Deep Direct Visual Servoing of Tendon-Driven Continuum Robots

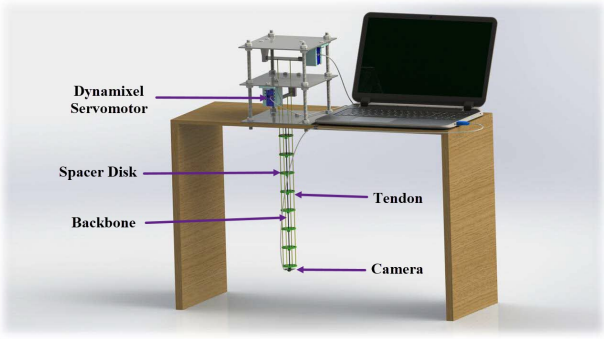
Ibrahim Abdulhafiz[1], Ali A. Nazari[2,*], Taha Abbasi-Hashemi[1], Amir Jalali[2], Kourosh Zareinia[2], Sajad Saeedi[2], and Farrokh Janabi-Sharifi[2]

***Abstract*— Vision-based control provides a significant potential for the end-point positioning of continuum robots under physical sensing limitations. Traditional visual servoing requires feature extraction and tracking followed by full or partial pose estimation, limiting the controller's efficiency. We hypothesize that employing deep learning models and implementing direct visual servoing can effectively resolve the issue by eliminating such intermediate steps, enabling control of a continuum robot without requiring an exact system model. This paper presents the control of a single-section tendon-driven continuum robot using a modified VGG-16 deep learning network and an eye-in-hand direct visual servoing approach. The proposed algorithm is first developed in Blender software using only one input image of the target and then implemented on a real robot. The convergence and accuracy of the results in normal, shadowed, and occluded scenes demonstrate the effectiveness and robustness of the proposed controller.**

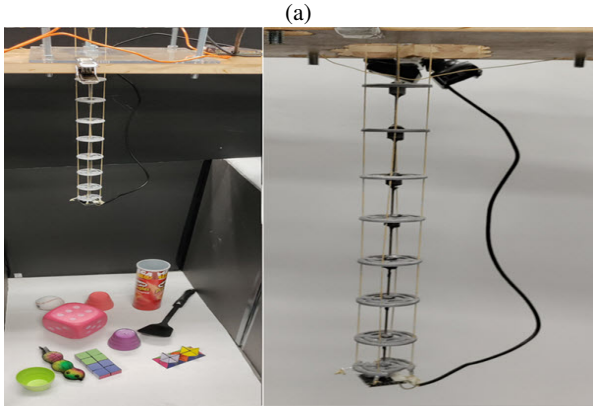
I. Introduction

Continuum robots (CRs) have become popular in recent years due to their continuum structure and compliance, enabling them to manipulate geometrically complex objects and work in unstructured and confined environments [1]. In particular, tendon-driven CRs have typically small diameter-to-length ratios [2], presenting great potential for their navigation in confined spaces such as body cavities [3].

Nevertheless, affected by the intrinsic compliance and the high number of degrees of freedom (DOFs), the control of CRs has been a challenge since their emergence. Both model-based [4] and model-free [5], [6] control approaches have been proposed. The issues related to modeling and sensing have contributed to the control challenge of CRs. Kinematic and dynamic modeling of CRs are ongoing research problems and often involve the iterative solutions of partial differential equations (PDEs) [7], [8], which are often contaminated with significant parameter uncertainties. Additionally, sensing presents its own set of challenges. For example, limitations due to the size, bio-compatibility, and sterilizability of common sensors limit their integration into CRs for many medical interventions [9]. Non-contact sensing methods such as vision-based techniques have thus found an important place in many interventions with CRs. In particular, imaging modalities are readily available in many medical interventions, circumventing the need for extra sensor integration [10]. Therefore, vision-based control methods provide attractive solutions by enabling the use of feasible sensing, and also direct end-point control of CRs with potential robustness to modeling uncertainties related to the robot and target object [10]–[13].



(a)



(b)

Fig. 1: (a) CAD model and (b) prototype of tendon-driven continuum robot. A sample of robot motion can be seen in the supplemented video.

Early methods of vision-based control, also called visual servoing (VS), relied on the image projection of geometric features such as points, lines, corners, edges, ridges, and blobs. Both eye-in-hand (EIH) and eye-to-hand (ETH) configurations were utilized [12]. Depending on the nature of error used in the control law, two basic approaches in VS have been realized, which include image-based visual servoing (IBVS) and position-based visual servoing (PBVS) [13]. Examples of VS approaches to CR control include the work of Wang *et al.* [14], [15] who selected the IBVS-EIH approach for kinematic control of a cable-

*Corresponding author; Email: ali.nazari@ryerson.ca
[1]Department of Electrical, Computer, and Biomedical Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada.
[2]Department of Mechanical and Industrial Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada.

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Discovery Grants 2017-06930 and 2019-05562 and the Ryerson Faculty of Engineering and Architectural Science Dean's Research Fund (FEAS DRF).

driven soft robot. They developed an adaptive PD tracking controller by knowing the intrinsic and extrinsic camera parameters, but they did not consider visual sensing accuracy in the modeling. Zhang *et al.* [16] modeled the statics of a cable-driven parallel soft robot and implemented an open-loop/closed-loop switching controller in an IBVS-ETH scheme. The open-loop controller was developed based on finite element modeling in the simulation environment, which is computationally inefficient for real-time control. System and imaging constraints were formulated in an IBVS-based visual predictive control with ETH configuration developed for tendon-driven CRs [10]. Model-less optimal feedback control in the IBVS-ETH scheme was used by Yip *et al.* [17] to control a planar tendon-driven CR in the task space. They estimated the image Jacobian using backward differencing.

Despite having several advantages, IBVS and PBVS methods have some disadvantages/limitations [12], [13]. The real-time pose estimation in PBVS schemes is always a challenge. Another significant challenge for feature-based methods is the extraction of image features, which may require camera pose measurement, robust feature extraction, feature matching, and real-time tracking, all of which are complex and computationally expensive [18]. The success of the feature-based visual servoing, in fact, depends on the tracking success and performance, *i.e.*, the speed, accuracy, robustness, and redundancy of the visual features [19].

Using non-geometric VS or direct visual servoing (DVS) is an alternative to eliminate the feature tracking requirement. For instance, Photometric VS [20] is a solution to the problem in a 2D scenario. It exploits the full image as a whole, uses the luminance of all pixels in the image, and avoids extracting geometric features of the image. Due to the redundancy of visual information in the control loop, DVS schemes are more accurate and robust than geometric feature-based VS methods [21]. Although these methods do not require feature extraction, their convergence is inferior to that of the classical VS methods [22].

Deep learning methods have been recently proposed to tackle the issues mentioned above. Examples include the work of Bateux *et al.* [22], [23], which is based on training a convolutional neural network (CNN) using images captured from different scenes of a target object along with their corresponding poses. The estimated poses were used in a resultant PBVS scheme to achieve real-time control of a rigid-link manipulator. The proposed method showed satisfactory results in both tested and unforeseen scenes [23]. Also, Felton *et al.* [24] proposed a deep network for end-to-end DVS in which the velocity of a camera mounted on a robot tip is predicted using a Siamese network. They trained the algorithm on a subset of the ImageNet dataset and tested its performance on a 6-DOF rigid-link robot. In spite of these studies, no study has been reported on investigating the DVS of CRs. There are significant challenges associated with CRs, which make their end-to-end DVS different and challenging. Examples include significant differences between kinematic and dynamic models from their rigid-link counterparts and the existence of considerable uncertainties associated with their models.

The objective of this paper is to develop the first deep learning-based end-to-end control of CRs utilizing DVS methods and its implementation in actuation space. Our contribution is as follows:

- Developing a deep learning-based direct VS algorithm. The deep network is structured by modifying the VGG-16 network. The model is then trained using a self-provided dataset (generated by Blender software), which includes variations of only one target image with normal conditions, illumination changes, and occlusions.
- Conducting extensive simulation studies in Blender in normal and perturbed conditions and then evaluating the controller's performance on a real robot. The algorithm is experimentally validated in a variety of scenarios including the normal operation of the robot within the full range of its workspace. The robustness is also analyzed against variations in the lighting in the environment and partial occlusion. Finally, our approach is compared with a classical IBVS approach.

II. Methodology

There exist many challenges in implementing VS on CRs. Unlike rigid robots with stable designs and well-defined kinematic models, the flexibility and soft nature of CRs make them susceptible to various modeling inaccuracies and extremely sensitive to noise and disturbance. Examples of modeling uncertainties include extreme hysteresis, backlash, dead zone, and high sensitivity to disturbance. Therefore, regressing the desired camera velocity is not sufficient for accurate control of CRs. To address these uncertainties, we propose a joint space VS scheme to localize the end-effector at a target image frame. This is accomplished by implementing an end-to-end deep learning model that directly computes the desired tendon velocities from camera images. In order to train the model robustly, a simulation environment is created to generate an appropriate training dataset.

Our methodology is based on employing a deep learning network that has been already trained but repurposing it by changing the last layer and tailoring it for the desired task. Using the image frames captured in real time by a camera, the network produces the raw Δq^* commands that can direct the robot to the desired target after a subsequent scaling by a proportional controller. The intended network is trained using a user-generated dataset of RGB images produced utilizing Blender software. The performance of the proposed method is evaluated through extensive simulation and experimental studies in normal and changing conditions to prove that the algorithm is robust against lighting changes and partially occluded environments.

A. Prototype Design and Development

As shown in Fig. 1a, the prototype CR has one section comprised of a flexible backbone made of spring steel, four braided Kevlar lines (Emmakites, Hong Kong) with a diameter of 0.45 mm as tendons, and spacer disks to route the tendons. The tendons were placed around the backbone

TABLE I: Prototype specifications.

Prototype's Part	Specification	Value
Backbone	Density (ρ)	7800 Kg/m^3
	Young's modulus (E)	207 GPa
	Length (L)	0.4 m
	Radius (r)	0.9 mm
Tendon	Breaking strength	31.75 Kg
USB webcam	Frame rate	30 fps
	Resolution	1080×7200 pixels
	Field of view (FOV)	19°

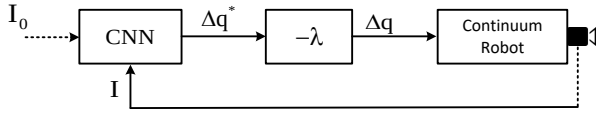


Fig. 2: Block diagram of the proposed visual servo controller comprised of a camera in an EIH mode, a tendon-driven CR, and a CNN model.

with an offset of 1.8 mm, and an angular distance of 90° from each other. The tendons were routed toward the robot tip by eight equally distanced spacer disks, which were 3D printed using PLA filament. The disks were solidly attached to the backbone using steel-reinforced epoxy adhesive with a strength of 3960 psi. A custom fixture was 3D printed to rigidly mount a 1080P HD webcam (OURLINK, CA, USA) on the robot tip in an EIH mode. The fixture was screwed on the last spacer disk such that it guarantees the minimum space between the camera and the robot tip while having no contact between the fixture and the camera's electronic board. The tendons were actuated using Dynamixel AX-12A servomotors (Robotis, CA, USA). Table I provides the prototype specifications.

B. Control Law

Classical VS approaches require complex Jacobian mapping, which is difficult to derive. Therefore, we aim to replace the entire mapping from image space to joint space with a learned model, such that the error between the current image frame, I, and the desired image frame, I_0, be minimized. As shown in Fig. 2, the output is multiplied by a gain, $-\lambda$, and fed into the CR. The control law is then stated as

$$\Delta q^* = f(I_0, I) \tag{1}$$

$$\Delta q = -\lambda\ \Delta q^* \tag{2}$$

where Δq and Δq^* are respectively the change and the desired change in tendon displacement in mm, I_0 is the target image, I is the current image, and $f()$ is a function of the target and the current images implemented on a modified VGG-16 network to output Δq^*.

C. Neural Network Design

In order to create an efficient neural network for our purpose, we utilized a VGG-16 backbone pre-trained on ImageNet to facilitate transfer learning [25]. Having been trained

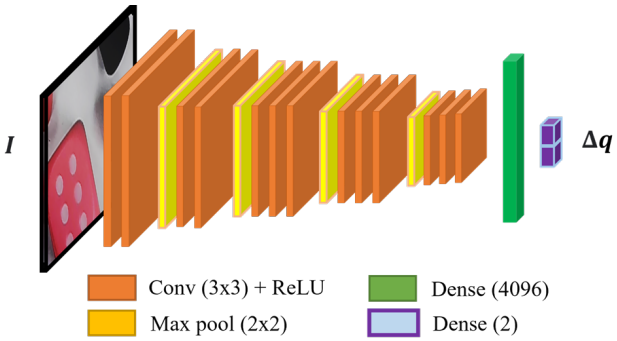


Fig. 3: Architecture of modified VGG-16 model.

on natural images, only the lower layers of the network will need to be trained to regress desired tendon displacements. In our model, the first 10 layers were frozen to speed training because they already contained low-level features from natural images. We modified this network by dropping out the last dense layer and replacing it with a dense layer with two outputs corresponding to q_1 and q_2. The activation function was set to linear (see Fig. 3). Various alternatives were considered to challenge this proposed model. Firstly, different backbones were considered, particularly ResNet50 and ResNet101. Secondly, we considered adding multiple dense layers to improve the nonlinear fitting of the CR model.

D. Simulation Environment

Training in the simulation provides various advantages over the real world. Not only are they much quicker in acquiring the data, but they also offer the opportunity to structure the environment to account for various noises and uncertainties, enabling the model to be more robust. On the contrary, attempting to learn the dynamics and uncertainties of the robot remains challenging in simulated environments. We resolved this problem by utilizing an open-source 3D computer graphics software called Blender and creating an environment that models the pose of the end-effector given a tendon displacement, q, value. This was achieved by using the forward kinematics of the robot to place and orient the virtual camera in the simulated environment. Being a single-section 2-DOF CR, we modeled the kinematics based on the constant curvature assumption, as presented by Rao *et al.* [26].

Whereas this approach ignores the dynamic effects of the CR, we propose that implementing robust vision control would allow the feedback loop to correct for most of the aforementioned challenges of the CR. Shadowing and occlusion were included to provide this robustness. Shadowing was achieved by adjusting the light source in the environment, whereas occlusion was achieved by placing black rectangles of random positions and dimensions within the image. Fig. 4 shows some samples from the simulation.

For acquiring the dataset, previous approaches made use of two sets of images; one randomly placed within a distance from the origin for general convergence, and the other one

(a) (b) (c) (d)

Fig. 4: Typical set of images used in simulation. (a) target image, (b) camera view at the tendon displacement of $(q_1, q_2) = (4, -3)\, mm$, (c) camera view with random lighting, and (d) camera view with occlusion.

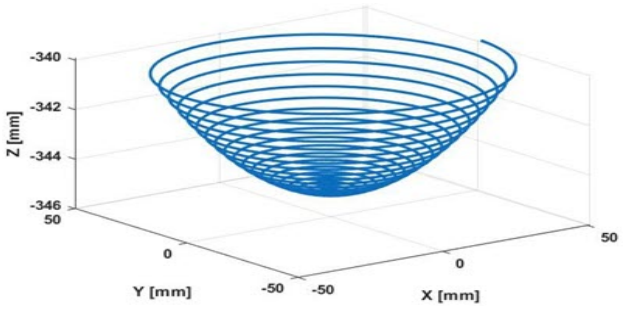


Fig. 5: Camera path used to generate the dataset. Assuming the camera is exactly mounted on the robot's tip, (x, y, z) are the coordinates of the camera's center point with respect to the robot's base.

very close to the origin for fine-tuning [23]. We thought this binary approach would produce noisier joint commands, and therefore we implemented a continuous method. The farther away the CR is from the origin, the sparser the dataset will be. Similarly, to produce more deterministic results, a spiral path was used to traverse all reaching points in the 3D world within a certain threshold. The intended path can stimulate nonlinearities of the robot very well while covering all quadrants of the robot's workspace even if there is no overlap, which is of our interest in experiments. The spiral path was generated using

$$q_1 = \frac{A}{n} x \, \cos\left(2\pi \frac{P}{n} x\right) \tag{3}$$

$$q_2 = \frac{A}{n} x \, \sin\left(2\pi \frac{P}{n} x\right) \tag{4}$$

where A is the maximum displacement of the tendon, P is the total number of periods the CR makes, n is the number of sample points, and x is an integer from 1 to n. Fig. 5 shows the generated spiral path. Using Blender, we created the environment and overlaid the desired scene. Employing a Python API, we moved the camera and light source around and captured images from the scene for training purposes.

The benefits of using a simulation environment instead of generating the dataset from the physical robot are twofold. Firstly, only one target image is required, and thus neither an exact camera is needed nor is having access to the physical robot required. So long as simple forward kinematics is known, Blender can be used to generate the entire dataset. This may be true, especially in medical procedures whereby only limited pre-operative image data is present in advance. Secondly, the use of simulation enables the dataset to be made robust to various lighting, noises, artifacts, and other disturbances.

E. Training and Validation

For training, we selected mean squared error (MSE) as the loss function because we designed our output activation function as linear. This was because we wanted to learn a one-to-one mapping of the ground truth control points. The ground truth of the dataset was generated based on equation (5), which keeps the ground truth between -1 and 1, allowing for better training. Similarly, this mapping produces smoother convergence profiles (as opposed to linear mapping).

$$q_{mapped} = \tanh\left(10\, q\right) \tag{5}$$

In comparison, we linearly mapped $-5\, mm$ and $5\, mm$ to -1 and 1, respectively, clipping any values beyond. However, such an approach would not penalize the optimizer as much near the origin. Since we aim to propose sub-millimeter accuracy, utilizing equation (5) would force convergence while producing a much smoother velocity profile.

The simulation environment was used to generate the dataset. To this end, 5000 images were acquired with a maximum amplitude of $7\, mm$ and a period of 20. Random lighting effect and random occlusion were included. These occlusions were represented as black rectangles overlaid at random positions to force the model to learn the full spatial features and make it more robust. As we used the classical VGG-16, the input image was RGB of size 224×224. The model was trained for 50 epochs with a batch size of 32 and a learning rate of $1e-5$ using Adam optimizer. The final MSE was determined to be $3e{-}5$. To validate our hypothesis, VGG-16 was swapped with ResNet51 and ResNet101. As expected, the training took substantially longer, and the MSE was inferior to that of the VGG. Similarly, two dense layers (1024 and 512, respectively) were added between the VGG and the final q output to test our hypothesis. Nonetheless, the training took longer without any significant improvement to the MSE. Training on an Nvidia Titan Xp GPU was reasonably fast, taking less than 20 minutes on 50 epochs to train. The model inference was also extremely fast, taking about $15\, ms$ per frame[1].

III. Experimental Results

The effectiveness and efficiency of the developed controller were tested in a variety of simulations and experiments. Here, we describe the simulations conducted using Blender software in order to test the robustness and accuracy of the controller. Following this, experimental studies are discussed in more detail, including the experiment design to cover all four quadrants of the robot workspace, as well as a discussion of the accuracy of the results. Finally, the deep

[1]The code and dataset will be made available publicly once the paper is accepted.

learning-based controller is compared to a classical IBVS controller to verify that the obtained results from CNN-based VS are significant compared to the classical one.

A. Simulation

Before conducting experiments in real-world scenarios, we performed some tests to validate the robustness and accuracy of the simulation. Since the kinematics did not incorporate nonlinear effects when generating the dataset, we needed to include various uncertainties to prove the model's robustness within the simulation.

1) Modeling Uncertainties: Since the constant curvature assumption does not hold true in all situations, other uncertainties and disturbances were added to the simulation. Regarding geometric uncertainties, derived from parameters of the robot including length, disk space, etc., Gaussian noise with a mean of 0 and standard deviation of $0.01\,mm$ was added to the output q values. Also, the outputs of the trained model were scaled to uniformly distributed random numbers in the range of 0.25 to 4. Random lighting was introduced to account for the vision uncertainties, and a region within the image was occluded with black rectangles. Instead of generating these random scene environments every iteration, we chose to regenerate these random uncertainties every 20 iterations to better model the changing conditions of the real-world environment.

2) Simulation Results: Simulating with the initial tendon displacements of $(q_1, q_2) = (6, -4)\,mm$ we noticed the CR is able to converge smoothly although less than 25% of the target image was visible at the starting position. Moreover, the change in lighting, as shown in Fig. 6, did not affect the convergence of the CR. More interestingly, adding occlusion (at times greater than 80%) did not destabilize the CR and, as noted with the raw network output in Fig. 7, was still able to counteract the Gaussian noise added to the actuation commands of the CR. A sample of simulation studies can be seen in the supplemented video. Having been successfully validated in simulation, the next section will extend it to the real-world environment.

B. Experimental Validation

In order to test the practicality of the proposed end-to-end model, we applied it to the experimental setup developed for the purpose to show its robustness to various noises and uncertainties. Fig. 1b shows the prototyped robot for the experiment.

1) Experiment Design: The physical environment was structured in a similar way to the simulation environment, as shown in Fig. 1b. To test the model's accuracy in converging the CR, the end-effector was moved to random positions, and the trained model attempted to minimize the difference between the current image frame I and the target image I_0 on which the model has been trained. The range of motion was limited to $\pm 10\,mm$ for each tendon to keep the scene within the camera's field of view (FOV). Note that the scene with which the model was trained was larger than the camera's FOV, which enabled our model to operate despite there being

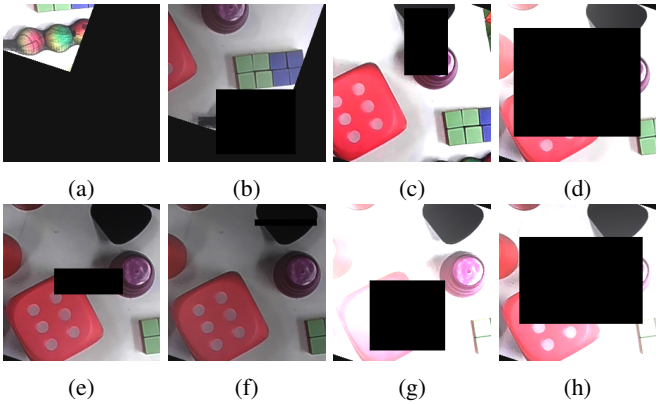


Fig. 6: Sequence of camera views in a typical simulation started at the tendon displacement of $(q_1, q_2) = (6, -4)\,mm$ at iteration numbers of (a) 1, (b) 25, (c) 50, (d) 75, (e) 100, (f) 225, (g) 250, and (h) 299.

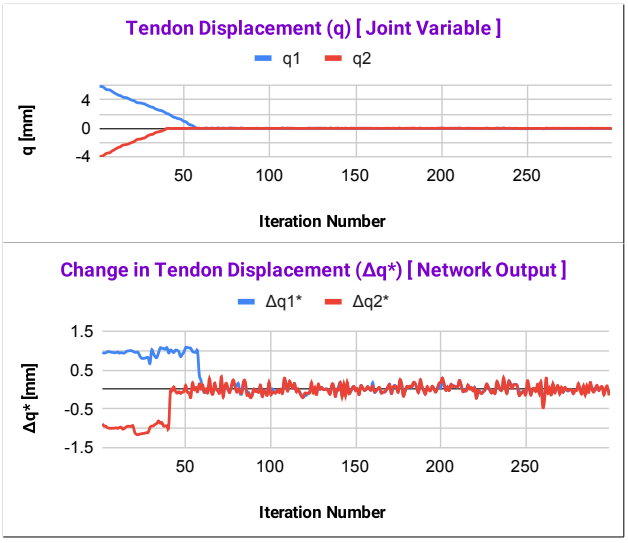


Fig. 7: Tendon displacement (top) and raw Δq^* commands from the network, before being multiplied by $-\lambda$, to stabilize the robot (bottom).

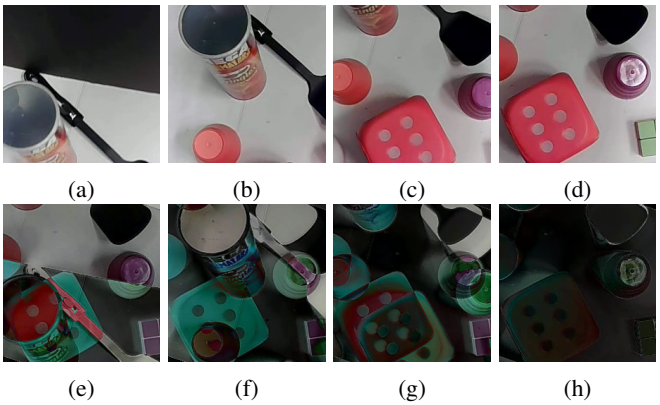


Fig. 8: (a-d) Sequence of camera views in a typical experiment started at the tendon displacement of $(q_1, q_2) = (5, -7)\,mm$, (e-h) Corresponding difference between the normalized target and intended images.

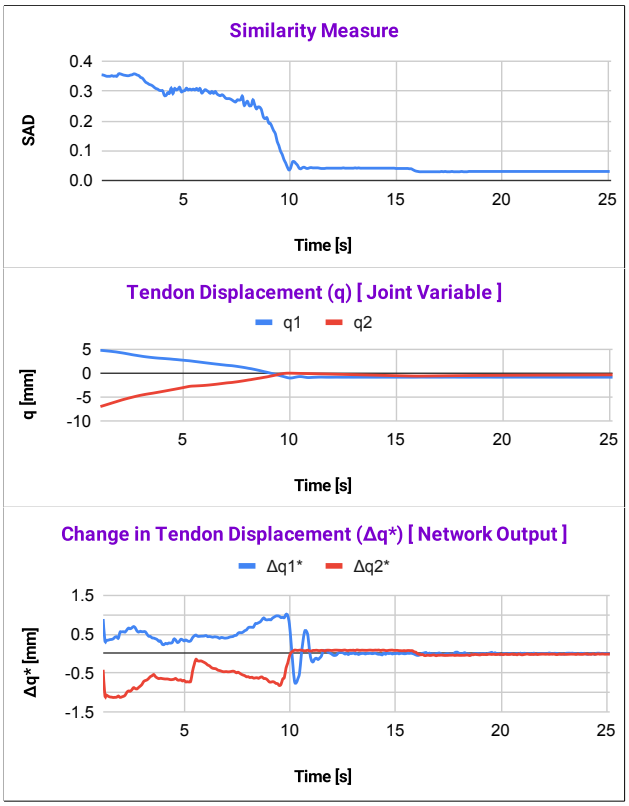


Fig. 9: SAD between I^* and I_0^* (top), tendon displacement at each iteration (middle), and the raw Δq^* commands (bottom).

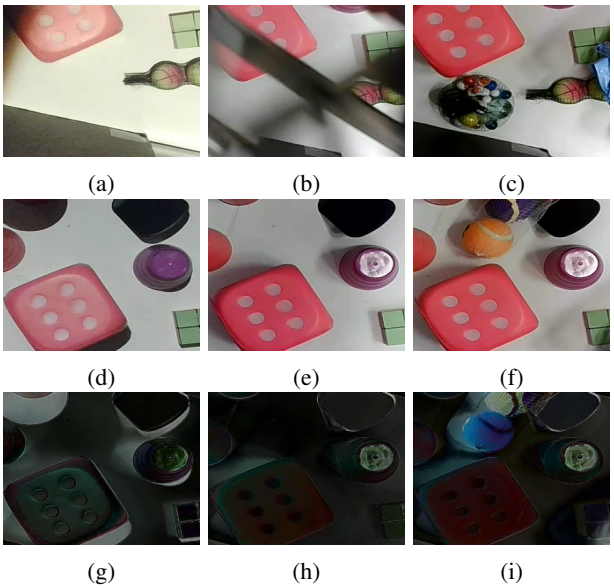


Fig. 10: (a-c) Initial views of the camera in the robustness analyses using dynamic lighting, dynamic occlusion, and partial static occlusion, respectively. (d-f) Corresponding converged views. (g-i) Corresponding difference between the target image and converged views.

no overlap with the target image. Further, to demonstrate the robustness of the controller, the robot was operated under dynamic lighting conditions, dynamic occlusion, and finally partial static occlusion.

2) Results and Discussion: Experimenting from the initial tendon displacements of $(q_1, q_2) = (5, -7)\,mm$, the first row in Fig. 8 shows that the CR converges to match the camera image to the target image. Normalizing and then subtracting the current and target images gives the images on the second row. Upon convergence, the overlap between the two images becomes highly precise. To evaluate the convergence quantitatively, the pixel-wise sum of absolute distance (SAD) between the normalized target and current images was calculated using

$$SAD = \sum |I^* - I_0^*| \tag{6}$$

where I^* is the normalized current image and I_0^* is the normalized target image. As shown in Fig. 9, the SAD value fails to approach zero, stating that the lighting environment and image exposure were slightly different.

Note that q_1 and q_2 values do not approach 0 in Fig. 9 despite that being the origin with which the target image was taken. This can be assigned to the dynamic effects of the CR and, more specifically, its hysteresis, which will be addressed in our future research. Without a computationally complex controller, our model can converge the camera frame to the target image by automatically compensating for the nonlinear effects. The raw Δq commands show that the model smoothly converges to the origin and stabilizes once it approaches.

In order to test the robustness, different lighting conditions and occlusions were considered. As seen in Fig. 10, the top row images show the CR at the starting position, $(q_1, q_2) = (-2, 2)\,mm$. The left, center, and right columns show the dynamic lighting, dynamic occlusion, and static partial occlusion scenarios, respectively. The bottom row shows the corresponding normalized differences with the target image at the final iteration for each scenario. Despite the uncertainties, the precise overlap confirms that the model has learned sufficiently through simulation alone and that it can robustly control the CR in an end-to-end fashion. Note that even though the simulation only used one 2D image of the target scene, which results in projection artifacts when simulating in 3D, the model successfully operated in the real 3D environment without additional modifications. Four examples of experimental validation and robustness analysis can be seen in the supplemented video.

C. Comparison with Classical VS

To determine the superiority of our approach, we conducted similar experiments using classical VS approaches. One initial downside of classical VS, as opposed to our approach, is the requirement of all features to be visible in the camera frame at all times. Hence, we were restricted to tendon displacements of $(q_1, q_2) = (\pm 2, \pm 2)\,mm$ to make sure the features always remain in the camera's FOV. For testing purposes, template matching of the pink dice circles was used in the classical approach.

TABLE II: Comparison between classical and CNN-based VS.

Quadrant	Metrics	Classical	CNN-based
#1 (+, +)	Initial q	$(2, 0.5)$	$(4, 4)$
	Final SAD	**0.046**	0.057
	Convergence (# of iterations)	**101**	191
#2 (+, -)	Initial q	$(1, -1.5)$	$(3, -5)$
	Final SAD	0.065	**0.058**
	Convergence (# of iterations)	88	**85**
#3 (-, -)	Initial q	$(-2, -2)$	$(-2, -3)$
	Final SAD	**0.036**	0.054
	Convergence (# of iterations)	105	**60**
#4 (-, +)	Initial q	$(-1, 2)$	$(-3, 6)$
	Final SAD	**0.027**	0.060
	Convergence (# of iterations)	228	**127**

1) Classical Control Law: Classical IBVS aims to minimize the pixel error between the current and the target features. In our case, four feature points were selected using a template matching algorithm. Thereafter, binary thresholding was used to extract regions of high feature similarity. Finally, the centroids of these features were extracted, resulting in a (u, v) coordinate for each of the features. Given four feature points, the classical image Jacobian for each feature was

$$L_x = \begin{bmatrix} \frac{f}{z} & 0 & -\frac{u}{z} & -\frac{uv}{f} & \frac{f^2+u^2}{f} & -v \\ 0 & \frac{f}{z} & -\frac{v}{z} & \frac{-f^2+v^2}{f} & \frac{uv}{f} & u \end{bmatrix} \tag{7}$$

where f and z are the camera's focal length and the image depth, respectively. The depth was considered constant, equal to 1 m. After calculating four Jacobian matrices corresponding to four features, the image Jacobian was found as

$$J_{img} = \begin{bmatrix} L_{x1} \\ L_{x2} \\ L_{x3} \\ L_{x4} \end{bmatrix} \tag{8}$$

In order to enhance the computational efficiency, the Jacobian matrix of the robot was approximated using a finite difference method [27]. To this end, the change in the joint space variable, Δq, was set to be 0.1 mm, which is small enough for a submillimeter accuracy. Then, the final interaction matrix, L_e, was computed as

$$L_e = J_{img} H J_{robot} \tag{9}$$

where

$$H = \begin{bmatrix} R_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & R_{3\times3} \end{bmatrix} \tag{10}$$

and R is the rotation matrix from the base frame to the camera frame. The final classical IBVS control law was

$$\dot{q} = -\lambda L_e^{+}(s - s^*), \tag{11}$$

where $\dot{q}$ is the velocity of the tendon, λ is a gain factor, L_e^{+} is the pseudo-inverse of the interaction matrix, s is the current feature and s^* is the target feature.

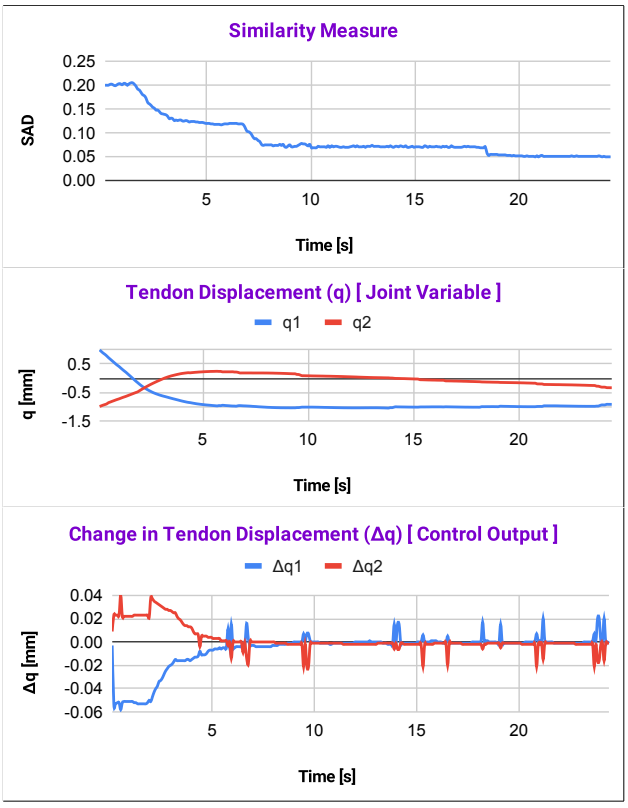


Fig. 11: Classical VS: SAD between I^* and I_0^* (top), tendon displacement at each iteration (middle), and the Δq commands (bottom).

2) Results and Discussion: As seen in Table II, our approach is more advantageous than the classical VS approach given its larger operating workspace. Despite having greater initial q values, our approach is capable of producing similar and, at times, more superior results. We chose to start at different initial configurations within each quadrant to show the robustness of convergence for both approaches.

A primary advantage of our approach is that it takes into account the entire image rather than particular features, making it more robust to noise, partial occlusion, and artifacts. Our model can converge in situations where the initial and target images do not overlap, thereby expanding the servoing workspace. Secondly, our end-to-end approach does not require a Jacobian of the robot and hence produces smooth trajectories throughout the servoing. This is contrary to the classical VS which requires the robot Jacobian to be known. However, CRs are known to have extensive dynamic effects, making accurate computation of the Jacobian challenging. In our case, the robot Jacobian was derived using simple kinematics and therefore did not take into account various uncertainties present in the CR. These typically result in slower convergence time and a noisier displacement profile when compared to our deep learning-based approach, as evident from Fig. 11. Consequently, the gain needs to be set much smaller to produce a smoother trajectory profile, thereby resulting in the slower convergence time observed.

IV. Conclusion and Future Work

In this paper, a deep direct visual servoing algorithm was proposed to control a single-section tendon-driven continuum robot in an eye-in-hand configuration. The advantage of our training approach is using single image and populating the images utilizing Blender software for generating a dataset for VGG-16 network. The dataset includes different views of the scene plus illumination change and occlusion for robustness analysis. The algorithm was tested on a real robot developed by the team and showed fast and accurate convergence in regular scenes. Also, the algorithm's robustness was verified in scenarios incorporating dynamic illumination changes as well as dynamic and static occlusions.

In our future work, the robot will be extended to multiple sections, and the dynamic effects of the robot motion such as hysteresis, tendon slack, backlash, dead zone, and external disturbances will be investigated. We will also extend the control approach to achieve faster convergence and improved robustness to the aforementioned dynamic effects.

References

[1] D. B. Camarillo, C. F. Milne, C. R. Carlson, M. R. Zinn, and J. K. Salisbury, "Mechanics modeling of tendon-driven continuum manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1262–1273, 2008.

[2] E. Amanov, T.-D. Nguyen, and J. Burgner-Kahrs, "Tendon-driven continuum robots with extensible sections—a model-based evaluation of path-following motions," *The International Journal of Robotics Research*, vol. 40, no. 1, pp. 7–23, 2021.

[3] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[4] M. Chikhaoui and J. Burgner-Kahrs, "Control of continuum robots for medical applications: State of the art," in *ACTUATOR 2018; 16th International Conference on New Actuators*. VDE, 2018, pp. 1–11.

[5] 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.

[6] T. da Veiga, J. H. Chandler, P. Lloyd, G. Pittiglio, N. J. Wilkinson, A. K. Hoshiar, R. A. Harris, and P. Valdastri, "Challenges of continuum robots in clinical context: a review." *Progress in Biomedical Engineering*, vol. 2, no. 3, p. 032003, 2020.

[7] J. Till, V. Aloi, and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat rod models," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 723–746, 2019.

[8] F. Janabi-Sharifi, A. Jalali, and I. D. Walker, "Cosserat rod-based dynamic modeling of tendon-driven continuum robots: A tutorial," *IEEE Access*, vol. 9, pp. 68 703–68 719, 2021.

[9] A. A. Nazari, F. Janabi-Sharifi, and K. Zareinia, "Image-based force estimation in medical applications: A review," *IEEE Sensors Journal*, vol. 21, no. 7, pp. 8805–8830, 2021.

[10] M. M. Fallah, S. Norouzi-Ghazbi, A. Mehrkish, and F. Janabi-Sharifi, "Depth-based visual predictive control of tendon-driven continuum robots," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2020, pp. 488–494.

[11] A. A. Nazari, K. Zareinia, and F. Janabi-Sharifi, "Visual servoing of continuum robots: Methods, challenges, and prospects," *The International Journal of Medical Robotics and Computer Assisted Surgery*, p. e2384, 2022.

[12] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[13] F. Janabi-Sharifi, L. Deng, and W. J. Wilson, "Comparison of basic visual servoing methods," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 5, pp. 967–983, 2010.

[14] H. Wang, W. Chen, X. Yu, T. Deng, X. Wang, and R. Pfeifer, "Visual servo control of cable-driven soft robotic manipulator," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 57–62.

[15] 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.

[16] 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.

[17] 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.

[18] É. Marchand and F. Chaumette, "Feature tracking for visual servoing purposes," *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 53–70, 2005.

[19] M. Ourak, B. Tamadazte, O. Lehmann, and N. Andreff, "Direct visual servoing using wavelet coefficients," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 1129–1140, 2019.

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

[21] L.-A. Duflot, R. Reisenhofer, B. Tamadazte, N. Andreff, and A. Krupa, "Wavelet and shearlet-based image representations for visual servoing," *The International Journal of Robotics Research*, vol. 38, no. 4, pp. 422–450, 2019.

[22] Q. Bateux, E. Marchand, J. Leitner, F. Chaumette, and P. Corke, "Visual servoing from deep neural networks," *arXiv preprint arXiv:1705.08940*, 2017.

[23] Bateux, Quentin and Marchand, Eric and Leitner, Jürgen and Chaumette, François and Corke, Peter, "Training deep neural networks for visual servoing," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3307–3314.

[24] S. Felton, E. Fromont, and E. Marchand, "Siame-se(3): regression in se(3) for end-to-end visual servoing," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 454–14 460.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2015.

[26] P. Rao, Q. Peyron, S. Lilge, and J. Burgner-Kahrs, "How to model tendon-driven continuum robots and benchmark modelling performance," *Frontiers in Robotics and AI*, vol. 7, pp. 1–20, 2021.

[27] K. Leibrandt, C. Bergeles, and G.-Z. Yang, "On-line collision-free inverse kinematics with frictional active constraints for effective control of unstable concentric tube robots," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3797–3804.