جـامعـة الشـارقـة
UNIVERSITY OF SHARJAH

Research Work Form

3rd Women in Research Forum

"Quwa: Women Empowerment for Global Impact"

Research Project Title :
Computer Vision Towards Functional Scene Understanding: Unpacking Activities of Daily Living Through AI in Low Vision (LV)

Research Category (Please choose one):

Research Category		Choose one
1.	Health Sciences	Engineering
2.	Pharmacy	
3.	Medicine	
4.	Dentistry	
5.	Engineering	
6.	Science	
7.	Computing & Informatics	
8.	Other disciplines with contribution to science and technology (i.e. Law, Sharia, and Islamic Studies, Fine Art and Design...)	
9.	Arts, Humanities and Social Sciences	
10.	Communication and Business Administration	

Category	Choose one
Community Impact	
Innovation	Yes
Industry Partnership	

Principal Investigator (PI)

Full Name: Lifan Yu Signature: Lifan Yu University Name: New York University Abu Dhabi

Academic Rank: Assistant Prof ☐ Associate Prof ☐ Professor ☐ PhD Student ☐ Master Student ☐ RA ☑

College: Department: Science

Telephone: +971 565019531 email: ly1164@nyu.edu

Other Researchers

1. Full Name: Yi Fang Signature: Yi Fang University Name: New York University Abu Dhabi

Academic Rank: Assistant Prof ☐ Associate Prof ☑ Professor ☐ PhD Student ☐ Master Student ☐ RA ☐

College: Department: Engineering

Telephone: +971 262 84891 email: yfang@nyu.edu

1. Research Project Overview

For people who have Low vision (LV), performing activities of daily living (ADL) is significantly more difficult than others. LV severely constrains one's mobility [1]–[3] and results in loss of independence ADL (i.e., eating, bathing, dressing, toileting, and transferring), followed by issues with unemployment [4], quality of life losses [5]–[7], and functional dependencies [8] that harm people's psychosocial well-being [9]. Although there are traditional methods to address ADL needs--assistive tools (i.e., canes), gadgets for preventing kitchen injury, and braille on facilities, they might not be sufficient in providing a high level of interactive guidance and real-time intelligent assistance. Now we are able to utilize artificial intelligence (AI), computer vision (CV), and versatile portable environment sensors for creating more intelligent and powerful assistive devices, we are able to move beyond current ADL assistive devices, with a paradigm shifting towards modern tools that will allow the visually impaired to regain the mobility losses associated with sensory deprivation, and stop the current downward 'spiral' of debility.

We are thus motivated to propose our solution **CV^{4}LV (Computer Vision for Activities of Daily Living through AI in Low Vision),** a wearable hardware-software integrated intelligent platform which provides real-time action prediction in one's immediate three-dimensional environment in order to assist in rehabilitating the visual awareness and living independence of people with LV. CV^{4}LV allows human-machine interaction and is composed of sensor systems embedded with algorithms on functional scene understanding through computer vision, and reinforcement learning for action decision strategy learning. Our system will provide action planning and guidance for people with LV in their daily activities, such as cooking, eating, opening doors, moving items around, etc., thus preventing them from indoor injuries due to collision, heat, cuts, etc. CV^{4}LV can help enhance their real-time perception of the surroundings and help them safely and independently perform daily activities. In both hardware and software designs, we intend to maximize the level of safe guidance for people with LV.

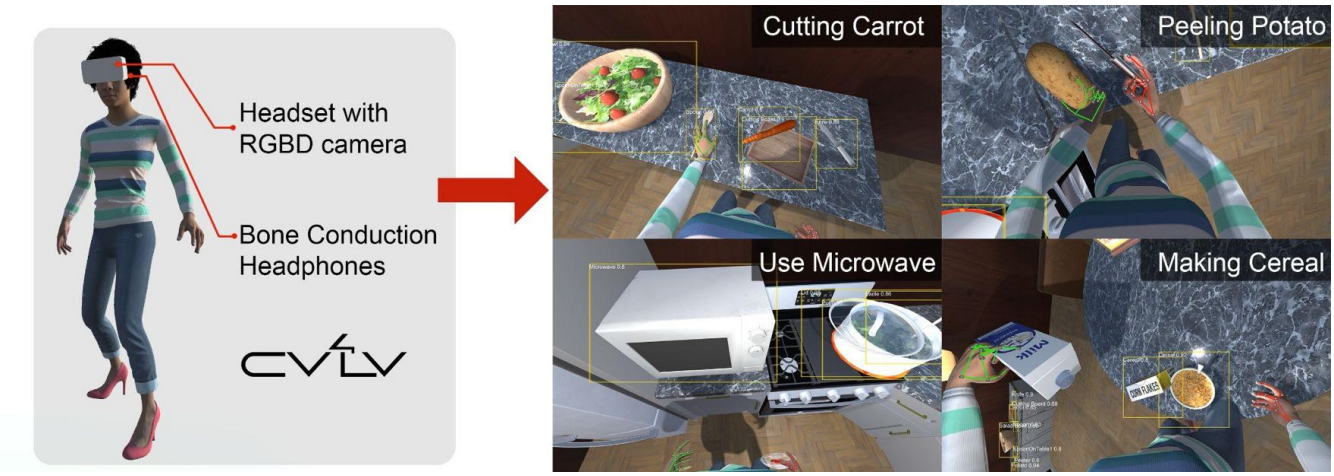


Figure 1: CV^{4}LV project overview

2. Research Project Objectives [should relate to Community Impact, Innovation or Industry Partnership]

The World Health Organization estimates there are 253 million people with vision impairment (VI) worldwide, 81% of which are 50 years old and above and 7.5% are below 15 years old [10]. There is a significant amount of need for ADL guidance for visually impaired people. The estimated economic impact of vision loss was $3 trillion dollars globally in 2010 [11], with conservative estimates of $75 billion dollars per annum in the US alone [12], [13]. Solving ADL problems for people with LV will bring a profound positive impact on society. Assisting them to live healthily and safely allows them to more actively and independently interact with people and environments around, and thus can enhance the well-being of our society.

Motivated by the massive need of people with LV to live healthily, our objective for the community is for CV^{4}LV to help people with LV interact with their surroundings in real-time, regain mobility losses, guide them through daily activities, and improve their living quality and psychological conditions. We aim to implement it into an accessible and convenient wearable assistive device to help enhance their physical and psychological health, allow more freedom and safety in their daily lives, and decrease the difficulty levels in their interaction with surrounding environments and integration into our society.

To ensure a safe and powerful assistive system to assist people with LV, our innovation objectives are focused on the software system of CV^{4}LV:

(1) Construct a game-like computerized engine as a 3D simulation environment for indoors scenarios which supports deep learning and reinforcement learning implementation and simulation of a person's indoors daily activities.

(2) Integrate human daily activity learning through LSTM with scene parsing to process real-time image input from the wearable RGBD camera to provide accurate and structured input for the reinforcement learning model.

(3) Introduce a deep reinforcement learning machine (DRLM) for safe decision making and action planning, to predict users' next steps in sub-activities, and safeguard users through their daily activities.

3. Literature Review

For assisting ADL needs of people with LV, traditional and computer vision methods are both put into practice. We will review their advantages and challenges.

Common traditional methods include the cane [14], kitchen gadgets which are helpful when making food/drinks [15], and individual supervisors who guide people with LV through their daily activities. While an individual supervisor is helpful, this solution is not always available due to high cost. Traditional assistive tools are limited in terms of real-time supervision.

In computer vision methods, visual imagery systems play a significant role [16]. Such systems integrate computer vision models with camera input and sensor input in order to process the surrounding environments where people see visually[16], and guide the users during navigation by giving warnings or instructions. One of which is the TYFLOS SYSTEM which consists of vision cameras, a laser range scanner, ear speaker, microphone and a portable computer [17]. This solution provides collision-free path-planning, detection of moving objects and description of the surrounding scene, through speech [17]. We also employ a visual imagery approach, processing real-time RGBD camera feed and transforming predictions from image into instructive speech, but instead of using a commonly seen stereo speaker, our solution uses a bone-conduction set to preserve the user's accurate perception of their surrounding sounds.

For indoors environments specifically, localization, path planning, and place recognition solutions widely exist[18], such as wearable devices which map out an unknown environment and locate the user location in it [19], [20]. Such solutions mainly focus on the surrounding environment and help people with LV navigate safely around indoors environments.

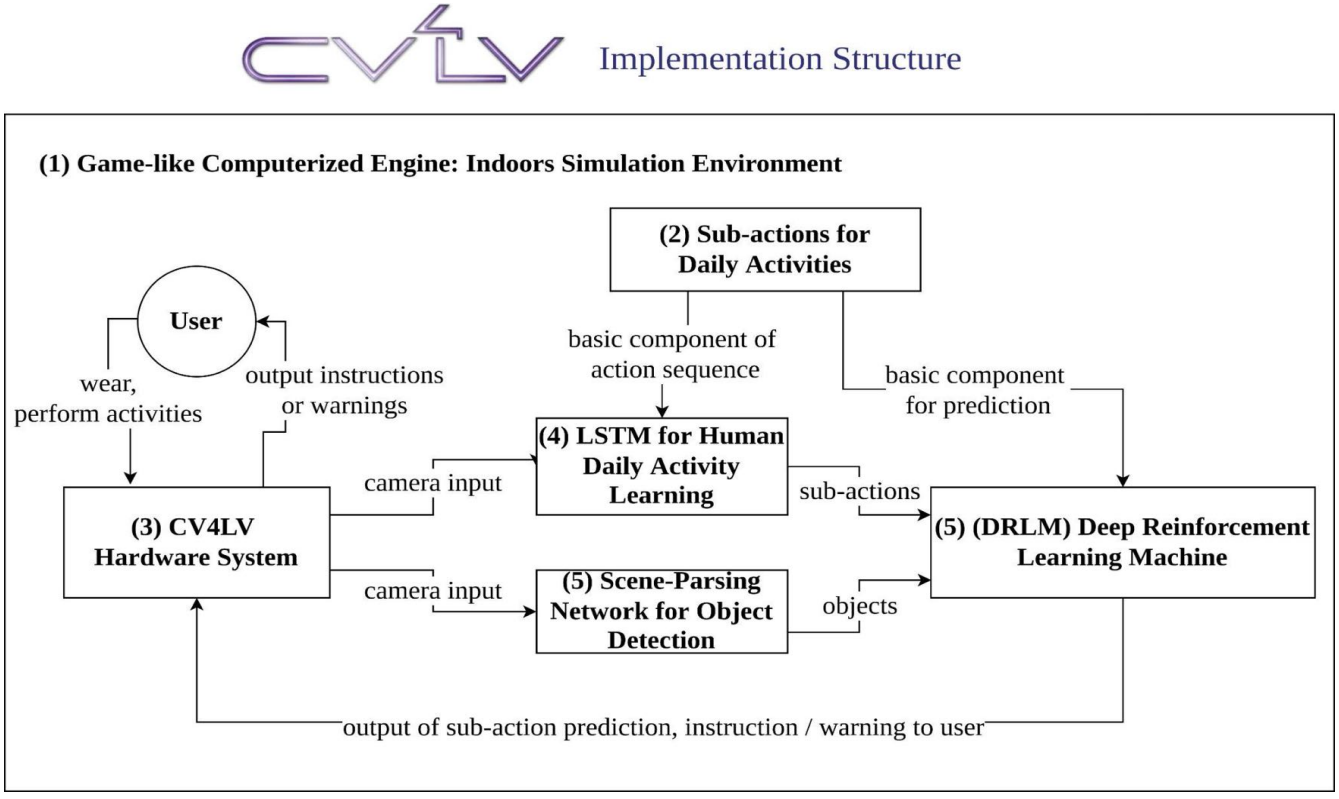
Through extensive literature review, it has been noticed that many of similar research projects are concerned with only localization, path planning or obstacle avoidance. There is a significant need for more research into actively predicting users' action sequences in certain indoors environments, which can be used to identify potential risks. This study aims to contribute to this research need.

This work will thus address the following existing challenges: (1) ADL assistance using deep learning with LSTM results and scene understanding as prior knowledge. (2) an accurate prediction of users' future movements. This is because while risks can be avoided by keeping the user away from objects and dangers, there are still risks of getting hurt during an ADL task. (3) A simulation environment with real-life characteristics which supports various machine learning algorithms and configurations of different 3D indoors scenarios is introduced in order to perform large-scale training and testing realistically without exposing any humans to risk.

4. Detailed Methodology

The methodology of **CV⁴LV** consists of 5 parts. The below figure shows the structure of the **CV⁴LV** implementation.

The **parts (1) to (5)** in the figure **correspond to the following subsections (1) to (5), respectively.**



(1) Game-like Computerized Engine: Indoors Simulation Environment

For running experiments and training the agent in indoors environments, a game-like computerized engine is implemented to simulate a variety of indoor environments. This simulation environment contains a living room, a kitchen, bedrooms, and other spaces commonly seen inside a home. It is equipped with furniture, cooking utensils, food, and other commonly seen items, and follows Newtonian dynamics. A simulated human character is built with a wearable camera on the head. This agent navigates around the simulation environment and performs daily activities, interacting with the objects around, reward and penalty can be generated easily without exposing people to real-world danger.

In this simulation environment, the agent needs to learn a series of actions to achieve the ultimate goal–-completing a specific ADL task. To achieve this, all items required for performing common ADLs are prepared and can be moved and used. The simulation environment is configurable in case of other training needs or additional ADL and addition and removal of objects. The interaction between the deep learning algorithm and the simulation environment is done via a Python Application Programming Interface (API) which utilizes the User Datagram Protocol (UDP). Through this API, the information parsed from the 3D simulation is fed as an input to the deep learning algorithm, and any prediction output from the deep learning algorithm is parsed and played by the agent in the 3D simulation environment.

The images in Figure 3 below demonstrate our simulation environment from different angles.

(a) Top view of the kitchen area

(b) First-person view of the kitchen area

(c) Top view of simulated kitchen and living room

(d) First-person view of the living area

Figure 3: Figures of the simulation environment

(2) Constructing Sub-action Sequences for Daily Activities

Each complete daily activity is decomposed into a sequence of sub-actions. The set of predefined sub-actions includes: reach, open, close, pick, place, release, pull, press, cut, stir, spin, shake, peel, each assigned with a Gray code, which is efficient for our hard-ware design since two successive actions in a sequential activity differ in only one bit in the action code. The table of sub-actions and figures demonstrating the corresponding simulations is shown in Table 1 below.

Code	0000	0001	0010	0011	0100	0101	0110	0111
Sub-action	reach	open	close	pick	place	release	pull	press
Simulation								

Code	1000	1001	1010	1011	1100	1101	1110	1111
Sub-action	cut	stir	spin	shake	peel	flip	pour	drink
Simulation								

Table 1: table of all sub-actions and their simulations

Examples of some actions commonly seen indoors ADL and their sub-actions are shown in Table 2 below.

Action	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9
Make cereal	reach	open	pour	close	place	reach	open	pour	close	place
Use oven	reach	open	reach	pull	place	release	shake	reach	close	
Cook soup	reach	open	reach	pick	stir	drink	stir	release	close	
Peel	reach	reach	peel	flip	peel	flip	peel	flip	peel	
Microwave food	reach	open	place	close	press	spin	press			
Cut vegetable	reach	cut	place							
Make salad	reach	stir	place							

Table 2: some commonly seen ADL sub-action sequences

The above definition of sub-actions is for action sequence learning of the LSTM and a structured prediction of sub-actions of the DRLM, which will be explained in the following sections.

OBJ OBJ

(3) CV^{4}LV Hardware System

The hardware of CV^{4}LV is a wearable system. A headset with a camera is used to capture live RGBD camera feed from the surrounding scenes. It can be oriented towards different directions following the user's head movements, and the scenes captured by the camera will thus resemble what is captured by eyes. In our scenario for indoors ADL, the user's hand movements serve as crucial input information. The camera on the headset is therefore able to capture hand movements, for users can orient their head to face the direction where their hands are at, which is known as proprioception -- the notion of self-perception to have both a sense of body orientation and position as well as a sense of body and limb motion [21].

An audio device will output warnings or instructions to the users according to predicted actions. We will integrate a wireless bone-conduction set with feedback element set which provide auditory as well as vibrotactile feedback channels, containing binaural bone conduction speakers. This design leaves the ear canal open for ambient sounds[22], thus providing safety and

convenience for our users, as they rely heavily on hearing for perceiving the surrounding environment. A microphone currently under development is integrated for oral communication-based voice recognition in order to capture user's exact needs and provide accurate assistance. For instance, when the user prompts with the voice command "Microwave food," the CV^{4}LV system will analyze the nearby objects, detect the microwave, and assist the user via voice in their tasks, for example. by replying "Place the food in the microwave one meter in front of you" . Likewise, depending on the user's actions and the objects detected, the upcoming action sequences are also predicted and suggested. The process discussed is shown in Figure 4 below.

(a) The user hearing the voice commands (b) The user carrying out the action

Figure 4 : CV^{4}LV Hardware System demonstration

(4) Human Daily Activity Learning Through LSTM

An LSTM (Long Short Term Memory network) method is proposed for human daily activity learning in CV^{4}LV. While Recurrent Neural Networks (RNNs) are commonly used in sequential processing tasks by investigating the complex spatial dependencies of input, the gradient vanishing bottleneck disables the normal RNN to solve the long-term dependency of the input sequence [23]. To this end, LSTM [24], a variant of RNN, is explicitly designed to avoid the long-term dependency problem since controlling gates with various functions are added. LSTM works tremendously well on a large variety of problems especially in the field of natural language processing such as speech recognition [25] and language translation [26].

In order to learn and understand the sub-activity sequence involved in a specific high-level activity, we proposed a model based on a many-to-many LSTM for interpreting the human actions of video frames over time, and each LSTM module consists of a memory cell and a number of input and output gates, controlling the information flow in a sequence, as well as avoiding important information loss.

For the extraction of sub-actions from a full action, a video consisting of the full process of completing a high-level activity is fed to the LSTM, passing through the hidden RNN layers, and a time-based action sequence is extracted. Illustration of learning the action sequence of peeling a potato is shown in Figure 5 below.

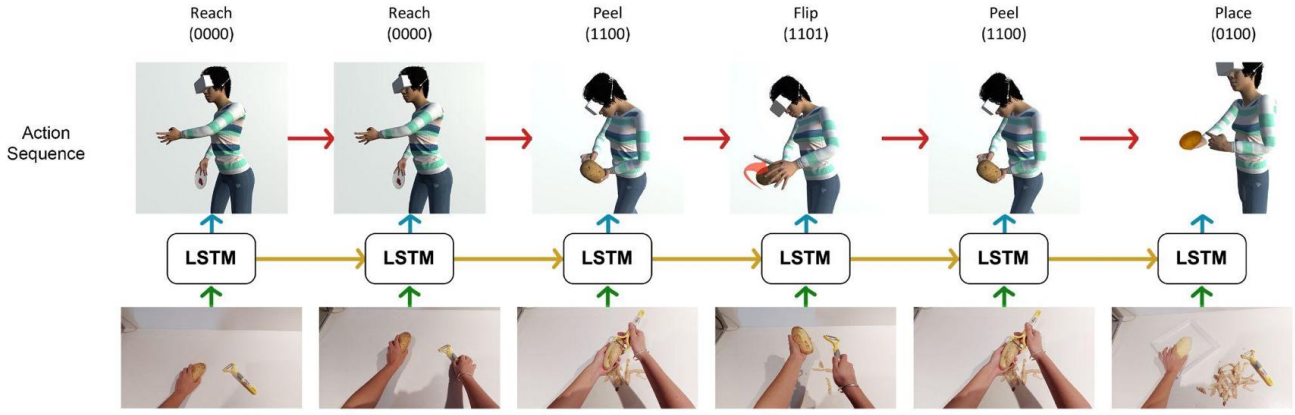


Figure 5: Illustration of learning the action sequence of peeling a potato using LSTM

(5) CV^{4}LV Software System and DRLM

The software system of CV^{4}LV consists of 3 main parts: (1) scene parsing network for processing real-time surrounding scenes (2) LSTM for human daily activity learning (3) DRLM for action sequence prediction, as shown in Figure 6 below.

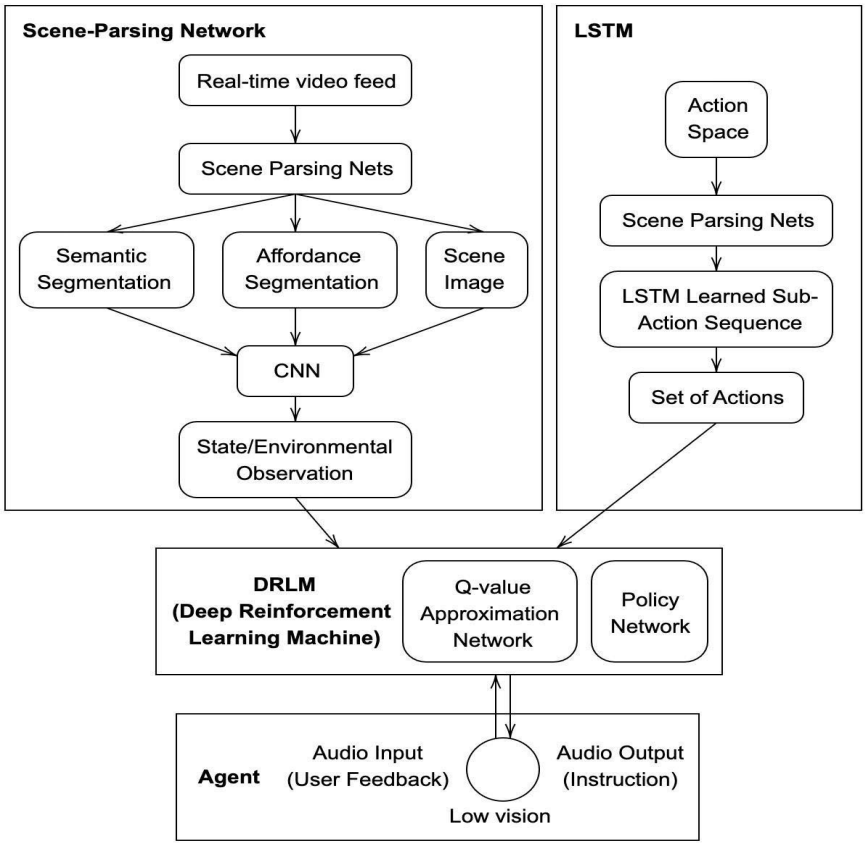


Figure 6: System architecture of CV^{4}LV software system

A scene parsing network is proposed in CV^{4}LV system for real-time 3D dynamic scene reconstruction and 3D semantic and affordance scene parsing. With both the semantic image and affordance image learnt from scene parsing networks and sub-activities sequences learnt from LSTM model based on human daily activity described above, a deep reinforcement learning machine (DRLM) is be introduced to gather all information extracted for intelligent decision making and action planning. This proposed DRLM involves the leveraging of the action sequence extracted from the LSTM model as prior knowledge. The benefits of this lays in three aspects:

1. It can accelerate the training process, reduce the time needed to build a trained model. Usually, the "trial-and-error" process in Deep Reinforcement Learning learns from having no knowledge and step by step trying different strategies and recording the reward feedback, which can be time-consuming compared to starting with the help of prior knowledge.

2. It can reduce mistakes the networks commit at the very first steps, thus improving the robustness of the representation and facilitating learning from raw data with more efficient and compact high level state representations [27]–[29].

3. It can help the model transfer from a simulated environment to the realistic world as the prior knowledge is common and universal in these environments.

CV^{4}LV coordinates the relations between "actions" and "objections" in an order. For example, after receiving the spoken assignment of "making cereal" from the user, the agent will start to analyze the scene images and apply corresponding policy to the state it is in. In this case, CV^{4}LV will first search for task-related items, such as cereal, milk, and bowl. The scene parsing network will cooperate with information regarding the localization and affordance of these objects.

Q-Learning is a reinforcement learning method which only takes a finite number of states (inputs) and a finite number of actions (outputs) [30]. It is modeled by a Markov decision process and uses a training mode - where the parameters are learned and an interference mode - where predictions are made. The parameters of Q-Learning are called Q-values. For the implementation of DLRM, Q-Learning algorithm is utilized to meet the project's needs. Q-Learning based reinforcement learning is appropriate for this project since the action space consists only of a finite number of sub-actions. The algorithm of DLRM in pseudocode form can be seen below in Algorithm 1.

Algorithm 1: CV^{4}LV Deep Reinforcement Learning Machine (DRLM) Algorithm

Data: Live RGBD camera feed from the headset of user, Voice Input for the action to be executed, action sequences learnt from LSTM, *Discount Factor* $\gamma = 0.75$, *Learning Rate* α=0.9 and `maximumTimeOut` = 5 seconds

Result: A predicted sequence of sub-actions

Set of sub-actions are extracted from the LSTM;
The individual sub-actions extracted are mapped to grey code;
Define the states s (inputs) from the sub-action sequence as `["microwave food", "cut vegetable", "make cereal", "make salad", "use oven", "cook soup", "peel"]` in respective Gray codes;
Define the actions a (outputs) from the sub-action sequence as `["microwave food", "cut vegetable", "make cereal", "make salad", "use oven", "cook soup", "peel"]` in respective Gray codes;
Initialize the Reward Table, $\mathbf{R}$, as an $n \times n$ 2-dimensional array of zeroes, where n is total number of sub-actions extracted
Automatically assign the weights 0 to the starting action and 1000 to the ending action, where the action is extracted from the user's voice input.
Initialize the Q-value table, $\mathbf{Q}$, as an $n \times n$ 2-dimensional array of zeroes, where n is total number of sub-actions extracted
while *Q-values are still updating* **do**
    Select a random state s_t from the possible states.;
    From that state s_t, perform a random action at that can lead to a next possible state, s_{t+1} , such that $R(s_t, a_t) > 0$, where R is the reward at s_t and a_t;
    Proceed to next state s_{t+1} and obtain the reward $R(s_t, a_t)$;
    Compute **Temporal Difference**, $TD_t(s_t, a_t)$, using $TD_t(s_t, a_t) = R(s_t, a_t) + \gamma \max(Q(s_{t+1}, a)) - Q(s_t, a_t)$ Update the Q-value by applying the Bellman equation $Q_t(s_t, a_t) = Q_{t-1}(s_t, a_t) + \alpha TD_t(s_t, a_t)$
    if *time taken for training* $\geq$ ***maximumTimeOut*** **then**
        return timeout;
    else
        continue loop;
    end
end
Perform the prediction for the action at state s_t by predicting the action at with highest Q-value for that state such that $a_t = \arg\max(Q(s_t, a))$

5. Experiments Set up & Results

Here a series of experiments set up is demonstrated. The following experiments are done stage by stage, arranged in the order of time implemented.

(1) **Testing the Indoors Simulation Environment,** usage, configuration, and the ability to support a variety of machine learning implementations;
(2) **Functional Testing of Hand Pose Estimation and Object Detection**
(3) **Testing Reinforcement Learning for Action Sequence Prediction** and its performance in predicting action sequences.

The **code of CV^{4}LV** is available on GitHub at https://github.com/SilvesterYu/CV4LV.git.

A **demonstration video** can be found in the ***demo*** section in each of the following subsections.

(1) Experiment 1: Testing the Indoors Simulation Environment

Experiment Summary:

This experiment is the construction of an indoors simulation environment. This environment is constructed in the Unity game engine. It is equipped with a kitchen and a living room with various objects commonly seen in a home and a simulated human character, which supports object detection, hand pose estimation and deep learning models.

Experiment Setup:

The sub-action animations of the agent are developed using the motion-captured data from Unreal [31], using animation retargeting technique to the agent model in Unity game engine. Other secondary actions such as walking, are extracted from the motion-captured data from Mixamo [32]. The sub-actions in each animation sequence is splitted using Unity Animation Sequence feature [33].

The simulation living room and kitchen environment is set up using 3D assets primarily from Unity Asset Store [34] and Sketchfab [35], two of the leading marketplaces for 3D assets.

A controller code for the agent is implemented via a C# script, which communicates with the DRLM via UDP communication and accepts the input and predictions from the deep reinforcement learning algorithm. All the possible states and actions of the agent in the simulation environment are modeled using a finite state machine. The state diagram of the relationship between the different actions of the agent can be seen in Figure 7 below.

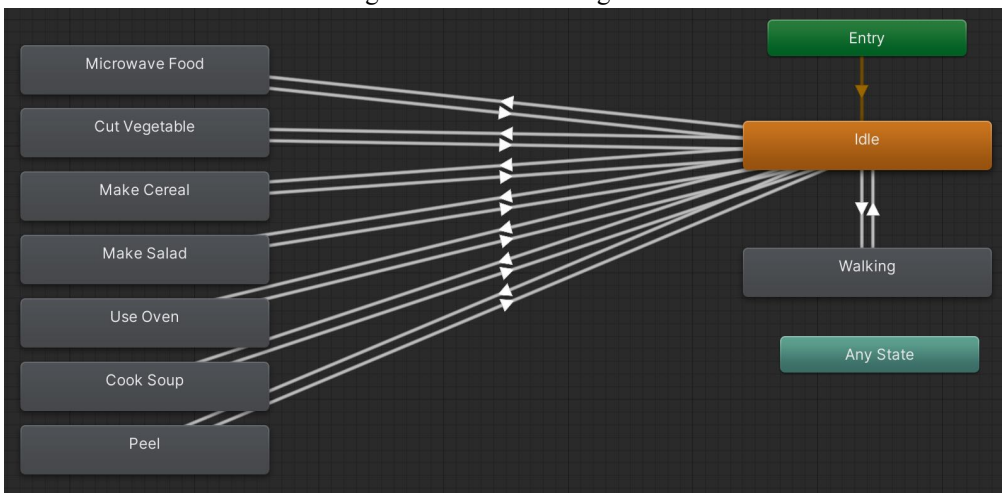


Figure 7: State diagram of all possible actions of the agent in simulation

Two additional C# scripts are also written for the simulation to communicate with the Mediapipe library [36] for proof-of-concept hand tracking and pose estimation (LSTM) and YOLOv5 [37] for detection of objects in the environment (proof-of-concept Scene-Parsing Network). YOLOv5 is chosen as the algorithm for proof of concept scene parsing since it can easily outperform existing object detection algorithms such as RetinaNet [38] and Single Shot Multi-Box Detector (SSD) [39], according to recent research literature [40].

Demo:

The demonstration video can be found in this link: https://www.youtube.com/watch?v=SsWW4wd2VYU The demo first shows a top view of the entire simulation environment, and then shows the simulated character and objects in the simulation. 2 views selected from the demonstration video can be seen in Figure 8 below.

Figure 8: simulation environment result

Result:

The simulation consists of three views. As shown in Figure 9 and Figure 10 below, the top left corner of the simulation consists of a third person camera view of the agent wearing the headset and performing actions. The main screen consists of the first person view of what the agent will see via the headset. The top of the simulation shows the output from the DRLM and the correct action sequence which should be predicted in discrete time steps. The real time object detection and hand pose estimation from YOLOv5 and Mediapipe [36] are rendered in real time on the first-person view of the simulation since these will also be seen by the user in real life when wearing the headset with CV^{4}LV system implemented.

Figure 9 below shows the DLRM (left) and the Simulation (right) running side by side before the action starts. The DLRM on the left predicts sub-actions based on input from the simulation environment.

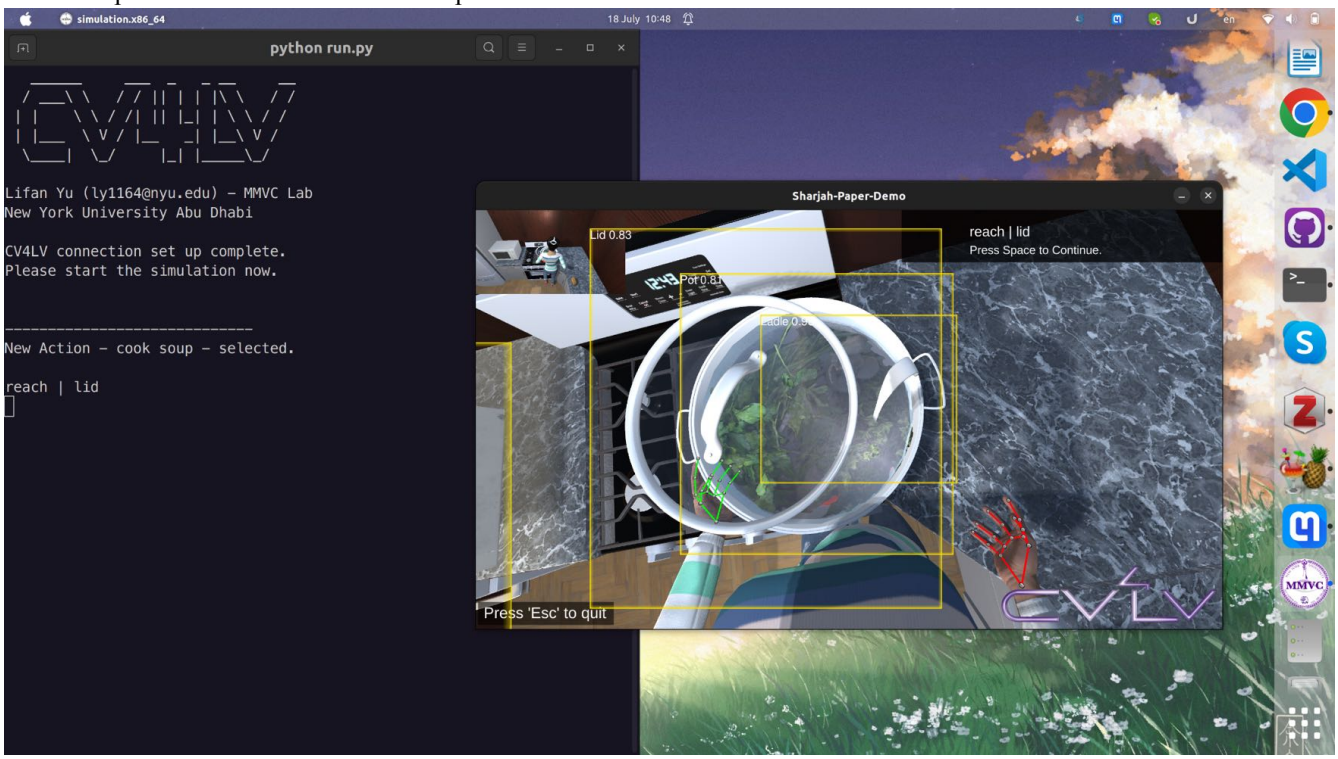


Figure 9: DRLM and Simulation running together (before a task)

Figure 10 below shows the DLRM (left) and the Simulation (right) running side by side, after the action has completed. The DLRM on the left shows all predicted sub-actions based on input from the simulation environment

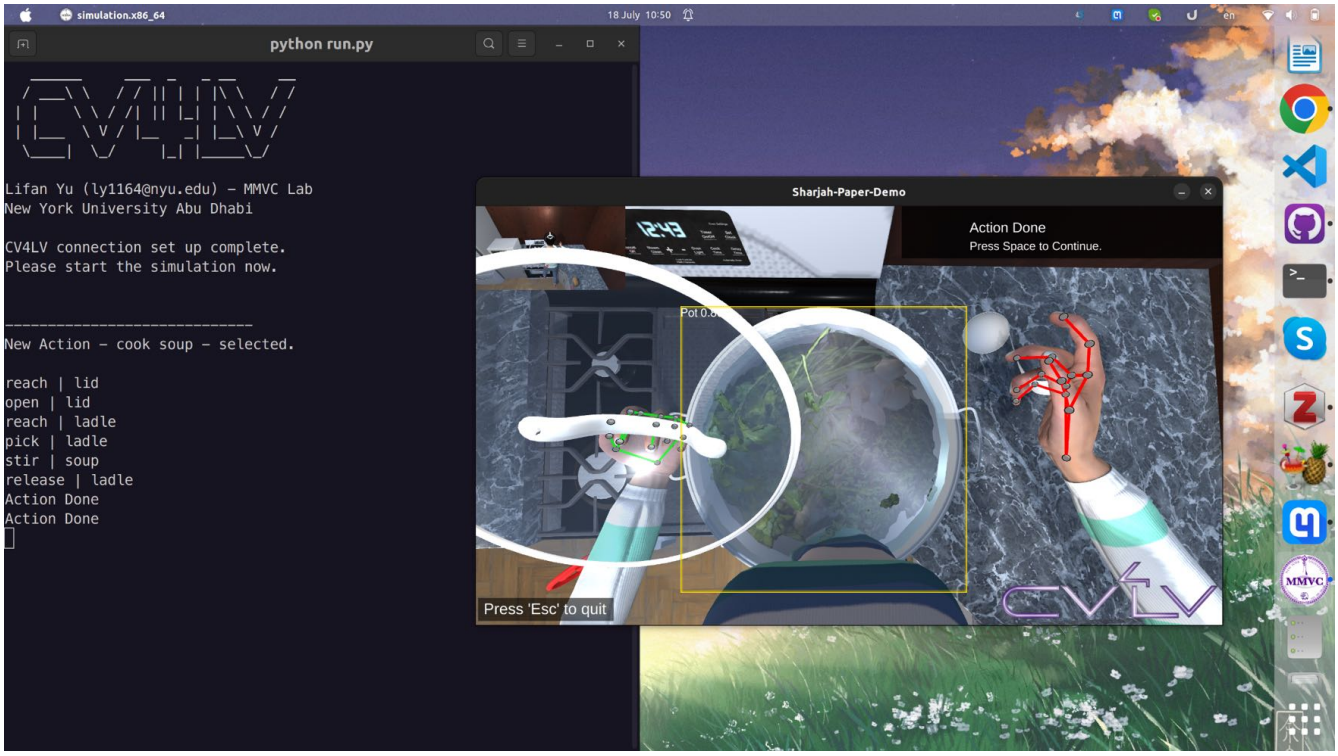


Figure 10: DRLM and Simulation running together (during a task)

This simulation is found to be very memory-efficient, as seen in the memory profile below in Figure 11. The entire simulation can be run only using 1.97 GB of system memory on CPU without requirement of any GPU, making it easily replicable.

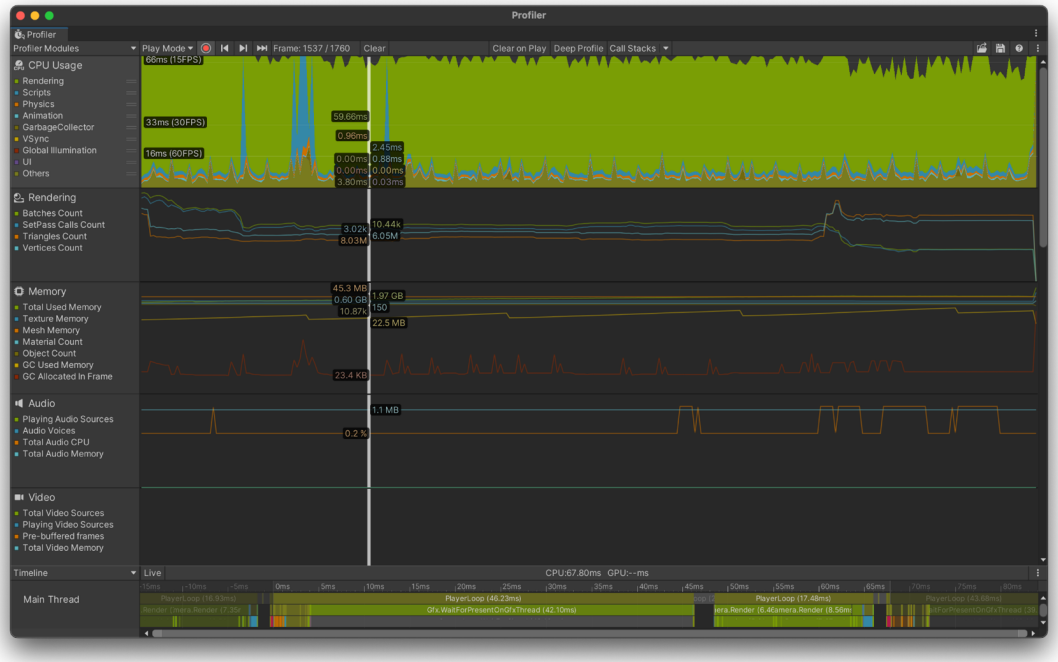


Figure 11: Memory profile of the simulation

(2) Experiment 2: Functional Testing of Hand Pose Estimation and Object Detection

Experiment Summary:

Here we test the functionality of hand pose estimation (LSTM) and object detection (Scene-parsing network). In order to accurately determine the user's hand pose and detect the objects in the camera input, we integrated into our simulation 2 state-of-the-art deep-learning models, namely, MediaPipe [36] for hand pose estimation and YOLOv5 [37] for object detection, which provided excellent results. Their detection results combined provide real-time and accurate input for the DRLM, which is shown in the following section.

Experiment Setup:

The full LSTM system is currently still being prototyped and is a part of the future long-term integration of the system. Hence, within the scope of this study, we are using MediaPipe library [36] for a proof of concept hand pose estimation algorithm, as a substitute for the full LSTM. Hand actions will be inferred from the coordinates of the hand landmarks detected from real-time image input. Although hand pose estimation for the action inference is used in this paper, the inputs and outputs of the current proof of concept remains identical to those of the LSTM proposed.

In the actual full implementation of the LSTM system, a large quantity of videos showing the hand movements of everyday activities will be recorded, analyzed, labeled, and the sub-actions will be extracted.

In a similar manner, the scene parsing network is also currently under development and it is a part of the future long-term implementation. Therefore, for the purpose of this study, an adequately robust scene parsing network is achieved through object detection using the state-of-the-art YOLOv5 algorithm [37]. Using our custom trained model, it can detect all 18 most commonly used objects in everyday life. Using this YOLOv5 implementation and the live RGBD camera feed, all the recognizable objects in the camera feed will be detected. The labels and the bounding boxes extracted from here can be used to infer the actions of the agent.

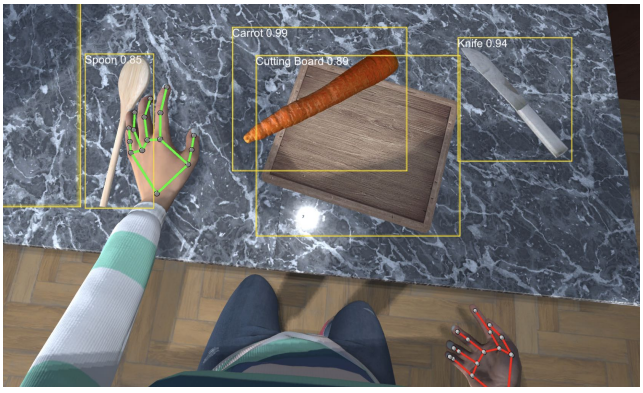
Demo:

Our simulated human agent performed a series of sub-actions with objects. The demonstration video of hand pose estimation and object detection can be found in this link: https://www.youtube.com/watch?v=a45zuNHR9OI

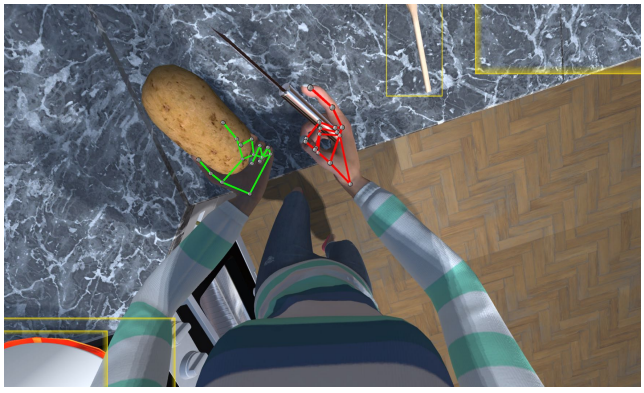
Result:

Both the hand pose estimation and object detection models integrate and perform well in the CV^{4}LV simulation environment, which provides a solid basis for further implementations. The computer simulation with the YOLOv5 scene parsing network and Mediapipe hand pose estimation LSTM implemented can perform the object detection and pose tracking of the agent in the 3D simulation very well.

The result of real-time object detection is shown in Figure 12 (a) below.
The result of real-time hand pose estimation is shown in Figure 12(b) below.



(a) Object Detection Result on Table



(b) Hand Pose Estimation during potato peeling

Figure 12: Results of Object Detection Hand Pose Estimation

(3) Experiment 3: Testing Reinforcement Learning for Action Sequence Prediction

Experiment Summary:

Here we test the Q-Learning algorithm of the DRLM (Deep Reinforcement Learning Machine) for sub-action prediction of daily life tasks. Experiments are run in the CV^{4}LV simulation environment and predictions are recorded. The model performs with high prediction accuracy.

Experiment Setup:

Once the simulation environment, the YOLOv5 proof of concept scene-parsing network and the MediaPipe hand pose estimation is implemented, the Q-Learning based Deep Reinforcement Learning Machine (DRLM) is implemented in Python according to pseudocode outlined in Algorithm 1. All sub-action sequences are mapped to Gray code for efficiency in hardware. Reward table values are automatically assigned based on the starting and ending actions of the main action sequence. Afterwards, the Python based DRLM is connected to the Unity 3D simulation environment through the User Datagram Protocol (UDP) to communicate the inputs and outputs. The prediction results will be shown in the 3D simulation environment.

Afterwards, the simulation is run for each main action sequence for 100 times, and the average numbers of correct and wrong predictions, as well as their standard deviations are recorded for quantitative evaluation of the effectiveness of CV^{4}LV system in a virtual simulation.

Demo:

The video of the CV^{4}LV action prediction can be in this link https://www.youtube.com/watch?v=aZxAg7kjqzE . In the demonstration video, several daily life tasks are demonstrated with the simulation running on the right side and real-time sub-action prediction output printed in the terminal on the left side.

Result:

Real-time outputs of CV^{4}LV in the 3D simulation environment are shown in the figures below.

Before performing a task, CV^{4}LV is able to identify the starting sub-action. For example, when the user reaches out a hand towards the carrot, with a cutting board and knife in the camera feed, the DRLM gives a prediction of “cutting carrot” (top-right corner), as shown in Figure 13 below.

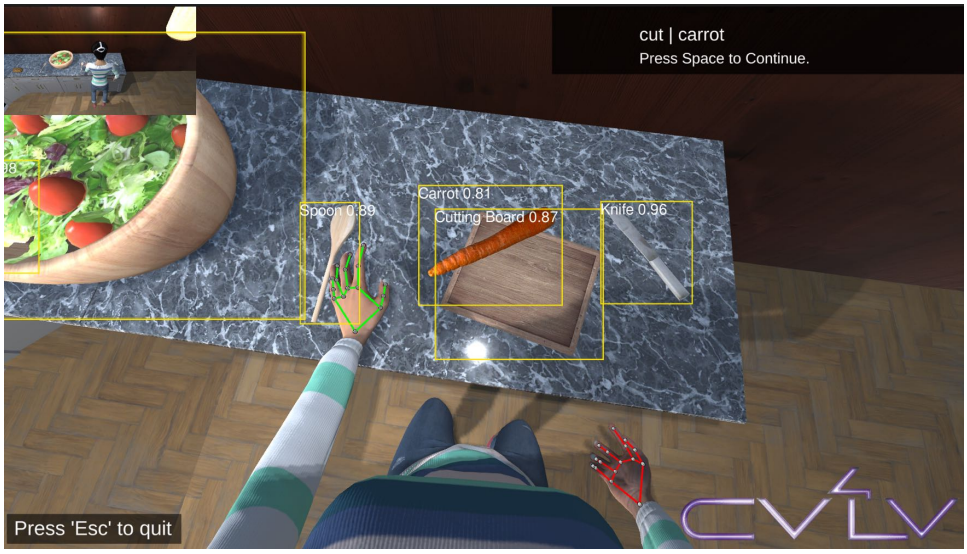


Figure 13: Prediction of cutting carrot action

When the user is in the process of “use oven” and when taking out a tray of cookies is detected, the DRLM predicts “place cookie tray”, as shown in Figure 14 below.

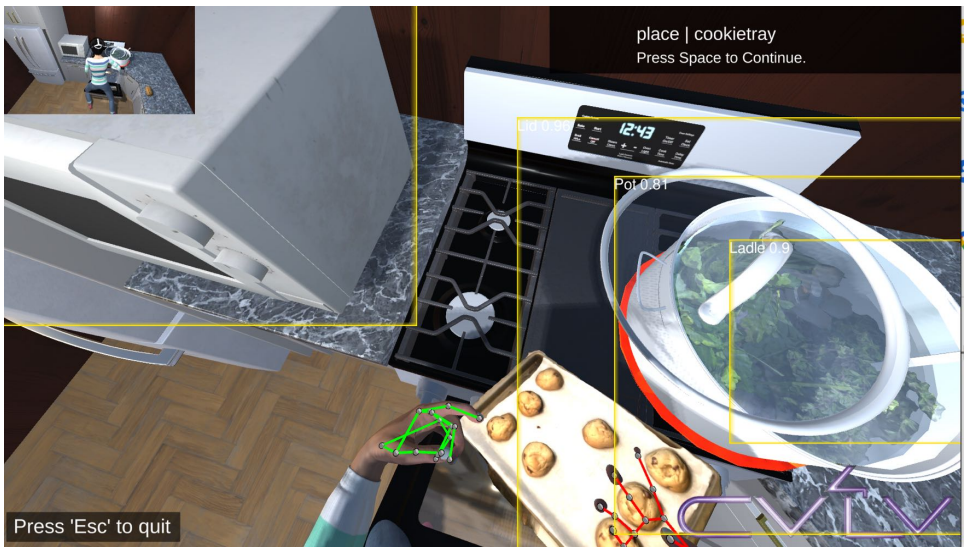


Figure 14: Prediction of placing the cookie tray

During the task “making cereal”, when the user is about to open a milk bottle, the model predicts that the user will “pour the milk”. The real-time result is shown in Figure 15 below.

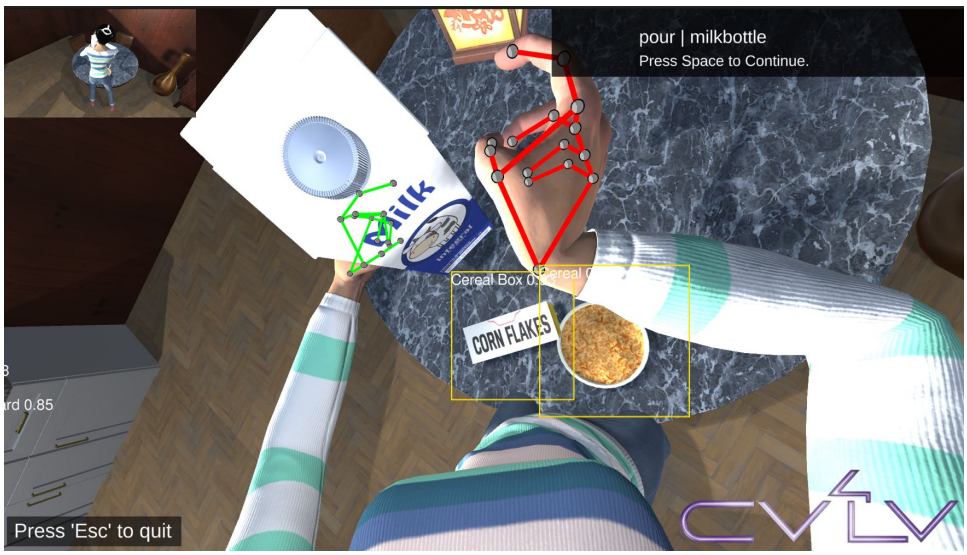


Figure 15: Prediction of pouring milk

CV^{4}LV can accurately predict the completion of a task. The figure below shows the user performing the “make cereal” task. After pouring cereal and pouring milk, the model predicts “done”, indicating that the task is completed. This is shown in Figure 16 below.

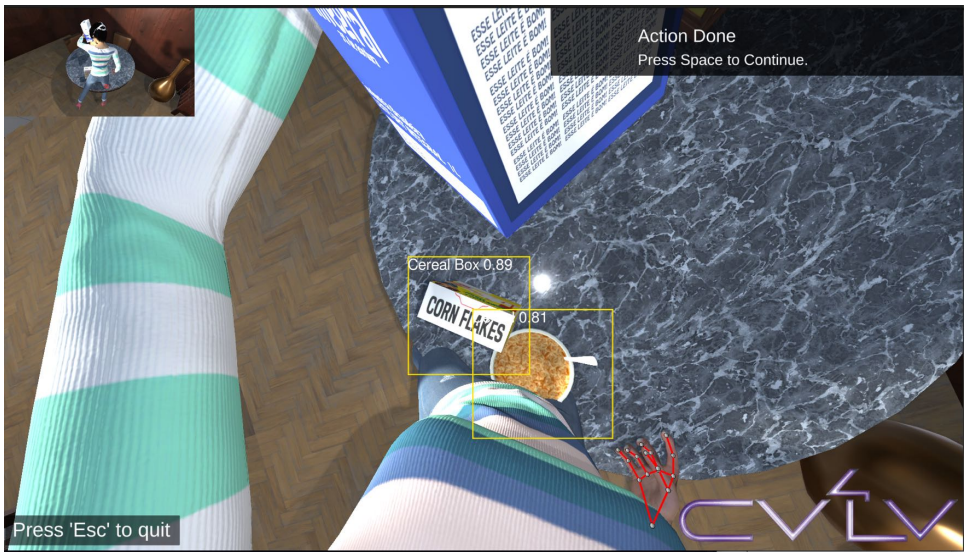


Figure 16: Recognition of the completion of an action

After the 100 simulation trials are done for each action, the results of the correct and wrong prediction percentages with uncertainties are shown in Figure 10 below. Generally CV^{4}LV algorithm predicts the entire action sequences well for all the actions with the correction prediction percentages between 83% to 94%. The correct prediction percentage is highest for the "Peel" action and the lowest for the "Make Cereal" action. However, the uncertainty is also highest for the "Peel" action and lowest for the "Cut Vegetable" action. In terms of the wrong prediction percentage, it is highest for "Make cereal" action and the lowest for the "Peel" action. It is hypothesized that the complexity of the task and the ease of objects being recognizable plays a role in causing these differences. However, it can be concluded that CV^{4}LV algorithm implementation right now in the proof of concept stage, has a very high correct prediction percentage and has an excellent potential to be implemented into a practical system in the future.

Results of average percentages of correct and wrong predictions over 100 runs for each action showing standard deviation are shown in Figure 17 below.

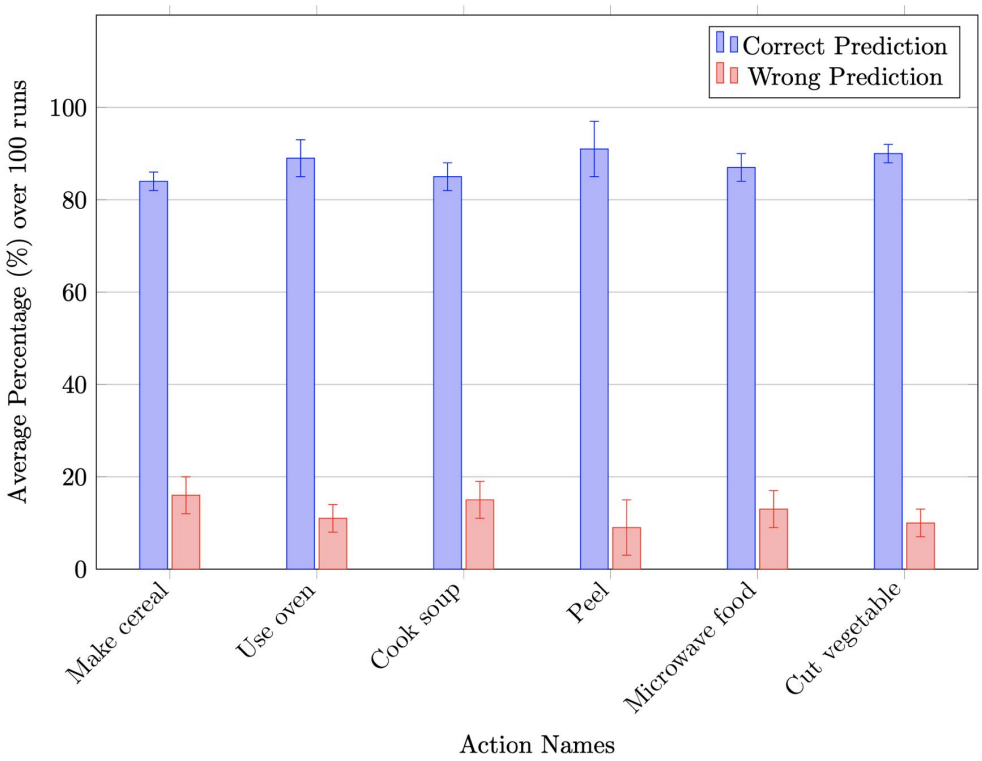


Figure 17 : Results of average percentages of correct and wrong predictions over 100 runs

Conclusion and Outlook

This study proposes an open-source solution CV^{4}LV (Computer Vision for Activities of Daily Living through AI in Low Vision) to assist people with low vision to safely perform activities of daily living through the use of deep learning. CV^{4}LV is an ongoing project and all implementations will be further developed and tested.

In the future development process, CV^{4}LV is expected to achieve a great level of accuracy and convenience for its users, in order to provide increased safety for people with LV in their daily lives, and help with the enhancement of their physical and psychological health.

Acknowledgements

This work is supported by the Multimedia and Visual Computing Lab (MMVC) at NYU Abu Dhabi.

6. References

[1] P. Y. Ramulu, C. Hochberg, E. A. Maul, E. S. Chan, L. Ferrucci, and D. S. Friedman, "Glaucomatous Visual Field Loss Associated with Less Travel from Home," *Optom. Vis. Sci.*, vol. 91, no. 2, pp. 187–193, Feb. 2014, doi: 10.1097/OPX.0000000000000139.

[2] S. Sengupta *et al.*, "Evaluation of real-world mobility in age-related macular degeneration," *BMC Ophthalmol.*, vol. 15, no. 1, p. 9, Dec. 2015, doi: 10.1186/1471-2415-15-9.

[3] S. W. van Landingham, J. R. Willis, S. Vitale, and P. Y. Ramulu, "Visual Field Loss and Accelerometer-Measured Physical Activity in the United States," *Ophthalmology*, vol. 119, no. 12, pp. 2486–2492, Dec. 2012, doi: 10.1016/j.ophtha.2012.06.034.

[4] C. E. Sherrod, S. Vitale, K. D. Frick, and P. Y. Ramulu, "Association of Vision Loss and Work Status in the United States," *JAMA Ophthalmol.*, vol. 132, no. 10, p. 1239, Oct. 2014, doi: 10.1001/jamaophthalmol.2014.2213.

[5] R. McKean-Cowdin, R. Varma, J. Wu, R. D. Hays, and S. P. Azen, "Severity of Visual Field Loss and Health-related Quality of Life," *Am. J. Ophthalmol.*, vol. 143, no. 6, pp. 1013–1023, Jun. 2007, doi: 10.1016/j.ajo.2007.02.022.

[6] R. Varma, J. Wu, K. Chong, S. P. Azen, and R. D. Hays, "Impact of Severity and Bilaterality of Visual Impairment on Health-Related Quality of Life," *Ophthalmology*, vol. 113, no. 10, pp. 1846–1853, Oct. 2006, doi: 10.1016/j.ophtha.2006.04.028.

[7] H. T. V. Vu, "Impact of unilateral and bilateral vision loss on quality of life," *Br. J. Ophthalmol.*, vol. 89, no. 3, pp. 360–363, Mar. 2005, doi: 10.1136/bjo.2004.047498.

[8] M. L. Popescu *et al.*, "Age-Related Eye Disease and Mobility Limitations in Older Adults," *Investig. Opthalmology Vis. Sci.*, vol. 52, no. 10, p. 7168, Sep. 2011, doi: 10.1167/iovs.11-7564.

[9] H. Court, G. McLean, B. Guthrie, S. W. Mercer, and D. J. Smith, "Visual impairment is associated with physical and mental comorbidities in older adults: a cross-sectional study," *BMC Med.*, vol. 12, no. 1, p. 181, Dec. 2014, doi: 10.1186/s12916-014-0181-7.

[10] "Vision impairment and blindness." https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (accessed Jul. 16, 2022).

[11] A. Gordois *et al.*, "An estimation of the worldwide economic and health burden of visual impairment," *Glob. Public Health*, vol. 7, no. 5, pp. 465–481, May 2012, doi: 10.1080/17441692.2011.634815.

[12] D. B. Rein, "The Economic Burden of Major Adult Visual Disorders in the United States," *Arch. Ophthalmol.*, vol. 124, no. 12, p. 1754, Dec. 2006, doi: 10.1001/archopht.124.12.1754.

[13] J. S. Wittenborn *et al.*, "The Economic Burden of Vision Loss and Eye Disorders among the United States Population Younger than 40 Years," *Ophthalmology*, vol. 120, no. 9, pp. 1728–1735, Sep. 2013, doi: 10.1016/j.ophtha.2013.01.068.

[14] P. Strong, "The history of the white cane," p. 3.

[15] F. J. Rowe, "Stroke survivors' views and experiences on impact of visual impairment," *Brain Behav.*, vol. 7, no. 9, p. e00778, Sep. 2017, doi: 10.1002/brb3.778.

[16] B. Kuriakose, R. Shrestha, and F. E. Sandnes, "Tools and Technologies for Blind and Visually Impaired Navigation Support: A Review," *IETE Tech. Rev.*, vol. 39, no. 1, pp. 3–18, Jan. 2022, doi: 10.1080/02564602.2020.1819893.

[17] N. G. Bourbakis and D. Kavraki, "An intelligent assistant for navigation of visually impaired people," in *Proceedings 2nd Annual IEEE International Symposium on Bioinformatics and Bioengineering (BIBE 2001)*, Bethesda, MD, USA, 2001, pp. 230–235. doi: 10.1109/BIBE.2001.974434.

[18] F. Hu, H. Tang, A. Tsema, and Z. Zhu, "Computer Vision for Sight," in *Computer Vision for Assistive Healthcare*, Elsevier, 2018, pp. 1–49. doi: 10.1016/B978-0-12-813445-0.00001-0.
[19] X. Zhang *et al.*, "A SLAM Based Semantic Indoor Navigation System for Visually Impaired Users," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2015, pp. 1458–1463. doi: 10.1109/SMC.2015.258.
[20] J. Xiao, S. L. Joseph, X. Zhang, B. Li, X. Li, and J. Zhang, "An Assistive Navigation Framework for the Visually Impaired," *IEEE Trans. Hum.-Mach. Syst.*, vol. 45, no. 5, pp. 635–640, Oct. 2015, doi: 10.1109/THMS.2014.2382570.
[21] S. Hillier, M. Immink, and D. Thewlis, "Assessing Proprioception: A Systematic Review of Possibilities," *Neurorehabil. Neural Repair*, vol. 29, no. 10, pp. 933–949, Nov. 2015, doi: 10.1177/1545968315573055.
[22] N. Patel, Y. Pan, F. Khorrami, J.-R. Rizzo, E. Wong, and Y. Fang, "Robust object detection and recognition for the visually impaired," presented at the the First International Workshop on Deep Learning for Pattern Recognition (DLPR2016), 2016.
[23] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network," 2018, doi: 10.48550/ARXIV.1808.03314.
[24] G. Lin, C. Shen, A. van dan Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," 2015, doi: 10.48550/ARXIV.1504.01013.
[25] A. Schwing and R. Urtasun, "Fully Connected Deep Structured Networks," Mar. 2015.
[26] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," 2016, doi: 10.48550/ARXIV.1612.01105.
[27] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," 2015, doi: 10.48550/ARXIV.1511.07122.
[28] A. Kundu, V. Vineet, and V. Koltun, "Feature Space Optimization for Semantic Video Segmentation," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 3168–3175. doi: 10.1109/CVPR.2016.345.
[29] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding," 2015, doi: 10.48550/ARXIV.1511.02680.
[30] H. De Ponteves, *AI Crash Course: a fun and hands-on introduction to machine learning, reinforcement learning, deep learning, and artificial intelligence with Python*. 2019.
[31] "Cooking Basics Mocap Animations in Animations - UE Marketplace," *Unreal Engine*. https://www.unrealengine.com/marketplace/en-US/product/cooking-basics-mocap-animations (accessed Jul. 17, 2022).
[32] "Mixamo." https://www.mixamo.com/#/ (accessed Jul. 17, 2022).
[33] U. Technologies, "Unity - Manual: Using Animation Events." https://docs.unity3d.com/Manual/script-AnimationWindowEvent.html (accessed Jul. 17, 2022).
[34] "Unity Asset Store - The Best Assets for Game Making." https://assetstore.unity.com/ (accessed Jul. 17, 2022).
[35] "Sketchfab - The best 3D viewer on the web," *Sketchfab*. https://sketchfab.com (accessed Jul. 17, 2022).
[36] "MediaPipe." https://mediapipe.dev/ (accessed Jul. 17, 2022).
[37] *ultralytics/yolov5*. Ultralytics, 2022. Accessed: Jul. 13, 2022. [Online]. Available: https://github.com/ultralytics/yolov5
[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," 2017, doi: 10.48550/ARXIV.1708.02002.
[39] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," 2015, doi: 10.48550/ARXIV.1512.02325.
[40] L. Tan, T. Huangfu, L. Wu, and W. Chen, "Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification," *BMC Med. Inform. Decis. Mak.*, vol. 21, no. 1, p. 324, Dec. 2021, doi: 10.1186/s12911-021-01691-8.
[41] N. Kiselov, hand-gesture-recognition-using-mediapipe. 2022. Accessed: Jul. 19, 2022. [Online]. Available: https://github.com/kinivi/hand-gesture-recognition-mediapipe

Terms & Conditions

1. The research work should not be published yet.
2. The prize should be spent on the research project such as conference expenses, publication fees, project resources, etc.
3. The research work should acknowledge the sponsor in its publication.
4. The applicant can't submit more than one research studies as main principal investigator.
5. The applicant should be female researcher in a university in UAE and GCC.
6. The applicant can be a female student, female faculty, or female research assistant.