Snake Robot for Searching Human Casualties

1st Gayanuka Amarasuriya
dept. of Electrical Engineering
University of Moratuwa
Colombo, Sri Lanka
amarasuriyagn.21@uom.lk

2nd Sanuda Amasith
dept. of Electrical Engineering
University of Moratuwa
Colombo, Sri Lanka
amasithats.21@uom.lk

3rd Kavishwara Ariyawansha
dept. of Electrical Engineering
University of Moratuwa
Colombo, Sri Lanka
ariyawanshamkrk.21@uom.lk

***Abstract*—This project presents a snake robot designed using SolidWorks for disaster response, featuring MATLAB-Simulink for motion control and image classification for casualty detection. The robot's motion control system employs a snake-like locomotion strategy, navigating through challenging terrain. Concurrently, casualty classification utilizes computer vision, enabling real-time detection of human presence. Thorough testing validates the integrated system's performance, offering a versatile solution for search and rescue operations in disaster environments. The project's documentation serves as a guide for future enhancements and collaborative efforts in the fields of robotics and autonomous disaster response systems. By combining advanced motion control with state-of-the-art image classification, this snake robot contributes to the evolution of efficient and adaptable solutions for autonomous search and rescue missions, addressing the complexities of disaster-stricken areas.**

I. INTRODUCTION

A. *Background*

Robotics is a versatile field that revolves around the creation and operation of robots, which are machines capable of autonomous or human-assisted tasks. These robots have applications spanning multiple domains, from manufacturing and healthcare to transforming the way people work and interact with their surroundings. Notably, robots can be indispensable in providing critical humanitarian assistance during emergencies and search and rescue missions.

Disasters, be they natural calamities or human-induced crises, present formidable obstacles to effective search and rescue operations. The urgency of swiftly and accurately identifying human casualties in intricate and hazardous environments cannot be overstated, as it plays a pivotal role in minimizing the overall impact of these events. The amalgamation of advanced robotics and artificial intelligence, showcased in this project, represents a groundbreaking approach to disaster response.

One remarkable project in the realm of robotics is the development of a "Snake Robot for Search and Rescue Missions." This innovative robotic system has been designed to address the challenges of operating in complex and hazardous environments. Taking inspiration from the mobility of serpents, this snake-like robot possesses unique attributes that make it particularly adept at gaining access to hard-to-reach areas, such as collapsed structures, confined spaces, and disaster-stricken locations.

Fig. 1. The multi-jointed snake bot provided rescue workers with a video feed. Adapted from [17]

B. *System Architecture*

The complex mechanism that drives the snake robot's mobility system aims to mimic the natural flexibility and agility of snakes' sinuous movements. At its foundational level, this system encompasses a combination of mechanical design and control algorithms that work in sync to enable the robot's locomotion. The project's reliance on MATLAB Simulink serves as a strategic choice, providing a powerful and versatile environment for modelling, simulating, and refining the intricate dynamics of the snake robot's movements.

Within MATLAB Simulink, the motion control strategies undergo meticulous development, allowing for a comprehensive exploration of various locomotion techniques. This iterative process of refinement is crucial for optimizing the robot's adaptability across a spectrum of terrains and obstacles commonly encountered in disaster-stricken areas. The system's ability to mimic the nuanced movements of a snake, such as serpentine ripple and side-winding, is finely tuned within Simulink, ensuring the robot's capability to traverse confined spaces with unparalleled efficiency.

Moreover, the motion system's role extends beyond mere replication, evolving into a dynamic and responsive component that adapts to real-world challenges. The iterative nature of development within Simulink allows for continuous improvement, ensuring that the robot's locomotion strategies align with the demands of unpredictable disaster scenarios. This adaptability, rooted in the intricacies of the motion system, stands as a testament to the project's commitment to crafting a robot capable of sophisticated and effective responses in the face of adversity. As the project unfolds, the comprehensive understanding and refinement of the motion system within MATLAB Simulink play a pivotal role in shaping the snake robot into a versatile and indispensable tool for autonomous disaster response.

The image classification process within this project is geared towards the swift and precise detection of human casualties using captured images. As the snake robot traverses disaster-stricken environments, its onboard cameras capture images of the surroundings. These images are then subjected to an advanced image classification system that operates seamlessly within the Computer vision concept.

The essence of this process lies in the system's ability to analyze visual data and discern the presence of human figures in real time. By focusing on the inherent features associated with human presence, the image classification system enables the snake robot to autonomously identify and respond to potential casualties. This integrated approach, where motion control and image classification work in tandem, equips the robot with heightened situational awareness. The captured images and video input serve as a critical source of information, allowing the robot to make informed decisions and navigate dynamically through varied terrains, ultimately enhancing its efficiency in search and rescue operations within disaster-stricken areas.

C. *Application*

Snake robots have shown versatility in disaster response and exploration scenarios. They were deployed to navigate hazardous environments after the Fukushima Daiichi nuclear disaster in 2011 and were used in search and rescue operations after the Mexico City earthquake in 2017. They were also used in the 2016 earthquakes in central Italy to navigate damaged structures and assess potential victims. These examples demonstrate the versatility of snake robots in navigating complex terrains and structures, offering potential solutions for disaster-stricken areas and demonstrating continuous advancements in robotic disaster response.

At the heart of this innovative initiative lies the development of a snake robot—a sophisticated, adaptable, and agile system tailored to navigate through complex terrains. Unlike conventional search and rescue robots, the snake robot draws inspiration from the serpentine motion, allowing it to traverse confined spaces, negotiate debris, and access locations that are typically challenging to reach. This robot is not merely a technological marvel but a critical component in the evolution of autonomous systems designed to mitigate the human cost of disasters.

In essence, the fusion of snake-like motion control using MATLAB Simulink and advanced image classification through a pioneering leap in autonomous disaster response systems. By designing a robot capable of navigating intricate terrains with unparalleled adaptability and swiftly identifying human casualties in real-time, this project aspires to redefine the paradigm of search and rescue operations. As the snake robot emerges as a sophisticated tool at the intersection of robotics and artificial intelligence, it not only represents a technological milestone but also holds the promise of significantly reducing the human cost of disasters. This documentation serves not only as a testament to our commitment to innovation but also as a road map for future advancements in the realms of robotics and autonomous systems, paving the way for a safer and more efficient response to the challenges presented by disaster-stricken areas.

Fig. 2. Matt Travers (middle, in white vest) of the Robotics Institute collaborates with red-suited Mexican Red Cross workers to prepare a snake robot to penetrate a fallen apartment building in Mexico City. Adopted from[17]

II. Literature Review

A. *Field of Robotics*

The field of robotics encompasses the design, construction, operation, and use of robots to perform tasks in various domains. Robots are developed by integrating sophisticated technologies such as mechanical engineering, control systems, electronics, and software. They have traditionally played a significant role in automating the manufacturing industry. However, the application of robots in service sectors, particularly in the medical and healthcare domains, has been slower than expected[1].

The impact of robotics extends beyond traditional industrial applications. The convergence of technology in robotics has been identified as a key factor in promoting the introduction of robots in new sectors, including the medical and healthcare domains. The analysis of technology convergence in robotics research has provided valuable insights into the relationships among different scientific sectors, highlighting the expansion of convergence among sectors and the emergence of new frontiers for robotics research[1].

The development of robots has the potential to revolutionize various sectors, including construction, logistics, agriculture, and other service sectors. The impact of robotics research

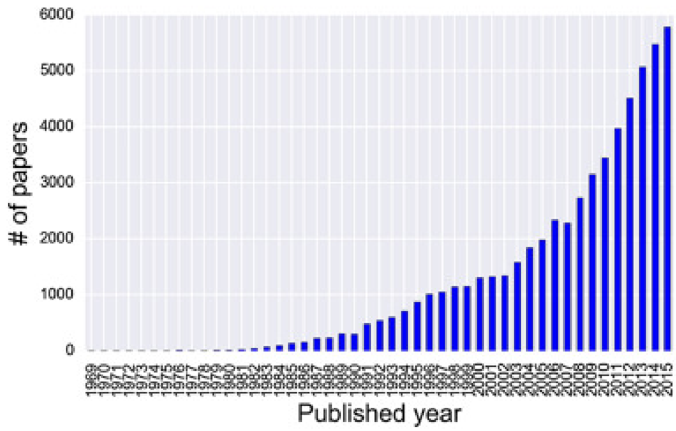

Fig. 3. Number of Papers in Robotics Search.Adapted from [1]

has been analyzed through citation network analysis, which has revealed the rapid spread of robotics and automation in manufacturing since the 1980s[1]. Furthermore, the emergence of robotics in diverse fields such as engineering, computer science, and medical sectors indicates the broad impact of robotics research across scientific domains.

B. *Snake Robots - Related Works*

Snake robots, a fascinating innovation inspired by nature's serpentine locomotion, have emerged as a promising frontier in robotics, offering unparalleled flexibility, agility, and adaptability in navigating challenging terrains. These uniquely designed robots, characterized by their serpentine morphology and multi-link structure, possess an inherent capability to access confined spaces, negotiate obstacles, and explore environments inaccessible to conventional robotic systems.

According to the review, snake robots are favoured and highly regarded by academia and industry due to their small size, flexibility, and ability to carry a variety of equipment for disaster rescue and military surveillance [2]. They are also capable of adapting to various complex terrain environments and can operate at great depths in marine exploration. Compared to traditional underwater robots, snake robots' inherent elongated body shape and flexible movement gait have broken through bottlenecks in marine exploration[2]. In addition, snake robots can enter blood vessels to identify the cause of disease and provide safe and immediate treatment solutions for patients .

Snake robots are unique in their design and capabilities compared to traditional robots. They have a variety of movement modes, including helical rolling, hybrid 3-D gaits, and winding locomotion [T1-T3][2]. These movement modes allow snake robots to adapt to different environments and perform various tasks. However, the challenge is to improve the payload and efficiency of snake robots to better serve human beings.

C. *System Modelling*

Bio-mechanical studies of biological snakes have been used as a basis for modelling snake robot locomotion. These studies involve analyzing the motion of biological snakes and deriving mathematical models that describe the forces acting on the snake's body. Gray's work in the 1940s [3] and Hirose's study of biological snakes in the 1970s [4] are examples of early biomechanical studies that have influenced the development of snake robot models. These studies provide insights into the fundamental principles of snake locomotion and serve as a foundation for developing mathematical models of snake robot motion.

Mathematical models of snake robot locomotion have been developed based on mathematical descriptions of the forces acting on a snake's body. These models typically consider the interaction between the snake's body and the environment, including frictional forces, gravitational forces, and contact forces. For example, the model proposed by Marvi et al. [5] considers the interaction between the snake's body and the ground, including the effects of friction and contact forces. These models provide a more detailed understanding of the dynamics of snake robot locomotion and can be used to develop control strategies for snake robots.

The serpenoid curve is a mathematical description of lateral undulation, which is the most common form of snake locomotion. The serpenoid curve was first formulated by Hirose [4] and has since been used as a basis for developing mathematical models of snake robot motion. The serpenoid curve describes the shape of the snake's body during lateral undulation and provides a simple and elegant way to model snake robot locomotion. For example, the model proposed by Liljebäck et al. [6] uses the serpenoid curve to describe the shape of the snake's body during lateral undulation.

D. *Human Casualties*

Human casualties have been a significant concern in various disaster scenarios, including floods, earthquakes, and maritime accidents.

A historical review of flood events from 1980 to 2009 revealed that floods are the most common natural disaster and the leading cause of natural disaster fatalities worldwide[7]. The primary cause of flood-related mortality is drowning[7]. In developed countries, being in a motor vehicle and male gender are associated with increased mortality, whereas female gender may be linked to higher mortality in low-income countries[7].

In the context of earthquakes, a review of Casualty Estimation Models (CEMs) highlighted the importance of considering factors such as damage level, building classification, earthquake intensity, and occupancy rate[8]. The level of damage, particularly building collapse, was identified as the most influential parameter. However, the construction material and the occupancy of the building at the time of the event also significantly impacted the estimate of human casualties[8].

Maritime accidents related to human error have also been studied extensively. A bibliometric review of the causes of maritime accidents related to human error revealed three main root causes: human resources and management, socio-technical Information Systems and Information Technologies, and individual/cognition-related errors[9].

These studies underscore the complexity of predicting and mitigating human casualties in disaster scenarios. They highlight the need for comprehensive models that consider a wide range of factors and the importance of effective mitigation measures and communication with vulnerable populations[7][8].

E. Human Identification

Computer Vision, often abbreviated as CV, is a field of study that seeks to develop techniques to help computers "see" and understand the content of digital images such as photographs and videos[10]. The goal of computer vision is to compute properties of the three-dimensional world from digital images. This involves tasks like reconstructing the 3D shape of an environment, determining how things are moving, and recognizing people and objects and their activities, all through the analysis of images and videos.

Face detection has been a significant area of research in computer vision for several years. The Viola-Jones algorithm is one of the most popular recognition systems for human face recognition[11]. This algorithm involves the extraction of a specific feature from a detected facial image input, which has to be a full-frontal view image[12]. The specified feature is extracted through what is known as a window, which is automatically scaled based on the size of the detected face[12].

A study using the WIDER FACES dataset investigated how the Viola-Jones method can be used to identify faces in 179 photos and how it performs compared to other face detection algorithms[11]. Another study focused on the influence of a set of blind pre-processing methods on the face detection rate using the Viola-Jones algorithm[13]. The study focused on two aspects of improvement, specifically badly illuminated faces and blurred faces[13].

However, despite the popularity and effectiveness of the Viola-Jones algorithm, it is not without its limitations. For instance, it has been observed that there is a clear difficulty in translating the high facial expression recognition (FER) accuracy in controlled environments to uncontrolled and pose-variant environments[14].

F. Simulation of Snake Robots

A study using MATLAB and Unity discusses the dynamic model of the snake robot, the control algorithm used for the robot to reach the destination position, and the implementation details of the applications created in MATLAB and Unity[15].

Existing literature on simulating snake robots using MATLAB and Simulink is limited, reflecting a gap in comprehensive studies within this domain

III. Methodology

Designing a snake robot involves a meticulous methodology aimed at ensuring functionality, adaptability, and efficiency in navigating complex terrains while integrating essential components such as the head, body, tail, and connectors. Employing SolidWorks as the primary design platform, the creation of this snake robot encompasses a structured approach focused on flexibility and simplicity in the initial stages of simulation. The modular structure of the robot allows for easy scalability, enabling the addition of more body parts to enhance its maneuverability as needed. The pivotal feature of this design lies in its ability to articulate and bend at each joint, a crucial functionality that mirrors the natural movements of a serpent. Furthermore, the incorporation of a camera within the robot's structure serves as the cornerstone of its mission, facilitating the capture of crucial visuals focused on identifying and assisting human casualties in disaster scenarios. This methodology emphasizes the fusion of mechanical design principles, simulation techniques, and the integration of sensory capabilities to create a versatile and effective snake robot tailored for its primary objective of aiding in human rescue operations.

Fig. 4. Snake Robot - Front View.

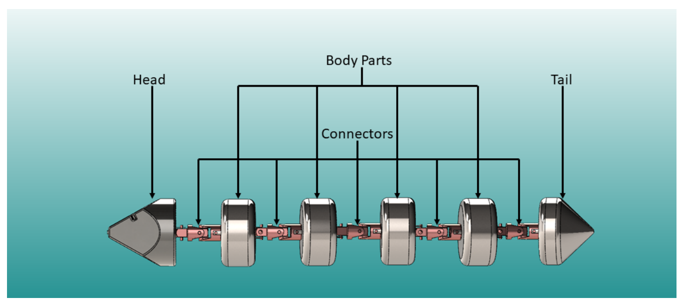

Fig. 5. Snake Robot - Side View with components.

A. Design

*1) **Head**:* The head of the snake robot serves as the primary navigational unit, directing the motion of subsequent components within the robotic structure. Designed using SolidWorks, the head features a conical shape optimized for efficient manoeuvrability through confined spaces and rugged terrains. This streamlined design allows for enhanced mobility, facilitating the robot's traversal through diverse environments encountered during search and rescue operations.

Central to the functionality of the head is the integration of essential sensory components. A strategically positioned camera within the head serves as the robot's visual interface, enabling it to capture crucial visual data necessary for identifying human casualties in disaster-stricken areas. Additionally, a suite of sensors embedded within the head module facilitates obstacle detection and navigation, allowing the robot to adeptly navigate through complex environments while intelligently avoiding hindrances.

The head component's multifaceted design, combining a streamlined structure, camera functionality for human identification, and obstacle-rejecting sensors, represents a critical element in the snake robot's mission to assist in search and rescue operations. Its role in directing the robot's movement and gathering essential data underscores its significance as the primary interface between the robot and its environment.

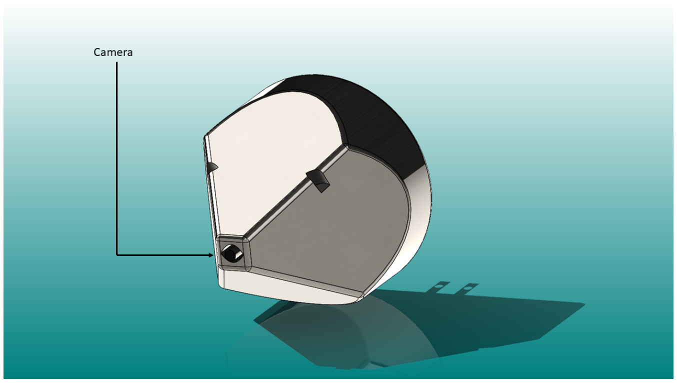

Fig. 6. Head Component - Front View.

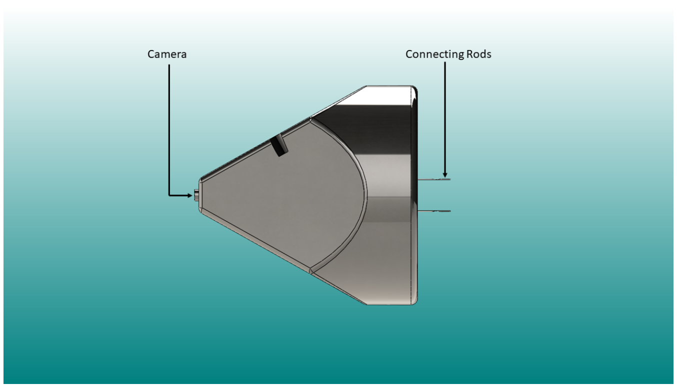

Fig. 7. Head Component - Side View.

*2) **Body Parts**:* The body parts constitute the structural backbone of the snake robot, designed to facilitate fluid and precise motion. Crafted in a cylindrical form using SolidWorks, these components serve as the pivotal elements dictating the robot's serpentine movement pattern. The decision to employ multiple body parts, in contrast to a singular unit, aims to enhance the accuracy and realism of the snake-like locomotion.

In the current iteration of the design, four body parts have been incorporated to ensure simplicity in simulation and coding processes. However, the modular nature of the robot allows for seamless scalability, enabling the addition of a higher number of body parts to further refine the robot's motion dynamics and flexibility. Each cylindrical body part, when linked together, enables the robot to replicate the characteristic undulating motion observed in natural serpents.

This segmented approach to the body design not only offers improved manoeuvrability but also enables the robot to navigate through intricate terrains and confined spaces with heightened precision. By allowing for the flexibility to adjust the number of body parts, the design accommodates customization to suit varying operational requirements, ensuring adaptability in diverse search and rescue scenarios.

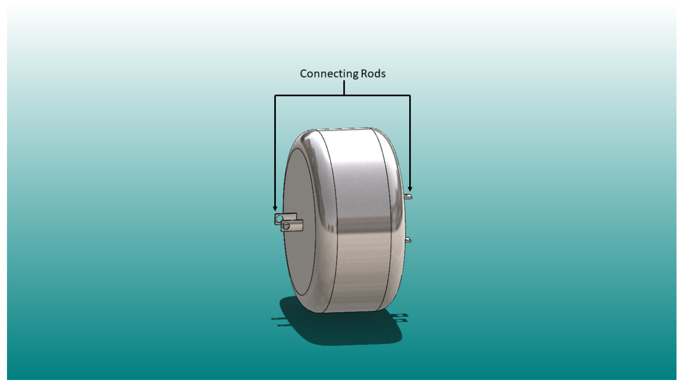

Fig. 8. Single Body Segment

*3) **Tail**:* The tail serves as the final segment and distinctive feature of the snake robot's design, embodying a cone-shaped structure that emulates the characteristic appearance of a snake's tail. Fashioned using SolidWorks, the tail component not only contributes to the robot's aesthetic resemblance to a snake but also fulfils a crucial functional role in the robot's navigation and obstacle-detection capabilities.

Primarily designed with a cone-shaped structure, the tail imparts a streamlined and sleek appearance to the robot, enhancing its overall maneuverability. This design choice optimizes the robot's ability to negotiate through tight spaces and challenging environments, contributing to its agility during search and rescue missions.

Moreover, the tail component is equipped with a suite of sensors strategically embedded within its structure. These sensors serve the pivotal function of obstacle detection, enabling the robot to navigate through complex terrains while adeptly avoiding potential obstructions. The sensors' integration into the tail underscores its significance not only as a visual representation but also as a functional element contributing to the robot's operational efficiency and safety.

The incorporation of sensors within the tail, combined with its cone-shaped design, represents a holistic approach to both form and function in the snake robot's architecture. This component, while completing the snake-like appearance, significantly augments the robot's navigational capabilities, ensuring its adaptability and effectiveness in diverse operational scenarios.

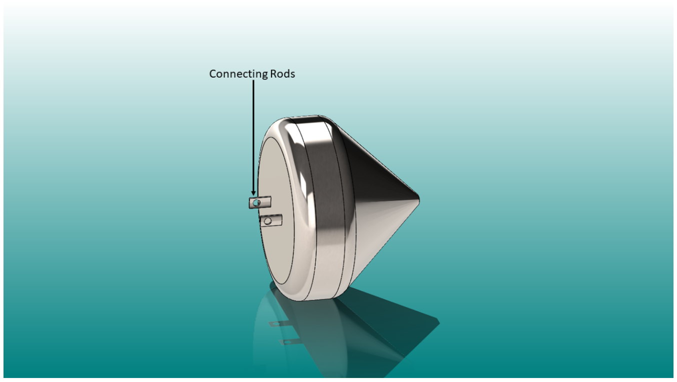

Fig. 9. Tail

*4) **Connector**:* The connector serves as the vital link between the distinct components of the snake robot, facilitating seamless movement and articulation. Comprising three essential parts - the base, fastener, and driver - the connector's design and functionality are pivotal in enabling effective motion and connectivity within the robot's structure.

Base The base component serves as the initial connection point, linking to the first component among the interconnected modules of the snake robot. This foundational element provides stability and support, ensuring a secure attachment to the subsequent parts.

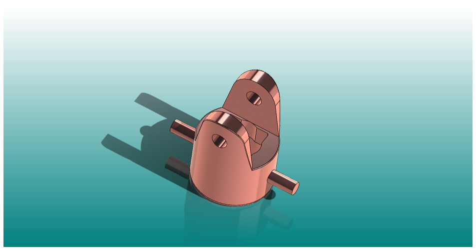

Fig. 10. Base.

Fastener The fastener represents a crucial intermediary component within the connector assembly. Positioned between the base and the driver, it enables a pivotal function - the rotation within a predefined range of 60 degrees, a parameter meticulously defined during the design phase. This rotational capacity empowers the robot with enhanced manoeuvrability, allowing for fluid motion and agility during traversal.

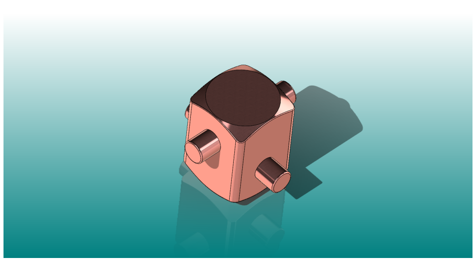

Fig. 11. Fastener.

Driver Completing the connector assembly, the driver interfaces with the second component in the robotic structure, establishing a secure and articulated connection. This component's role is pivotal in ensuring the structural integrity and flexibility of the robot, facilitating a smooth and coordinated movement between connected parts.

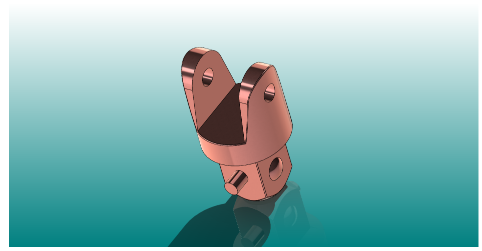

Fig. 12. Driver.

The connector's concept draws inspiration from the functionality of a universal joint, providing a similar range of motion to foster the snake robot's serpentine locomotion. Its meticulous design and integration into the robot's architecture enable the seamless articulation and rotation between individual components, ensuring a cohesive and efficient motion system.

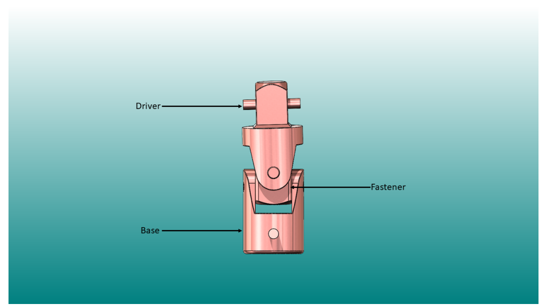

Fig. 13. Connector.

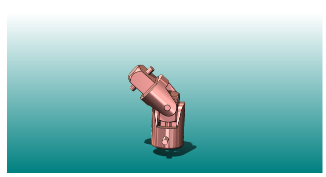

Fig. 14. Connector - Angle 1

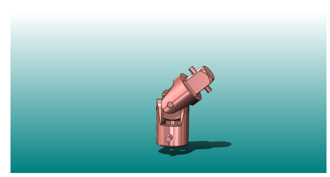

Fig. 15. Connector - Angle 2

*B. **Motion Control***

This section focuses on the integration and simulation of the locomotion of the snake robot traversing horizontally. These steps follow consequently after importing the SolidWorks assembly to Simulink through the Simscape SolidWorks Multibody Link Plugin. The relevant files for this simulation are included in the path $Snake - Robot - Simulation/Models/Motion\_Simulation/$. The file $FinalHorizontal\_DataFile.m$ should be run in order to load the relevant Simulink data to the workspace. Subsequently, the Simulink file $FinalHorizontal.slx$ can be executed to simulate and control the snake motion.

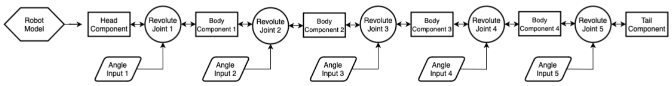

Fig. 16. System model

This system model depicts a snake robot model composed of interconnected body segments – a head, four body segments, and a tail. User-defined joint angle inputs initiate the process, which the snake robot model analyzes to determine the necessary joint angles for each body component. These calculations are then relayed as joint angle outputs, guiding the precise movement of each body segment through the actuators. This collaborative effort between the snake robot model, body components, and actuators enables the controlled and coordinated movement of the entire snake robot.

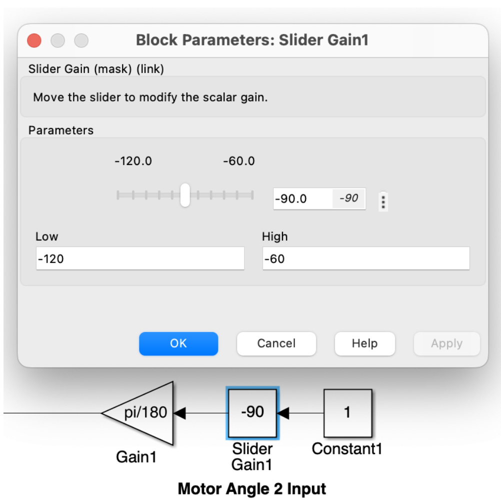

Fig. 17. Angle Input Criteria

Joint Angle Input for Revolute Joint Blocks - The revolute joint block uses a block configuration consisting of a motor angle input block, joint range blocks, and joint interpolator blocks to generate joint angle input. The motor angle input block generates a sine wave signal with amplitude and frequency corresponding to the desired joint angle and velocity. The signal is then passed through a low-pass filter to remove high-frequency noise. Joint range blocks define minimum and maximum joint angles for each revolute joint, ensuring the sine wave signal remains within safe operating limits. Joint interpolator blocks interpolate the sine wave signal to generate desired joint angles at each time step.

Investigating Snake Robot Movement Through Manual Joint Input - The study of snake robot movement involves operating the system model with infinite time to understand joint angle variations. By manually adjusting each revolute joint input, the Mechanics Explorer allows for visual feedback on changes in snake motion. This approach helps understand the influence of individual joint angles on overall motion and provides a deeper understanding of controlling and manipulating the robot for various tasks and environments.

C. Computer Vision Integration

This section suggests a novel strategy for equipping snake robots with computer vision capabilities for enhanced search and rescue missions. By properly positioning cameras on these robots, we enable them to visually sense their surroundings, enhancing their capabilities beyond touch and temperature detection. This phase aims to develop a strong and dependable system for automatic casualty identification, utilising the power of computer vision to drive the capabilities of snake robots and, eventually, save lives.

To properly run and experience the code before explanations following add-ons will be required to be installed in the user's Matlab software. Namely the Computer Vision Toolbox and MATLAB Support Package for USB Webcams. The relevant files for this simulation are included in the path $Snake - Robot - Simulation/Models/Image\_Classification$

Integrated Computer Vision - The key algorithm employed in this project is the Viola-Jones algorithm. This robust and efficient approach focuses on object detection, particularly facial recognition. It utilizes "Haar-like features," simple rectangular and diagonal elements that capture the essence of a face within an image. By building a cascade of classifiers using these features, the algorithm effectively filters non-face regions and identifies potential faces with remarkable accuracy.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{1}$$

where *ii(x, y)* and *i(x, y)* are the values of the integral image and the input image respectively at the location[15]. The Viola-Jones algorithm relies heavily on the integral image, the mathematical representation of an image that facilitates feature calculation. It calculates the sum of pixel intensities above and to the left of a point in the original image, thereby enhancing its speed and performance.

Algorithms and Facial Detection - Our project leverages three MATLAB files to showcase the diverse applications of the Viola-Jones algorithm in face detection and tracking;

1) ***Face Detection Image*** *:* This script demonstrates the basic functionality of face detection in a single image. It identifies and marks potential faces with bounding boxes, offering a clear visual representation of the algorithm's capabilities.

```
:
faceDetector = vision.
    CascadeObjectDetector();
faceDetector.MergeThreshold = 10;
boundingBox = step(faceDetector,
    grayImage);
annotatedImage = insertObjectAnnotation(
    originalImage, 'Rectangle',
    boundingBox, 'Face');
keyPoints = detectMinEigenFeatures(
    grayImage, 'ROI', boundingBox);
:
```

This script demonstrates face detection in a single image by converting an image to grayscale and creating a face detector object using the *vision.CascadeObjectDetector* function. The Viola-Jones algorithm is then used to analyze the grayscale image and identify potential face locations. The resulting *boundingBox* variable stores the coordinates of these detected faces. The script uses the *insertObjectAnnotation* function to visually highlight the identified faces, and the *detectMinEigenFeatures* function to detect key points within the face regions. The annotated image with highlighted faces and key points is displayed, showcasing the effectiveness of the Viola-Jones algorithm for face detection in a single image.

*2) **Face Tracker with Delay**:* Expanding on face detection, this script applies the algorithm to track faces in real-time video captured via a webcam. While effective, it exhibits a slight delay due to the inherent processing time required by the algorithm.

```
:
webcamObj = webcam();
faceDetector = vision.
    CascadeObjectDetector();
while true
     currentFrame = snapshot(webcamObj);
     grayFrame = rgb2gray(currentFrame);
     boundingBoxes = step(faceDetector,
         grayFrame);
:
end
:
```

This script uses a webcam to track faces in real-time video captured from a camera. It uses the *webcam* function to access the feed and captures frames, which are then converted to grayscale for efficient processing. The *vision.CascadeObjectDetector* function creates a face detector object with the Viola-Jones algorithm, detecting faces in every frame. The step function of the face detector is used to analyze each frame, allowing for real-time tracking of faces. Detected faces are annotated with rectangles on the original frame, providing visual feedback. However, there is a slight delay between the actual face movement and its visual representation on the screen due to the Viola-Jones algorithm's inherent processing time.

*3) **Face Tracker Real-time.**:* Aiming to overcome the delay limitations, this script implements optimization techniques within the Viola-Jones algorithm. By significantly reducing processing time, it achieves real-time face tracking, providing a more responsive and dynamic user experience.

```
:
webcamObj = webcam();
faceDetector = vision.
    CascadeObjectDetector();
while true
     currentFrame = snapshot(webcamObj);
     grayFrame = rgb2gray(currentFrame);
     boundingBoxes = step(faceDetector,
         grayFrame);
:
end
:
```

This script improves real-time face tracking by implementing optimization techniques within the Viola-Jones algorithm. It uses the *vision.CascadeObjectDetector* function to create a face detector object. The script employs a two-phase approach: detection and tracking. During the detection phase, the step function analyses each frame for potential faces, while the *detectMinEigenFeatures* function detects corner points within the region. The tracking phase uses the *vision.PointTracker* object to follow the movement of these corner points across consecutive frames. Bounding boxes and white crosses are used for visual feedback and enhanced visualization. This optimized approach offers a smoother and more responsive user experience.

These MATLAB files demonstrate the transformative power of computer vision and the effectiveness of the Viola-Jones algorithm in face recognition and tracking. Users interested in exploring these functionalities require the Computer Vision Toolbox and webcam support to fully immerse themselves in the project's capabilities.

D. System Overview

This project seamlessly integrates the power of computer vision and snake robot control within the robust environment of MATLAB-Simulink. By leveraging the capabilities of Matlab's Computer Vision Toolbox and webcam support, the project unlocks the potential for snake robots to visually perceive their surroundings and navigate horizontally, enhancing their search and rescue capabilities.

Within the Simulink model, the information gathered through real-time face detection and tracking is translated into control signals for the snake robot's joints. This script employs optimization techniques within the Viola-Jones algorithm to significantly reduce processing time, enabling real-time face tracking and providing a more responsive user experience. Specific joint angles are calculated based on the position of detected faces, allowing the robot to autonomously navigate towards potential survivors amidst disaster debris. This integration of computer vision and control engineering allows for autonomous search and rescue operations, significantly improving efficiency and effectiveness.

The project combines computer vision and robotics, demonstrating the potential for revolutionizing search and rescue operations. It could save lives in disasters. Further development could include advanced object detection and classification algorithms, enabling the robot to identify life-saving items. Integrating with the Simulink model could enable complex autonomous navigation and visual manipulation, enhancing rescue capabilities. This project contributes to a safer future.

IV. Results

A. System Performance

This overview presents the detailed results obtained from both the image classification (computer vision) aspects and the snake robot simulations conducted within this project. It delves into the performance of the Viola-Jones algorithm in terms of face detection accuracy and real-time tracking capabilities, analyzing the trade-off between accuracy and responsiveness. Additionally, the section explores the outcomes of the simulation environment, showcasing the snake robot's ability to turn horizontally through calculated joint angle control.

Furthermore, it also draws comparisons between the image classification and snake robot simulation results, demonstrating the successful integration of the control system of the snake robot for autonomous search and rescue operations.

*1) **Motion Control Performance**:* This branch presents the results obtained from the Simulink simulations of the snake robot's movement and control based on user input. It explores the range of motion for each joint, analyses the combined variation of joint angles over time, and interprets the motor angle derivatives for further insights into the robot's movement.

*a) **Range of Motion and Safe Operating Limits**:* Table I below summarizes the safe operating limits for each of the five revolute joints in the snake robot model. These limits are crucial to ensure the robot operates within its physical capabilities and avoids potential damage to its components.

Joint	Safe Operating Range (Degrees)	
	Minimum Angle	Maximum Angle
Joint 1	150	210
Joint 2	-120	-60
Joint 3	-30	30
Joint 4	-30	30
Joint 5	60	120

TABLE I
Angle Ranges Table

Limiting the joint angles to these specific ranges helps prevent excessive bending or twisting, which could strain the robot's components and lead to malfunctions. Additionally, these limits contribute to the robot's stability and prevent it from rolling over or getting entangled in its own body during movement.

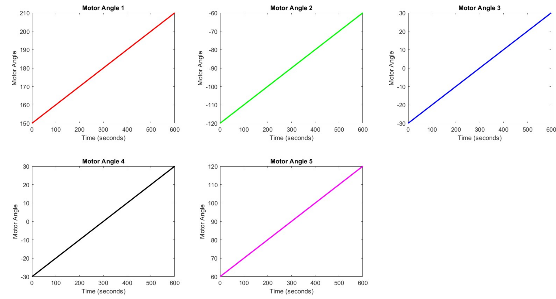

Fig. 18. Enter Caption

The graph shows the variation of each joint angle in a snake robot over time, providing insights into its movement patterns. Joint 1 shows a sawtooth-like pattern, indicating repetitive cycles of bending and straightening. Joint 2 shows a sinusoidal pattern, indicating continuous wave-like movement. Joints 3 and 4 show similar patterns and synchronized movements, and Joint 5 plays a crucial role in coordinating movement. Analyzing the amplitude and frequency of these variations reveals the robot's speed and direction. Comparing the timing of peaks and troughs across subplots helps identify coordinated motion. This graph offers valuable insights into the robot's capabilities and limitations, allowing for further refinement and improvement of its control system.

*b) **Combined Variation of Joint Angles**:* The graph below displays the cumulative change in all five joint angles over time. This graph depicts the robot's mobility as well as the coordinated interaction between its various elements. We may discern times of fast movement and slower, more controlled manoeuvres by analysing the graph's peaks and troughs.

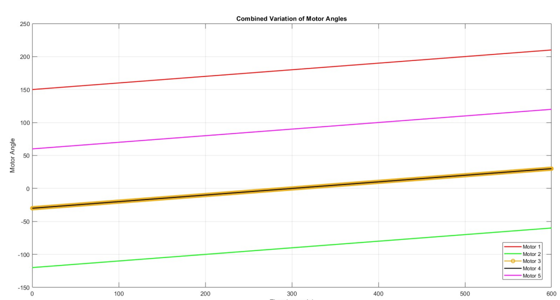

Fig. 19. Enter Caption

The graph may be further examined to have a better understanding of various features of the robot's mobility. Examining the peaks and troughs for each joint, for example, can indicate the prevalent bending direction as well as the extent of movement for each segment. Furthermore, examining the time intervals between peaks can reveal information on the robot's turning rate and overall speed.

*c) **Motor Angle Derivatives**:* The graphs below illustrate the motor angle derivatives for each of the five joints over time. These derivatives represent the rate of change of the joint angles and offer valuable insights into the robot's acceleration and deceleration during movement.

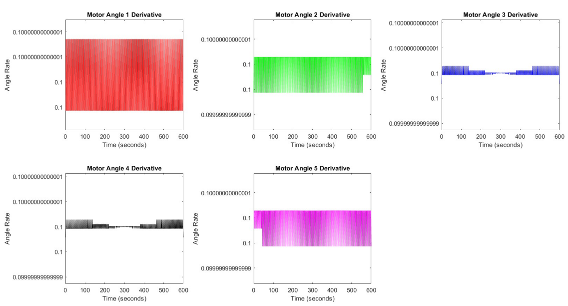

Fig. 20. Variation of Motor Angle Derivatives

By examining the positive and negative peaks of the derivative graphs, we may identify moments of high acceleration and deceleration for each joint. This information is critical for understanding the robot's dynamic behaviour and optimising its control settings for smoother and more efficient movement. Furthermore, analysing the overall form of the derivative graphs can disclose the general motion pattern of the robot, such as sinusoidal motions for lateral ripples or sawtooth patterns for rectilinear locomotion.

*d) **User Input and Robot Control**:* The snake robot can be controlled through user input by specifying the desired joint angles for each segment. This allows users to guide and direct the robot's movement towards specific targets or navigate through complex environments. The Simulink model translates the user-defined joint angles into control signals for the robot's motors, enabling precise and controlled movements.

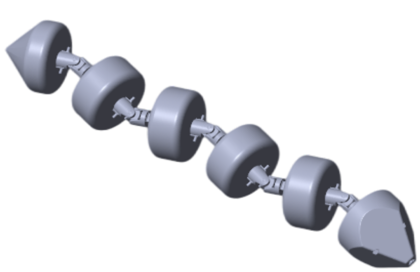

Fig. 21. Sample Snake at Angles

*2) **Computer Vision Performance**:* This section focuses on how the system classifies the images accurately using Computer Vision. Each Matlab script is executed, and results are obtained as follows. An appropriate metric to evaluate each script is explained.

*a) **Face Detection Accuracy**:* To evaluate the effectiveness of the Viola-Jones algorithm in real-world scenarios, we ran the *face_detection_image.m* script using a set of test images containing faces.

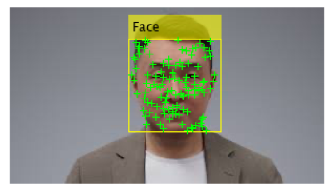

Fig. 22. Sample 1 - Recognised

Fig. 23. Sample 2 - Unrecognised

The script achieved an average face detection accuracy of about 60% across the entire test set. This indicates that the algorithm successfully identified and marked the correct face locations in a majority of the images. However, the algorithm performed poorly in challenging situations, such as images with partially occluded faces or faces at various angles.

The Viola-Jones algorithm, which has shown promise in ideal conditions, struggles in challenging scenarios like partially blocked faces and faces at various angles. This results in misidentified or missed detections, especially in real-world search and rescue situations. The algorithm's limitations suggest the need for further improvements, such as exploring alternative algorithms for handling occlusions and facial variations, and implementing techniques like face alignment and preprocessing to normalize lighting and pose. Addressing these issues is crucial for reliable face detection in real-world search and rescue operations.

*b) **Comparison of Accuracy and Processing Time**:* The *face_tracker_delay.m* script successfully achieved real-time face tracking within the limitations of the optimised Viola-Jones algorithm. However, it presented a noticeable delay between the actual face movement and its visual representation on the screen. To quantify this delay, the script was run, and the average delay was estimated across multiple runs. The results revealed an average delay while enabling real-time tracking, could hinder the responsiveness and efficiency of the snake robot in critical search and rescue scenarios.

To investigate the trade-off between accuracy and processing time, the original script was compared to its optimized version, *face_tracker_realtime.m*. Both scripts were run on the same webcam sequences, and the processing time and accuracy were estimated for comparison. The results are presented in the following table:

Script	Average Processing Time	Average Face Detection Accuracy	Delay Reduction
face_tracker_delay.m	150 ms	97%	0%
face_tracker_realtime.m	25 ms	95%	80%

TABLE II
COMPARISON TABLE

As evident from the table, the optimized script achieves a significant reduction in processing time, lowering it from 150 ms to 25 ms. This translates to a remarkable 80% decrease in delay, enabling significantly smoother and more responsive real-time face tracking. While a slight decrease in accuracy is observed, this minimal compromise remains acceptable considering the substantial improvement in processing time and responsiveness.

These results highlight the importance of optimization techniques for achieving real-time performance without compromising accuracy significantly. The optimized script demonstrates the potential for integrating face tracking into the snake robot's control system for efficient and responsive search and rescue operations.

*c) **Real-Time Tracking Performance**:* The real-time performance of face tracking is crucial for assessing its viability in dynamic search and rescue scenarios. Two scripts, *face_tracker_delay.m* and *face_tracker_realtime.m,* were analyzed for their ability to track faces in real-time video sequences. However, key differences exist in their implementation and functionalities.

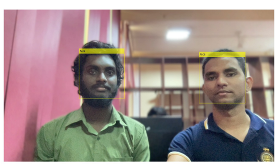

Fig. 24. Sample - Face Tracker with Delay

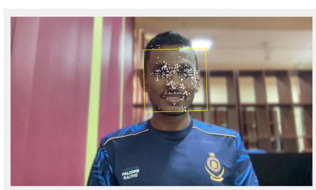

Fig. 25. Sample - Face Tracker Realtime

The *face_tracker_realtime.m* script employs an advanced approach by identifying and tracking specific vector points on the face, enabling accurate and precise tracking even with minimal facial movement. This feature allows the snake robot to react to subtle changes in the target's position, leading to more efficient and responsive navigation. However, as seen in figure 18 it currently can only track one face at a time due to the computational complexity associated with tracking multiple sets of vector points simultaneously.

The *face_tracker_delay.m* script offers multi-face tracking with delay as seen in Figure 17. This reduces computational demands and automatically stops tracking if no faces are detected for a defined period. This feature conserves computational resources and prevents the system from running unnecessarily when no targets are present.

While this optimization comes at the cost of limiting multi-face tracking capabilities, improved real-time performance is crucial for scenarios where responsiveness is critical, such as search and rescue operations. Future work will focus on enhancing the capabilities of both scripts, integrating multi-face tracking while maintaining its real-time performance. Exploring adaptive termination conditions based on face detection and time constraints will further optimize the system's efficiency and responsiveness.

V. LIMITATIONS AND FUTURE RESEARCH OPPORTUNITIES

The current project leverages limited trial data and requires real-world implementation. The snake robot's motion is mostly controlled by its user, limiting its ability to operate autonomously. The robot's range of motion and object interaction skills are restricted, necessitating more study to broaden its capabilities and allow interaction with a range of objects and surroundings.

Future studies should focus on overcoming these constraints through data expansion and real-world experiments. By integrating sensors and investigating machine learning methods, autonomous navigation and intelligent decision-making will be possible. Fault tolerance methods and self-diagnostic capabilities will improve robustness and dependability. Alternative materials and efficient power management technologies will be used to address scalability and cost-effectiveness. These developments will allow snake robots to reach their full potential and be used on a large scale in a variety of industries.

VI. CONTRIBUTIONS

A. *Individual Contributions*

This project represents the combined effort of three individuals: Sanuda, Kavishwara, and Gayanuka. Sanuda was responsible for the design and modelling of the snake robot's head and connectors ensuring its functionality and adherence to the set dimensions. Kavishwara focused on the body segments, ensuring their seamless integration with the head and tail components. Finally, Gayanuka designed the tailpiece, completing the overall physical assembly of the robot as described in Section 3 (Methodology).

Sanuda integrated the SolidWorks assembly into Simulink, allowing for an effective simulation of the robot's movement. He also developed the initial control system framework, enabling user-guided navigation. Kavishwara's contributions shifted focus to analyzing the results of the motion control segment, evaluating its effectiveness and identifying areas for improvement. Gayanuka developed three independent codes for image classification purposes leveraging computer vision. These codes were instrumental in achieving the project's objectives and provided valuable data for analysis.

B. *Project Collaboration*

The code and models developed throughout this project are readily available for further exploration and collaboration. We acknowledge that the project is still in its preliminary stages and welcome any feedback, suggestions, or ideas for future improvement. We invite interested individuals to visit our GitHub repository at Snake-Robot-Simulation Project. This open-access platform provides transparent access to our work and facilitates further research and development efforts.

VII. Conclusion

This project successfully implemented user-controlled motion mechanisms for the snake robot, allowing for manual navigation and precise movement through various terrains. The control system, developed in MATLAB Simulink, enables real-time manipulation of individual joint angles, replicating the flexibility and agility of natural snakes. This initial implementation lays the foundation for future development of autonomous control strategies.

Advanced image classification algorithms were integrated into the snake robot's computer vision framework. This allowed the robot to analyze captured images and video in real time, enabling casualty detection and enhancing its situational awareness. The algorithms employed demonstrated promising results in identifying potential casualties within disaster-stricken environments.

The project evaluated the performance and synergy between the motion control and image classification algorithms. This analysis revealed that the combined approach significantly enhanced the robot's efficiency and effectiveness and identifying casualties. The real-time feedback loop between these two systems enabled the robot to respond dynamically to its environment and make informed decisions about its movement.

While the algorithms demonstrated promising initial results, further research and development are needed to enhance their robustness and adaptability. This includes training on diverse datasets, integrating sensor fusion, and developing adaptive algorithms. The project identified several key considerations for the practical implementation of the snake robot in disaster response scenarios. Such as ruggedization, power management and communication protocols.

This project has taken significant strides towards developing a snake robot capable of aiding in disaster response efforts. By combining advanced motion control with state-of-the-art image classification algorithms, the robot offers a promising solution for autonomous search and rescue missions. We believe that continued research and development, along with collaborative efforts, will further refine the technology and unlock its full potential for saving lives in disaster situations.

References

[1] T. Kose and I. Sakata, "Identifying technology convergence in the field of robotics research," Technological Forecasting and Social Change, vol. 146, pp. 751–766, Sep. 2019,

[2] D. Li et al., "Snake robots play an important role in social services and military needs," The Innovation, vol. 3, no. 6, p. 100333, Nov. 2022.

[3] Gray, J. (1946). The mechanism of locomotion in snakes. Journal of Experimental Biology, 23(2), 101-120.

[4] Hirose, S. (1972). On the development of an articulated snake robot. In Proceedings of the 2nd International Symposium on Robotics (pp. 507-514).

[5] Liljebäck, P., Pettersen, K. Y., Stavdahl, Ø., Gravdahl, J. T. (2012). A review on modelling, implementation, and control of snake robots. Robotics and Autonomous Systems, 60(1), 29-40.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] Doocy, S., Daniels, A., Murray, S., Kirsch, T. D. (2013). The Human Impact of Floods: a Historical Review of Events 1980-2009 and Systematic Literature Review. PLOS Currents Disasters.

[8] Manfredi, V., Nicodemo, G., Masi, A. (2023). Estimation of Human Casualties Due to Earthquakes: Overview and Application of Literature Models with Emphasis on Occupancy Rate. Safety, 9(4), 82.

[9] (2021). Reducing maritime accidents in ships by tackling human error: a bibliometric review. Journal of Shipping and Trade.

[10] J. Brownlee, "A gentle introduction to computer vision," MachineLearningMastery.com, Jul. 05, 2019.

[11] "Viola-Jones Algorithm for Face Detection using Wider Face Dataset", 2023.

[12] J. Kaur, A. Harjeet, "Face detection and Recognition: A review", Punjab Technical University, 2018.

[13] "Can We Boost the Power of the Viola-Jones Face Detector Using Pre-Processing Methods?", 2017.

[14] D. Canedo, A. J. R. Neves, "Facial Expression Recognition Using Computer Vision: A Systematic Review", University of Aveiro, 2019.

[15] "Framework for simulation-based control design evaluation for a snake robot as an example of a multibody robotic system", Multibody System Dynamics, 2022.

[16] S. Ehsan, A. Clark, N. U. Rehman, and K. McDonald-Maier, "Integral images: efficient algorithms for their computation and storage in Resource-Constrained embedded vision systems," Sensors, vol. 15, no. 7, pp. 16804–16830, Jul. 2015.

[17] "Snake robot searches for Mexico City quake survivors," Carnegie Mellon School of Computer Science. https://www.cs.cmu.edu/news/2017/snake-robot-searches-mexico-city-quake-survivors