ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

Semantic Communications for Unmanned Aerial Vehicles

Team #25

YU LIU (yul9@illinois.edu)
CHANG SU (changsu4@illinis.edu)
CHENHAO LI (cl89@illinis.edu)
TIANZE DU (tianzed2@illinis.edu)

Sponsor: Meng Zhang
TA: Xiaoyue Li

May 23, 2023

Acknowledgement

Thanks to Prof.Zhang Meng for providing fund and equipment support for the project, and for giving suggestions and pointing out the direction of our project in the weekly communication. Thanks to Prof. Mark Butala, Prof. Timothy Lee, Prof. Chenhui Shao and other professors for affirming our project and caring about our progress in the weekly group meeting. Among them, Prof. Timothy's suggestions helped us to finally determine the application scenario of the project, which had a significant impact on the promotion of the project schedule. Thanks to Teaching Assistant Li Xiaoyue and Teaching Assistant Wang Yi for patiently answering our questions and providing necessary help. Thanks to Liu Ziang, Zhang Bohao, Han Zifei and other students who volunteered to help us shoot the data set. Without their help, we could not improve the dataset and make the model achieve the current effect.

Abstract

Our project enables semantic communication on an unmanned aerial vehicle(UAV). Our UAV takes video using its camera and sends it to the Raspberry Pi on it. Raspberry Pi then extracts the semantic information of the video, and the semantic information will be sent to the computer to achieve communication. The goal of our semantic extraction is to get what the basketball player is doing at this point in the video. Finally, our semantic communication accuracy can reach more than 85%. It can be said that the main significance of our project is to prove that the emerging concept of semantic communication can be achieved.

Contents

1 Introduction

1.1 Purpose

Existing communication systems are mainly based on Shannon's information theory, and they are mostly developed to maximize data-oriented performance indicators, such as communication data rate, while ignoring content-related information [1]. In this case, people start to think about semantic communication. Semantic communication breaks through the traditional theoretical framework of Shannon's information theory improving transmission rate and accuracy, and transforming the content of communication into the meaning of information more valuable to human beings, thus fundamentally transforming the existing communication architecture into a more universal intelligent and human-oriented system [2].

The unmanned aerial vehicles (UAV) currently on the market can only fly and take pictures, and transmit the pictures or videos to the receiver using traditional means of communication [3]. But in many cases, the direct transmission of images from UAV is a huge waste of power and transmission. So, our goal is to develop a UAV technology that allows the UAV to transmit images and videos using semantic communication. More specifically, our UAV can build on the capabilities of existing UAV to process a sample of the image taken, extract specific semantics, and convey its symbolic representation to the target receiver. In this way, all we need to transmit is a sentence instead of a whole video. We hope this technique will be much faster than transmitting each complete image directly.

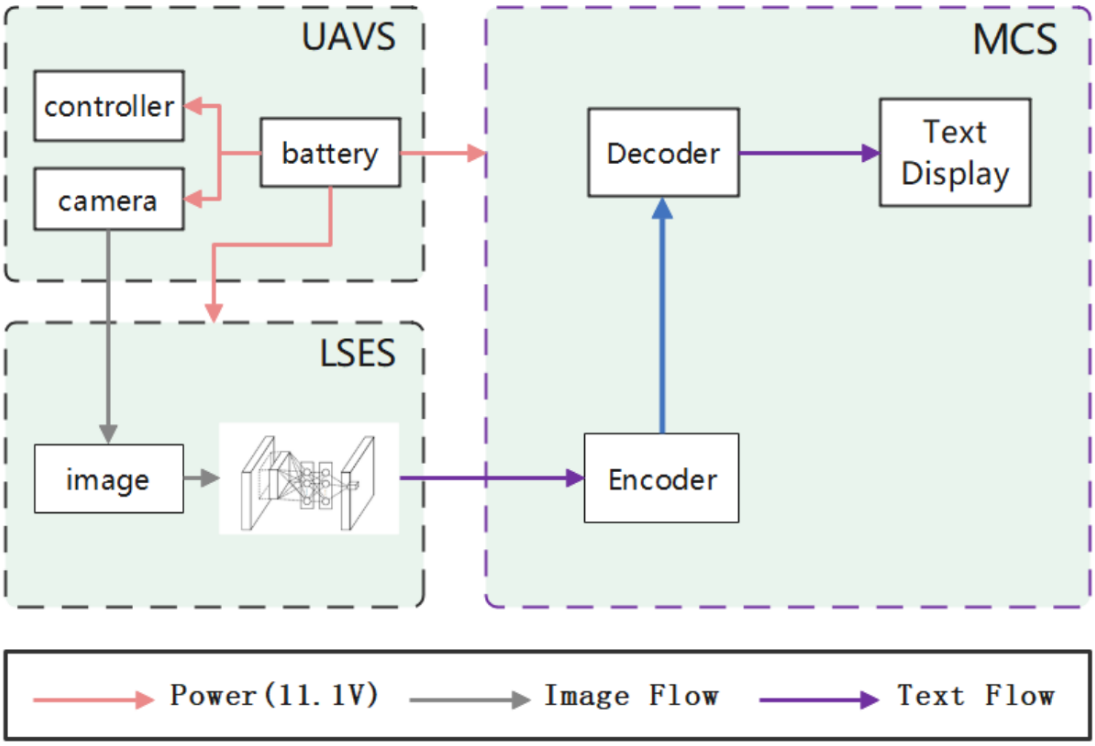

Figure 1: System Block Diagram

And the Block Diagram of our project is shown in Fig. 1.

1.2 Functionality

Our high-level project functionality includes:

- The UAV must be able to carry camera and the Raspberry Pi to move around. The UAV can hover up to at least 5 meters in the air and take videos with a resolution of 1920 by 1080. Through this, we can use UAV to achieve the basketball court recording.
- The Raspberry Pi must understand the videos from camera and extract useful semantic information. In our case this means what the basketball player is doing. The UAV should be able to predict the types of actions with over 70% accuracy. Through this, we can extract video semantic information. This will pave the way for the subsequent transmission of semantic information. Besides, the process time should be less than 8s for one video.
- The WIFI chip will must transmit semantic information to the receiver successfully. The computer will show the finally semantic information. The faster transmission speed than the traditional communication method and stronger anti-interference ability is highly appreciated. The time required to transfer each semantic information should be less than 1s.

1.3 Subsystem Overview

Figure 2 is the top-level diagram of our project.

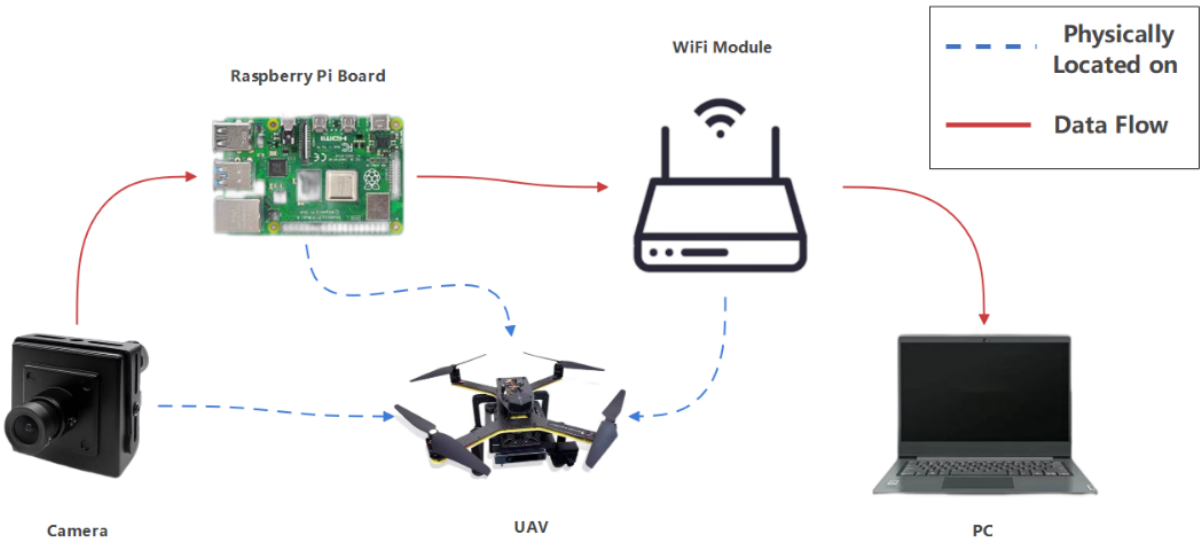

Figure 2: Top-level Diagram

1.3.1 UAV mechanical, balance and dynamic Subsystem (UAVS)

The UAV mechanical, balance and dynamic Subsystem includes a power module, a controller module, and a camera. The power module, which includes a distributor board and a lithium battery, is to supply stable voltage for every device and other subsystems on UAV. The controller module, PIXHAWK, will receive signal and control the four propellers, which are used to control the UAV's movement. The camera will take images, which will be used as input data for lighting semantic extraction subsystems (LSES).

1.3.2 Lighting Semantic Extraction Subsystem (LSES)

Given the video from camera of UAVS as input, LSES are designed to extract semantic information of the video and generate a descriptive sentence about basketball actions in the video. LSES uses Raspberry Pi on the UAV to extract and transmit semantic information. LSES could utilize advanced computer vision algorithms and machine learning techniques to accurately detect and identify the people and their actions in the video, providing valuable insights for a range of applications. Our project will focus on basketball game at the gym. The semantic information extracted by LSES will serve as the input of the mutual communication subsystem (MCS).

1.3.3 Mutual Communication Subsystem (MCS)

MCS accepts the text information extracted from images by LSES. This subsystem converts text into a bits signal and transmits it to another smart device over a physical channel. MCS consists of two separate parts: the transmitter on UAV and the receivers on smart devices, for example a computer, which are connected by the physical channel. The subsystem includes a encoder and a decoder. And finally, the text information will be displayed on a screen. The text has similar semantic information. Communication should be quick and without losing semantic information.

2 Design

2.1 Design Procedure

2.1.1 UAV mechanical, balance and dynamic Subsystem (UAVS)

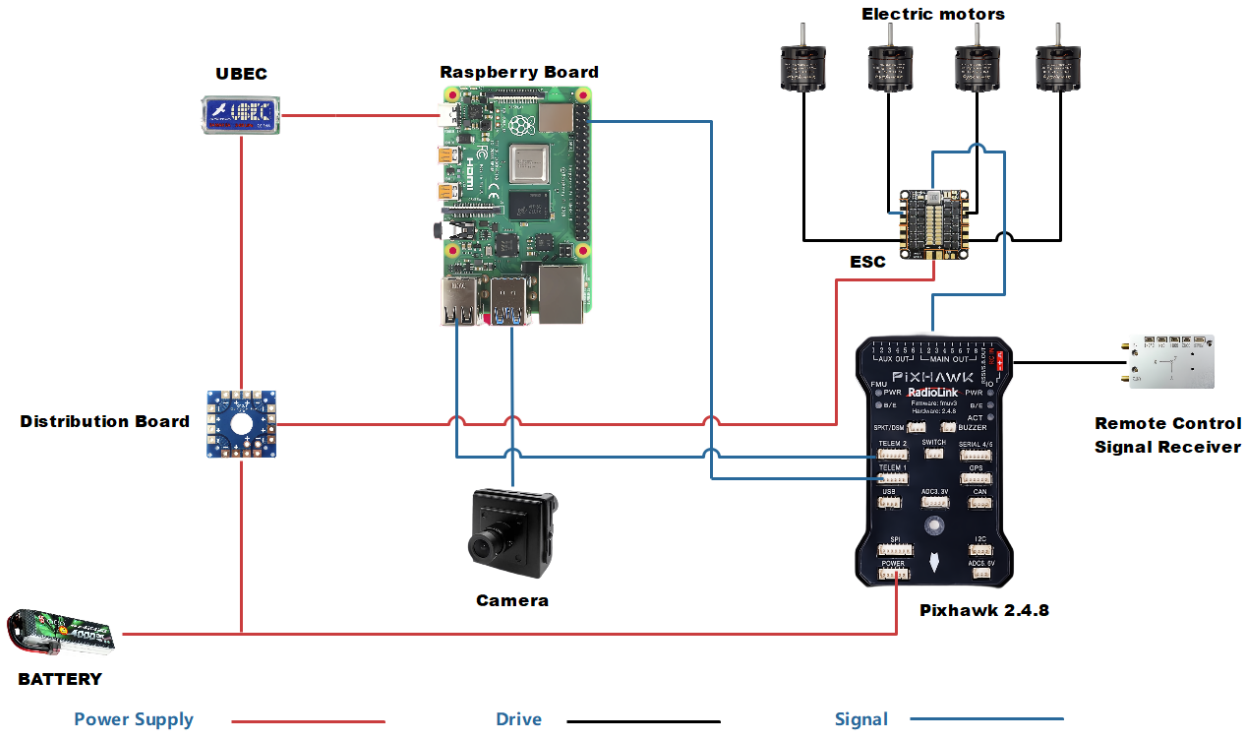

Figure 3: Structure of UAV

After considering the cost and project requirements, we decided to add the accessories shown in the Fig. 3: Firstly, we chose to use the 4G version of the Raspberry Pi board for the on-board computer because the 2G version could not meet the memory requirements of our entire model and the 8G version was too expensive (120% of the price of the 4G version and 150% of the 8G version based on the price of the 2G version). In comparison, the 4G version of the Raspberry Pi was able to meet the memory requirements of our model and was within our budget, so we chose this version of the Raspberry Pi as our on-board computer.

For the flight control we chose Pixhawk version 2.4.8. One of the reasons was that our UAV is a miniature UAV and Pixhawk was sufficient for the attitude control of the UAV and there were open source websites available to guide the assembly of the UAV, so we chose this flight control.

For the camera module, after testing the weight capacity of the drone and the input pixels of the project model, we finally chose a smaller monocular camera with up to 1080p pixels as our video capture tool. Again the cost was acceptable. For the battery, we decided to

use a stable and fast charging and discharging lithium battery, which is small enough to power the drone, as the endurance should not be shorter than fifteen minutes and the weight should not be too heavy.

2.1.2 Lighting Semantic Extraction Subsystems (LSES)

Since the design document, under the suggestion of the course instructor, we change our application scenario. We will choose a single kind of sport, basketball, and recognize some specific action in the game, for example, passing, laying up, shooting. This change makes our project much more challenging, since we need to analysis video instead of image to understand semantic meaning. We need to change our network. The network architecture we envisioned at the beginning was YOLO [4], but the YOLO network was not suitable for video motion detection.

Then we need to choose the appropriate semantic extraction network. Before I can do that, I need to prove the feasibility of semantic communication. The semantic channel capacity of a discrete memoryless channel [5] is expressed as

$$C_s = \sup_{p(Z|X)} \left\{ I(X;V) - H(Z \mid X) + \overline{H_S(V)} \right\} \tag{1}$$

For Eq. 1, $I(X;V)$ is the mutual information between the source, X , and the transmission task, V. Here $p(Z|X)$ is the conditional probabilistic distribution that refers to a semantic coding strategy with the source, X, encoded into its semantic representation, Z, and $H(Z|X)$ means the semantic ambiguity of the coding. $\overline{H_S(V)}$ is the average logical information of the received messages for the task V. Then here we can see that if we could make $\overline{H_S(V)}$ be bigger than $H(Z|X)$, the semantic channel capacity could be always bigger than 0. That means the receiver can handle the semantic ambiguity. For our design, this is easy to accomplish. As long as the accuracy of semantic information extraction model reaches more than 50%, it can be realized.

The network we need needs to have the following characteristics. First, the network needs to be suitable for motion detection. This is the key need to accomplish our semantic extraction. Secondly, the complexity of this network needs to be as small as possible. This is because eventually we need to do the target detection work on the Raspberry Pi. The CPU and GPU capabilities of the Raspberry Pi are relatively poor to complete the computing process for large models.

After searching the Internet and reading several related papers, I found two networks that meet the above requirements, namely 3D Convolutional Network[6] (C3D) and 2D Convolutional network [7]. The final semantic extraction model I choose is 3D Convolutional Networks. The 3D Convolutional Networks is better than 2D convolutional network because that whether 2D convolution is applied to a single image or multiple images, the output is a two-dimensional result, so the time series information will be lost when used for video recognition; 3D convolution solves this problem well, it preserves both temporal and spatial information [6].

In general, the advantage of this model is that it can extract information from the time sequence, so it performs well in the face of action recognition. And this model architecture is relatively simple, which is suitable to run on Raspberry Pi. And the running time of the model is relatively small.

In addition, I have done simulation validation experiments of this model on the computer. I ran the 3D Convolutional Network on a public dataset UCF-101 [8] and see if the accuracy of the model reaches a high level after multiple epochs. UCF101 is a motion recognition dataset of realistic action videos, providing 13,320 videos from 101 action categories.

Figure. 4 is the simulation result of 3D Convolutional Network. This shows the accuracy of the trained model in the training set and verification set. It can be found that after training several epochs, our model can reach 70% accuracy, which indicates that the 3D Convolutional Network is very suitable for action recognition.

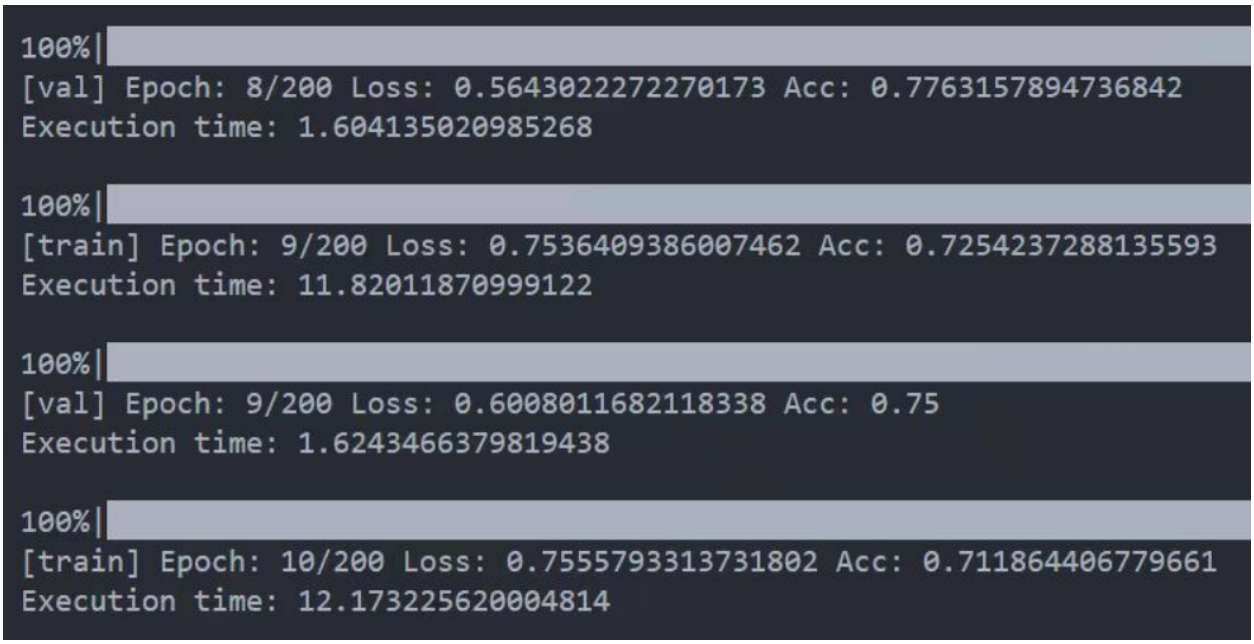

Figure 4: The result of LSES

More importantly, we cannot any suitable dataset. Since we are shooting basketball videos on the UAV, any basketball videos not shot from the top view or non-basketball videos shot from the top view cannot be used as our dataset. Thus, we need to collect and label the dataset by ourselves, which has huge amount of work. The details of the recording of dataset will be discussed in Section 2.2.2.

2.1.3 Mutual Communication Subsystem (MCS)

Driven by deep learning, natural language processing (NLP) has achieved great success in analyzing and understanding large volumes of linguistic text. We also try to use a

new perspective for the communication system from the semantic level, and proposes a semantic transmission system based on a deep learning network, which is called deep learning enabled semantic communication systems (DeepSC) [9], for text transmission. On the basis of transformer, the goal of DeepSC is to minimize semantic errors, restore sentence meaning, not traditional bit or symbol error communication. The advantage of DeepSC compared to other NLP model is that its architecture is not so complex and you can run it on Raspberry Pi.

2.2 Design Details

2.2.1 UAV mechanical, balance and dynamic Subsystem (UAVS)

Figure 5: Initial UAVS

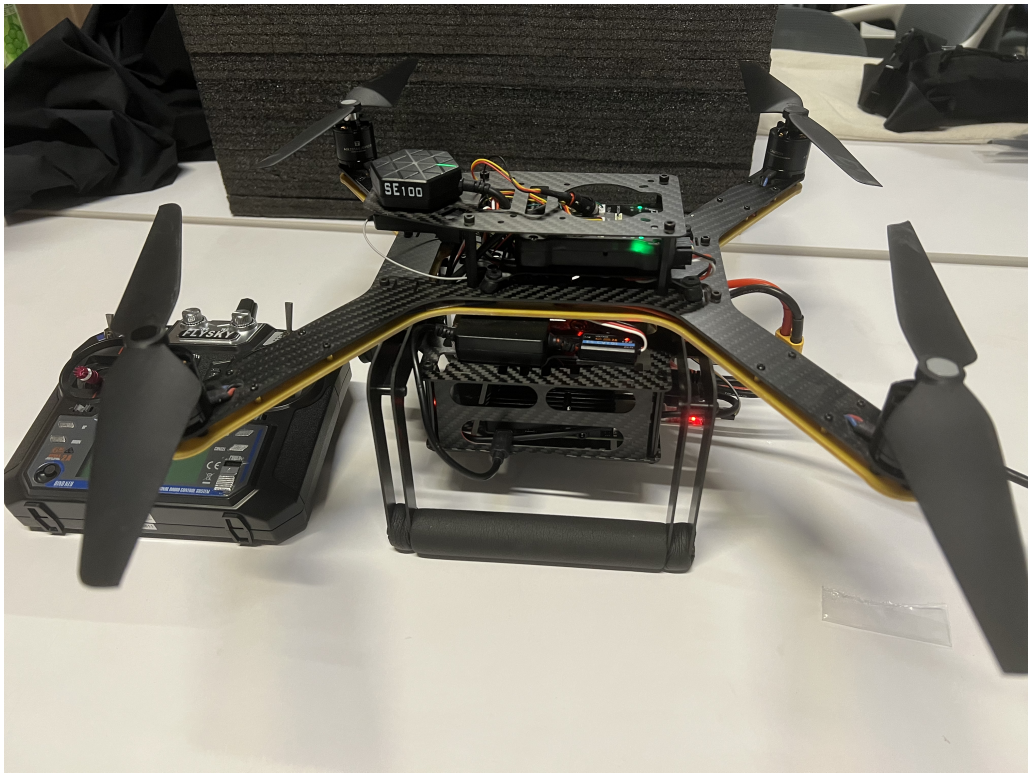

Figure 6: Modified UAVS

I will describe how the drone was assembled from its initial state of just motors and electrical speed controller on the Fig. 5 into a drone that meets the requirements of our project on the Fig. 6. As described in the alternative, we chose the right components for the project, taking into account cost and requirements. During the assembly process, we solved several problems: Firstly, the allocation of component positions. In accordance with the 445 battery safety specification document (plus references), we separated the battery from the on-board computer and flight controls as far as possible to ensure electrical safety. In addition, by consulting open source tutorials, we were able to correctly install the flight controls so that they could accurately control the attitude of our quadrotor drone. We have also installed a cooling fan on the Raspberry Pi to ensure that the on-board computer does not overheat and cause accidents. The second is the balance calibration. As the weight of the drone is not balanced in all directions, it is necessary to adjust the output of each motor by controlling the flight control through the remote control's trim button to enable the drone to fly in a balanced manner. The third issue is the camera lens angle. Through continuous test flights and test shots, the camera's pitch angle at 5m flight height was determined and fixed. Once the above major problems were solved, the UAV was created to meet the requirements of the project as shown in the diagram on the right.

2.2.2 Lighting Semantic Extraction Subsystems (LSES)

In this part, I will describe the design description and justification of LSES from two aspects: dataset and model.

Just like what I said above, we shot the dataset by ourselves. While one student takes pictures with a 3m high selfie stick, another student and I shoot, lay up and pass the basketball on the basketball court. And Fig. 7 is one frame of the dataset that we shot.

Figure 7: One frame from the dataset

We ended up taking more than 800 videos from different angles. Then we labeled the dataset into three categories: ”Shoot”, ”Lay Up” and ”Pass”. Also, as shown in Fig. 7, we shot the data set with someone else playing ball in the background. I think this can be interpreted by the model as noise from the data set, which helps to make our model more robust.

The architecture of 3D Convolutional Network is like this:

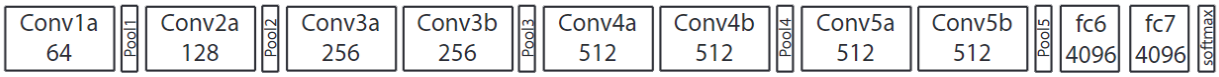

Figure 8: 3D Convolutional Network architecture[6]

From Fig. 8 we can see that the network takes the edited video clips as input, and the resolution of all videos is adjusted to 128*171. The video is also split into 16 frames that do not overlap each other and are used as network input. The network has 5 convolutional layers and 5 pooling layers, two fully connected layers and a softmax loss function layer to predict action labels. The number of filters in the five convolutional layers is 64, 128, 256, 256, 256, respectively. All convolutional layers have suitable padding and stride to ensure that the input to output of convolutional layers does not change in size. The

pooling layer is processed by 2*2*2 kernels to reduce the output by a factor of 8. The two fully connected layers have 1024 outputs, and then we train the network from scratch, here using the least gradient algorithm with a starting learning rate of 0.003.

For our project, I modified the fully connected layers output slim to the number of our labels. To be specific, that is 3. Besides, I changed the probability of the dropout from 0.5 to 0.3 to make the fitting of the model better. Since we would run the code on Raspberry Pi, we want to minimize the model size and computational time. Therefore, here I delete two layers of convolutional networks that do not change the output dimension. The reason is that the network accuracy will not decrease much after deleting, but the operation time can be greatly reduced. Here I did the testing, using both the modified network and the pre-modified network on the UCF-101 dataset [8]. The result is that after 20 epochs, the modified network accuracy rate is 76.5%, and the average running time per epoch is 10 minutes and 26 seconds; the pre-modified network accuracy rate is 80.2%, and the average running time per epoch is 13 minutes and 35 seconds. As you can see, although the accuracy doesn't decrease much, the computation time decreases a lot.

2.2.3 Mutual Communication Subsystem (MCS)

MCS accepts the text information extracted from images by LSES. MCS consists of two separate parts: the transmitter on UAV and the receivers on smart devices, which are connected by the physical channel.

The structure of DeepSC is shown in Fig. 9. The transmitter consists of two parts, a semantic encoder and a channel encoder, to extract semantic information from it and guarantee the successful transmission of semantic information on the physical channel. The receiver has corresponding decoders.

In the process of training, we uses two loss function parts, as shown in Eq. 2, the first part is cross-entropy loss to minimize the semantic difference between the input sentence and the output sentence by training the entire system. The second part is a loss function for mutual information, which maximizes the obtained data rate. A parameter λ ($0 \leq \lambda \leq 1$) is also be added as the related weight for both parts.

$$L_{total} = L_{CE}(s, \hat{s}) + \lambda L_{MI} \tag{2}$$

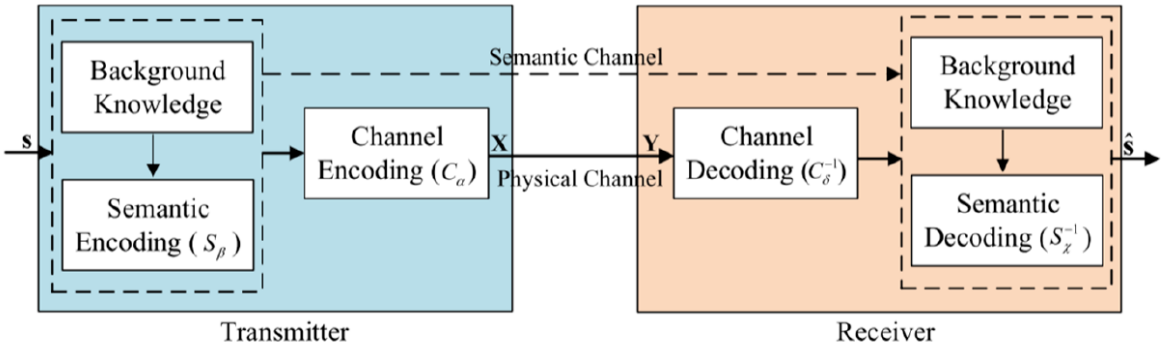

Figure 9: Structure of DeepSC network [9]

Additionally, we designed a graphical user interface (GUI) to serve as the endpoint for Mutual Communication System (MCS). The main interface is shown on Fig. 10.

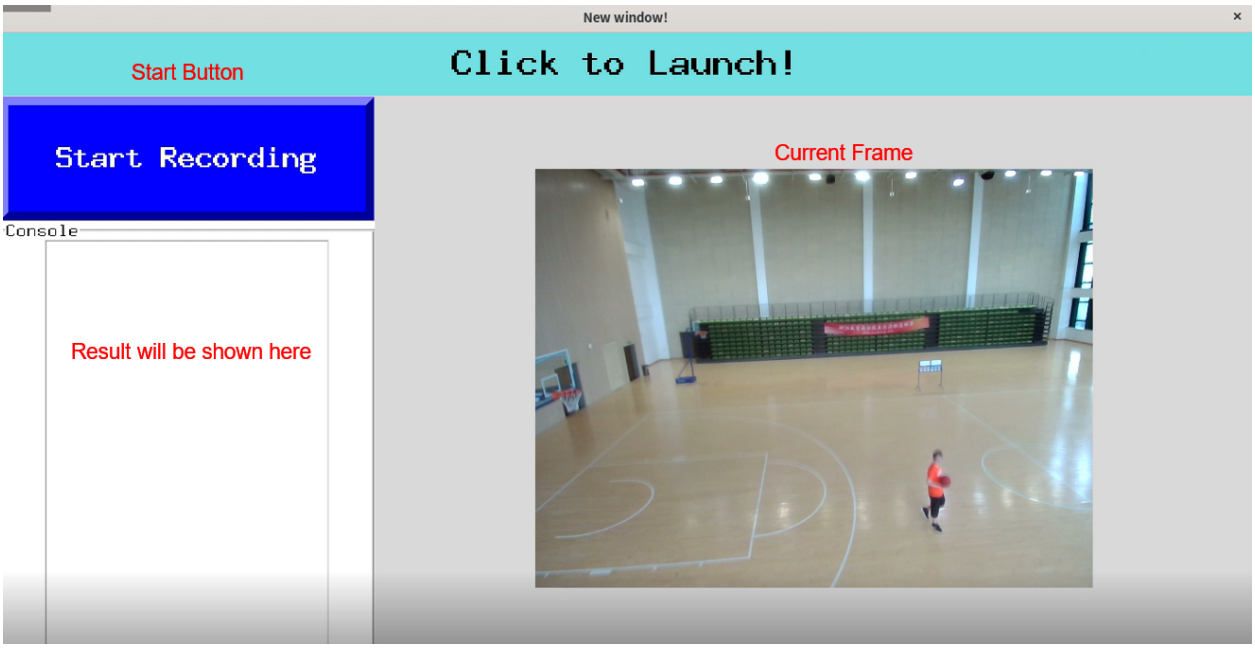

Figure 10: GUI Start Window

Our GUI is able to:

1. Display the current frame shot by the camera sensor embedded on the UAV mechanical, balance and dynamic Subsystem (UAVS). This enables real-time monitoring of the signal strength of our MCS as well as helping UAV manipulator adjust the orientation of our drone.

2. Asynchronously send "Start Recording" instruction to the Raspberry Pi on LSES. When LSES receives the instruction, it will start collecting the frames from the camera sensor

and further assemble them to form a 3s video clip. This video clip will be fed to our C3D model in LSES to inference the action of the player(s) on the basketball court.

3. Besides forming the video clip, LSES will also asynchronously send collected frames to the GUI, which enables our GUI to replay the recorded videos. This feature is helpful in verifying the performance of our model in LSES as well as the communication part of our MCS.

4. After inferencing, the action as well as its probability will be sent to our GUI, which will be properly formatted and displayed on the GUI console, as is shown in Fig. 11. If the confidence score is lower than a preset threshold, a warning will be fired to notify the user.

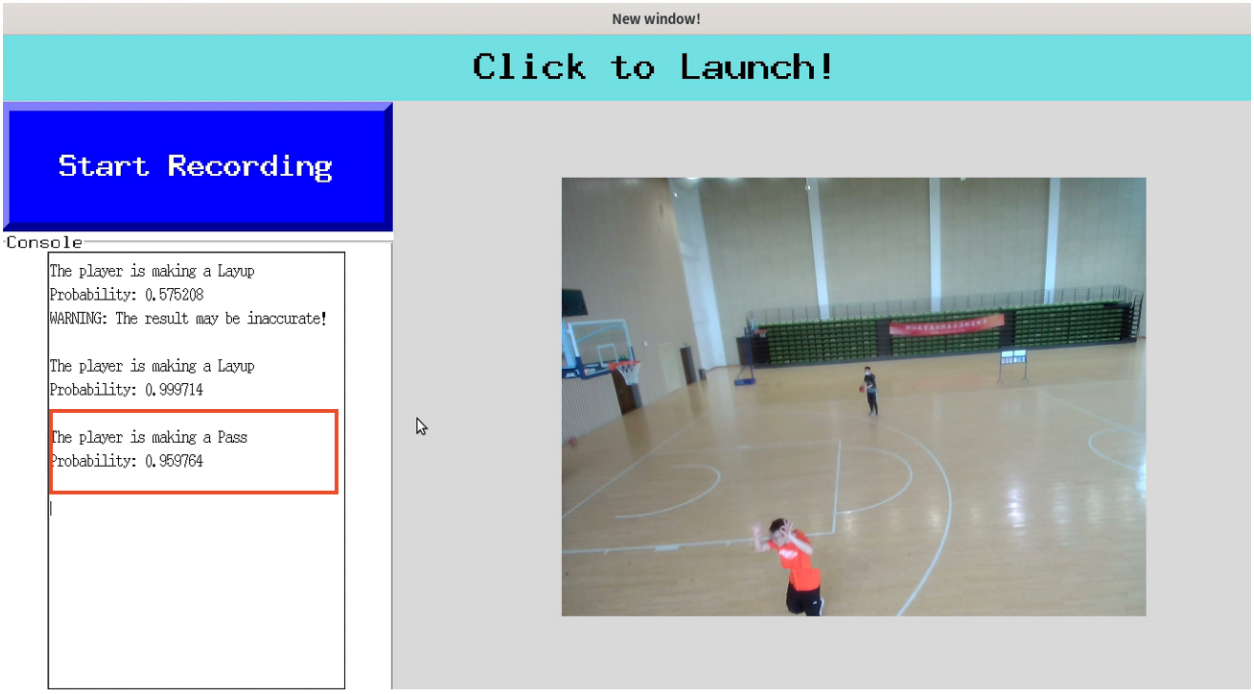

Figure 11: GUI Running Window

Our GUI is designed with the best practice of multi-process programming in mind. Specifically, we assigned one process for each independent task and constructed pipes for bidirectional message passing and dataflow. The event loop in GUI main process is responsible for launching all child processes, receiving messages and data from LSES and UAVs and handling requests from user. This parallelism can significantly improve user experience by providing minimal latency between user action and visual feedback.

3 Requirements and Verification

3.1 UAV mechanical, balance and dynamic Subsystem (UAVS)

3.1.1 Completeness of Requirements

Our power supply module can power both the drone motors and the on-board computer, the Raspberry Pi, and allows both parts to work properly at the same time. After assembly, the UAV can perform a series of flight manoeuvres such as take-off and landing, hovering and cruising, and fly in a smooth attitude, with all flight manoeuvres in 3D space. The camera can be used to shoot video at the specified altitude 5m, and the video is clear and shake-free. The athlete's movements can be recognised in the footage.

3.1.2 Appropriate Verification Procedures

The output voltage of the lithium battery was first measured using a meter at 11.8V, which was in line with the battery calibration voltage. Afterwards, the voltage supplied to the Raspberry Pi after ultra battery elimination circuit (UBEC) was measured to be 5V, again in accordance with the UBEC instructions, and the whole power supply module function was verified.

When testing the flight function of the UAV, the UAV was first made to take off normally and rise to a specified height and then adjust the throttle to hover. The hovering time of one minute did not reveal any obvious imbalance such as tilting of the UAV body, and the results did not change after several test flights. Afterwards, the UAV was flown in different directions in the horizontal and vertical planes respectively, and the aircraft was in a manoeuvrable state and the remote control response was timely. When landing, the aircraft could also be steered to land at the designated location. This verified that the UAV was in good flying condition.

The recorded video was put into the LSES and the system was able to recognise the movements made by the athlete in the video taken by the UAV.

3.1.3 Quantitative Results

For quantitative results, the voltage supplied to the Raspberry Pi after UBEC is 5V. The video shot by camera on the UAV was recorded at 30 fps in 1080p. The UAV can fly to an altitude of 5m and hover. All of these quantitative results are in line with our requirements.

3.2 Lighting Semantic Extraction Subsystems (LSES)

3.2.1 Completeness of Requirements

TThe requirements for LSES are that it could identify the actions of players on the basketball court with an accuracy of more than 80%. Besides, the running time of the model should also be small enough. Our expectation is to process one video in about 1 second on

computer and 8 seconds on Raspberry Pi. Finally, the LSES can fulfill these requirements well. More detailed requirements can be found in Appendix B.

3.2.2 Appropriate Verification Procedures

First, I did the training model part on the computer. The model divides the dataset taken by us into three parts: training set, test set and verification set. At this stage, I tried our best to adjust the model parameters to make the accuracy of the model on the training set as high as possible and pay attention to the accuracy of the model on the test dataset and the verification dataset.

Our initial design is that in addition to correctly identifying the three basketball actions, the LSES should also be able to determine whether the video is about basketball or not. In order to do this, in addition to the initial three labels: ”shoot”, ”lay up” and ”pass”, I added a label called ”Nothing”. The videos of this label means that no basketball-related actions are included.

During the model training phase, I added some non-basketball-related videos for training and marked them as ”Nothing”. After about 200 epochs of training, the accuracy of the model reached its highest value, and the accuracy of the model on the training and validation sets is shown in Fig. 12.

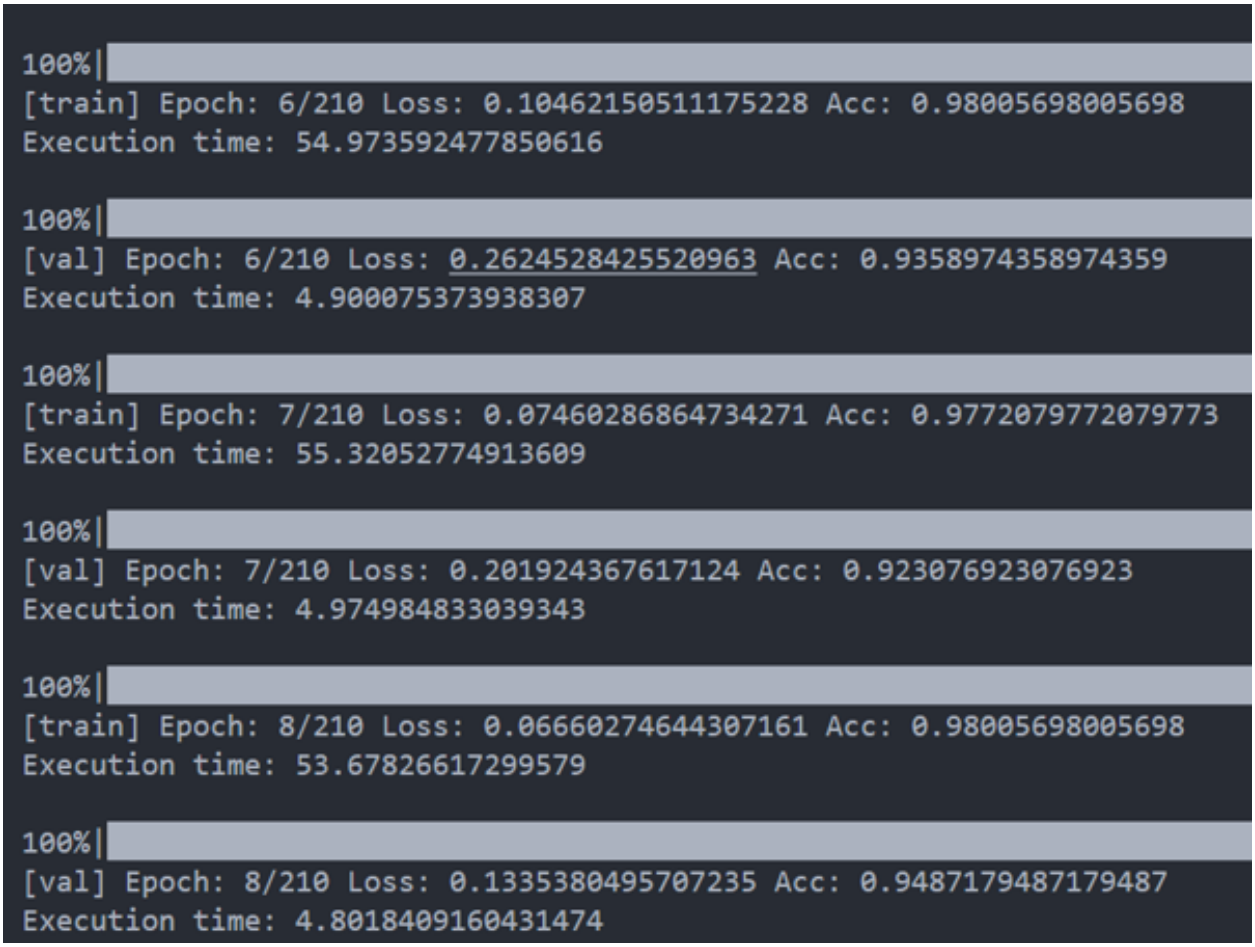

Figure 12: 4 labels C3D Model Training Result

It can be seen that the model achieves an accuracy of over 90% on both the training and validation sets. I then used the model for some tests, for example, I transmitted a video of a baseball to the model and the model output was "Nothing". This shows that our model works very well with videos that are not about basketball. However, during the testing phase I discovered a problem with our existing model, which is that some basketball-related videos are identified as "Nothing" when the video is captured by our UAV. I think that is because of the angle problem and UAV shaking when the video is filming, the video captured by the UAV differs so much from the dataset that the predictions would become inaccurate.

To solve this problem, we decided to remove the 4th label and use only the basketball dataset we shot for training. We then retrained our model, and this time the accuracy of the model improved further. After 200 epochs, the result of the model on the data set is shown in Fig. 13.

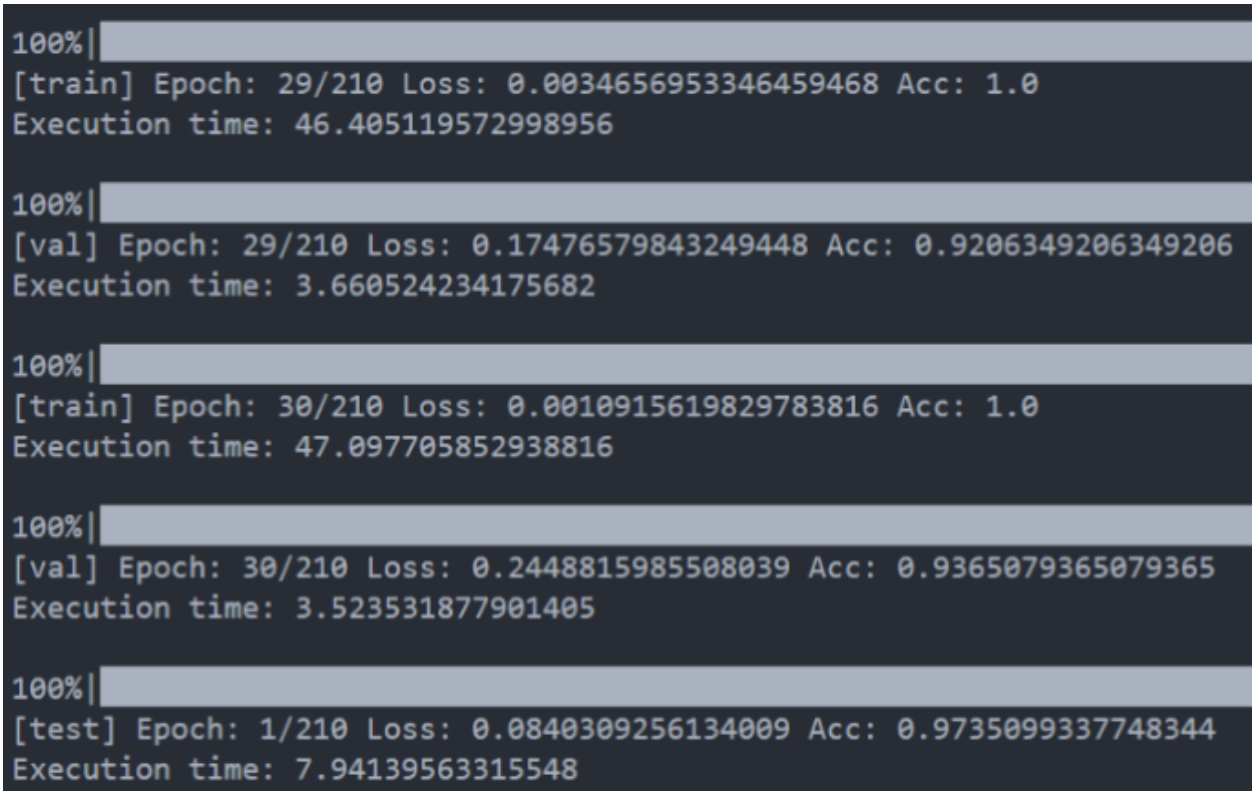

Figure 13: 3 labels C3D Model Training Result

From Fig. 13 you can see that the accuracy of our C3D network can reach 100% on the training set, and more than 90% on the verification set and test set. Subsequently, I wrote the inference script to test the validity of the new model. First I tried it on my own computer. The inference script could use our trained model to extract the semantic information from the video and recognize the types of basketball actions of the video.

Then I put the script on Raspberry Pi and used the UAV to shoot video for motion recognition. And I record the accuracy of our predictions. On Raspberry Pi, the prediction accuracy of the model is roughly the same as that of the computer, at more than 85%, as long as the UAV does not shake badly due to air currents, causing the picture to shake or be unclear.

3.2.3 Quantitative Results

First of all, Fig. 14 shows the change of the accuracy of our model in the training stage.

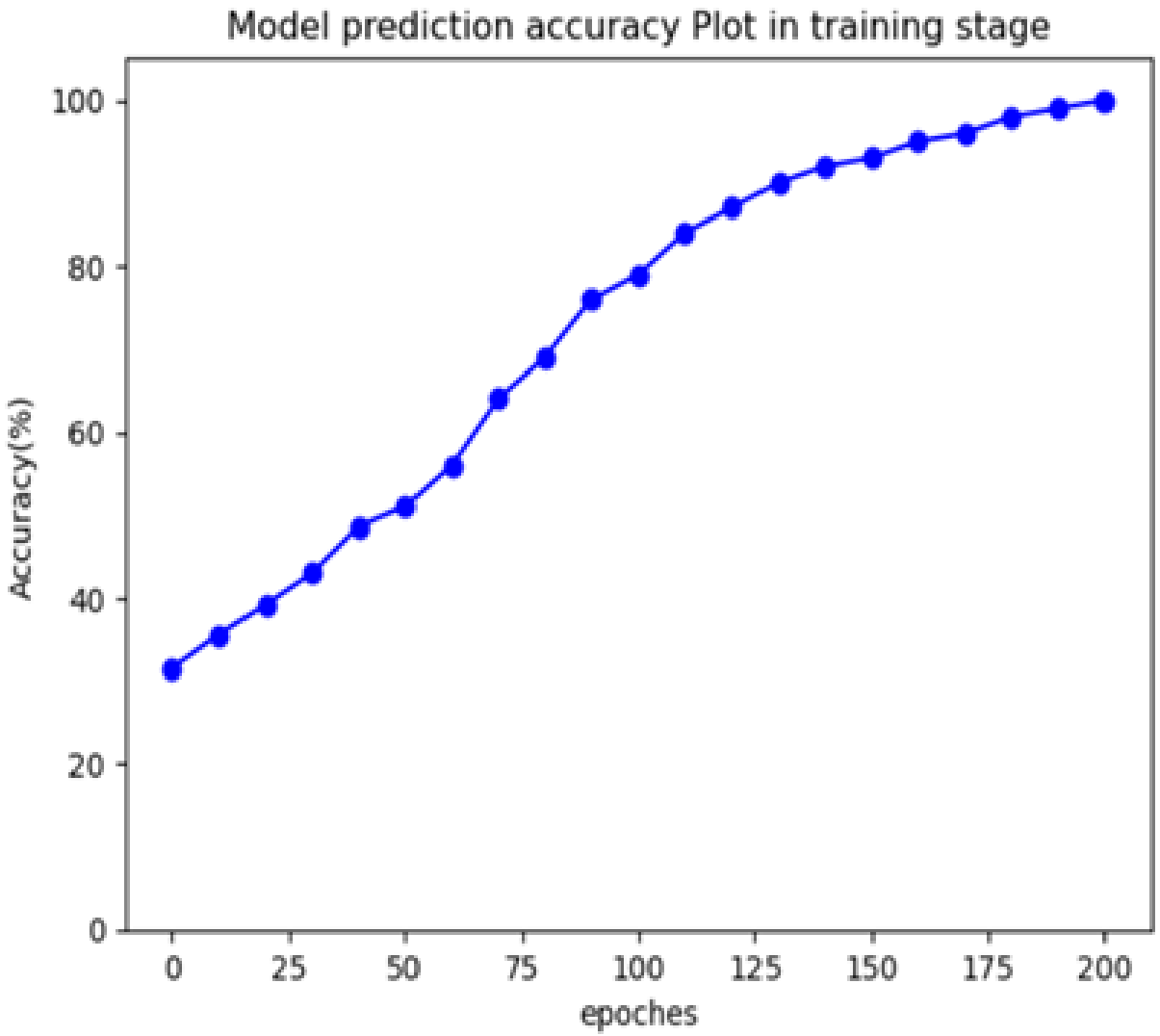

Figure 14: The Training Process of C3D Model

As can be seen from Fig. 14, at around 200 epochs, the prediction accuracy of our model gradually increases from 30% to 100%. This proves that our model performs very well in the training stage and the accuracy of the model can be maintained at a very high level in the end.

Figure 15 shows the accuracy of our model on the UAV when processing video from the UAV's camera.

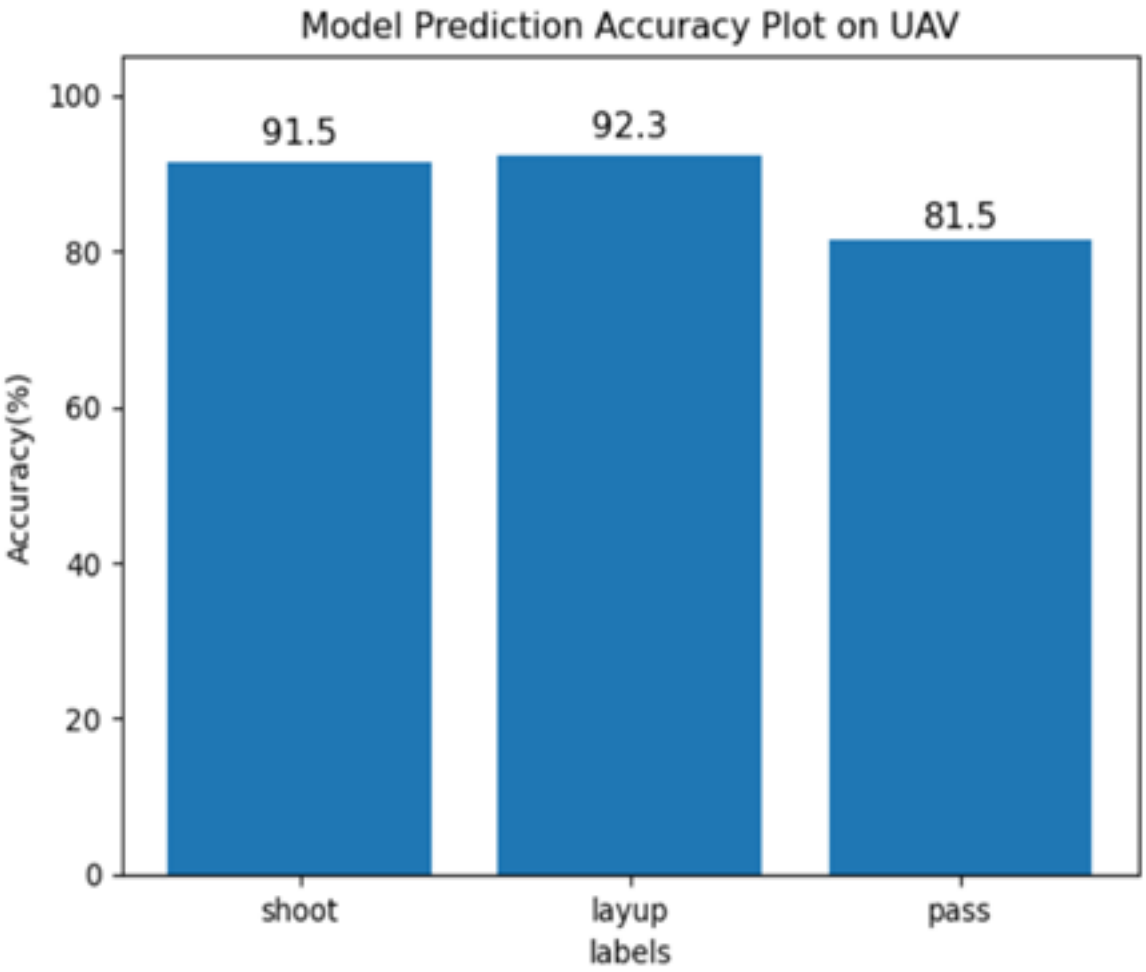

Figure 15: The test result on UAV

As can be seen from Fig. 15, for the two types of videos ”Shoot” and ”Lay up”, the prediction accuracy of the model can reach more than 90%; for the videos of ”Pass”, the prediction accuracy of the model can reach more than 80%; in general, the prediction accuracy of the model can reach more than 85%. This accuracy rate is higher than our accuracy requirement 80%.

On the NVIDIA GeForce RTX 3090 graphics card, our model running time was about 0.5 second per video. This processing speed is capable of processing videos from UAV in real time and predicting results. However, the Raspberry Pi GPU model is VideoCore VI. Due to the limited hardware conditions of Raspberry Pi, it takes about 7.5 seconds to run a script file on Raspberry Pi. This result is very much in line with our initial expectations and it meets our requirements.

3.3 Mutual Communication Subsystem (MCS)

3.3.1 Completeness of Requirements

The requirements for MCS are that it could transmit the text output of the LSES accurately and efficiently. For our finished subsystem, the transmission time is less than 0.5 second, and the semantic information during the transmission process does not change significantly. We use bilingual evaluation understudy score (BLEU), defined as Eq. 3, to evaluate the performance of our network. The result is very good.

$$logBLEU = min(1 - l_s/l_{\hat{s}}, 0) + \sum_{n=0}^{N} u_n logp_n \tag{3}$$

In Eq. 3, $l_s, l_{\hat{s}}$ are the length of input and output sentences. u_n is the weights of n-grams and p_n is the n-grams score.

3.3.2 Appropriate Verification Procedures

For network verification, we divide the dataset into three parts: training set, test set and verification set. In our test, we will use our test data, which will not be used in our training, to test the model's performance under different epoch. Meanwhile, I will test some sentences used in basketball games, for example, the player shoots the ball to test it on Raspberry Pi. The criteria is the value of the BLEU.

The GUI can verify the functionality of all our subsystems in a visual way. For UAVs, we can test the functionally of the hardware sensors and control system by looking at the frames on the GUI main window. If there exists no glitch between adjacent frames and frames shot by our camera sensor include players and basketball stand in the center, then the UAVs works as desired. For LSES, we can compare the inferenced result displaying on the GUI console with the recorded video clip to assert the accuracy of our C3D model. If the accuracy for each action is greater than 0.9, then we can assert the functioning of LSES.

For MCS, we can verify the efficiency of data transmission by measuring the latencies during transmissions. Firstly, we measure the time difference between a user clicks "start" and the inferenced result displaying on the GUI console (denoted as L), which can be done by an accurate timer. The time difference should be less than 5s. Secondly, we measure the time difference between MCS sending a "start" instruction and LSES receives it (denoted as L_{send}). This can be measured by taking the difference between the frame id of the first frame in the recorded video clip and the recorded frame id when user presses the start button, and then multiplying by $1/fps$, where fps is the frame rate of the video. We do several experiments and take the average among these calculated numbers.

3.3.3 Quantitative Results

The final result of this subsystem is not bad. After training, the BLEU of our DeepSC network can reach 90% on the test dataset. The result for our test under different signal noise

ratios (SNR) is shown in Fig. 16. We noticed that the BLEU score increases as the SNR increases and the noise decreases, and it exceeds 90% when the SNR reaches 18dB. This is already a noisy environment in communication, compared to 70dB under normal conditions, but our network still has a good performance. The result we reproduced has certain differences between this in paper [9], but the difference is not large. The reproduction is successful.

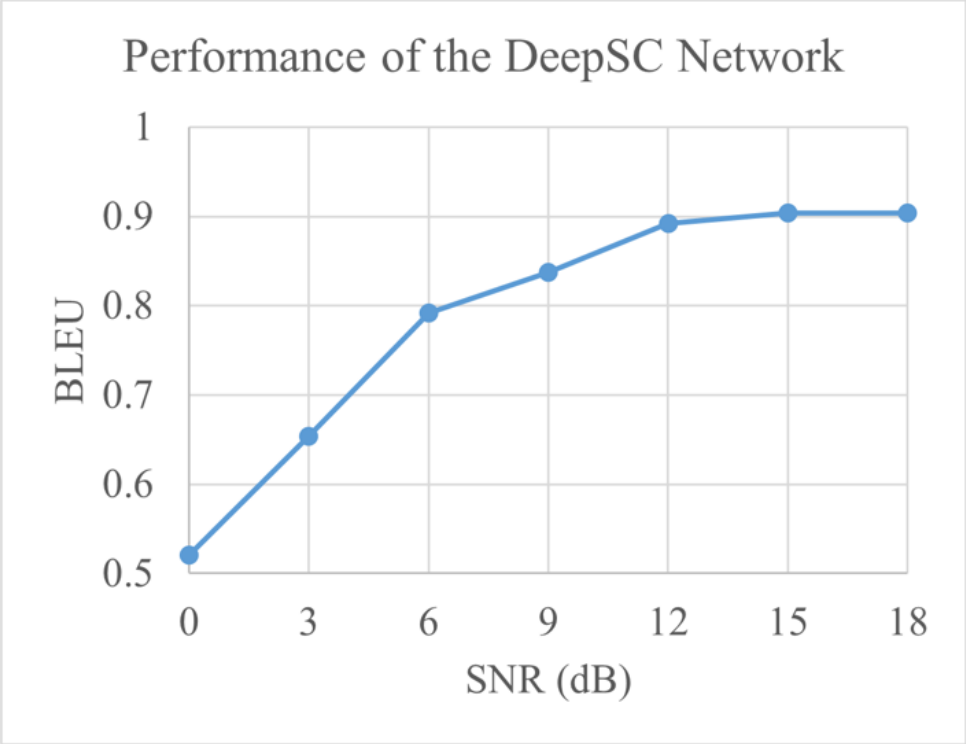

Figure 16: DeepSC performance

And using GUI, we measure the latency between LSES sending the result and MCS receives it (denoted as L_{recv}), which could be calculated by the Eq. 4:

$$L_{\text{recv}} = L - L_{\text{send}} - L_{\text{inf}} \tag{4}$$

In Eq. 4, L_{inf} is the inference time on LSES and could be accurately measured by using the 'time' module in python library. After calculation, our latency is 0.42s, which is less than 0.5s. It is a desirable latency for transmitting one sentence with about 10 words.

4 Cost and Schedule

4.1 Cost

First, for our labor cost, we assume everybody's hourly wage is ¥100/hour, and we need to work for 10 hours/week for all four people. And we need to do this for the following 10 weeks this semester. So for this part, our fixed development cost is :

$$4 \cdot \frac{20CNY}{hr} \cdot \frac{10hr}{wk} \cdot 10wks \cdot 2.5 = 20000CNY$$

Then, since only one person is needed to operate the drone, we don't need a lot of bulk. For the parts and manufacturing prototype costs, it is estimated as ¥2946 which is shown in Tab. 1:

Table 1: Cost Table

Part	Vendor	Cost (prototype)(unit:¥)	Cost (bulk)(unit:¥)
Professional aerial photography UAV (CK10pro)	Taobao	1888	50
8GB Raspberry Pi (4B; generic)	Taobao	728	20
200W pixels Monocular Camera (Reshi Technology)	Taobao	150	20
WiFi module (Small R Technology; MT7620)	Taobao	50	40
Total	Taobao	2816	130

Then we add two parts together, our total development cost should be ¥20000+¥2946=¥22946.

4.2 Schedule

Figure 17 shows the schedule of the work done throughout the semester by our four teammates.

Week	Yu Liu	Chenhao Li	Chang Su	Tianze Du
3/20/23	Write Design Document 2.1 and 2.2	Write Design Document 2.3 and 2.4	Write Design Document Part 1 and 3	Write Design Document Part 4
3/27/23	Learn the use of the Raspberry Pi	Look for object detection algorithms	Find the appropriate dataset	Purchase the required parts
4/3/23	Simple programming on the Raspberry Pi	Run through object detection algorithms on the computer	Find the appropriate semantic segmentation algorithm	Add parts on the UAV
4/10/23	Run object detection algorithms on the Raspberry Pi	Find the right means of communication for UAV	Run the semantic segmentation algorithm on the computer	Design and construction of UAV balancing systems
4/17/23	Enable communication between drones and other smart devices	Enable communication between drones and other smart devices	Implement the semantic segmentation algorithm on the Raspberry Pi	Design and construction of drone power systems
4/24/23	Carry out the final inspection of the part for which he is responsible	Carry out the final inspection of the part for which he is responsible	Carry out the final inspection of the part for which he is responsible	Carry out the final inspection of the part for which he is responsible
5/1/23	Test flights of UAV, detection and analysis of errors	Test flights of UAV, detection and analysis of errors	Test flights of UAV, detection and analysis of errors	Test flights of UAV, detection and analysis of errors
5/8/23	Prepare for Mock demo	Prepare for Mock demo	Write the Final Report draft	Write the Final Report draft
5/15/23	Detect the overall effectiveness of the project	Detect Lighting Semantic Extraction Subsystems	Detect Lighting Semantic Extraction Subsystems	Detect UAV mechanical, balance and dynamic Subsystem
5/22/23	Write Final Report	Write Final Report	Prepare for Final Presentation	Prepare for Functionality Demonstration Video

Figure 17: Schedule for teammates

5 Conclusion

5.1 Accomplishments

In the end, our project successfully fulfilled all of the high level requirements. Our UAV was successfully equipped with a camera and Raspberry Pi, and still maintained its balance and could fly up to 5m in the air for video recording. The Raspberry Pi on the UAV can extract semantics from basketball videos captured by cameras and identify the movement categories of basketball players in the videos. Semantic extraction process takes less than 8 seconds. And the recognition accuracy rate is more than 85%. In addition, UAV can transmit semantic information to the computer and deliver it to the user through our well-designed GUI interface. The complete transmission time is controlled within 1 second.

5.2 Uncertainties

From Section 3.2.3 we can see that although the training accuracy of our model is almost 100%, the final accuracy on UAV is only about 85%. I think this is because only a few of our teammates participated in the shooting of our dataset, and the diversity and number of the data set was insufficient. In addition, video from a mobile phone is still different from video shooting by a UAV in the air. This equates to a lack of training set and the fact that the training set and the test set are not the same, so it is not unusual for the model to be less effective on the test set than on the training set.

As for the reason why the prediction accuracy of ”Pass” category is only 81.5%, which is slightly lower than that of the other two categories, I think that is because in the training stage, the number of videos labeled as ”Pass” is lower than that of the other two categories of videos, which leads to worse prediction results of the model in this category. This problem can be solved by photographing more ”Pass” data sets and training the model.

5.3 Future Work

- Dataset. Our dataset is mostly shot by our teammates playing ball, so the diversity of the dataset is not good enough. If we can hire more people to take part in the shooting of our dataset, the effect of our model will be better.
- GPU. The GPU of Raspberry Pi is VideoCore VI. If we can change it to a stronger GPU, the computing speed will be improved.
- Algorithm model. We were limited by the Raspberry Pi hardware and the size of the model had to go small. If we have a better GPU on Raspberry Pi then we can use a bigger model and a better algorithm.

5.4 Ethical Considerations

A number of potential ethical and safety issues had to be considered in our project. First of all, both the UAV and the Raspberry Pi board need to be powered by batteries, which cannot be replaced by other power sources. So the stability and safety of the batteries are an important part of ensuring the success of the project. According to the ECE445 battery safety document [10], we will understand the battery specifications before installing the battery, test the battery circuit packaging, charging and discharging, and the operating temperature, and pay attention to the isolation from other work areas such as the transmission module and the Raspberry Pi board to avoid impact.

In addition to the manipulation method, when choosing the flight area and time period, we also need to ensure that the drone will not cause threats and interference, and avoid flying in densely populated areas and flight-restricted areas. When flying, we need to comply with local regulations and rules to ensure that my project is operating within legal limits. Because our tests and demonstrations are conducted on the campus of Zhejiang University, according to the school's guidelines [11], we need to apply to the school in advance before the drone flight.

According to the Institute of Electrical and Electronics Engineers(IEEE) Code of Ethics 1 "to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment" [12], we promise that the data set used in the project will seek the permission of the owner, and the collected images will also be cleared after use in order to protect information security. The final result of the project cannot be used in any scenario that infringes on public information and privacy.

References

[1] X. Luo, H.-H. Chen, and Q. Guo, "Semantic communications: Overview, open issues, and future research directions," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 210–219, 2022.

[2] T. Han, Q. Yang, Z. Shi, S. He, and Z. Zhang, "Semantic-preserved communication system for highly efficient speech transmission," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 245–259, 2022.

[3] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Communications magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, 2016. arXiv: 1506.02640 [cs.CV].

[5] Z. Qin, X. Tao, J. Lu, and G. Y. Li, "Semantic communications: Principles and challenges," *arXiv preprint arXiv:2201.01389*, 2021.

[6] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[7] J. Bouvrie, "Notes on convolutional neural networks," 2006.

[8] K. Soomro, A. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, Dec. 2012.

[9] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.

[10] U. of Illinois ECE445 Course Staff. "Safe practice for lead acid and lithium batteries." (2016), [Online]. Available: https://courses.engr.illinois.edu/ece445zjui/documents/GeneralBatterySafety.pdf (visited on 03/21/2023).

[11] Z. University. "Drone school flight." (2020), [Online]. Available: http://xwfw.zju.edu.cn/wsbsdt.php?cmd=sx_content&blsxm=347300&dxlb=&fwxz=&bjlx=&fwms=&sldz=&fwzt=&zuixin_tuijian=&orderby=&jianhua_flag=&shoulibm=&one_run=&keyword=%E6%97%A0%E4%BA%BA%E6%9C%BA&p=1 (visited on 03/23/2023).

[12] IEEE. "Ieee code of ethics." (2020), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/07/2023).

Appendix A Standard Abbreviations

Unit or Term	Symbol or Abbreviation
unmanned aerial vehicles	UAV
UAV mechanical, balance and dynamic Subsystem	UAVS
Lighting Semantic Extraction Subsystem	LSES
Mutual Communication Subsystem	MCS
3D convolutional network	C3D
natural language processing	NLP
graphical user interface	GUI
signal noise ratios	SNR
Institute of Electrical and Electronics Engineer	IEEE
ultra battery elimination circuit	UBEC
bilingual evaluation understudy	BLEU
deep learning enabled semantic communication systems	DeepSC

Appendix B Requirements & Verification Table

Below is the Requirements & Verification Table for our project.

Subsystem	Requirement	Verification
Unmanned Aerial Vehicles mechanical, balance and dynamic Subsystem (UAVS)	The ultra battery elimination circuit (UBEC) could provide 5.8V from an 11.1V source.	We will measure the output voltage from UBEC using an oscilloscope, ensuring that the output voltage stays within 5% of 5.8V.
	UAV can still fly smoothly and quickly after adding devices	We will do some fly tests. Such as let the UAV quickly rise into the air and hover. The success criterion is that the UAV can move and hover smoothly in three dimensions.
	UAV must be able to hover up to at least 5 meters in the air and take clear videos	We will do flying tests and take some simple videos. Such as let the drone fly to an altitude of 5m and take pictures. The success criterion is that it can hover up steadily at the required height, and the picture is clear and not blurred by movement.
Lighting Semantic Extraction Subsystem (LSES)	LSES needs to understand video information about player's actions on video at a high accuracy of at least 70%.	We will run our own LSES model on the computer first and the expected accuracy is above 70%.
	LSES should be small but efficient, which can be carried on Raspberry PI and extract information quickly for about totally 8s for each video.	The test on UAV will be given to check data transmission between subsystems and to test the accuracy in real scenarios. The accuracy is expected to be at least 70%. The detection is expected to finish in average of 8s for one video.
Mutual Communication Subsystem (MCS)	MCS needs to successfully transmit text information without any error from UAV to smart devices and display it on a screen.	The first test is to transmit some sample words from one smart device to another by the required time. Also, another test on UAV will be given to check data transmission between subsystems.
	The time for transmission should be less than 0.5s for each sentence with about 10 words	The delay due to the transmission should be less than 0.5s.