VISUAL SERVOING FOR POSE CONTROL OF SOFT CONTINUUM ARMS

BY

SHIVANI KAMTIKAR

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2022

Urbana, Illinois

Adviser:

Professor Girish Chowdhary

ABSTRACT

For soft continuum arms, visual servoing is a popular control strategy that relies on visual feedback to close the control loop. 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 research 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. Integrated and modular approaches for estimating the actuations from the image are proposed and are experimentally compared. A proportional control law is implemented to reduce the error between the desired and current image as seen by the camera. The model together with the proportional feedback control makes the described approach robust to several variations such as new targets, lighting, loads, and diminution of the soft arm. Furthermore, the model lends itself to be transferred to a new environment with minimal effort.

To Aai, Baba and Sarthak, for their love and support.

ACKNOWLEDGMENTS

Throughout this journey, I have received a great deal of support from various people. A very big thank you to my advisor, Professor Girish Chowdhary, for his continued guidance and support. His lab provided the perfect environment for this work. His constant encouragement and feedback helped me advance in my research and go beyond my comfort zone.

I would also like to thank Professor Krishnan for all the excellent ideas that he proposed during our meetings. I would like to thank Dr. Naveen for his constant support and guidance. Frequent discussions with him helped me come up with new ideas and solutions.

The help from my other teammates, Samhita and Ben, was instrumental in making this research see the light of the day. For this, I am grateful. Professor Saurabh Gupta's courses were extremely helpful in keeping me up-to-date with the current literature in my field of research. The papers discussed in these classes helped me gain relevant knowledge in my field of research and pushed me in the right direction. Senior Ph.D. students in my lab, Arun and Anwesa, advised, guided, and inspired me throughout this journey. A big thank you to them.

I would also like to thank all the supporting organizations that provided funds for this work. This thesis is based on the work supported in part by Artificial Intelligence for Future Agricultural Resilience, Management, and Sustainability (AIFARMS) National AI institute in agriculture supported by Agriculture and Food Research Initiative (AFRI) grant no. 2020-67021-32799/project accession no.1024178 from the USDA National Institute of Food and Agriculture, by USDA-NSF NRI grant USDA 2019-67021-28989, NSF 1830343, and by joint NSF-USDA COALESCE grant, USDA 2021-67021-34418.

Big thanks to my family for their love and support. Finally, I could not have completed this thesis without the support of my friends. Special thanks to Pranav, Apurva, Rahul, Revanth and Mrignyani for their constant encouragement. Thank you.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Soft continuum arms (SCA) [1] 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.

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. With a camera (eye-in-hand configuration) at the distal tip of the SCA acting as a feedback sensor, the pose errors can be reduced. 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. This thesis describes a system for visual control of soft robotic arms with adequate accuracy that has not existed before, and therefore this contribution is novel and expected to be useful to soft robotics as well as other visual control problems. This research proposes the use of NN for visual servoing in SCA using two approaches: *integrated* and *modular*. These approaches are used to find the actuations (control parameters) required for the SCA to reach the desired goal position/location. One of the key motivations for this work is to avoid (or minimize) the use of position sensors on the SCA to control it with reasonable accuracy.

First, we discuss the emergence of SCAs and then move on to the challenges of controlling SCAs and the problem statement at hand.

1.1 EMERGENCE OF SOFT CONTINUUM ARMS (SCA)

Soft Continuum Arms (SCA) have been gaining a lot of traction from the robotics community recently due to their flexibility, adaptability, safe interaction with the environment, and low manufacturing cost [2][3][4]. SCA have great potential to be used in search and rescue [5], delicate handling of objects [6], and human assistive devices [7]. SCA can find use in many tasks in the agricultural sphere as well. They can be used for picking berries and other fruits from plants [8]. They have many advantages over conventional rigid arms due to their soft bendable structure. They can reach in the interiors of many plants and other objects without causing damage to themselves and their surroundings. They also have nearly infinite degrees of freedom as opposed to conventional robotic arms that have limited degrees of freedom. Due to this, SCAs can perform a myriad of tasks, many of which are not possible by conventional robots. In this work, we make use of the BR^2 soft continuum arm [9]. Fig. 1.1 shows the BR^2 SCA used in this work in different configurations.

1.2 CHALLENGES IN CONTROLLING SOFT CONTINUUM ARMS

The challenges in SCA control can be attributed mainly to the difficulties in modeling and sensing [4] its deformed shape. These challenges arise due to the nearly infinite degrees of freedom of SCA. Current modeling methods are either simplistic with a constant curvature assumption that work only in a 2D plane or valid for SCAs with short lengths [10]. On the other hand, Cosserat models [11] require expert knowledge for their implementation and therefore have been less explored by the community. In addition, even with effective models, there aren't cost-effective sensors [12, 13] to get the spatial position feedback of SCAs. Obtaining the mapping between the actuations or the control inputs of the SCA and the position is an important aspect of controlling the SCA. In works such as [14][15], there is no one-to-one mapping between the actuations and the position of the arm as each x,y,z position can be reached with multiple input actuations. Moreover, in these works, orientation is not considered and hence these methods can only control the position and not the orientation. However, in the work presented in this thesis, there is an underlying assumption that there is a one-to-one map between the actuations and the tip camera image. Thus, the position as well as the orientation is being controlled by this method. However, this adds challenges in the control of SCA since the mapping is not precise, considering the the accuracy of the pressure regulators used (0.5% Full Scale (F.S) hysteresis and repeatability of SMC ITV00312UBL). In other words, it is not guaranteed that the tip will reach the same 3D position and orientation when actuated to the same pressures making it prudent to have an accurate control system.

There is also a possibility of the SCA taking multiple equilibrium configurations under the influence of a heavy payload beyond 50 grams, and higher actuation pressures where buckling of one of the unpressurized actuators can occur. All of this poses more challenges in controlling the soft arm. The manipulator used in this work is operated in pressure ranges where there are no such extreme effects.

1.3 VISUAL SERVOING AS A METHOD TO CONTROL SCA

Visual servoing is a popular strategy to control or manipulate the arm of a robot using visual inputs. It involves giving visual inputs to the robot system and using these visual inputs, controlling the motion of the arm of the robot. Visual servoing has been well studied and validated in conventional rigid arms but has not been extensively validated on SCA. This is due to the inherent complexities in modelling and sensing of the SCA. However, visual servoing is a promising method to achieve accurate 3D pose control of SCA. The average

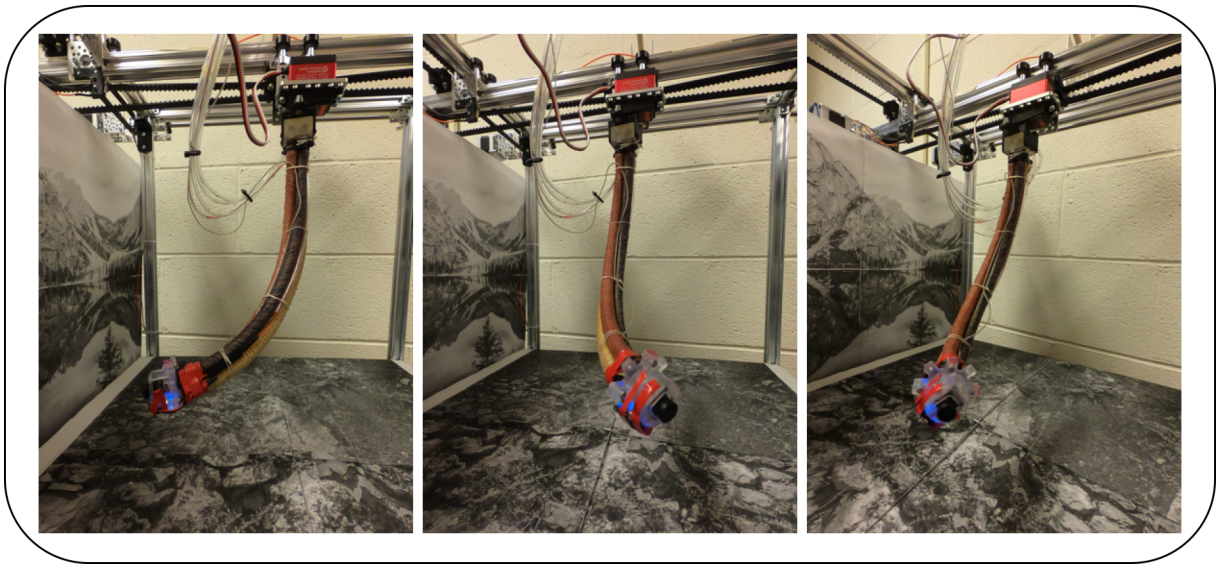

Figure 1.1: BR^2 Soft Continuum Arm in different configurations

error for current SOTA methods of controlling SCA is 2.5-3 cm. The method proposed in this work reduces this error to less than 2cm. Hence, it is shown that visual servoing is a reliable method to control SCA.

This thesis has demonstrated the use of soft arms for the precise reaching of a 3D target using a single image input in a structured environment. The thesis has also demonstrated the robustness of the system in various conditions and have also demonstrated the adaptability of the system in a new environment. This work can be used for several structured environment tasks that require delicate handling of items such as cookie placing and boxing (food industry) and light bulb boxing (manufacturing industry) where there is a need for fragile manipulation of the target objects.

The remainder of the thesis has been divided into the following chapters:

- Chap. 2 describes the background of control and visual servoing.
- Chap. 3 describes the overview of the proposed approach and the main contributions of this work.
- Chap. 4 and Chap. 5 explain the main implementation details of the approach along with experimental setup, data collection, network design and the control architecture.
- Chap. 6 describes the various experiments that were conducted to test the robustness of the system.
- Chap. 7 discusses the important observations and notable points from the research.

- Finally, we conclude in Chap. 9

This thesis summarizes the research on visual servoing of soft continuum arm, which has led to the following publication in :

S. K. Kamtikar, S. Marri, B. T. Walt, N. K. Uppalapati, G. Krishnan, and G. Chowdhary, "Visual servoing for pose control of soft continuum arm in a structured environment," IEEE Robotics and Automation Letters, 2022 [16]

CHAPTER 2: RELATED WORK

Considerable work has been done in the following areas:

1. SCA Control
2. Visual Servoing
   (a) Classical Visual Servoing
   (b) Neural Networks for Visual Servoing
   (c) Visual Servoing for SCA

Since SCA control and visual servoing are the main focus points of this thesis, these topics are discussed in more detail below.

2.1 SCA CONTROL

The challenges in soft arms such as difficulty in modeling and sensing, make it relatively difficult to replicate the work done on rigid arms. Many papers assume a steady-state assumption: under force equilibrium, the complete configuration of the soft arm can be modelled using a low-dimensional state space representation. Most kinematic steady-state models assume that the configuration space of a 3D SCA can be modeled using 3 dimensions, more commonly known as constant curvature (CC) approximation [17][18]. Due to the difficulties involved in cable-driven actuators, researchers started using sensors in order to compensate for the modeling uncertainties [19][20]. In [20], a configuration space controller is proposed which uses sensory information about the configuration and the sensory information about the joint variables to achieve asymptomatic tracking of a stationary configuration target [21]. Many works then progresses on to focus on more kinematically complex formulations by extending the CC model to a variable constant curvature (VCC) approximation where the curvature of each segment depends on the radius of the segment creating a high-dimensional configuration space [22][23]. Recent model-based techniques are based on the design of the arm. In [24],a closed-loop task space controller was applied. There is a question of scalability in such works. Currently, model-based methods are most popular for control of SCA.

Model-free approaches for control of SCAs are relatively new in the field. Many use the inverse kinematics models directly [25]. [26] presents a robust and accurate generic approach for closed-loop task space control. However, there exist scalability problems with this as well.

It was found that using simple neural networks helped the systems perform better than when computationally complex methods were used.

Papers like [27] investigate the use of a commercial fiber optic shape sensor for sensing the shape of the soft arm as it deforms. The sensor also attempts to detect the collision locations as the arm touches the obstacles along with the environmental shapes and material stiffness. Such methods are not cost-effective due to the use of expensive sensors. Simulation environments such as Elastica [28] have been proposed to model soft arms and perform complex tasks such as maneuvering through obstacles to reach a target. The work makes use of reinforcement learning which seems like a popular choice in many control papers. In [14][15], a Kirchoff rod model for training a reinforcement learning (RL) control policy on the BR2 arm was used. The RL policy was used for position control and accuracy was limited to a range between 2 cm to 3 cm (without control of the orientation). Our approach for SCA control makes use of visual servoing using neural networks, the background for which is explained below.

2.2 VISUAL SERVOING

2.2.1 Classical Visual Servoing

Visual servoing by its name is to control a system using vision. Classical visual servoing extracted features like points or lines using early computer vision techniques, and control was designed based on these features as seen in [29], [30]. This limited the types of objects that can be used, the environment lighting conditions, and are heavily dependent on the reliability of feature extraction methods. The introduction of using luminance of all pixels in the image [31] addresses the issue of object limitations, but still requires camera calibration. [32] on the other hand, represented images with principal component analysis that greatly reduces the dimensions and [33] used a moments-based approach to extract features. All these methods still require fine-tuning for different applications.

2.2.2 Neural Networks for Visual Servoing

As feature extraction techniques in computer vision improved with the advent of neural networks, so did their applicability in visual servoing. A related paper in this area [35] made use of deep neural networks like AlexNet [36] and VGG [37] to learn the relative pose that is fed into the control policy. This thesis work is primarily inspired by this approach. More advanced deep learning models like LSTMs [38], GANs [39] are seen in [40], [41] respectively.

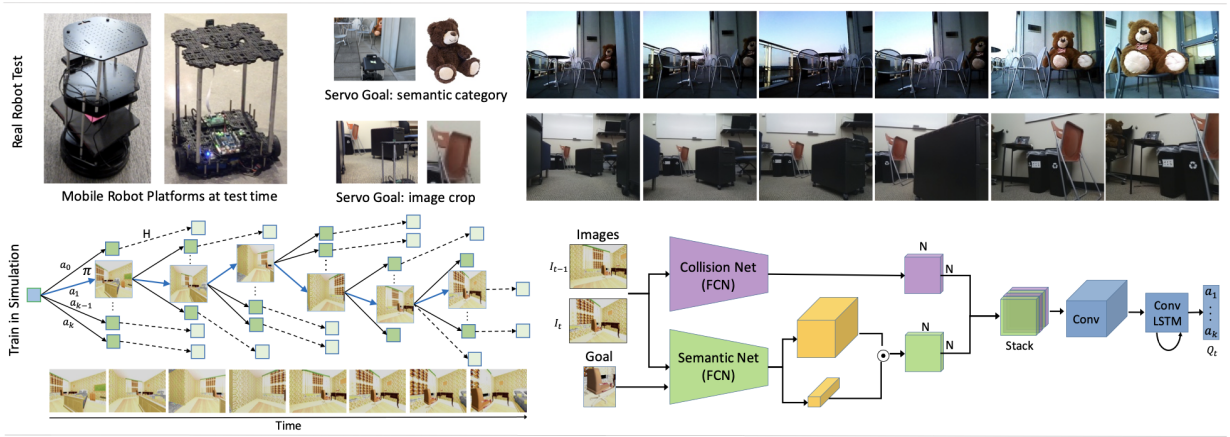

Figure 2.1: Workflow of method used in [34]. This method learns collision-free goal reaching entirely in simulation using RL techniques.

[42] on the other hand implemented a hybrid control policy where open-loop odometry was used as a coarse policy and a visual feedback policy was used to close the final error gaps to reach the targets. However, the above-mentioned works focused mainly on rigid arm visual servoing for which the system model is already known.

Reinforcement learning is popular choice in many vision-based visual servoing works [34][40]. In [34], an agent is trained to choose an action that takes it closer to the goal specified at the beginning of the experiment, all while avoiding obstacles. It incorporates high-level semantics using previously-labeled human data. The challenge here is to get a huge amount of labeled image data. This can be possibly solved by creating a self-supervised system with our robot to collect data with little to no human-intervention. DroNet [43] makes use of the concept of learning-by-demonstration predict the collision probability and the steering angle. This can be replicated in our system by having a neural network predict the controls of the arm (pressure, angle etc.) and the collision probability. Many visual servoing methods on rigid arms are done in simulation without extensive testing on the real robot [41][44]. These methods are challenging to implement as creating a simulation environment for SCAs is difficult due to many sim2real issues. [45] used a combination of reinforcement learning, kinematic modelling, and visual feedback to develop a controller for a target-driven task. The robustness of this approach is questionable since there are multiple modules and the CNN and RL are separated and the time taken for both of these is large.

In 3D visual servoing, feature tracking is followed by pose estimation or 3D reconstruction. Methods like [46][19] reconstructed the robot using its homography and kinematics. Problems with 3D reconstruction of the scene or the robot include computational complexity since it requires a large amount of image data to reconstruct scenes or the robot.

2.2.3 Visual Servoing for SCA

Visual servoing for SCAs has gained a lot of traction recently, due to their difficulty in modeling and pose control. Works like [47], [22] 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 [48] in order to track the camera motion but was limited to 2D space. Works like [49] focus on an adaptive controller using the CC model for manipulation of SCA in constrained environments. They investigate tasks in which the SCA interacts with the environment which causes the kinematic model to change. Many works focus on deforming the SCA to a specific shape and form [50]. The work focuses on an online estimation of the deformation Jacobian that maps the motion of the robot to the deformation caused by it. They compare this work to various model-free and model-based methods. In the work [51], they do visual tracking of the arm using a simulation model. A kinematic model of the SCA is obtained. This method relies heavily of the simulation model of the SCA which is difficult to build.

In this thesis, we are interested in eye-in-hand image-based visual servoing in a 3D framework. We rely on neural-networks to handle the feature extraction and mapping to actuation. The control policy then computes the error between the predicted actuations of current and target images.

CHAPTER 3: OVERVIEW

Visual servoing for SCAs has gained a lot of traction recently, due to their difficulty in modeling and pose control. Works like [47], [22] 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 [48] in order to track the camera motion but was limited to 2D space. In this work, the author is interested in eye-in-hand image-based visual servoing in a 3D framework. The research relies on neural-networks to handle the feature extraction and mapping to actuation. The control policy then computes the error between the predicted actuations of current and target images.

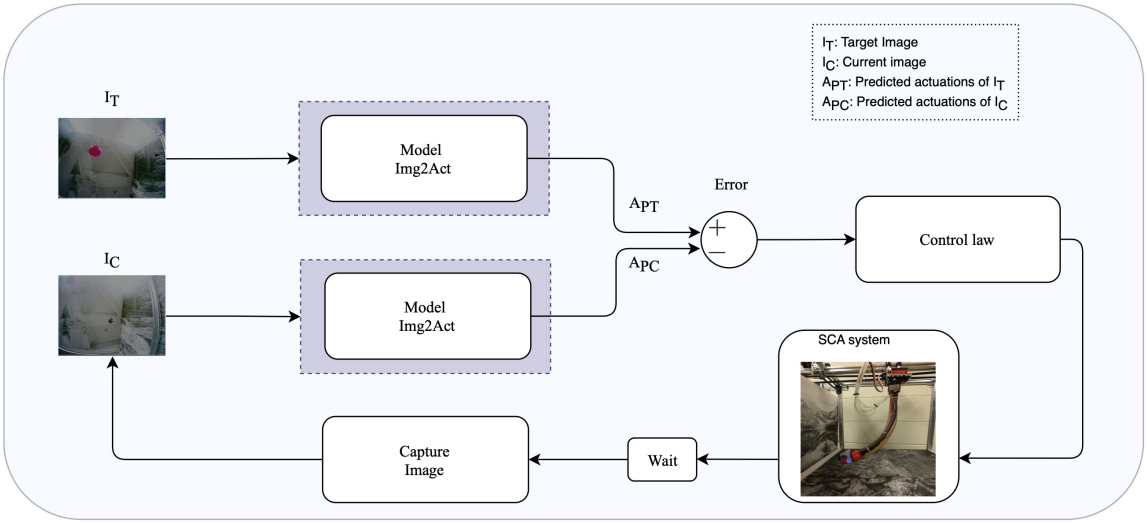

Figure 3.1: Workflow of our method to reach the target image given current image.

This work propose two approaches, *integrated* and *modular*, to estimate the pose of the soft manipulator, and control it using visual servoing in a structured environment. The integrated approach predicts the actuation directly for a given input image, which is useful when the environment is changed. The modular approach on the other hand, first predicts the pose for the given image and then maps the predicted pose to actuation which is particularly useful when the SCA is changed. Both these frameworks take a single RGB image, I, and predict the control inputs (actuations) required to reach the corresponding pose of the soft arm (current pose). Using this information the error in the geometrical features of the current and target images, as well as the error between the current and target actuations is calculated. These errors are reduced by using visual feedback to estimate the control commands needed to reach the desired target pose. Through experiments, the research shows that both the approaches perform well, with the *integrated* approach being robust

to various changes such as light intensity, diminution of SCA, added weights, etc. Fig. 3.1 shows the overall workflow of the proposed approach.

The main contributions of this work are listed below:

1. Controlling the 3D position and the orientation of the soft continuum arm (SCA) using an end tip camera.

2. Demonstrating the feasibility of control using a single tip camera, thereby avoiding the need for multiple sensors that currently limit SCA control.

3. Two different methods are presented, integrated and modular, that are generalizable and adaptable to different conditions in a structured environment.

CHAPTER 4: IMPLEMENTATION

4.1 EXPERIMENTAL SETUP

The experimental setup consists of five connected systems: Soft Continuum Arm (SCA), gantry, electrical control board, computers, and magnetic sensor. The SCA(Fig. 4.1(c)) is made of three Fiber Reinforced Elastomeric Enclosures (FREE)[52] - one bending, two rotational (one clockwise(CW) and another counterclockwise (CCW)) and is referred to as a BR^2 [9]. It has an individually controllable pneumatic actuator for each FREE. The gantry (Fig. 4.1(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 was reset between tests and data collection runs to reduce error accumulation. A servo motor (DS3218MG, DSSERVO) joins the SCA to the gantry and controls $\theta(\pm 90$M.S.). 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.

4.2 DATA COLLECTION

A 1200 TVL camera (Caddx Firefly, Micro FPV Camera w/ VTX), which is a low-cost, lightweight (4.2 grams), small form-factor camera was mounted on the distal tip of the SCA and images from the camera at various views were collected by moving the soft arm and gantry. The setup of the soft arm is given in Fig. 4.1(c). The process is automated and the inputs are given in the form of actuations, such as pressures (b, r), x, y and angle (theta).

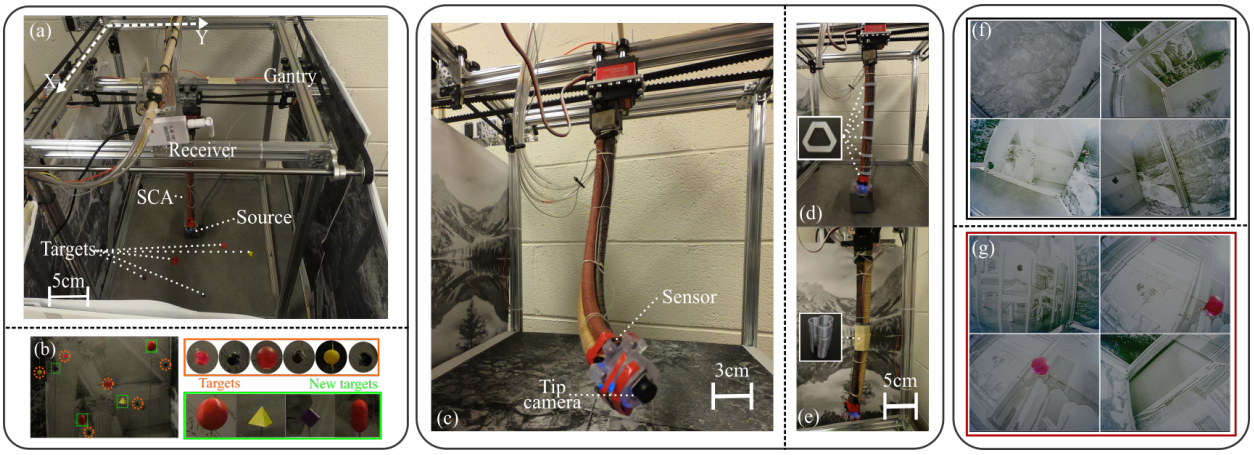

Figure 4.1: Experimental setup: (a) BR^2 SCA attached to a rotating servo that can move in X and Y direction in the gantry along with the targets and the wireless receiver to receive the tip camera image. (b) Four new targets (not seen in training) along with the targets used for training. (c) BR^2 SCA 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) Four sample images used for the training with first background and (g) four sample images used for the training with the second background.

Images of the scene are captured at discrete configurations throughout the workspace while state data (actuations and sensor readings) is collected to self-annotate the images. A few examples of images taken by the camera are shown in Fig. 4.1 (f) and (g). The images have a resolution of 640x480 pixels.

Initial attempt at data collection consisted of a workspace with a plain background, with no distinguishable features. This kind of data did not prove to be useful as there were no features to extract. The dataset consisted of images that looked similar to each other and recognising textureless images was not possible for the neural network as it could not differentiate between the actuation to image mappings. This setup was changed to include backgrounds with outdoor scenes as seen in Fig. 4.1. The background added essential features and textures that made it possible for the images in the dataset to look different.

4.3 METHOD SELECTION

4.3.1 Pairs of Image Method

The first method that was tried for the visual servoing task was estimation of relative pose between any two images in the dataset. This method was chosen to find the relative

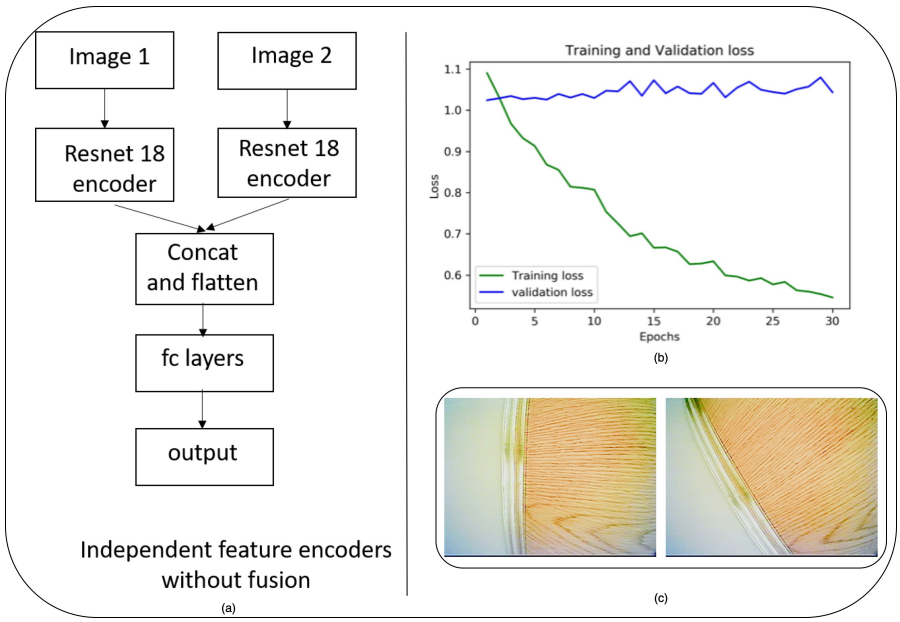

Figure 4.2: Pairs of images method: (a) Independent encoder architecture where 2 images are fed to independent encoders and then concatenated, (b) Training and validation plot and (c) Images taken from two different configurations of the SCA

pose between any two poses (configurations) of the SCA in order to get the transformation between these two poses. This would eventually be used to get the relative pose between the current and the target image. Images and their corresponding actuations were collected and the difference between the actuation values of each pair of images was calculated. The dataset hence consisted of pairs of images and their corresponding relative poses as labels. Fig. 4.2(a) shows the block diagram of the model that was used. Very soon, it was found out that this method was not useful. The validation loss fluctuated between a range (was almost constant) so it seemed like the model was not learning as seen in Fig. 4.2(b). Further analysis showed that the failure of this method could be caused by the following reasons:

1. The images didn't have enough texture for the model to learn anything. There was little or no overlap between the image pairs and hence, finding the relative pose between two images was difficult. The data collection method was crude and the dataset consisted of images that looked very similar to each other i.e., the dataset was not rich. Moreover, 40% of the dataset consisted of images of corners of the workspace that had little or no difference between them as seen in Fig. 4.2 (c)

2. The target objects in the workspace were very small as compared to the rest of the image and they are not clearly visible in the images. However, this turned out to be false as proved by the work done later in this thesis.

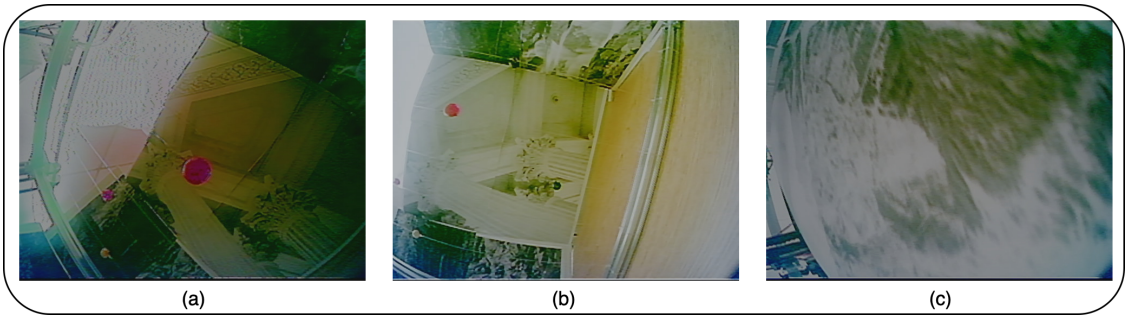

Figure 4.3: References image method: (a) Target image, (b) Reference image, and (c) Initial image

3. While testing, it was observed that the SCA moved randomly because for two different configurations (two different combinations of actuations), it was seeing the same image. Unless we give some prior knowledge, or we have some texture/overlap, the arm won't orient itself correctly. In Fig. 4.2 (c), the SCA doesn't know which which direction (orientation) to do to since the images look essentially the same.

This method was eventually discarded.

4.3.2 Reference Image Method

Due to the failure of the previous method, the reference image method was explored. This method was inspired by the work done in [35]. Here, the relative pose between the current and a reference image is predicted along with the relative pose between the target and the reference image. These two relative poses are then used to predict the relative pose between the current and the target image. This reference image could be any image in the workspace that has some overlap between the current and target image. In this work, the reference image was selected such that it can see some part of the target image so that there is some visual overlap between the images. The network model used in this method was similar to the model used in 4.3.1 as shown in Fig. 4.2. This method makes use of an additional step and is more complex that the final method selected as explained in the rest of the thesis. This method was discarded due to the following reasons:

1. The model was severely overfitting. The model seemed to be having problems during inference while finding relative pose between the current and target image (given a fixed reference image)

2. There is a necessity of providing a fixed reference image. This is not very practical because the environment will keep changing and so will the reference image. Hence,

collecting reference images for every new environment is not practical.

3. Finding relative pose between the image pairs (current, reference, and target) was complex and took long during inference.

To overcome these problems and to simplify the problem statement, the method of predicting the absolute actuation values (control of SCA) given image inputs was proposed. This method is described in this thesis as the primary method used for the visual servoing task on SCA and explained in detail in the subsequent chapters.

4.4 NETWORK ARCHITECTURE

Due to the ability of Deep Convolutional Neural Networks (CNNs) to automatically extract features from large training datasets, they have shown to be effective in various computer vision applications, such as image recognition [36], segmentation and also have been studied to estimate the pose of a robot manipulator given image inputs [35]. Inspired by this, this work uses VGG16 [37], to estimate the input actuation values required to reach a specific pose of the soft manipulator arm using image inputs. VGG16 [37] is originally trained for classification task on 1.2 million ImageNet images that has around 138 million parameters. Since this task is not exactly similar to the image classification task, this work modifies the final few layers, performed transfer learning by using previously trained VGG16 weights on some layers and fine-tune it on the data used in this research which effectively helped the network to learn new features pertaining to the task at hand. AlexNet[36] and ResNet[53] were also experimented with. However, AlexNet is a much older type of neural network and it did not give good results. ResNet on the other hand proved to be more compplex than required for the task at hand. Research also found that freezing the first 12 layers of the VGG-based network and retraining the remaining layers gave optimal results in terms of loss and error. Experiments were also done by unfreezing all CNN layers, freezing all CNN layers, freezing and adding a combination of CNN and fully connected layers. However, all of these experiments had issues like overfitting, underfitting, problems in generalization etc. In addition to the above changes, 2 fully connected layers (with 64, 32 units, respectively) with ReLU non-linearity were added to the network. To aid regularization, batch normalization layers, dropout layers after the dense layers were added. l_1 and l_2 regularizers were also applied to all the dense layers to decrease overfitting with 0.0001 and 0.0005 as their respective regularization factors. This network is called VSBaseNet. Fig. 4.5(a) shows the complete network architecture of the base network, VSBaseNet.

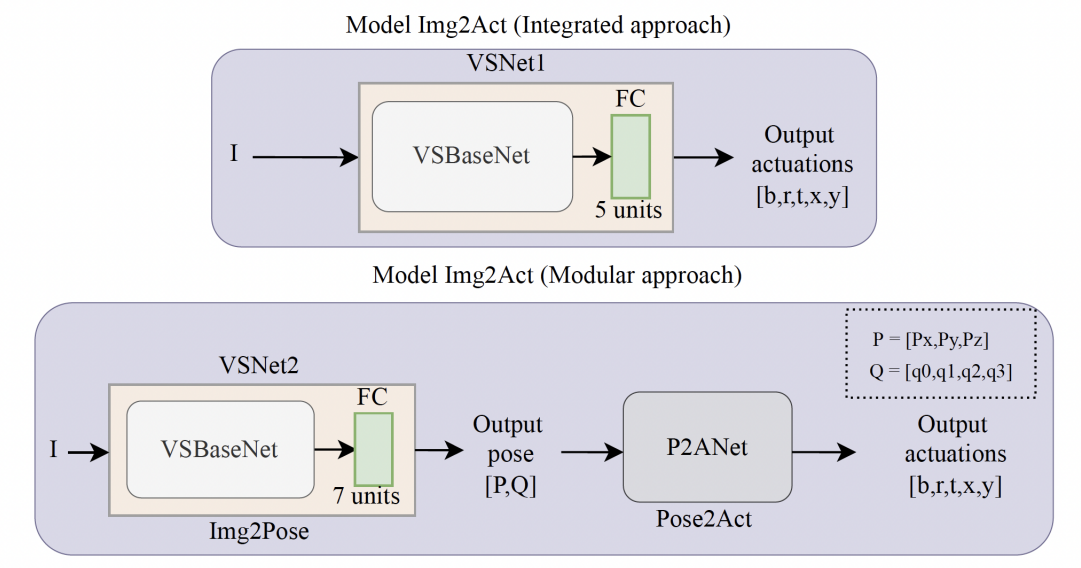

Figure 4.4: Two different approaches (modular and integrated) for obtaining a mapping from image to actuations (Img2Act)

Two different approaches namely, *integrated approach* and *modular approach*, were used to predict the actuations from the input image. These two approaches were implemented and tested in order to see their effectiveness in various scenarios as shown in section 2. The workflows of both the approaches are given in Fig. 4.4 and their network architectures details are given below.

4.4.1 Integrated Approach:

In the integrated approach, the network directly outputs the actuations given an input image, I. Here, the network used is VSNet1 which consists of the base network, VSBaseNet, along with a dense output layer with sigmoid activation. Since the work deals with a regression task, the final dense layer consists of five units that output 5 floats corresponding to the five input actuations: bending (b), rotation (r), theta (t), and the gantry (x and y). The details of VSNet1 are given in Fig. 4.4 and Fig. 4.5.

4.4.2 Modular Approach:

For the modular approach, we divided the image-to-actuation step in two parts (modules): image-to-pose, and pose-to-actuation. The image-to-pose (Img2Pose) module takes in a single image (taken at the current arm pose), I, and outputs the pose information. The VSNet2 network is used to take an input image and output the pose in the form of a vector

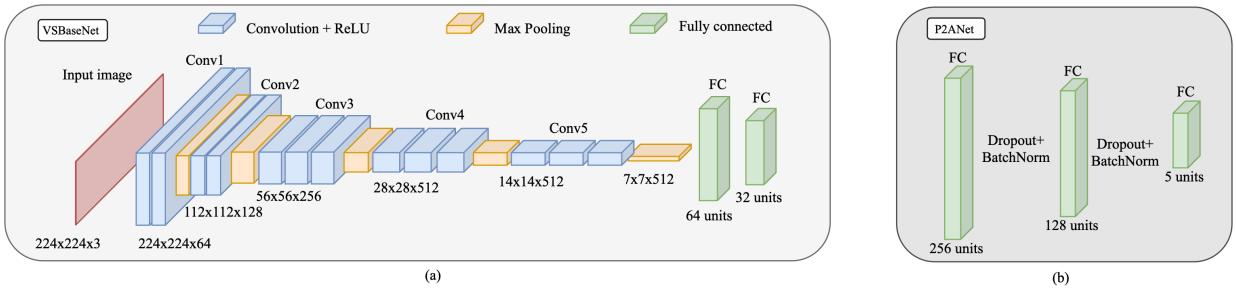

Figure 4.5: (a) Network architecture of VSBaseNet and (b) Network architecture used for the pose to actuation mapping (P2ANet).

comprised of the position and orientation (quaternion) information, $[p_x, p_y, p_z, q_0, q_1, q_2, q_3]$. This pose information is fed as input to the P2ANet network which predicts the corresponding mapping of actuation inputs in the form of another vector consisting of actuation values $[b, r, t, x, y]$. The network architecture of VSNet2 is similar to VSNet1 (it uses the same base network, VSBaseNet), except the last (output) layer, which has 7 units corresponding to the 7 output floats. The P2ANet consists of 3 dense layers with 256, 128, and 5 units respectively along with ReLU non-linearity in the first dense layer and sigmoid activation the last layer. Batch normalization and dropout layers were added to aid regularization. The network architecture of VSNet2 and P2ANet is given in Fig. 4.4, Fig. 4.5 (a) and (b).

4.5 TRAINING

4.5.1 Dataset

Using the self-annotated data collection method built in this work, a total of 7980 images corresponding to different poses were collected. The absolute pose data with respect to the initial configuration was also noted for each of the images. The system 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. This sensor is flexible, lightweight ($<$ 2 g), has a positional accuracy of less than 1mm and does not hinder or alter the performance of the soft arm. 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 both approaches, image data was used to predict the actuations (integrated approach) or pose (modular approach) of the soft arm. The range of values for each of the 5 actuations were as follows: Bending (b): 14 to 22 psi (discrete values with steps of 2 psi) (96.5 to 151.7

kPa in 13.8 kPa steps); Rotation (r): -18 to 18 psi (discrete values with steps of 2 psi)(-124.1 to 124.1 kPa in 13.8 kPa steps); Theta (t): +6 to -6 degrees (discrete values with steps of 2); x: 14, 16 and 18 cm (discrete values); y: 14, 16, 18, and 20 cm (discrete values).

The dataset is divided into training, validation and testing sets with 4910, 1676, and 2394 images respectively. The ground truth values for the integrated approach consist of the absolute actuation values corresponding to the pose of the soft arm for each image. A CSV file containing 5 columns corresponding to each of the actuation values was created, and then split into training, validation and testing label files for training purposes. This method was repeated for the image-to-pose part of the modular approach where the ground truth values consisted of the pose information. This entire data collection process is automated.

4.6 LOSS FUNCTION AND OPTIMIZATION

The 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 either the pose $(p_x, p_y, p_z, q_0, q_1, q_2, q_3)$ or the 5 actuations (b, r, t, x, y). To regress absolute values of pose or actuations, the use of mean-squared error (MSE) loss function which computes the mean of squared errors between the ground truth values and the predictions was done.

$$loss(I) = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{4.1}$$

Here, Y_i corresponds to the ground truth actuations whereas, $\hat{Y}_i$ corresponds to the predicted actuations of the input images. Other kinds of loss metrics, such as Mean Absolute Error (MAE) loss, were also experimented however, MSE gave the best results in terms of representing the accuracy of the predictions.

Experiments were done with SGD and Adam optimizer for training and found that Adam optimizer converged faster and with less oscillation. Best results were achieved 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:

$$\eta_n = \eta_{n-1} * \frac{1}{1 + decay * n} \tag{4.2}$$

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:

$$decay = \frac{\eta_0}{N} \tag{4.3}$$

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

CHAPTER 5: CONTROL ARCHITECTURE

5.1 FORMULATION OF CONTROL LAW

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, which could also be dependent on the path taken by the manipulator[9]. (ii) Inaccuracies in the trained model to fit the pose to actuations could also lead to large deviations from the target. To overcome these inherent errors, the following control update was integrated, where the error between the current predicted actuations and the target actuations at various iterative steps are fed back into the input until the tip converges to the target image (I_T) within reasonable accuracy, as shown in the Fig.3.1:

$$A_{RC}(k+1) = A_{RC}(k) - \lambda(A_{PC}(k) - A_{PT}) \tag{5.1}$$

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 the arm operates in a quasi-static manner in each iteration step and at the end of each step k, it is made to reach static equilibrium where all the external forces are balanced by the actuation forces. The current image for the next iteration is taken only after this equilibrium is reached after 6 seconds and hence the *wait* after system actuations as shown in figure 3.1. 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, λ_r for the x, y and θ variable and λ_s for the b, r variables in order for efficient and smooth convergence.

5.2 ESTIMATION OF λ

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. It is empirically obtained that the number of iterations required to reach a test image to obtain the actuation error (MSE_a) less than 5 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

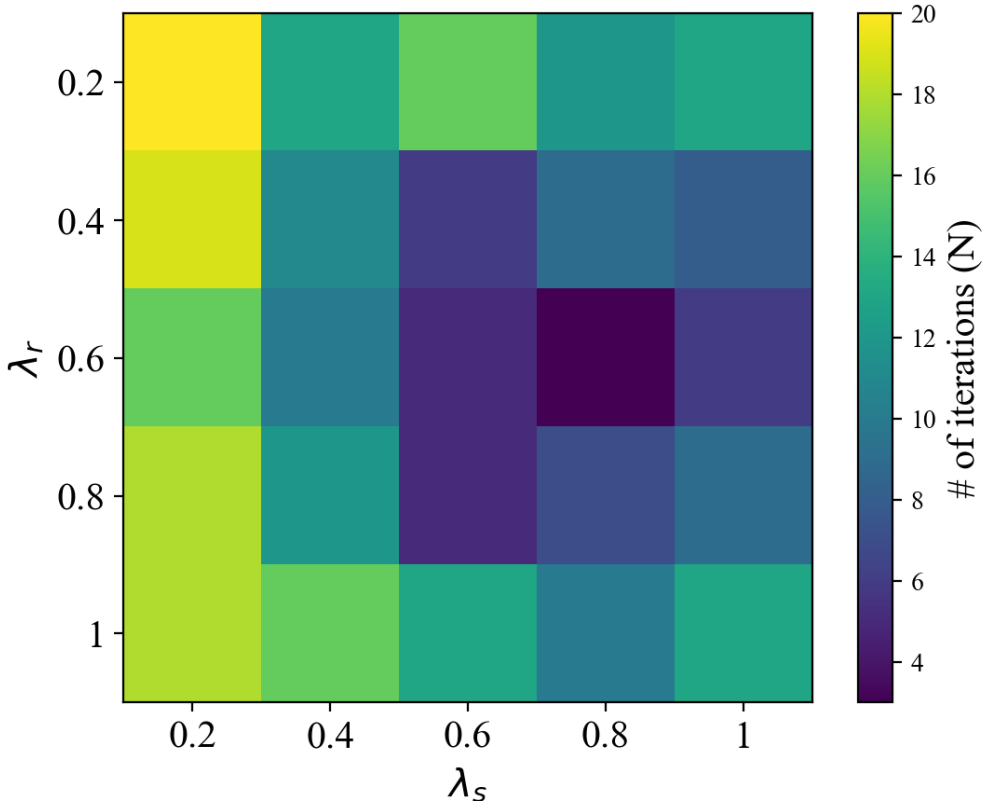

Figure 5.1: Empirical estimation of λ_r and λ_s

following validation tests is set to $[\lambda_r, \lambda_s] = [0.6, 0.7]$.

CHAPTER 6: EXPERIMENTS AND RESULTS

6.1 FORMULATION OF NORMALIZED, UNITLESS METRIC

During the earlier naive testings, an MSE metric was used which was formulated as follows:

$$MSE_a = \frac{1}{N}\sum_{i=1}^{5}\left(a^i_{observed} - a^i_{target}\right)^2 \tag{6.1}$$

Here, N = 5 corresponds to the 5 actuations - b (kPa) (i=1), r (kPa) (i=2), t (radians) (i=3), x (m) (i=4), y (m) (i=5); $a_{observed}$ is observed actuation, a_{target} is target actuation. However, it was realized that without proper units, it is difficult to gauge the accuracy of the system. The system controls the five input actuations (b, r, t, x, y) of the SCA, all of which have different units.

A normalized MSE metric that is scale-invariant and unit-less is formulated in order to represent the accuracy of the system is shown in Eq. 6.2 [16]. In this equation, each term is divided by the resolution i.e., the minimum change a state can undergo. Here, N = 5 corresponds to the 5 actuations - b (kPa) (i=1), r (kPa) (i=2), t (radians) (i=3), x (m) (i=4), y (m) (i=5); $a_{observed}$ is observed actuation, a_{target} is target actuation and $a^i_k = 0.1$ is the scaling factor $\forall i \in \{1, 2, 3, 4, 5\}$. All states are rounded off to their first decimal point and hence 0.1 (0.1 kPa, 0.1, radians, 0.1 m) is the scaling used. Based on this metric, we define the stopping condition for all the tests conducted to be $MSE_a < 5$ or when the number of iterations (N) reaches 15. These values were empirically decided with two criteria: a) reduce the translation and rotation error and b) reach the target image in a reasonable number of iterations.

$$MSE_a = \frac{1}{N}\sum_{i=1}^{5}\left(\frac{a^i_{observed} - a^i_{target}}{a^i_k}\right)^2 \tag{6.2}$$

6.2 INTEGRATED APPROACH

Thirty (n = 30) random points in the operating range/workspace of the SCA system were collected and their pose $(x, y, z, q0, q1, q2, q3)$ information is recorded with the Polhemus magnetic sensor. VSNet1 (shown in Fig 4.5 is used for reaching the desired target images. For each test, the SCA system starts with a random initial configuration.

The target image, current images at different iterations, and the final image (when the

Table 6.1: Comparison between Integrated and Modular approach

Method and number of tests (n)	Avg. MSE_a (normalized - no units)	Avg. Euclidean dist. error (cm)	Avg. rotation error (radians)
Integrated approach (n=30)	5.587*	1.6481	0.2325
Modular approach (n=15)	6.489*	1.8002	0.4261

stopping condition of $MSE_a < 5$ was reached) for one of the test cases is shown in Fig 6.1. It took eleven iterations for it to reach the desired stopping condition. From the position and rotation error plots in Fig. 6.1, it can be observed that the error was reduced to less than 2 cm in six iterations. In the remaining iterations, the system transitions to further reduce the error. The accuracy of this approach is also shown with the quantitative metrics of average MSE in actuations, average Euclidean distance error, and average rotation error between the final and target image for all the 30 tests as reported in Table 6.1. We would like to highlight that for two of the test cases where the arm looks at the ground with no features initially, it reached with 77 and 30.7 MSE_a at the 15th iteration leading to average $MSE_a = 5.587 > 5$.

Fig. 6.2 shows the histogram of translation and rotation errors for the 30 test points. 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 first converted to rotation matrix in order to use the Eq.6.3.

$$e(R_1, R_2) = cos^{-1}\Big(\frac{trace(R_1 R_2^T) - 1}{2}\Big) \tag{6.3}$$

6.3 MODULAR APPROACH

The modular approach (as in Fig.4.4)) was tested on fifteen random points (n = 15) in the workspace within the range of the SCA and the gantry. The pose information for all the test images was recorded using the Polhemus magnetic sensor. VSNet2 predicts the pose given an input image and the P2ANet outputs the corresponding actuations for the predicted pose. The quantitative metrics using the 15 tests is given in Table 6.1. The target image, current images at different iterations, and the final image (when the stopping condition of

Table 6.2: Results of experiments (integrated approach)

Method and number of tests (n)	Avg. MSE_a (normalized - no units)	Avg. Euclidean dist. error (cm)	Avg. rotation error (radians)
New targets in workspace (n = 6)	2.828	1.1108	0.0858
Lighting changes (n = 10)	3.485	1.0690	0.0857
Diminution (n = 10)	3.777	1.4491	0.1350
Uniform load (n = 10)	3.296	1.3274	0.0975
Change in background (n = 5)	0.778	1.4212	0.1252

$MSE_a < 5$ was reached) for one of the test cases is shown in Fig.6.1. As seen in Fig.6.1, the final image obtained after converging in 12 iterations is a little farther from the desired target image, however the orientation is much closer to the desired orientation using this method. It was also observed that two of the tests with MSE_a of 172.3 and 44.7 at the 15th iteration, resulting in average $MSE_a = 6.489 > 5$. Fig.6.2 shows the translation and rotation errors for the 15 test points.

6.4 NEW TARGETS

The integrated approach is tested new targets (as shown in Fig. 4.1(b)) inserted in the workspace. 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 target image, current images at different iterations, and the final image (when the stopping condition of $MSE_a < 5$ was reached) for one of the test cases is shown in Fig.6.1(a). As seen in the position error plot in Fig. 6.1(b), the error reduced to less than 2 cm in three iterations and converges to the new target image in 11 iterations. The quantitative metrics using the six tests are given in Table 6.2. Fig. 6.2(c) shows the histogram of translation and rotation errors for the six test points.

6.5 ROBUSTNESS TO LIGHT CHANGES

The robustness of the integrated approach against light exposure changes was tested with an extra light source in the environment, thus making it brighter. Tests were conducted at an average illuminance of 341.4 lx compared to 155.4 lx for training and other testing (Light Meter Model R8130, Reed Instruments). The results for one case are shown in Fig. 6.1(a)-(b). For this case the target image was reached in six iterations. The quantitative metrics

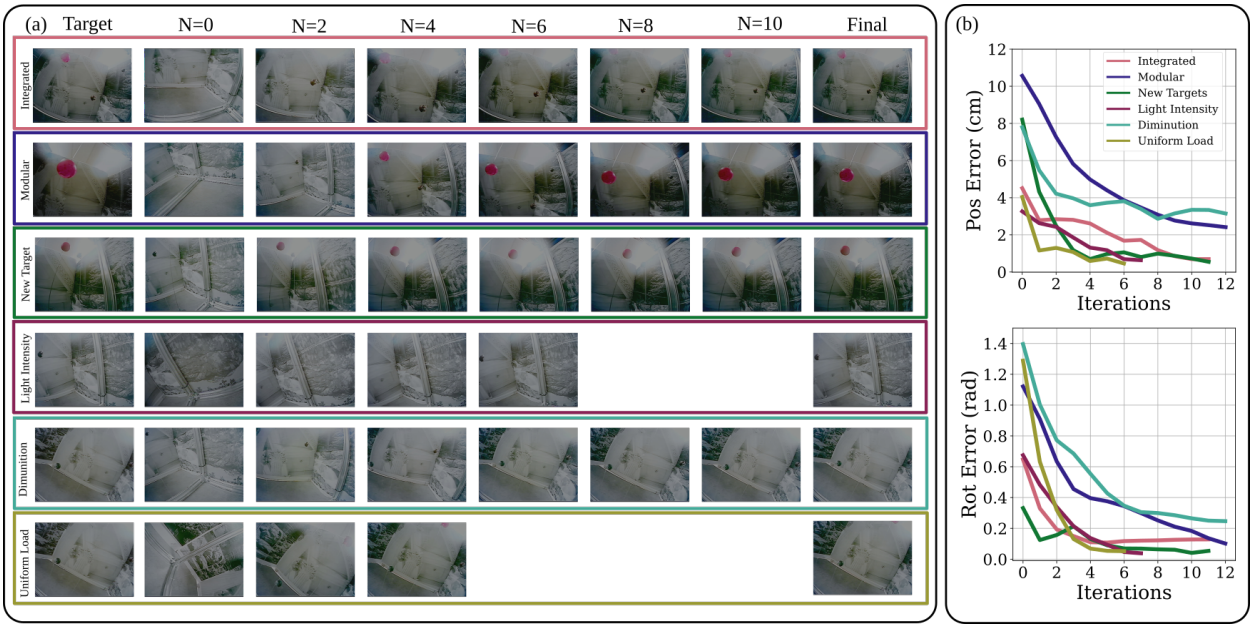

Figure 6.1: Results (for one case each): (a) The target, current images at different iterations (denoted by N) and the final image when the stopping condition $MSE_a < 5$ was reached and (b) the corresponding position and rotation error over iterations for integrated, modular, new targets, light intensity, diminution and uniform load.

using the ten tests are given in Table 6.2. Fig. 6.2(d) shows the histogram of translation and rotation errors for the ten test points (n = 10).

6.6 EFFECT OF DIMINUTION

For this experiment, the functionality of the SCA was restricted by attaching 3D printed clips to its mid section as shown in Fig. 4.1(e). These clips restrict the bending functionality of the SCA in the sealed section of the arm. The integrated approach was tested on 10 different random images. The results of one test case are shown in Fig. 6.1(a) and (b). As seen in the Fig. 6.1(b), the SCA reached the target image in 12 iterations. The quantitative metrics using the 10 tests are given in Table 6.2 along with the histogram of translation and rotation errors for the 10 test points (n = 10) in Fig. 6.2(e).

6.7 UNIFORM LOAD

Six uniform rings of 1.4 grams each were added on to the SCA equidistantly along the length as shown in Fig.4.1(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

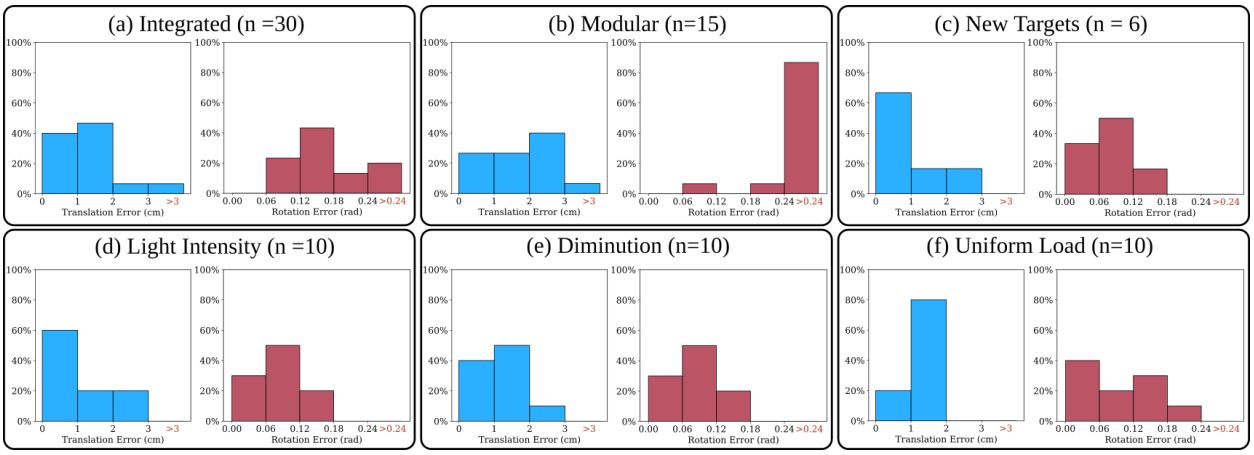

Figure 6.2: Histogram of translation and rotation errors obtained for the test cases of (a) Integrated (30 points), (b) Modular (15 points), (c) New Targets (6 points), (d) Change in light intensity (10 points), (e) Diminution of SCA functionality (10 points), and (f) Uniform load (n = 10 points).

locations. A total of ten experiments were conducted. The integrated approach was used for this experiment, where results of one of the tests with stopping condition $MSE_a < 5$ is shown in Fig. 6.1(a). The target was reached accurately with loads in six iterations. The total added weight is around 25% of the total weight of the SCA. The quantitative metrics using the ten tests (n = 10) are given in Table 6.1 along with the histogram of translation and rotation errors for the ten test points in Fig.6.2(f).

6.8 ADAPTABILITY TO A NEW ENVIRONMENT

In order to test the transferability and adaptability of the system to new environments, the background of the structured environment was changed. Previously unseen images were added in the background of the setup and additionally included images on the ground (bottom of the environment). With the new background, data was recollected as described in Section IID. The 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 $MSE_a < 1$ and maximum iterations as 15. The results of two cases are shown in Fig. 6.3 (b) 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. It was also observed that retraining VSNet1 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. 6.3(a).

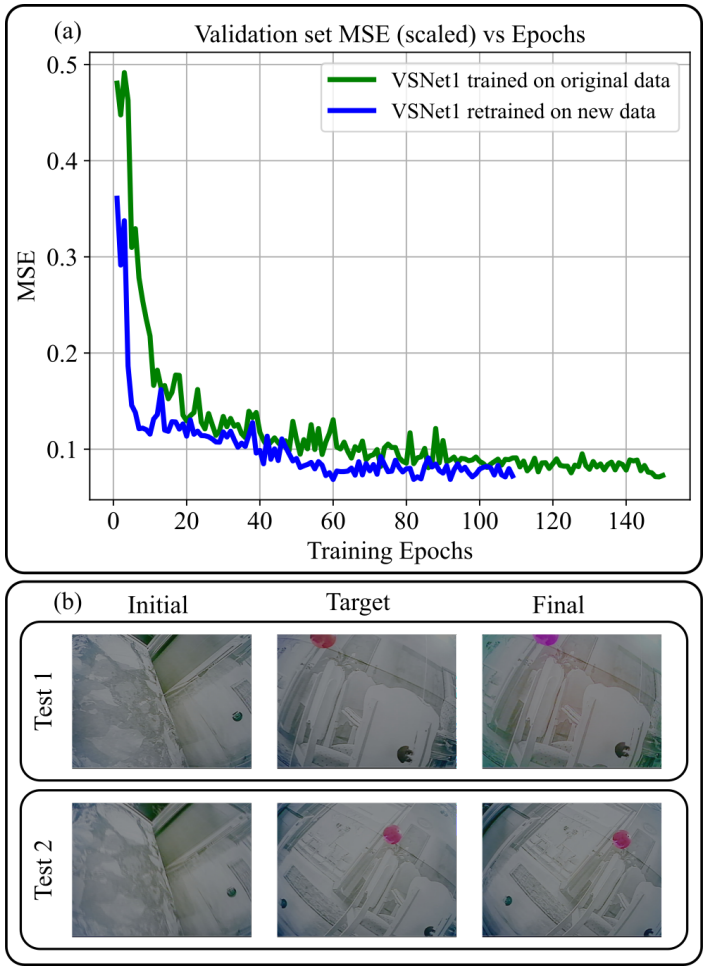

Figure 6.3: Results: (a) Validation set MSE trend for original data trained on VSNet1, and new data retrained on VSNet1 and (b) The initial, target and the final image when the stopping condition $MSE_a < 1$ was reached.

CHAPTER 7: DISCUSSION

In this thesis, we demonstrate that visual servoing using deep neural networks leads to accurate and **robust** control of a soft continuum arm, which is otherwise known to be hard to control using model-based techniques. The work showcased two approaches for deep-learning based visual servoing of SCAs, the first utilizing an *integrated* (image to actuation) approach, and the second utilizing a *modular* approach (image to pose and pose to actuation). In the integrated approach as seen in Fig. 6.2(a), 90% of the data has less than 2 cm translation error (approximately the diameter of the SCA) and 80% less than 0.24 radians for the rotation. The test cases with higher error occurred on the extremities of the workspace (edge of the gantry in this case). Such errors are likely a result of no features in background in two different parts of the workspace causing the model to get confused between them. In these cases, the gantry bottom had a plain background and the model was confused for a similar image on the other corner of the gantry. This can be addressed by having a non-plain background on all sides of the operating region. Excluding these outliers reduces the average translation error to less than **1.4 cm**.

The modular approach can be useful when either the SCA is changed (by retraining P2ANet alone) or the background is changed (retraining VSNet2 alone). Although the modular approach does a reasonable job in reducing the errors for more than 50% of the data, from Fig. 6.2(a) and (b) it was found to be less accurate compared to the integrated approach. This may be due to errors that accumulate due to the intermediate pose estimation step. One thing to note here is that in both approaches the architecture directly computes the control actuation, as such, this indicates that deep learning based visual servoing can be directly utilized in a control architecture with a simple linear control law. The reasonable tracking from the architecture indicates that further optimization of control was not necessary for this problem setup which was focused on the static reach problem. However, optimization and learning-based-control could be interesting directions for future work in problems like dynamic tracking, or trying to reach objects that are not reachable with static actuation by using the arm's momentum.Although the modular approach does a reasonable job in reducing the errors for more than 50% of the data, from Fig.6.2(a) and (b) it can be observed that the integrated approach does better than the modular approach with less translation and rotation errors. The modular approach can be useful when either the SCA is changed (have to retrain only P2ANet) or the background is changed (have to retrain only VSNet2). This is a useful approach in cases where either of the above mentioned changes

are of frequent occurrence.

From the histogram plots for different cases Fig. 6.2(c-f), the integrated approach is robust to several changes the SCA may encounter (such as loads, disturbances and diminution) for performing different real-world tasks. The 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 [14], the image based method also controls the orientation of the SCA where the rotation errors were less than 0.24 **radians** for 100% of the data and no abrupt changes in actuations were noticed leading to smooth convergence of the end effector to the target. Furthermore, the system worked satisfactorily well in a **new environment**, considering the model was not fine-tuned to the new dataset. The data collection was efficient for a new background since it's **self-supervised**. It was observed that retraining VSNet1 took fewer steps and converged faster. Since the retraining of the model was done with images where the ground is visible, the system was able to converge upon encountering the ground during testing. Experiments were performed on the new background with a more rigid stopping condition ($MSE_a < 1$) and it was found that the method is capable of performing more accurately with a stricter stopping condition. A few points were also tested in the new background with the previous model (trained on the original dataset), but it did not converge. This ascertains the claim that retraining the VSNet1 with new data was required. Since the work has a self-supervised system, collecting data and retraining on a new background can be done in a few hours.

CHAPTER 8: FUTURE WORK

Reaching goal objects while avoiding obstacles using visual servoing in a semi-structured or an unstructured environment, is an essential step towards taking the work closer to more real-world dynamic scenarios such as in agricultural settings (eg: autonomous harvesting). While work has been done in goal-based reaching and collision detection, most of it is based on rigid robots. The challenges in soft arms make it relatively difficult to replicate the work done on rigid arms. Work on SCA has been done in simulation where the authors use reinforcement learning algorithms to avoid obstacles[28]. Works on traversability of robots [54][55] rely on observations made by the robot while it is navigating in an environment. This method can be used to predict the general traversable and untraversable areas of an environment and plan an optimal path to the goal.

Planned work for the near future: Considering current literature on reach-avoid using visual servoing, we propose a system that attempts to reach a given target in a cluttered environment. The proposed system will have two cameras: global camera and tip camera. The system will have obstacles in the form of vertical bars to mimic stems in a plant. We make the assumption that the global camera can see the target, obstacles as well as the arm. The global camera will be used initially to manipulate the arm from an arbitrary position to a position where the tip camera can see the target. The tip camera will take images of the scene from various angles (different configurations of the arm). The images obtained from the tip camera and the global camera will be used to recreate the scene using structure from motion algorithms. This 3D reconstruction will be used to get waypoints and a trajectory to the target from the current position.

After getting the trajectory, the main challenge is getting the arm follow the path without colliding with the obstacles. An object detection system that will be used to detect the target object. After the tip camera spots the target, reinforcement learning will be used to reach it using visual inputs. The action space consists of the controls of the arm such as, pressures, extrusion. The reward will be based on whether the arm reaches the target or not. In real-world agricultural settings, we have traversable obstacles such as leaves and untraversable obstacles such as stems. To replicate this, the obstacles in our system will also be classified as traversable and untraversable. Based on this classification, the robot will make the decision of whether it can take the next step or if it needs to retract.

I intend to investigate visual servoing in cluttered environments where the soft arm leverages its flexibility and interaction with the obstacles in reaching desired regions during my

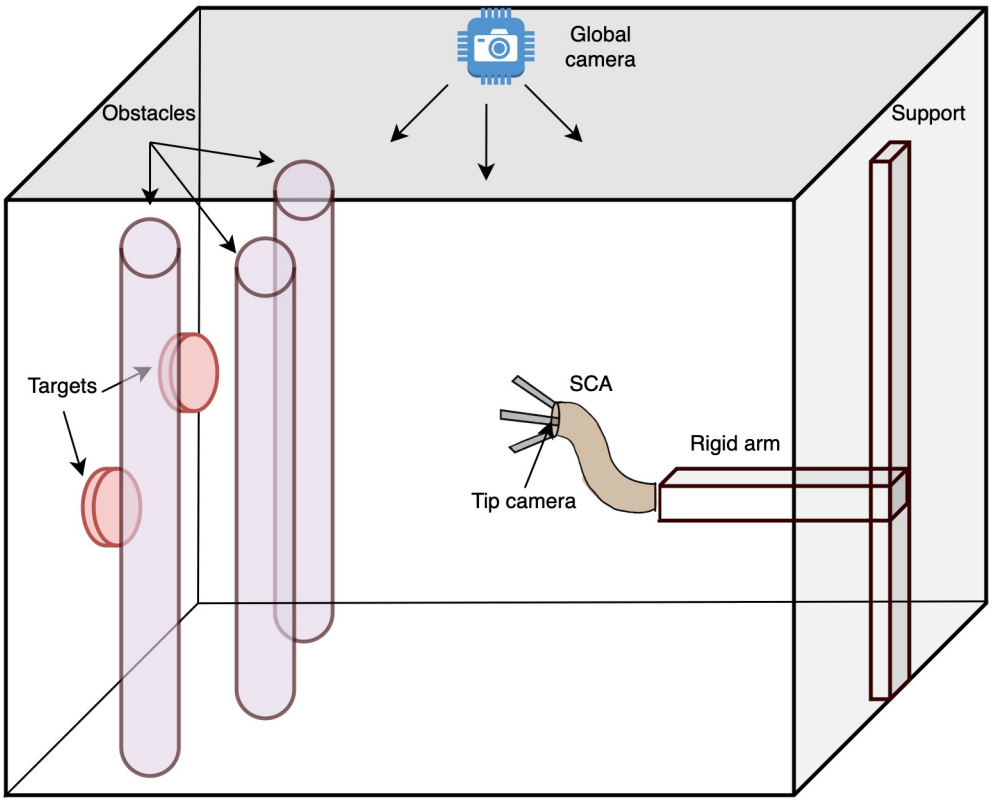

Figure 8.1: Proposed system for future work

Ph.D. The eventual goal of this research is to apply it in autonomous harvesting scenarios to pick berries using a hybrid arm (with soft plus hard components).

CHAPTER 9: CONCLUSIONS

To conclude, this research 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. This method includes a feedback controller, on top of the 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. The research demonstrated the proposed 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 the approach through various types of experiments ranging from change in environment lighting, new targets in the environment, restricting the functionality of the arm to adding uniform load. Additionally, the system not only controls the **position** of the arm but also the **orientation** as compared to [14]. The work also verified the **transferablility** of the 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 this system for various settings without any need for manual labeling since the data collection for training the prediction model is automated.

While the investigation was limited to the **quasi-static** response of the SCA, in the future we will explore visual servoing in dynamic environments for which the work will leverage the recent advances in spatio-temporal neural networks [38]. In future work, we would like to validate the effectiveness of the modular approach by changing the SCA that has a different architecture than the BR^2 SCA used in this work. Furthermore, acquiring a target image is limited to random exploration or a teaching policy method currently. In future work, we would like to give a query object as the target to which the arm should reach [34]. This work is restricted to controlling the soft arm moving with zero collisions with its environment. With obstacles, the data collection process will no longer be automatic as shown in this work. Therefore, in future work, we intend to investigate visual servoing in cluttered environments where the soft arm leverages its flexibility and interaction with the obstacles in reaching desired regions.

REFERENCES

[1] J. Hughes, U. Culha, F. Giardina, F. Guenther, A. Rosendo, and F. Iida, "Soft manipulators and grippers: a review," *Frontiers in Robotics and AI*, vol. 3, p. 69, 2016.

[2] N. K. Uppalapati and G. Krishnan, "Design and modeling of soft continuum manipulators using parallel asymmetric combination of fiber-reinforced elastomers," *Journal of Mechanisms and Robotics*, vol. 13, no. 1, p. 011010, 2021.

[3] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Applied bionics and biomechanics*, vol. 5, no. 3, pp. 99–117, 2008.

[4] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[5] S. Neppalli, B. Jones, W. McMahan, V. Chitrakaran, I. Walker, M. Pritts, M. Csencsits, C. Rahn, and M. Grissom, "Octarm-a soft robotic manipulator," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 2569–2569.

[6] B. T. Phillips, K. P. Becker, S. Kurumaya, K. C. Galloway, G. Whittredge, D. M. Vogt, C. B. Teeple, M. H. Rosen, V. A. Pieribone, D. F. Gruber et al., "A dexterous, glove-based teleoperable low-power soft robotic arm for delicate deep-sea biological exploration," *Scientific reports*, vol. 8, no. 1, pp. 1–9, 2018.

[7] A. Zlatintsi, I. Rodomagoulakis, P. Koutras, A. Dometios, V. Pitsikalis, C. S. Tzafestas, and P. Maragos, "Multimodal signal processing and learning aspects of human-robot interaction for an assistive bathing robot," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 3171–3175.

[8] R. Bogue, "Fruit picking robots: has their time come?" *Industrial Robot: the international journal of robotics research and application*, 2020.

[9] N. K. Uppalapati and G. Krishnan, "Design and modeling of soft continuum manipulators using parallel asymmetric combination of fiber-reinforced elastomers," *Journal of Mechanisms and Robotics*, vol. 13, no. 1, 2021.

[10] T. George Thuruthel, F. Renda, and F. Iida, "First-order dynamic modeling and control of soft robots," *Frontiers in Robotics and AI*, vol. 7, p. 95, 2020.

[11] M. Gazzola, L. Dudte, A. McCormick, and L. Mahadevan, "Forward and inverse problems in the mechanics of soft filaments," *Royal Society open science*, vol. 5, no. 6, p. 171628, 2018.

[12] B. Shih, D. Shah, J. Li, T. G. Thuruthel, Y.-L. Park, F. Iida, Z. Bao, R. Kramer-Bottiglio, and M. T. Tolley, "Electronic skins and machine learning for intelligent soft robots," *Science Robotics*, vol. 5, no. 41, 2020.

[13] T. G. Thuruthel, B. Shih, C. Laschi, and M. T. Tolley, "Soft robot perception using embedded soft sensors and recurrent neural networks," *Science Robotics*, vol. 4, no. 26, 2019.

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

[15] N. K. Uppalapati, B. Walt, A. Havens, A. Mahdian, G. Chowdhary, and G. Krishnan, "A berry picking robot with a hybrid soft-rigid arm: Design and task space control," *Proceedings of Robotics: Science and Systems, Corvalis, Oregon, USA*, p. 95, 2020.

[16] S. K. Kamtikar, S. Marri, B. T. Walt, N. K. Uppalapati, G. Krishnan, and G. Chowdhary, "Visual servoing for pose control of soft continuum arm in a structured environment," *IEEE Robotics and Automation Letters*, pp. 1–1, 2022.

[17] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *Journal of robotic systems*, vol. 20, no. 2, pp. 45–63, 2003.

[18] M. N. Boushaki, C. Liu, and P. Poignet, "Task-space position control of concentric-tube robot with inaccurate kinematics using approximate jacobian," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5877–5882.

[19] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury, "Task-space control of continuum manipulators with coupled tendon drive," in *Experimental Robotics*. Springer, 2009, pp. 271–280.

[20] B. A. Jones and I. D. Walker, "Practical kinematics for real-time implementation of continuum robots," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1087–1099, 2006.

[21] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft robotics*, vol. 5, no. 2, pp. 149–163, 2018.

[22] F. Xu, H. Wang, W. Chen, and Y. Miao, "Visual servoing of a cable-driven soft robot manipulator with shape feature," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4281–4288, 2021.

[23] T. Mahl, A. Hildebrandt, and O. Sawodny, "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," *IEEE transactions on robotics*, vol. 30, no. 4, pp. 935–949, 2014.

[24] B. Conrad and M. Zinn, "Closed loop task space control of an interleaved continuum-rigid manipulator," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1743–1750.

[25] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, "Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 823–834, 2015.

[26] M. C. Yip and D. B. Camarillo, "Model-less feedback control of continuum manipulators in constrained environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 880–889, 2014.

[27] K. C. Galloway, Y. Chen, E. Templeton, B. Rife, I. S. Godage, and E. J. Barth, "Fiber optic shape sensing for soft robotics," *Soft robotics*, vol. 6, no. 5, pp. 671–684, 2019.

[28] N. Naughton, J. Sun, A. Tekinalp, T. Parthasarathy, G. Chowdhary, and M. Gazzola, "Elastica: A compliant mechanics environment for soft robotic control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3389–3396, 2021.

[29] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.

[30] F. Chaumette and E. Malis, "2 1/2 d visual servoing: a possible solution to improve image-based and position-based visual servoings," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, 2000, pp. 630–635 vol.1.

[31] C. Collewet and E. Marchand, "Photometric visual servoing," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 828–834, 2011.

[32] K. Deguchi, "A direct interpretation of dynamic images and camera motion for vision guided robotics," in *1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems (Cat. No.96TH8242)*, 1996, pp. 313–320.

[33] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1116–1127, 2005.

[34] F. Sadeghi, "Divis: Domain invariant visual servoing for collision-free goal reaching," *arXiv preprint arXiv:1902.05947*, 2019.

[35] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, "Training deep neural networks for visual servoing," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3307–3314.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

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

[38] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

[39] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[40] F. Sadeghi, A. Toshev, E. Jang, and S. Levine, "Sim2real view invariant visual servoing by recurrent control," 2017.

[41] O. M. Pedersen, E. Misimi, and F. Chaumette, "Grasping unknown objects by coupling deep reinforcement learning, generative adversarial networks, and visual servoing," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 5655–5662.

[42] S. Paradis, M. Hwang, B. Thananjeyan, J. Ichnowski, D. Seita, D. Fer, T. Low, J. E. Gonzalez, and K. Goldberg, "Intermittent visual servoing: Efficiently learning policies robust to instrument changes for high-precision surgical manipulation," 2020.

[43] A. Loquercio, A. I. Maqueda, C. R. Del-Blanco, and D. Scaramuzza, "Dronet: Learning to fly by driving," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1088–1095, 2018.

[44] Y. Li and J. Košecka, "Learning view and target invariant visual servoing for navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 658–664.

[45] W. Liu, Z. Jing, H. Pan, L. Qiao, H. Leung, and W. Chen, "Distance-directed target searching for a deep visual servo sma driven soft robot using reinforcement learning," *Journal of Bionic Engineering*, vol. 17, no. 6, pp. 1126–1138, 2020.

[46] V. K. Chitrakaran, A. Behal, D. M. Dawson, and I. D. Walker, "Setpoint regulation of continuum robots using a fixed camera," *Robotica*, vol. 25, no. 5, pp. 581–586, 2007.

[47] F. Xu, H. Wang, J. Wang, K. W. S. Au, and W. Chen, "Underwater dynamic visual servoing for a soft robot arm with online distortion correction," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 979–989, 2019.

[48] X. Wang, G. Fang, K. Wang, X. Xie, K.-H. Lee, J. D. L. Ho, W. L. Tang, J. Lam, and K.-W. Kwok, "Eye-in-hand visual servoing enhanced with sparse strain measurement for soft continuum robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2161–2168, 2020.

[49] H. Wang, B. Yang, Y. Liu, W. Chen, X. Liang, and R. Pfeifer, "Visual servoing of soft robot manipulator in constrained environments with an adaptive controller," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 41–50, 2016.

[50] R. Lagneau, A. Krupa, and M. Marchal, "Active deformation through visual servoing of soft objects," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8978–8984.

[51] Z. Zhang, T. M. Bieze, J. Dequidt, A. Kruszewski, and C. Duriez, "Visual servoing control of soft robots based on finite element model," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2895–2901.

[52] N. K. Uppalapati and G. Krishnan, "Towards pneumatic spiral grippers: Modeling and design considerations," *Soft robotics*, vol. 5, no. 6, pp. 695–709, 2018.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[54] L. Nardi and C. Stachniss, "Long-term robot navigation in indoor environments estimating patterns in traversability changes," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 300–306.

[55] S. Hosseinpoor, J. Torresen, M. Mantelli, D. Pitto, M. Kolberg, R. Maffei, and E. Prestes, "Traversability analysis by semantic terrain segmentation for mobile robots," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 1407–1413.