-reinforced elastomers. *Journal of Mechanisms and Robotics*, 13(1).

Uppalapati, N. K.; Walt, B.; Havens, A.; Mahdian, A.; Chowdhary, G.; and Krishnan, G. 2020. A berry picking robot with a hybrid soft-rigid arm: Design and task space control. In *Proc. Robot.: Sci. Syst.*

Wang, X.; Fang, G.; Wang, K.; Xie, X.; Lee, K.-H.; Ho, J. D. L.; Tang, W. L.; Lam, J.; and Kwok, K.-W. 2020. Eye-in-Hand Visual Servoing Enhanced With Sparse Strain Measurement for Soft Continuum Robots. *IEEE Robotics and Automation Letters*, 5(2): 2161–2168.

Xu, F.; Wang, H.; Chen, W.; and Miao, Y. 2021. Visual Servoing of a Cable-Driven Soft Robot Manipulator With Shape Feature. *IEEE Robotics and Automation Letters*, 6(3): 4281–4288.

Xu, F.; Wang, H.; Wang, J.; Au, K. W. S.; and Chen, W. 2019. Underwater Dynamic Visual Servoing for a Soft Robot Arm With Online Distortion Correction. *IEEE/ASME Transactions on Mechatronics*, 24(3): 979–989.