Critical Design Report
by FELEREST

Contents

1 Executive Summary 3

2 Introduction 4

3 Overall System Description 5
a) Description . . . 5
b) Requirements . . . 6

4 System Architecture 9
a) Server - Client Model . . . 9
1. Reasons and Requirements . . . 9
2. Software Design . . . 10
b) Tests and Results, Future Test Plans . . . 12
c) User Interface . . . 12

5 Image Processing - Computer Vision 17
a) Classification . . . 17
1. Subsystem Description . . . 17
2. Explanation of CNN for the Interested Reader . . . 19
3. Tests and Results, Future Test Plans . . . 21
4. Requirements . . . 26
5. Modifications . . . 27
6. Compatibility Analysis . . . 27
b) Identification . . . 27
1. Subsytem Description - Requirements . . . 27
2. Explanation and Comparison of Methodology . . . 27
3. Data Collection and Sources . . . 28
4. Feature Descriptor Implementation Details . . . 30
5. Tests and Results . . . 31

6 Electrical Design 39
a) Subsystem Description . . . 39
1. Supply Unit . . . 40
2. Raspberry Pi and Arduino . . . 42
3. Servo Motor . . . 42
4. Measuring the Food Level . . . 42
5. Measuring the Weight . . . 43
b) Requirements . . . 43
c) Tests and Results, Future Test Plans . . . 44

1. Test Procedure . . . 44
2. Test Results . . . 44
d) Modifications . . . 46
e) Compatibility Analysis . . . 47

7 Mechanical Design **48**
a) Subsystem Description . . . 48
1. Connection between electronic devices . . . 48
2. Food Mechanism . . . 51
b) Requirements . . . 53
c) Tests and Results, Future Test Plans . . . 53
1. Food Mechanism . . . 53
2. Exterior Design . . . 56
d) Modifications . . . 57
1. Food Mechanism . . . 57
e) Compatibility Analysis of Mechanics . . . 57

8 Power Analysis **58**

9 Cost Analysis **60**

10 Deliverables **62**

11 Conclusion **63**

Appendices **65**

- Modification issues are not systematically reported.

- compliance with requirements?

1 Executive Summary

?

Cats and humans are historical friends. It is estimated that there exists 200-600 million cats (as pets) around the world [3]. 80 million cats alone are owned by humans living in USA [6]. Although the existence of many cats, the current feeding diets are far from enough. They are either fed only when their owners are around or they are faced with the challenge of finding food for themselves in the case of street cats. These conditions are not hygienic for either humans or cats. The mission of Felerest is to create a safe and clean environment for the closest companions throughout history: cats and people.

In order to solve this problem Felerest proposes an autonomous cat feeding system. This system aims to feed cats without any human attraction other than charging the system and inserting food into the reservoir. This system will be able to adjust itself to the eating habits for every single cat and will give the specified amount of food among many other features.

2 Introduction

On the challenging path of forming a novel engineering application, the Felerest team has come closer to achieving the final version of the Smart Connected Cat Feeding & Monitoring System. This report examines each system, sub-systems and the components forming these sub-systems with a top to down structure. In the end the reader will have knowledge about: test results of these parts and their performances, what the final version will be, any changes in the design and its reasoning. The mentioned parts will be indicated through flow charts, block diagrams, and 3D drawings where applicable. The examination of each part will be done in detail mentioning technical details, work principles, proof that requirements are satisfied, discussions on compatibility, design reasoning, any trade-offs and their justifications, problems encountered and how our exceptional team of engineers have overcome it. These sum up the theoretical backbone of the project, how this knowledge has been put into application and problems faced in implementation, the unique and creative solutions overcoming these problems will additionally be present in this report.

For the report to be comprehensive to the reader the report will be in top-down structure as mentioned. To assure this, an overall view of the system will be given as a starter. This will then be followed with sub-system explanations. The aforementioned details given will be inclusive for both the overall and sub-system explanations. Within these sub-system explanations each component forming it will also be explained in detail. In the latter parts, a general compatibility and cost analysis will be done. In this compatibility analysis the reader will be introduced to the signal interfaces and their descriptions. The server-client model can also be included as a part of this interface although it is mentioned as a sub-system under the caption 'System Architecture'. The cost analysis will be broken down in detail with explanations. The final element of the report will be to discuss the power distribution in the overall system. The calculations done in power usage will be a proof of the satisfaction of a specific requirement which is the battery usage of at least 5 hours. Our highly talented team has exceeded this limit to provide our customers for a longer duration of functioning. Last but not least, the recent and updated Gantt Chart will be provided in the Appendix section.

To sum, this report concludes the possible design ideas presented in the previous report (Conceptual Design Report). The reader will be confident why the final design was preferred over other choices, why adjustments have been made if there are any and will have an intuition of how the project will function and look like in the end. Since the current status of the project is almost ready for mass production, the visualization of many sub-systems and the overall design (outer shell) of the system is ready for view and will be supported with photos throughout this report.

3 Overall System Description

Poor overall sylem description

a) Description

This section does not clearly describe what this system does!

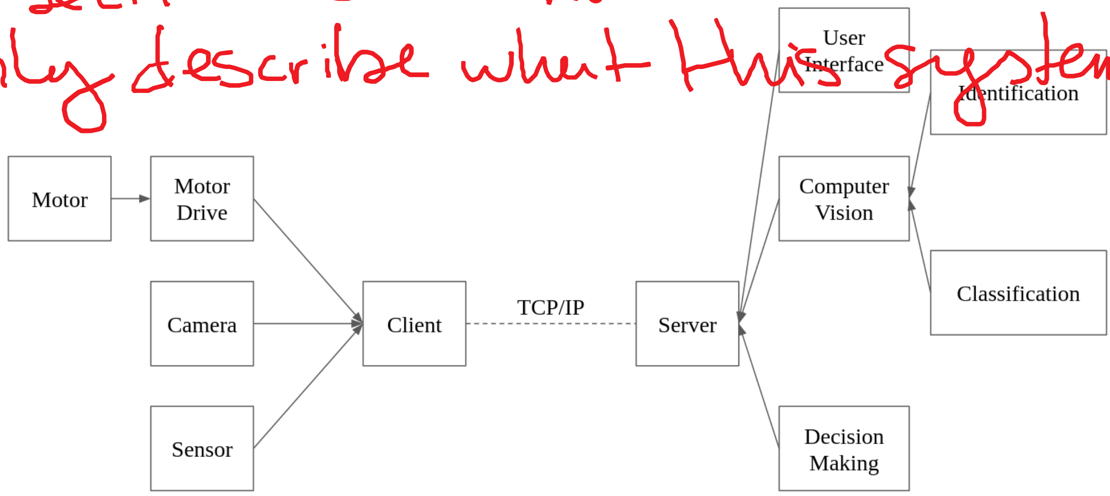


Figure 1: Overall System Parts

This report consists of top down approach which explains the overall description, then diving into smaller parts and sub-parts of them. Overall system ~~description~~ is best demonstrated with the figure in 1. There are mainly 2 parts, server and client respectively, in addition to the connection between them. Server client model will be investigated deeper in the a) section with justifications. For now, it is enough to know that the system has 2 parts connected by a computer network implemented over the internet, and these parts have special sub-modules and missions assigned them. System details will be explained in the next sections.

misleading

System design is made on requirements and best fits to these requirements. Customer wants, security, safety, as well as technical musts and edge conditions are determined and solutions are developed. A list of requirements for the project is given as below in addition to requirements developed in each component which are specified in the section of the corresponding component.

- The product should give response in a few seconds so as not to allow a cat to escape
- The product should work for long hours on battery ?

Before going into parts in detail, it is important to make some points clear about the rules and techniques used in the system. Black box approach is the first property each module and team member understands deeply. Each module is seen as black box and each of them can be separated without any loss in functionality. Therefore, any addition from the beginning or a problem that may arise can be solved locally without changing the overall structure of the product and only having a little work since each team member is specialized in the specific part.

Other than system components, information flow and decision making are the two crucial parts of the system. Figure 2 shows how information flows and choice is made by the system. Camera captures the image and it is transferred to the client, Raspberry Pi which is then send through TCP/IP channel to the server side. Server has components to process this information and take action based on the system requirements. The image processor actually consists of two independent phases, classification and identification. Data is processed in classification program which is then sent to identification in case of a cat detected. The cat image is then identified in identification stage with image descriptors. This way, the cat name, or id, is determined and it is returned to the decision control module. Calculations based on database and decision control algorithms, a choice is made. Then the actions are going to be taken.

?

This design allows better manipulation of data and less computational burden since only specific data, images classified as cats will be exposed to identification process which requires greater computational capability. Moreover, cats can be identified in a few seconds in total so that the system can feed each cat before they quit the scene.

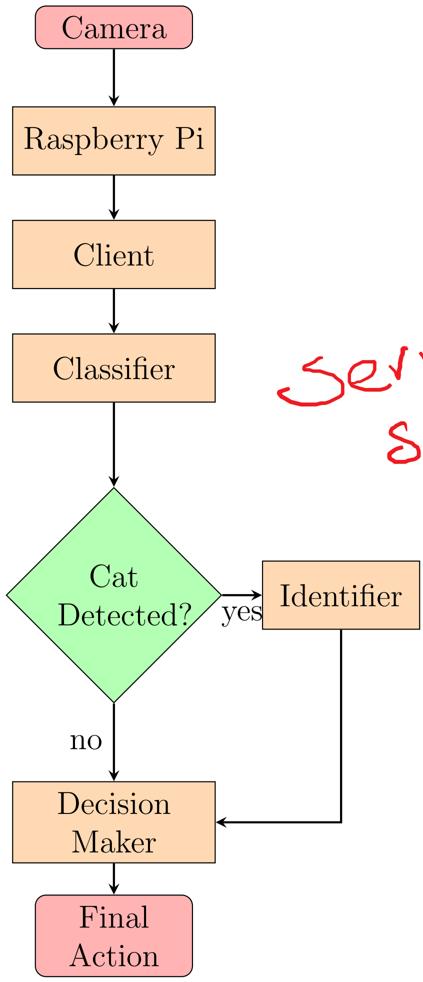


Figure 2: Information Flow Chart

server side ?

b) Requirements

- **Computer Vision:**
- Differentiate between cats and dogs.

- Clearly recognize scenes without any dogs.
- Detect how many cats are present.
- Have reliable confidence for the classifications.
- **Electronics:**
- The subsystem should be rechargeable and the charging time should be less than 5 hours.
- Rechargeable batteries should be non-removable.
- Battery duration should be at least 5 hours.
- The subsystem should be wired correctly and properly.
- The subsystem should be able to operate during charging.
- The subsystem should be able to operate Raspberry Pi and Arduino.
- The subsystem should be able to drive the servo motor.
- Power consumption of the subsystem should be minimum.
- Status of the food supply should be observable.
- Battery level for both charging and operation modes should be observable.
- The subsystem should be able to measure the amount of given food.
- **Mechanics:**
- Total system should be lifted by an ordinary person.
- Volume of the box should be enough for all electronic devices to fit in.
- Box should be durable for harsh outdoor conditions.
- Food reservoir should be provide at least 100 meals for an ordinary cat.
- There should be maximum %5 waste of food.
- Amount of food remaining, battery chargers while loading and in operation should be visible.
- Angle and the position of the camera must be adjusted properly in order to identify cats and dogs from 1.5 meter.
- There must be no obstacle in front of the camera unit, so that it can capture accurate photographs.

- Camera unit should be adjusted properly so that it should be protected from animal attacks.

4 System Architecture

The architecture is the most crucial part of the system that designates the relationships between each components. How components are connected, their interaction, and the behaviour of the system with the reasoning for the requirements and problems discussed. How system make decision, handle situation, manage all parts of it are all explained in this section.

This section explains computation steps, more detailed data communication strategy, software architecture, and user interface design specifications as well as critical implementation details. Following two sections, section a) and section c) explains these aspects in detail.

a) Server - Client Model

mostly irrelevant discussion.

1. Reasons and Requirements

Image processing and real time decision making are computationally expensive tasks which come with computational burden. There are also basic requirements which are important for the regular and stable operation of the system. In order to satisfy those requirements and reduce cost in addition to more flexible operation, server - client model best fits for the product. There are many reasons regarding the model structure from the point of customer and the company.

Customer needs and demands are changing continuously, even faster than it had been in past, such that customers are now looking for augmented product features, after-sale services, customer support, maintenance, repair, etc. Moreover, in the scale of thousands of product sales, it is apparent that massive customer support due to failures or malfunctions, questions and suggestions related to the product will require a big budget in the company. In addition, customer desire for monitoring the system and cat statistics are going to be satisfied through a server at present. All of these requirements and specifications make it reasonable to use server - client model.

Server - client model provides very powerful ways to satisfy augmented product requirements. After sale services can be ensured over network; any errors, warnings, malfunctions can be solved in real time, even before customer recognizes it. Digital world also enables customer to find their wants and answers for their products. Sensor data, images, statistics, problems are gathered and manipulated on servers make it possible to satisfy customer needs much faster and quicker. Customized experience for the users can easily be done, end even after the sale they can be reached and powerful customer experience can be achieved. This model also allows the Felerest group to remotely control, update, and collect data in a single way. New updates, modifications, features can be added easily and cheaper. Therefore, server - client model forms the project core and used in practice. Some of requirements for this sub-system is given in the itemized structure shown.

- Real time decision making capability
- Transfer data without any trouble and loss
- React to environmental changes accordingly
- Detect dangerous situations and take cautions
- Collect sensor, camera data for later analysis
- Process data efficiently
- Use reasonable amounts of bandwidth
- Manage database, dump unnecessary information

2. Software Design

The software combines all the parts and creates a backbone for the operation. Modular architecture with agile development is used throughout the project. python is used because of it's flexibility and it's rich library support as well as a big community. New libraries always can be found and errors are easily be solved with the help of community. Moreover, it's abstraction design make it the proper language for the backbone. Note that many programming languages are used in practice, C, bash scripting, JS, PHP; however python is the base language that runs the system.

The system can be explained best in the overall system diagram in 33. Controller is the mainframe that governs the whole system. Single server - multiple clients model is used so that the requirements can be satisfied effectively. Each product represents a client, and all of their data is collected in a server for processing, storage, and management. This allows the system to have greater computational capacity and storage size.

Client Client is the product itself as mentioned earlier. Peripheral communication and environmental controls are the jobs of this module as explained in overall design. Client have three sub-modules which are GPIODriver, CameraDriver, and Client object which separates into Command Client and Video Client. GPIODriver is the controller for the input/output whereas CameraDriver is an abstraction class over Raspberry Pi camera for an easier and clear implementation. Client objects are for video and command transfer which is explained in the next paragraph, namely

"Data Communication". As mentioned earlier, all of these abstractions are helpful for connecting various components.

Data Communication Data communication consists of every data, no matter type of it. The design suggests two parallel communication channels over TCP/IP protocol which is the only protocol used in data communication for reliability and cheaper implementation cost. The transfers use two separate ports over TCP/IP protocol which can be configured for customization but using 10002 and 10003 by default. These ports do not belong to any known service, therefore port filtering or any other service running in parallel will not be affected.

Everything comes with a cost, data communication does also. Opening data to the internet makes it vulnerable for possible attacks and customer privacy becomes an important issue. Some strategies are developed so as to cope with all of these challenges. Turkey puts some regulations on data transactions one of which is the "Protection of Personal Data" (6698 Sayılı Kişisel Verilerin Korunması Kanunu, Madde 3). This code states that without approval of the customer, it is not possible to store these data on abroad servers which is why servers will be located in Turkey and data privacy on an international scale will be solved. Moreover, encryption for both data, frame or video, and command are encrypted through 4096-bit RSA keys which are considered to be secure [4]. Implementation is based on SSH tunnelling, using OpenSSH.

Server Server resembles to the client. It uses sub-modules to create the server module. Controller, Classifier, Identifier, and User Interface are the parts creating the server side. Encapsulation of the software is done with python3 in which the most of the code is written. On the other hand, modules requiring high speed, computation burden are written in C, or they are compiled from C libraries such as SIFT, neural networks. Server is responsible for decision making, data storage, statistics, and human interaction for further product updates.

Data, as mentioned earlier, is another problem. It's security, transmission, storage creates problems for even very large servers. Solving these problems requires more strict specifications. For the final prototype, Linux file system with "pickle" objects are used for databases. However, this does not mean the final product is going to have simple file systems. Encrypted SQL databases, distributed storage techniques are future work after the prototype which involves data science and security engineer optimizations. For the prototype, these are omitted and it is assumed the server disk is completely secure, or it is run under a human supervision.

b) Tests and Results, Future Test Plans

Tests are measured with software tools nethods, top, iotop, and system monitor. The system results are given for the personal computer located in Computer Engineering Department Laboratories in METU. Taken 10 data points are averaged to get these results.

what is tested?

```
121 KB/s network usage on average
1.7 lag in data transmission
0.0625 errors / min
Memory (server) : 624 M
Memory (client)  : 1.99 M
CPU (server) : 13.3 %  (i7-4770S)
CPU (client)  : 2.2 %  (ARM1176(+)
```

c) User Interface

This module deals with the delivery of information regarding the device and its cats to the owner (user) of the device via the Internet. Following are the required features:

- Device's food supply level
- Device's battery level during charging
- Device's battery level during operation
- Device's feeding log
- Cats' profiles

An explanation and reasoning of the requirements is due:

Food Supply: Food supply level information helps the user decide when to refill the food tank. How the information is obtained is explained in detail in Section 6.

Battery Level: This is essential for any rechargeable product. Detailed explanation will follow in Section 6.

Feeding log: Each device has a feeding log of all cats that have been fed from it since its setup.

Cat profiles: These profiles are initially handled by the Identification module. (see 4.b). Each profile includes a name, an image and an individual feeding log. Furthermore, the name attribute is initially 'name' and can be edited later on by the user of the device. Nonetheless, the user cannot edit his/her cats' images: the images are provided by the device's camera. Feeding log is also not to be edited.

For the interface, a Python web framework called Flask is used. There were a few reasons behind this choice:

- Our user interface and web designer, Furkan, had more experience in Flask than in other frameworks.
- Since Flask uses Python in the background, CPU time is less than other platforms such as Nodejs.
- There is no need for an additional server, since a command as simple as 'flask run' can let us run our website on our computers.
- Because the communicated modules are mostly developed in Python, the communication is seamless.

The user interface, prototype of which is currently hosted on felerest.pythonanywhere.com, consists of 3 main pages: devices page, cats page, settings page, all of which can be accessed only with the user's password.

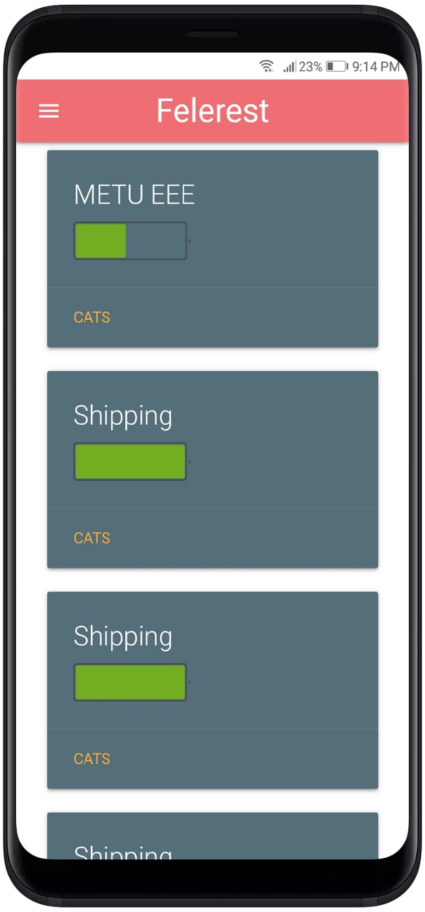
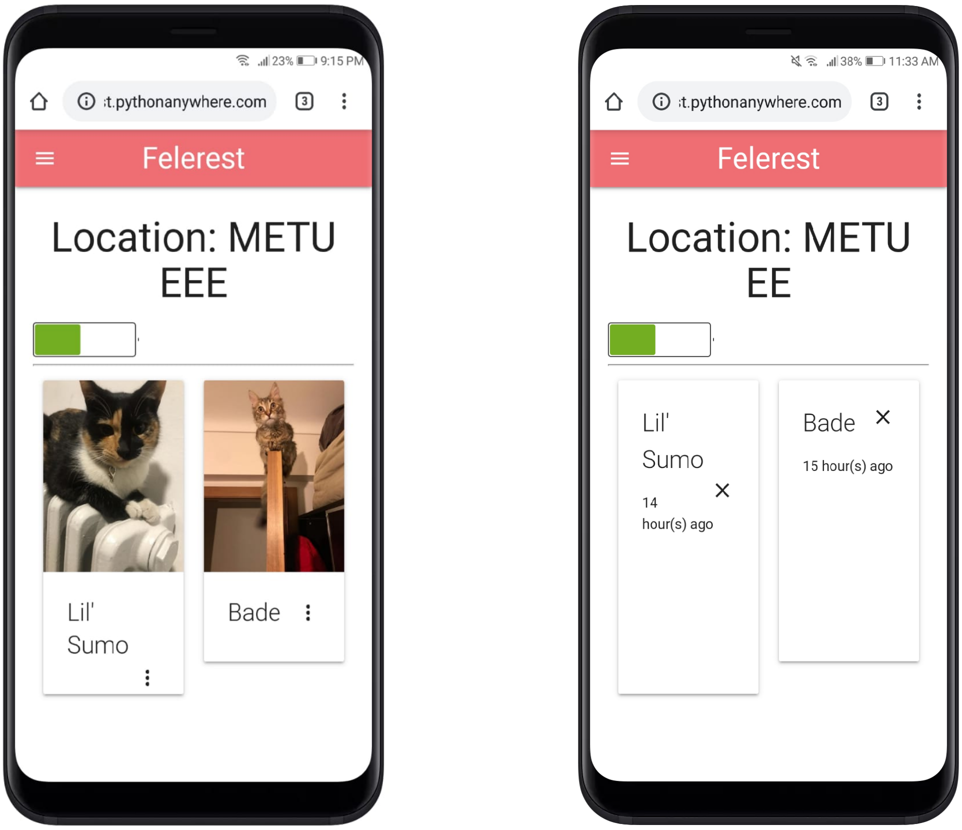


Figure 3: User Devices Page

The devices page (Figure 3) displays all devices that the user owns as cards in a grid. Each card shows that particular device's location, battery level, and food supply level. The food supply level indicator will be added eventually. There is also a link at the bottom of each card that takes the user to that device's cats page. There will also be another link that displays the entire feeding log of the device.

Cards in the figure labeled Shipping are just dummy devices to test the behavior of the grid with multiple cards.

The cats page (Figure 4) displays all cats that belongs to a device. On the page, the location, battery level, and food supply level of the device is displayed again for convenience of the user. Moreover, each cat has its own card that displays a name



(a) Device Cats Non-revealed (b) Device Cats Revealed

Figure 4: Device Cats Page

and an image. When the user clicks on a cat's card, the last feeding time of the cat is revealed, along with a link to the cat's feeding log. Currently, only the last feeding time is implemented.

The last essential page, the settings page, can be seen in Figure 5. Here the user can change the locations of devices and names of cats that are affiliated with him/her. The changes that are done here are implemented instantly. Locations/names 'METU EEE', 'Lil' Sumo', and 'Bade' have all been given as inputs to this page.

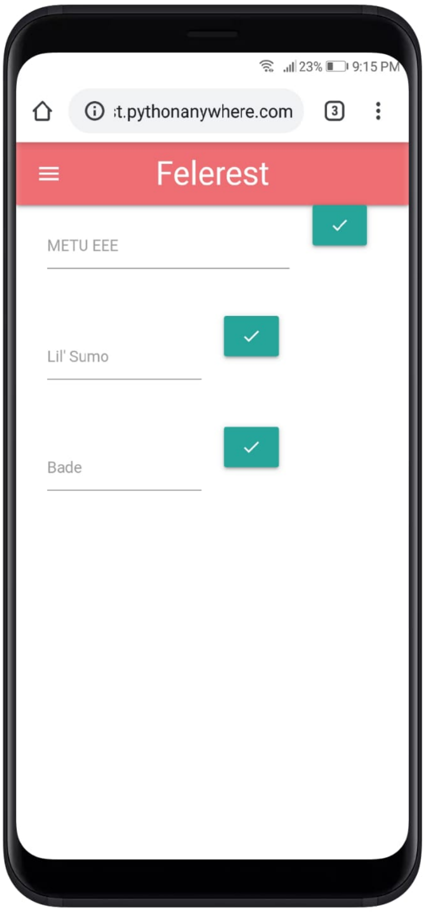


Figure 5: User Account (Settings) Page

5 Image Processing - Computer Vision

a) Classification

In this module, the following will be discussed:

- Subsystem Description
- Tests and Results, Future Test Plans
- Design Modifications
- Compatibility Analysis

First, the subsystem will be described in detail. The explanations won't be too technical for example principles of convolutions neural network, how it works, the layers required to construct it will be omitted. However, the general working principle of the code, some basic information about the chosen classification method (CNN), the general structure of the code (any pre or post processing methods), some important parameters that define how successful classification is will be explained in the description.

Then, the tests conducted on the classification code, the meaning of these tests, the meaning of the results and the success measures will be explained in detail. The tests that will be conducted will also be mentioned with explicit details.

Any kind of design modifications that were done to either the classification algorithm or the physical parts concerning this module (sub-subsystem) will be described afterwards. These include any extraction, addition, minor or major modifications.

In the last part, the compatibility analysis will mention how the interface of classification is regards to other correlated parts of the overall system and how it communicates well with other modules or subsystem. The requirements that were expected from this module will be mentioned and a proof will be done to show these requirements are satisfied.

1. Subsystem Description

As mentioned in the earlier reports, Convolutional Neural Networks were preferred as a method for classification. 6 is a very basic way of schematizing the network structure and the purpose of the classification algorithm.

As depicted in figure 7 the algorithm is somewhat straight forward. In the initialization procedure first the weights of the system are obtained, later the frame is passed through the neural network to classify where there are any objects that belong to the 80 different classes or not. However, not all of these classes are important to

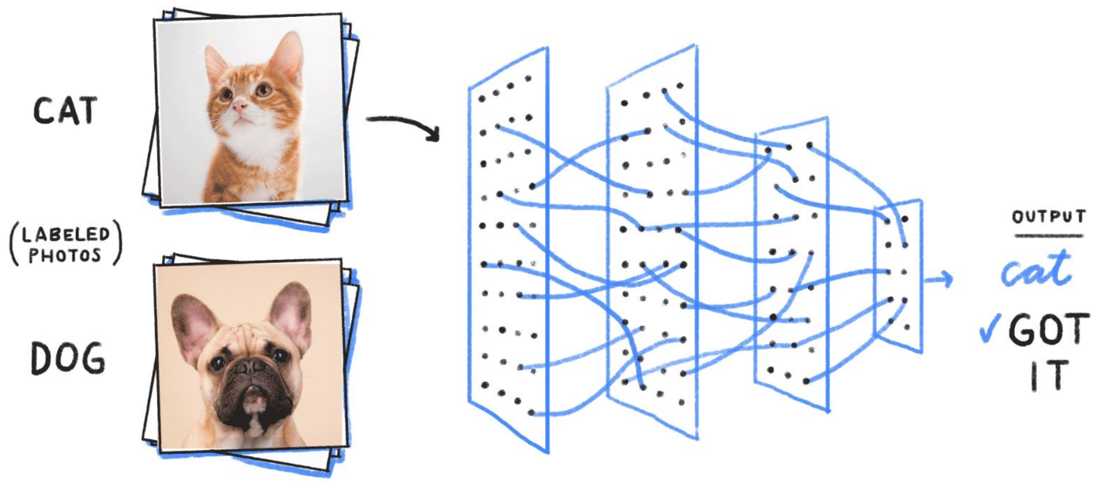


Figure 6: Simple Model of Cat-Dog Classification using CNN [5]

us. Therefore, the list containing the detected classes are compared with the desired classes which are cats and dogs. If there is a dog present, the system returns the string 'dog' no matter the presence of a cat. If there are cats present, the system returns the string 'cat'. If the debug mode is activated, the algorithm puts every classified object in a box with the confidence rate written on top. It also writes the time that has passed to do this procedure to every frame. This way, it can easily be defined what the network is doing wrong so the problem is easily detected, fixed and improved. In addition to this, the algorithms contain logging info lines which log what the system is currently doing to a separate file at all times. This was when an error is encountered it can be easily avoided.

If we are to itemize the basic principles of this algorithm and to discuss the extreme cases the following would hold:

- If there is a dog present the algorithm detects it independent of number of dogs and cats present at the same time.
- If there are only cats present; first it detects the presence of a cat, second it detects how many cats are present at the same time.
- If there is nothing present (no cats and no dogs) then nothing is done.

This itemization basically covers the backbone of the classification module where the input is the image and there are 3 states that carry two outputs each.

Example outputs are as follows:

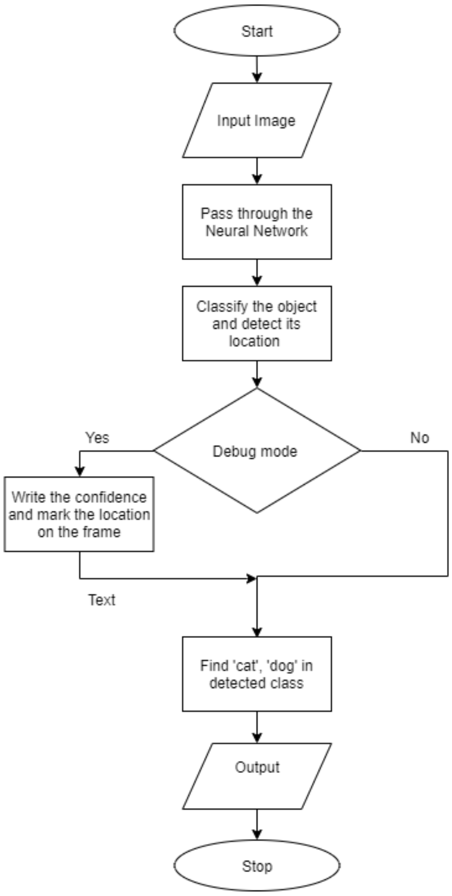


Figure 7: Algorithm Flowchart of the Classification Subsystem

- O = (‘cat’, ‘3’)
- O = (‘dog’, ‘NA’)
- O = (‘NA’, ‘NA’)

The second variable in each outcome is only important for the case where a cat is detected in the image. For other cases where there is a dog or nothing is present at all than the quantity doesn’t have an importance therefore it doesn’t matter if it is ’NA’ or any other random number.

2. Explanation of CNN for the Interested Reader

A basic NN architecture can be seen in figure 8. Inputs are not only images, but also the environmental features such as food case weight and animal weight. All of them are not going to be implemented, and most of them will be model dependent, for different applications of the product, different model features will be included. Figure shows all possible features that will be included in the neural network prediction model. More basic and simpler network architecture is given in figure 9, the networks we are going to be using are more complex and more layered structures which are basically the same in principle but requires more work in practice. The neural network architecture is trained, validated, and tested before it gets ready to be used. There are a lot of ready to use open source repositories which provide very powerful trained networks. The implementation of the classification model in this project takes advantage of the already trained networks since forming these networks and training them require a lot of time investment, effort and sources. Our group focuses on finding innovative solution by taking advantage of already existing solutions and advancements. This puts our team on a different level in the competitive market of cat feeding systems.

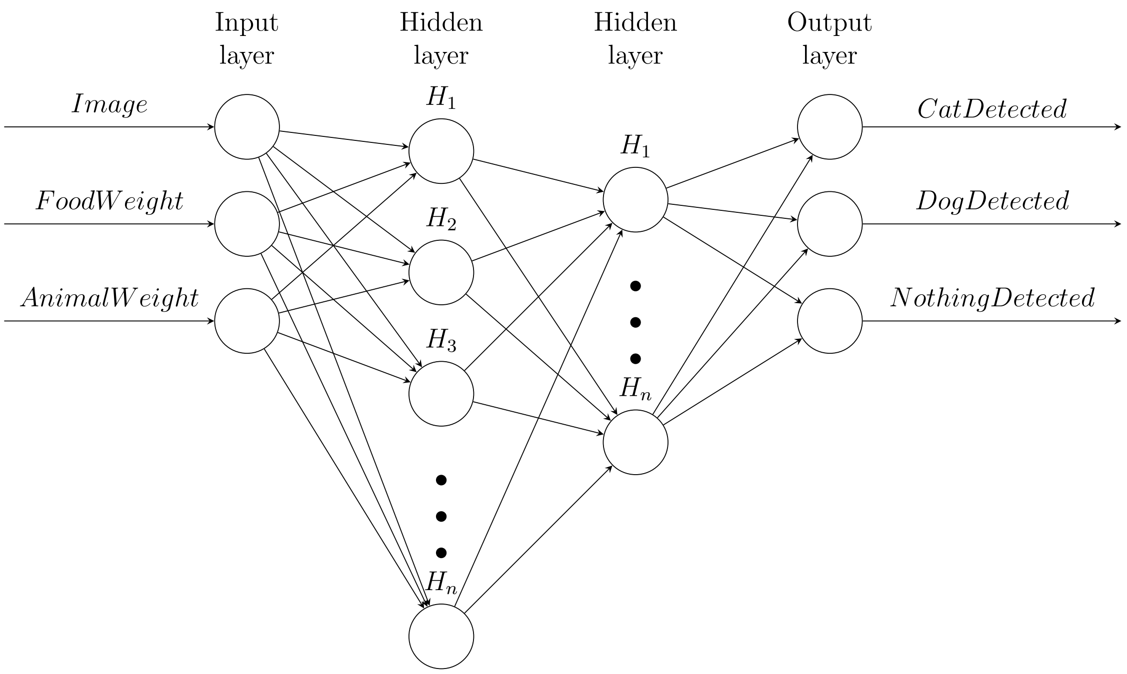


Figure 8: Proposed Neural Network Architecture

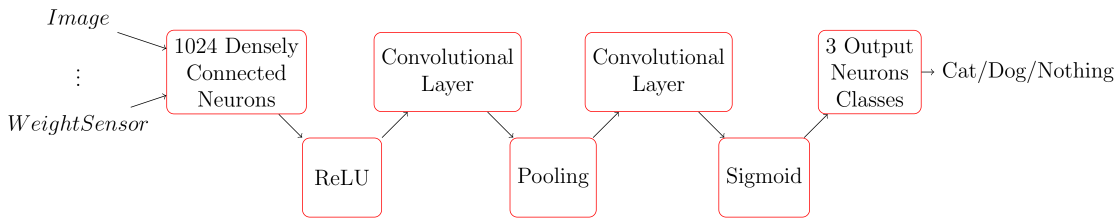


Figure 9: Simple Neural Network Architecture Design

3. Tests and Results, Future Test Plans

In this part, the conducted tests will be discussed in detail alongside planned future tests. Information regarding each test will be given in the following order: 'What is the test?', 'Why is it important?', 'How was the test conducted?', 'The meaning of the results and its significance'.

1. **Test: How confident are we on detecting the correct class for cats?**

   - **What is it?** Confidence is a property of the classification via CNN. It gives us the measure of how certain the network is in classifying it in the correct category. This test is to figure out what is the average confidence in correctly categorizing cats.

   - **Why is it important?** This test is important in determining the confidence threshold for the algorithm. This way it is possible to correctly eliminate cases where confidence is low. We are then left with correctly detected cats with high probability and no cat is left that is below this threshold.

   - **How was it conducted?** In this test we investigated photos with single cats present. We collected the confidence for these images and found the estimate confidence. The photos were taken with a camera and they were of real cats. Unfortunately the cats present were 3 different types due to the lack of finding willing real cats to test with. These cats' photos were taken in different angles and positions.

   no mathematical definition of confidence

   - **Results** There were 51 photos in total and the average confidence of cats detected in these photos were approximately 84%. It should be noted that this is a successful and high result. The threshold when for these photos were around 50%, meaning that any cats detected with a lower confidence than 50% were omitted and regarded as a failure. The average result being 84% implies that 50% was a successful result since it exceeds it largely. Figure 10 is a great example of this. Please take regard to the image quality in this case. As can be observed the cat is very dark in this image and the contrast is low. The cat is also very close to the camera and only a small portion of the nose and left ear can be easily observed even with a human eye. This demonstrates the performance of the classification algorithm.

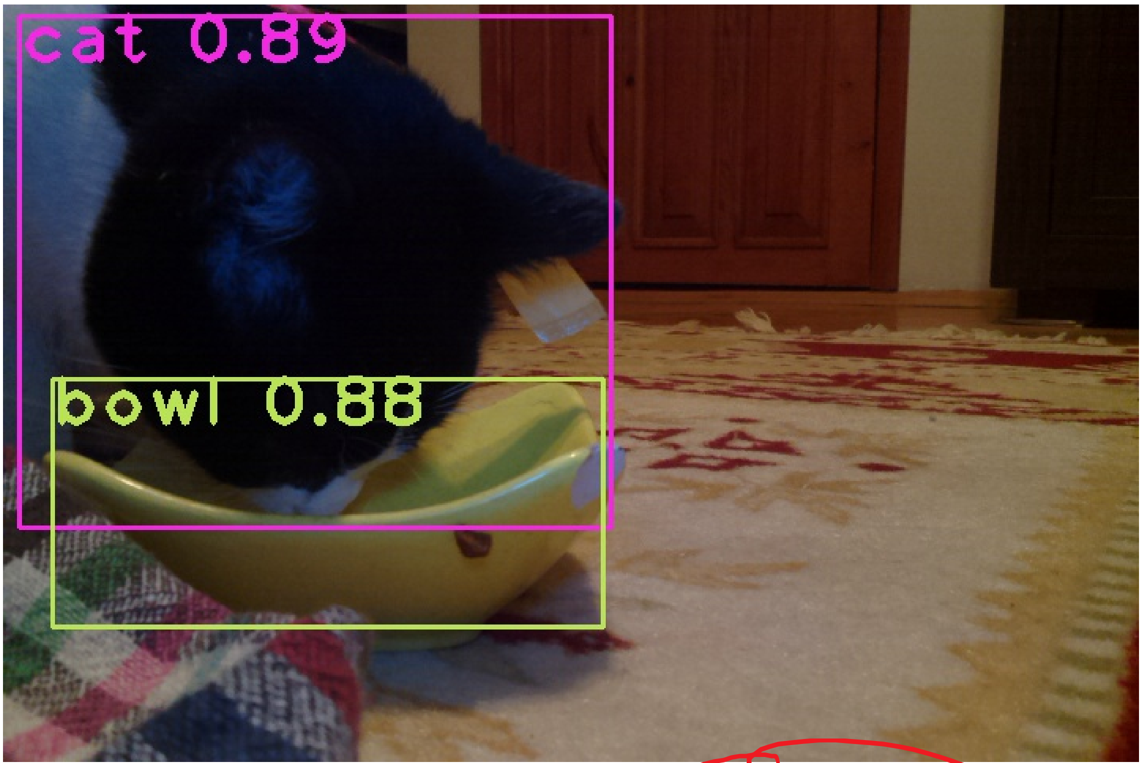


Figure 10: A cat being detected with confidence of 89%. ?

2. **Test: How current is the false positive cases?**

   - **What is it?** False positives are the worst cases where a negative situation is detected as a positive situation. In this module if dogs are detected as cats this is the most undesirable scenario. So, we want to run images of dogs and see if these dogs are detected as cats. If so, we want to detect the likelihood of false positive cases occurring. This might give us a specific cause where we can limit the code to avoid false positives.

   - **Why is it important?** This is important because then dogs are fed with cat food which increases the likelihood of the dog revisiting the feeding system for food. We need to detect false positive cases so a solution can be found to prevent this case.

   - **How was it conducted?** Since finding real dogs are harder than finding cats we had to test with only images of dogs. The important restriction to be kept in mind is that the photos found online are usually photo-shopped and doesn't give a good representation of reality.

- **Results** There were 6 different dog photos, by capturing different orientations and distances of these photos with the camera, we analyzed 24 cases. Out of these cases there was only a single case where a dog was detected as a cat. We believe that in real life cases this will be reduced even further to perhaps 1 false positive in a 100 dogs. Since there are multiple frames sent per second and a final result is obtained from these images this result will not cause a problem. Because in the final result the dog will be detected eventually in one of the frames taken over a small interval of time and food will not be distributed.

The aforementioned two tests alongside more detailed analysis is presented in the table below 11. Tests are done in extreme conditions where light is very little, and view angle is narrowed by intent. The 'correct' column indicates that the corresponding class has been identified correctly and 'accuracy' gives an average of their confidences. 'Wrong' indicates if there the class present in the photo was detected as another class, for example; a cat was present and a dog was detected. 'Unclassified' corresponds to a class being present and the algorithm not detecting anything at all, in other words not realizing the presence of a cat or a dog. 'TOTAL' is the number of photos taken with the camera and the test was ran over these photos. There were 112 cat photos present with real cats and photos of cats. However, in this test data there are multiple photos of the same cat with different backgrounds, light settings, contrasts, orientations, positions and such.

In result, the cases that are either unclassified or wrongly detected does not have significance. This is because while the camera is ON, the algorithm only processes frames taken from each time interval. This time interval is determined by the speed of the algorithm (both the preprocessing an the time it takes for the algorithm to process the frame) and to some extent the capability of the camera. Since there will be x many frames processed in a second there will be higher likelihood of correctly classifying the object. In other words, a single frame that is wrongly classified or unclassified will be ignored and have no importance. To be certain that this won't be an issue and to have an idea about the processing speed, it best to some tests.

3. **Test: The processing speed of the classification algorithm**

   - **What is it?** This test will determine how many second it takes to run the classification algorithm on a single image. It will give us the rate.

no math def.

	Correct	Accuracy	Wrong	Unclassified	TOTAL
Cat	91	84%	15	6	112
Dog	23	79%	1	0	24

Figure 11: Accuracy table for cat and dog classes

- **Why is it important?** As mentioned before, this test is important for the next step of the computer vision part. After classification it should be made certain by somehow mediating over a selected interval of time. This part of the process can even be determined perhaps after the identification part and before commanding the mechanic part for food flow. This way the accuracy is surely increased.
- **How was it conducted?** For this test, we placed time stamps throughout the code to first determine the most time consuming part of the code. After doing this we evaluated the time consuming sections of the code to decrease the time further. After repeating this process a several time, we outputted the total time spent over a single frame. An important note to be made here is, the initialization part of the code was omitted. The initialization is done only once in the beginning when the camera is turned ON or the system is started over. For example, these parameters are weights that need to be downloaded once in the very beginning. Basically a single time stamp was placed in the code after the initialization part and at the very end where the code is executed or started over. Then the time elapse was collected by subtracting the result of these two time stamps.
- **Results** The rate was found as **3 frames per second**. In other words a single frame took nearly 0.3 second to process. As mentioned earlier, at the beginning the time for each frame to be processed was longer (more than 0.5 seconds) which sometimes meant not more than a single frame was processed in a second. By adapting the code for better performance, the achieved final time elapse was significantly smaller.

It should be noted that the time required for single frame's processing is a varying parameter. The detection speed changes under conditions where there is light exposure, many object, color variations, movement, contrast changes, fast moving objects that cause blurring in pixels and such.

4. **Test: The distance from the camera for successful classification**

   - **What is it?** This test is to have an idea about the camera's distance range and the classifications capabilities. At the end we will be able to determine how successful we are in classifying close range objects and objects that are far away.

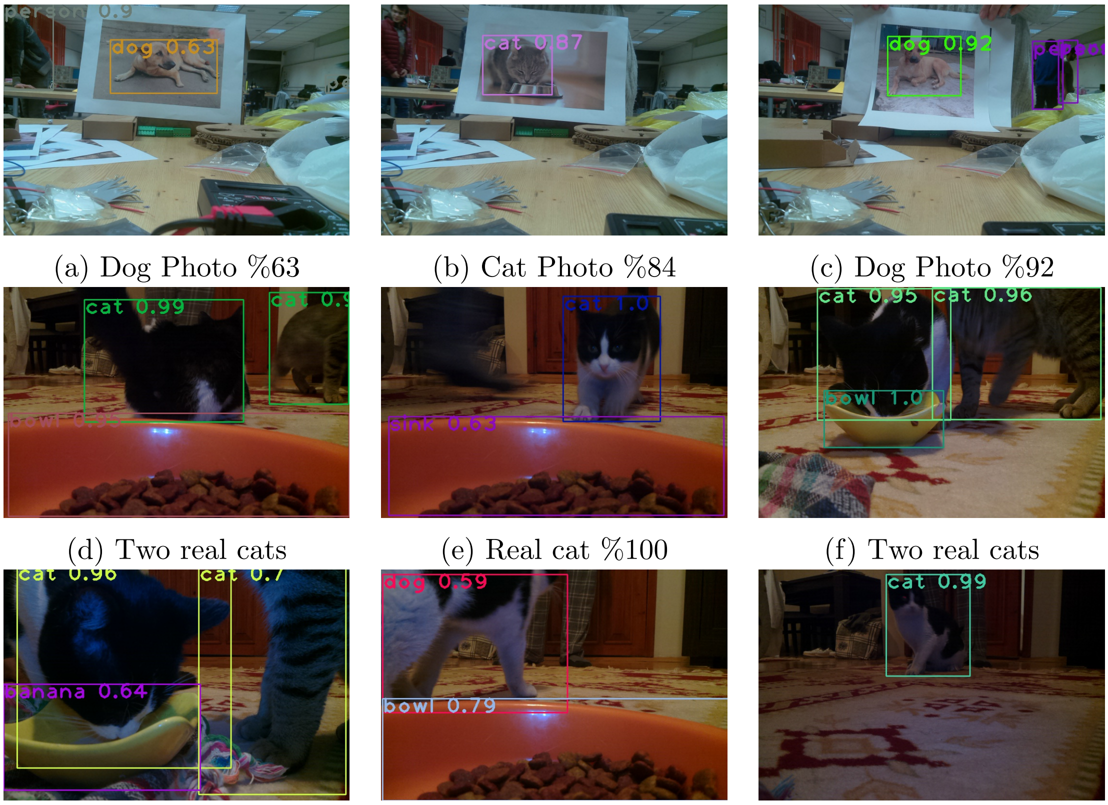
   - **Why is it important?** This is important for two reasons. One is for the standby state of the machine. For low power consumption the camera will be off at idle, when there are no pets approaching the box. However, there will be a sensor which detects movement. When a movement is detected the camera will turn on and start recording. The information of farthest detection is important such that the movement sensor will have a similar distance range. For compatibility ans successful operation this is crucial. The second reason is for the developers. This knowledge is important so the requirements are satisfied. A short range is undesired since detection and classification becomes difficult. This might cause cases where the cats are present but its undetected because ts out of range. Therefore, the range should be somewhat large.

   - **How was it conducted?** An image of a cat is held in front of the camera at a very close proximity and waited for processing, then the image is moved further away from the camera inch by inch. At a distance where classification is unsuccessful is marked and measured. A similar experiment is done to detect the the lower distance boundary of detection. Of course the images are chosen from the data-set with high confidences such that the reason for non-detection is not dues to that specific cat photo and the test is conducted under close to ideal conditions.

   - **Results** Detection distance is **5-150 cm for cats** and **10-180cm for dogs**. This range was found to be sufficient for robustness and high performance.

In figure 12 9 photos from the test data set was chosen. There are photos with two real cat, single real cat, photos of dogs, photos of cats and a cat detected as a dog. Please take notice to the image qualities, the small areas (parts of the bodies) that are visible to the camera and high accuracy nevertheless.

The entire data set is given in this link. Some of the tests were done on images obtained from google randomly. These images were printed on A4 paper in large scale and held in front of the camera at various angles from various distances. Because of the enlarged photos, low quality of the printer and the images itself there were some uncompensated noise factors. Some were done on real cats as indicated.



(a) Dog Photo %63 (b) Cat Photo %84 (c) Dog Photo %92

(d) Two real cats (e) Real cat %100 (f) Two real cats

(g) Two real cats (h) Real cat detected as dog (i) Real cat %99

Figure 12: 3 x 3 Photos from the tested data-set

Future Tests:

- Test on real dogs.
- Test time elapse for each frame when identification part is introduced.

Additional Note: As mentioned in the previous report, there were preprocessing and fine tuning that was planned. Due to the unexpected high results in confidence, accuracy and performance. Fine tuning was seen unnecessary, especially in regards to time investment and large data set that was necessary. As for the preprocessing, this was achieved during one of the tests when trying to reduce frame time. The image resolution was made smaller and no change in contrast and color of the frame was decided to be necessary.

4. Requirements

- Differentiate between cats and dogs.

- Clearly recognize scenes without any dogs.
- Detect how many cats are present.
- Have reliable confidence for the classifications.

All of the requirements are satisfied as already explained in the earlier sub-subsections.

numbering?

5. Modifications

No modifications were done.

6. Compatibility Analysis

This is not a computibility analysis.

The compatibility was tested during the tests since the input of the algorithm is from the camera and the output is to the identification module.

Pre-processing ensured the compatibility from the camera to the algorithm and the initial promising results of the identification module satisfied the compatibility of the classification module with the identification module.

b) Identification

1. Subsytem Description - Requirements

Identification is another subsystem of the overall system. System should identify cats and feed them accordingly with a high accuracy. Identification is put just after the classification so that only cats are recognized and no more computation is required. There are different techniques in identification process such as histograms, feature descriptors, neural network based identification, and landmarks.

- Identify cats with accuracy greater than 0.9.
- Recognize new cats and add them to database.
- Work in a reasonable time, aimed less than the second.
- Occupy space less than 1GB on the server for database.
- Be capable of adjust data.
- Advance database as more sample arrives.

2. Explanation and Comparison of Methodology

Every method has its own advantages and disadvantages. Comparing them and selecting the best is best explained in the following table representation. Tables

Histogram Based Identification	
Advantages	**Disadvantages**
Very fast train time	Relatively slow identification time
Easy to implement	Open to noise because of light, angle etc.
Very powerful with different color cats	Bad performance with similar colored cats
Small database requirement	Relatively low accuracy
Fast update-train time	Confusion as database enlarges

Table 1: Advantages and Disadvantages on the Histogram Based Identification

Neural Network Based Identification	
Advantages	**Disadvantages**
Very fast identification time	Very slow train time
Learning based, easier	Open to noise
More powerful with more data	Bad performance with few data
Very small database requirements	Very slow update-train time
	Fine-tune requirements for different environments
	Useless with single-shot data

Table 2: Advantages and Disadvantages on the Neural Network Based Identification

1, 2, 3, and 4 give the comparisons for histogram, neural network, landmark, and feature descriptor based identification algorithms.

Comparison tables are self-explanatory. However some clarifications are needed. There are two train processes where train is conventional train with existing data and update-train is train for the new coming cat, image data. Neural network based identifiers require very big update-train, which is the most important drawback of them.

3. Data Collection and Sources

At the end of this comparison, it should be noted that different methods have their advantages and disadvantages. Therefore, it is also possible to combine them in a way that the final decision takes considerations of different domains exhibiting

Landmark Based Identification	
Advantages	**Disadvantages**
No train required	Another control of comparison required
Easy to implement	No past experience
Powerful with different kinds of cats	Lack of train data
Small database requirement	Bigger identification time with 2-steps
	Very little available resources

Table 3: Advantages and Disadvantages on the Landmark Based Identification

Feature Descriptor Based Identification	
Advantages	**Disadvantages**
Robust to illumination and angle	Very slow train time
Relatively fast identification time	Low accuracy if no features present
Powerful with detailed cats	Wrong matches with
Useful with different environments	
Fast update-train time	
Relatively higher performance for single shot	

Table 4: Advantages and Disadvantages on the Feature Descriptor Based Identification

different characteristics. Unfortunately, at the level of this prototype, without gathering any data, it is impossible to estimate the data behaviour in real application. What is the cat frequency, how photos are taken, what is the angle of camera, what is external lighting etc. All of these questions require answers to finish the complete fine tune of the system.

All of the systems described can be used for different environments. However, lack of data makes it really difficult to make a decision at this step when there is no working prototype present. Therefore, the most robust one is selected as it is the one which minimizes the risk, but it is pointed that there is not a single decision but a set of decisions to use different methods in accordance and assigned weights so that best results can be obtained. It should also be noted that different methods are implemented, and based on the prototype and product data, continuous switch between all of these methods are possible. More clearly, there are hyper-parameters that defines how the identification methods affect the final decision on the cat id. However, all of these adjustments and fine-tuning process is left after the prototype has been finished since no data found on the internet or another resource completely, or even nearly defines the behaviour and characteristics of data collected by the system. Finally, from now on, feature descriptors are assumed to be a generic solution to the problem. However it is always the case that a hybrid approach will be used in the actual product and the final prototype.

4. Feature Descriptor Implementation Details

Feature descriptors are used in identification. SIFT and ORB implementations are used and feature vectors are created for the interest points that are calculated based on known algorithms SIFT and FAST. For commercial purposes, ORB is decided to be used; however, SIFT is used in research of the product because of its slightly better performance. Note that SIFT is patented which is why ORB is going to be used in the mass production software of the product.

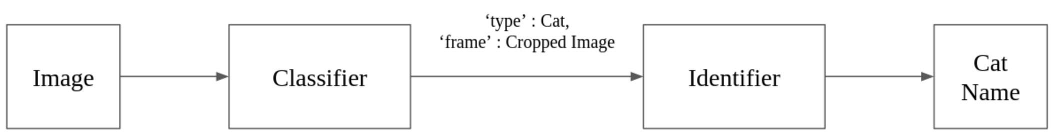


Figure 13: Classifier Identifier Connection

Identifier module requires an image with strict bound. Cropped image is directly sent to the Identifier module from Classifier module. This strict bound both prevents excessive data coming from background and smaller set of vectors to be processed. Since the angle that cats coming to eat the food is deterministic and more or less the same, background details are mostly eliminated, and an image consists only of cat head. This way, computation burden and noise are minimized.

Effect of background can be seen from figure 14 and 15. Most of the features come from background. However, this image does not properly show the correct case where cats will come directly to camera. Therefore, background will lose its importance. In other words, classifier crops images in practice such that there is no background and disturbing effects. Some of the samples for this process, is given in figure 16. Note the Facebook database which will be explained in the next section "Test Methods and Results" consists of images with backgrounds, this is not the case thanks to the configuration of the mechanical system and classifier.

5. Tests and Results

It requires extra effort to make tests to determine how the sub-system does well. Both the lack of electronic and mechanical background makes it very hard to collect data directly from the camera in addition to the mobility of cats. Both finding and capturing a cat picture requires a great effort and it only allows a few sample data to be collected. Therefore, there left two possibilities; finding online cat images and using a static object. Online images are mostly consisting of random cat images that are not identified or a specific cat only appears in the pictures. Therefore, a data set is created based on the cat images shown on the computer screen and shots taken from the screen as well as Facebook group volunteers that gives samples for different cats with different names. Consequently, there exist two separate data sets, one of which is set of images for the same cat on the screen, few number of classes (cat identities), but a lot of samples, and the other one is a data set with various classes of cats but with a limited data.

3 different cat images found on the internet where neither of them is the same cat. These cat images are displayed on the computer screen and total of 100 shots were created for each cat image that is displayed. Among these 100 images there are only 60-80 possible samples which include cat, or classified correctly. Since camera gets blurry when moving and sometimes it points to different directions, it was not always possible to take pictures of cats. Therefore, only 40 - 60 sample is generated for the cat image displayed on the screen. It should be noted, to show background effect, different backgrounds, different display positions and different displays are used to display image.

Cat windows inside these images are extracted using the classification code and database created. 5 samples per class are separated for test set. Note that since

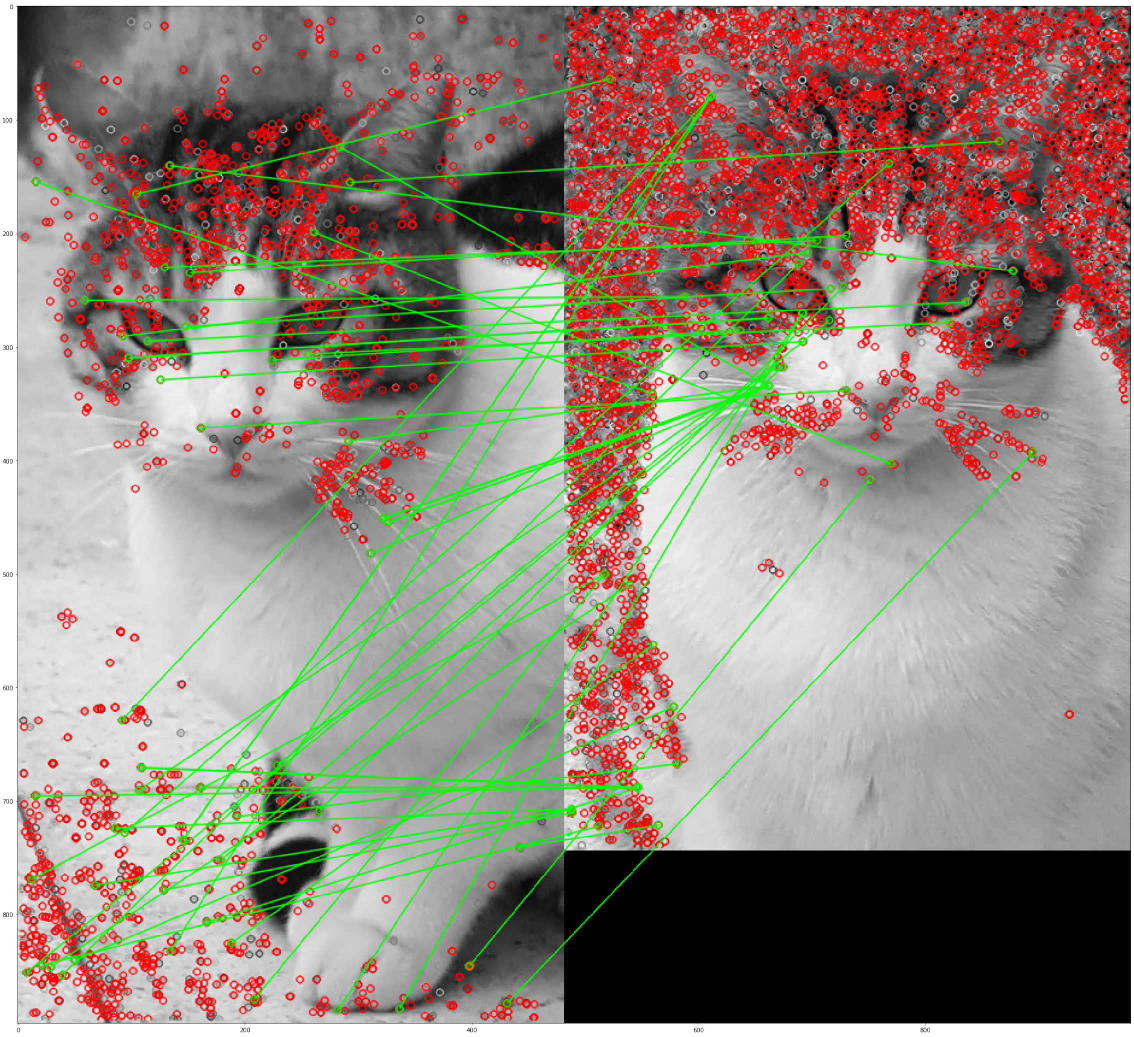


Figure 14: Background Interest Points

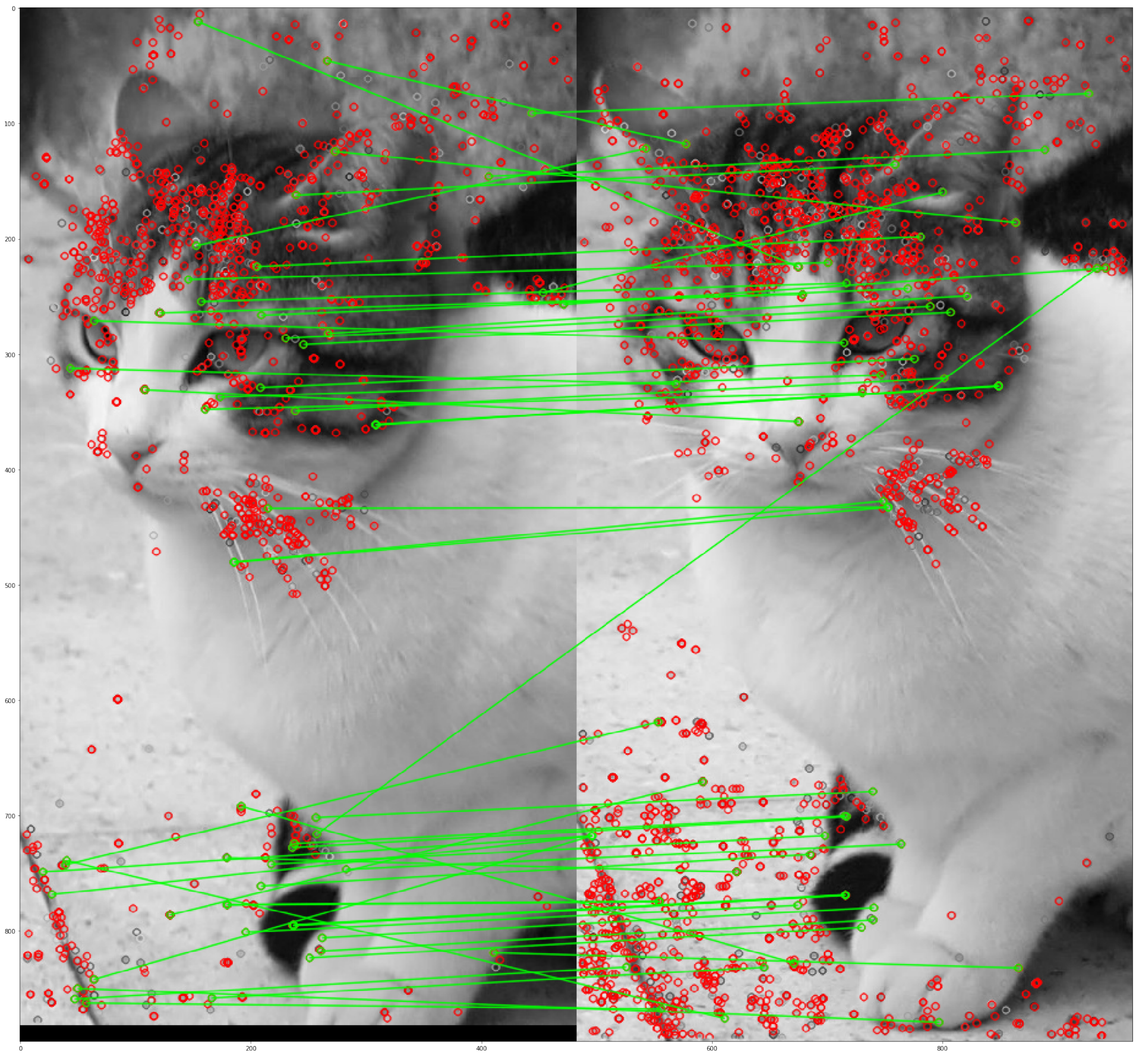
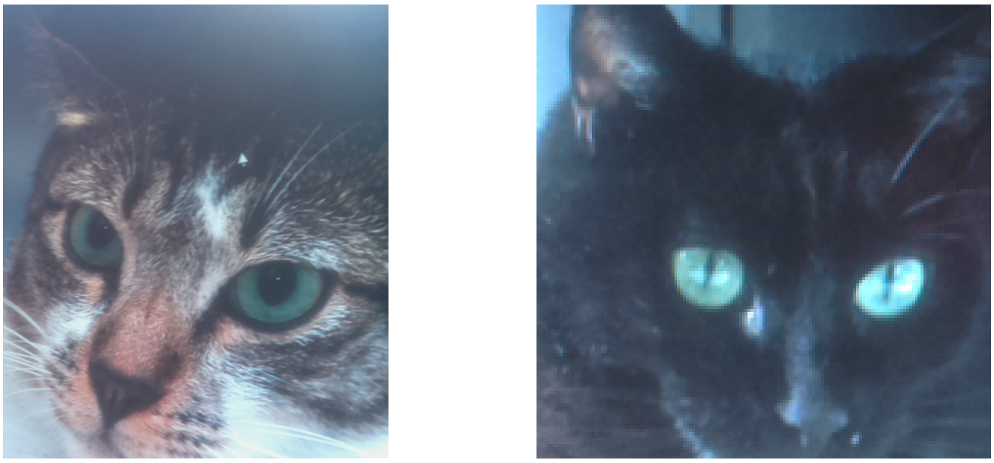


Figure 15: Background Match



(a) Cat 1

(b) Cat 2

Figure 16: Sample Images Generated by Classifier on Real Camera Shots

there are no hyper-parameters currently optimized, there is no validation set selected and only train and test sets became in use.

The other database consists of 17 different classes and 4-7 samples for each class. This is a good example of few data identification. The data set is divided to train and test sets, where test set consists of 1 sample of each class since the data is very limited. This seems a biased approach; however, system is designed for the conditions of excessively limited data. Tests are done on both data sets for different approaches and the hybrid approach at the end, and the results are given in tabular and graphical forms.

There are three independent tests are done on two data sets explained above. Confusion matrix representation is given in figure 17 and 18 for two different data sets explained above. Moreover, technical information on database provided in figures 19 and 20 for this data sets. Note that in case of very little data, there are hardly SIFT vectors found. Test results show that, with correct camera position, and enough data accuracy can easily can go around 1.0 whereas different cameras with different angles make it to reduce even 0.3125 which is the case in figure 18.

Above discussion distinguishes the behaviour on two separate sets. However, there is no single control variable to measure the actual performance. Since these data sets differ in both size and capture techniques, another data set is created which is small in train images but data collected with the same camera in different environments. This data set is tested for single shots. Single shot of a cat is taken from the camera and remaining images are put as test set. In other words, for 3 different classes, there is only 1 train sample, and 40-60 test samples. The results are astonishing in

Confusion Matrix

		Predicted			
		Cat 1	Cat 2	Cat 3	Cat 9
Ground Truth	Cat 1	5	0	0	0
	Cat 2	0	5	0	0
	Cat 3	0	0	5	0
	Cat 9	0	0	0	5

Figure 17: Confusion Matrix for Set 1 - Test 1

Confusion Matrix

		Predicted														
		Selin	Seda	Melis	Haza	Ezgi	Esra	Esr	Ekin	Dilara	Ceren	Bahar	Ase	Cem	Deniz	Ezg
Ground Truth	Selin	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Seda	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	Melis	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	Haza	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	Ezgi	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	Esra	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
	Esr	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	Ekin	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	Dilara	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	Ceren	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	Bahar	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	Ase	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	Cem	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	Deniz	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
	Ezg	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Figure 18: Confusion Matrix for Set 2 - Test 2

```
Number of classes in total : 3
Number of vectors per class :
        bir                              : 347024
        dokuz                            : 662830
        uc                               : 52026
        ------------------------------   : ____+
        Total                            : 1061880
```

Figure 19: Database Information for Test 1

```
Number of classes in total : 17
Number of vectors per class :
        ase                              :   164
        bahar                            :   154
        cem                              :    10
        ceren                            :   118
        deniz                            :     2
        dilara                           :    18
        duygu                            :    24
        ekin                             :    24
        esr                              :   156
        esra                             :  4130
        ezg                              :  1082
        ezgi                             :    18
        haza                             :     0
        hazal                            :     0
        melis                            :    46
        seda                             :    40
        selin                            :   148
        ------------------------------   :  ____+
        Total                            :  6134
```

Figure 20: Database Information for Test 2

```
Number of classes in total : 3
Number of vectors per class :
        bir                              :   308
        dokuz                            :   146
        uc                               :   136
        ______________________________   :  ____+
        Total                            :   590
```

Figure 21: Database Information for Test 3

```
Detailed accuracy report :
        Cat name selin with accuracy    : 1.000000
        Cat name seda with accuracy     : 1.000000
        Cat name melis with accuracy    : 1.000000
        Cat name haza with accuracy     : 1.000000
        Cat name ezgi with accuracy     : 1.000000
        Cat name ezg with accuracy      : 1.000000
        Cat name esra with accuracy     : 1.000000
        Cat name esr with accuracy      : 1.000000
        Cat name ekin with accuracy     : 1.000000
        Cat name dilara with accuracy   : 1.000000
        Cat name deniz with accuracy    : 1.000000
        Cat name ceren with accuracy    : 1.000000
        Cat name cem with accuracy      : 1.000000
        Cat name bahar with accuracy    : 1.000000
        Cat name ase with accuracy      : 1.000000
        Cat name bir with accuracy      : 0.978261
        Cat name dokuz with accuracy    : 0.932584
        Cat name uc with accuracy       : 1.000000
Calculated accuracy is 0.9597701149425287
```

Figure 22: Accuracy Tests with New Cats to the System

this part, accuracy is around 0.96 which is given in figure 22. In addition, new cats which are the cats the system did not see before are identified as 'None' meaning it is a new cat. The accuracy is exactly 1 which is because a threshold is dynamically adjusted for the train samples with a little loss in accuracy of the known cats, only around 0.02 - 0.07 as can be seen in figure 22.

Test results show that the system is capable of handling the cat identities with a higher accuracy. Despite the little data sources, more than 300 images and 20 classes are used in the test procedure and the results show an obvious superiority for the feature descriptors. Identification times, model train, database creation time are given below for operative perspective.

- **Database creation (164 images) (54 per class) :** 6 hours 30 minutes 33 seconds
- **Database creation (74 images - 4.3 per class) :** 12 minutes 45 seconds
- **Database creation (3 images - 1 per class) :** 4.65 seconds
- **Image match (average) :** 0.4 seconds

The tests are done on a regular old personal computer without any acceleration or turbo technology. Single core - single thread is used in the process for better accuracy.

```
Architecture:                     x86_64
CPU op-mode(s):                   32-bit, 64-bit
Byte Order:                       Little Endian
Address sizes:                    43 bits physical, 48 bits virtual
CPU(s):                           8
On-line CPU(s) list:              0-7
Thread(s) per core:               2
Core(s) per socket:               4
Socket(s):                        1
NUMA node(s):                     1
Vendor ID:                        AuthenticAMD
CPU family:                       23
Model:                            24
Model name:                       AMD Ryzen 5 3500U
                                  with Radeon Vega Mobile Gfx
```

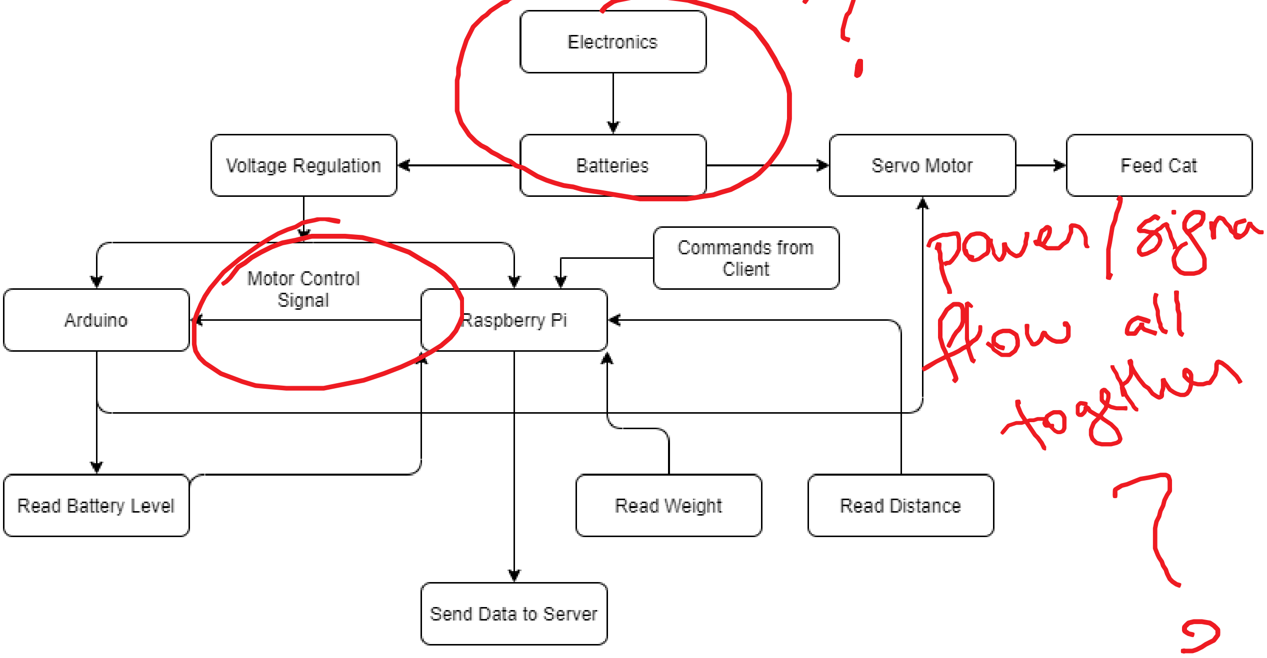
6 Electrical Design

In this module, the following will be discussed:

- Subsystem Description
- Requirements
- Tests and Results, Future Test Plans
- Design Modifications
- Compatibility Analysis

a) Subsystem Description

In this subsystem, all electronic components are connected and powered up for the use of other subsystems. The subsystem contains batteries, voltage regulators, battery chargers, relays, diodes, a microcomputer, a micro-controller and a servo motor. Flowchart of this subsystem is shown in Figure 23.



?

power/signal flow all together ?

Figure 23: Flowchart of the Electronics Subsystem

1. Supply Unit

Supply connections for the motor and controller units are shown in Figure 24. Two separate battery units are designed for motor and controller units to obtain better motor performance.

High motor torque is required for the rotation of the mechanical gate. 2 Lithium-Ion Batteries are connected in series to obtain a voltage in the input voltage range of the MG996R servo motor. Parallel and series combination of diodes are added at the end of this unit. Parallel diodes increase the maximum passing current of the node, and serial diodes are used to keep the voltage in the input range. Relays are added to decrease power consumption and increase charging speed. The relays RL1 and RL2 shown in 24 are used for parallel charging operation while the outer box is connected to the power outlet with a micro-USB adapter. Also, while the box is not plugged in, the batteries become serially connected, as explained above. However, in this way, the motor is not able to operate during the charging operation. Another relay RL3 is added to solve this problem. The power of the motor is supplied directly from the power outlet during the charging operation, and it is supplied from the batteries while the box is not plugged in. Furthermore, the relay RL4 is connected to decrease power consumption. It is controlled from the Raspberry Pi and turns on the motor when a cat is detected.

For Raspberry Pi and Arduino, a constant stable voltage input is required, so a 5V step up voltage regulator is used. Two Lithium-Ion batteries are connected in parallel to increase usage time and the batteries are directly connected to the step up voltage regulator. Both Raspberry and Arduino are supplied from this regulator.

For charging operation, 3 different charger circuits are connected parallelly as seen from the Figure 24 and all batteries are charged up synchronously.

Also, two resistors are used to read analog voltages from the Arduino.

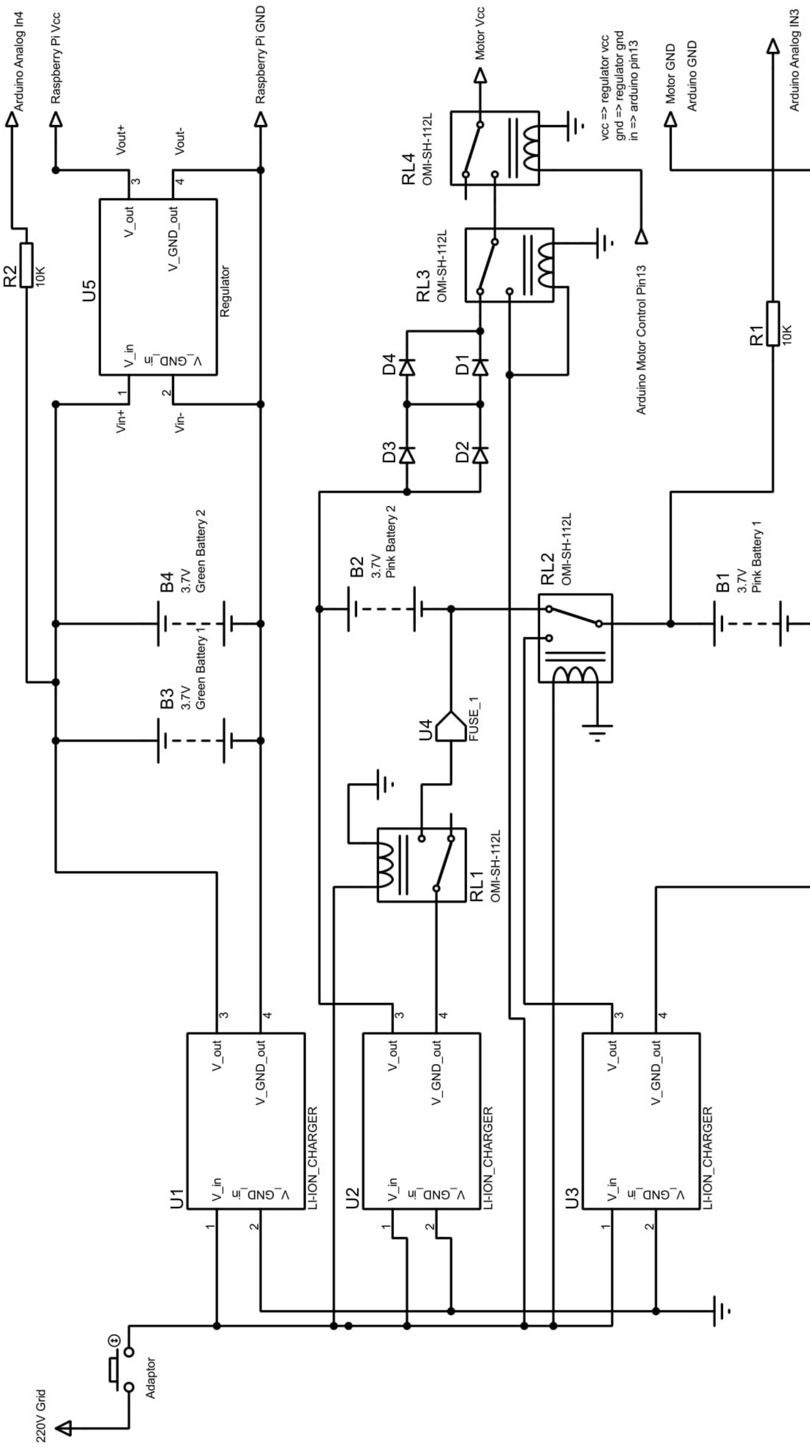


Figure 24: Circuit Design of the Supply Unit

2. Raspberry Pi and Arduino

Raspberry Pi is the main control mechanism of the project and its operations are listed below.

- Receiving images from the camera module and transferring them to the server.
- Controlling the rotation of the motor by sending a control signal to Arduino through a serial communication channel, and this command opens the Relay 5 shown in Figure 24 and generates a PWM signal.
- Measuring the distance coming from the ultrasonic sonar sensor, converting the measurement to volume, and sending it to the server.
- Measuring the weight of the given food and sending the data to the server.
- Receiving the analog battery voltages from the Arduino by using I2C serial communication protocol to indicate battery level in the web interface.
- Controlling the LEDs which indicate the detection of cats/dogs.
- Controlling the operation of the light bulb according to the time.

Arduino is the second controller used in the project and its operations are listed below.

- Controlling the servo motor by generating PWM signals when rotation command is received from Raspberry Pi.
- Receiving the analog voltage data from the batteries and transmitting the data to the Raspberry Pi.

3. Servo Motor

Servo motor rotates the food gate according to the amount of food calculated for each different cat, and it is controlled by changing the duty cycle of the PWM signal generated in Arduino. The motor unit is explained in more detail in the mechanical design part of this report.

4. Measuring the Food Level

The users are informed about the remaining food level in the reservoir. The food level is measured by the ultrasonic sonar sensor HC-SR04, which measures the distance between the food and sensor. The sensor generates an ultrasonic sound around 40kHz, and the amplitude of this sound is very low in order not to disturb the animals. The sensor contains two different transducers. The first transducer generates the sound wave, and the second transducer receives the echo of the transmitted wave. The distance is calculated by the following formula:

$$\text{distance} = \text{time} \times \text{speed of sound} \ / \ 2 \tag{1}$$

The generated sound is transmitted from the transducer, and then the wave bounces back from the object; the returned wave is received from the second transducer. The distance is divided by 2 since the sound makes a round trip between the sensor and the food. Sound speed is taken as 340 meters per second. The measured distances are proportioned with respect to the reference level, and volume measurement is obtained.

5. Measuring the Weight

A weight sensor is used in order to obtain the amount of food given to the cats. The sensor has three input wires. The sensor returns analog voltage values with respect to the applied force on the sensor. This analog voltage should be converted into digital data. An analog to digital converter HX711 integrated circuit is used, and the digital data is read through a port in the raspberry pi and scaled according to the measured weights.

b) Requirements

The requirements for the electronics subsystem are listed below.

- The subsystem should be rechargeable and the charging time should be less than 5 hours.
- Rechargeable batteries should be non-removable.
- Battery duration should be at least 5 hours.
- The subsystem should be wired correctly and properly.
- The subsystem should be able to operate during charging.
- The subsystem should be able to operate Raspberry Pi and Arduino.
- The subsystem should be able to drive the servo motor.
- Power consumption of the subsystem should be minimum.
- Status of the food supply should be observable.
- Battery level for both charging and operation modes should be observable.

- The subsystem should be able to measure the amount of given food.

c) Tests and Results, Future Test Plans

1. Test Procedure

- Measure the voltage of batteries.
- Charge the batteries with charger circuits until the voltages are balanced.
- Connect batteries to regulators.
- Measure the output voltage of regulators for different voltages inside the voltage range of battery.
- Connect relays.
- Power up the Raspberry Pi, Arduino, and the motor.
- Connect all sub-units properly.
- Measure the distance with sonar sensor.
- Measure the weight of given food.
- Drive the motor.
- On-Off control of the light bulb.
- Obtain images from the camera.
- Measure the maximum and minimum camera view distance.
- Drive the motor while camera is active.
- Measure the distance while motor and camera are active.
- Measure the weight while distance sensor, motor and camera are active.
- Measure the currents for power analysis.

2. Test Results

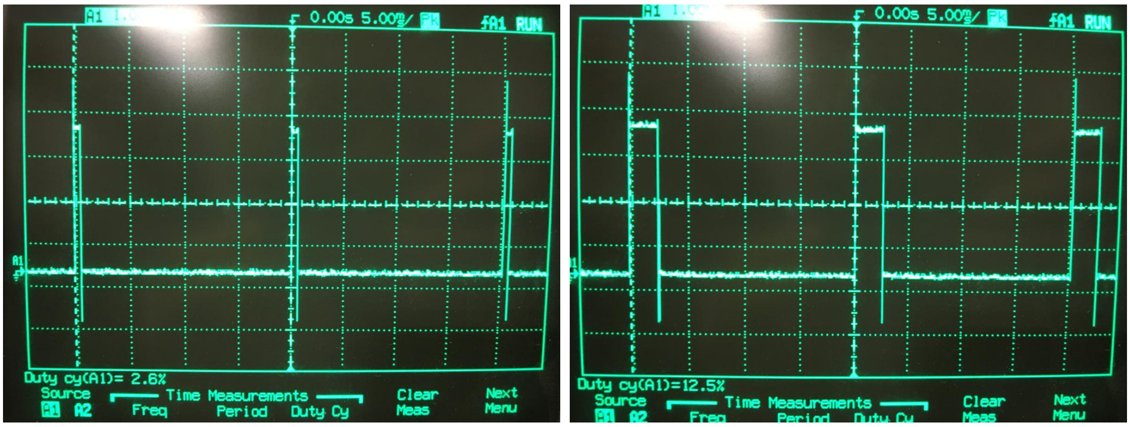
- Li-Ion batteries operates properly. Output voltage is between 3.7V and 4.2V.
- Parallel charging for two different power supply units with 3 charger operates correctly.
- Batteries work while charging. The subsystem requirement for re-chargeability is satisfied.

- 5V 1.2A regulators return 5.35Volts when the input is 3.7V and 5.45 Volts when the input is 4.2V. This is undesired, however does not cause any faults or performance loss for the operation of raspberry pi and arduino.
- Relays work correctly.
- All connections are done in a board and placed in a fixed layer. For charging operation, a micro-USB socket is placed outside of the box. Thus, the subsystem requirement for non-removable batteries is satisfied. Also, solders and connectors are used in wiring, and cable connections are optimized and hidden.
- Raspberry Pi, Arduino and Motor operates without a performance loss.
- Measured distances are correct if the distances are away from the sensor 2 centimeters and results are shown in Figure 25.

```
('Distance:', 3.69, 'cm')
('Distance:', 4.41, 'cm')
('Distance:', 6.07, 'cm')
('Distance:', 7.53, 'cm')
('Distance:', 8.57, 'cm')
('Distance:', 9.48, 'cm')
('Distance:', 10.5, 'cm')
('Distance:', 11.75, 'cm')
('Distance:', 12.48, 'cm')
('Distance:', 13.07, 'cm')
('Distance:', 14.82, 'cm')
('Distance:', 15.61, 'cm')
('Distance:', 16.94, 'cm')
('Distance:', 18.35, 'cm')
('Distance:', 18.62, 'cm')
('Distance:', 21.57, 'cm')
('Distance:', 22.89, 'cm')
('Distance:', 24.93, 'cm')
```

Figure 25: Measured Distances From Ultrasonic Sonar Sensor

- Weight sensor does not read small values, a better resolution sensor will be implemented.
- Motor is rotated according to the entered amount of food properly. Generated PWM signals can be seen in Figure 26.
- High quality images are obtained.
- A dog can be detected from the taken image from the camera with a distance range of 10 cm to 180 cm away from the box. Average detection distance is



(a) PWM Signal Duty Cycle = 2.5   (b) PWM Signal Duty Cycle = 12.5

Figure 26: Generated PWM signals from the Raspberry Pi

110cm.

- A cat can be detected from the taken image from the camera with a distance range of 15 cm to 153 cm away from the box.
- The system operates properly while motor and all sensors are working. The jittering issue mentioned in the conceptual design report is solved.
- The battery unit draws 0.84 A when they are fully discharged. Charging and operation times are mentioned in the power analysis section of the report.
- Currents are measured and written in the power analysis section of the report. The drawn currents for the units are reduced by using relays and power consumption is decreased.

d) Modifications

According to the test results, some modifications are applied to the system, as listed below.

- Fuses are added to obtain a safer system.
- Motor noise is eliminated. In the previous design, the motor is driven from raspberry pi, and raspberry generates an unstable PWM signal while the camera is operating. Arduino UNO is implemented in the system to obtain a stable PWM signal. The command still comes from the raspberry pi, and it is transmitted to Arduino by a serial communication channel using I2C communication protocol. When the command arrives arduino in milliseconds, the

motor rotates properly.

- Relays are added to reduce power consumption and get rid of adjustable voltage regulator. The relay connections are shown in Figure24.
  - Adjustable voltage regulator did not work properly due to an increase in its temperature. We used relays and got rid of this regulator. The relays RL1, RL2, and RL3 are used for this purpose. Servo motor MG996R operates in the voltage range between 4.8 V and 7.2V, and out batteries supply voltage in the range 3.7V-4.2V. We find a solution with 3 relays and 4 diodes for this problem. While the motor is supplied the batteries, two batteries are connected in series supplying 8.2V, this voltage is reduced 1.4 Volts by connecting series added diodes. Two parallel connected diodes are connected in series with another parallel connected diodes. Thus, voltage is reduced by 1.4 volts, and the motor can get the required current from the batteries. For charging circuit, relay1 and relay2 are switched and the series connected batteries are isolated. Batteries are charged separately, and with relay RL3, the motor supply is also switched and motor is supplied from the power outlet.
  - Relay RL4 shown in Figure24 is connected to reduce power consumption. The motor supply connections are turned of with this relay until a turn on command comes from the raspberry pi.

e) Compatibility Analysis

It is observed that the electronics system is reliable when it is connected with the other systems. All systems operate at the required performance and no error is observed. Although the system is working properly, the control mechanism of the motor will be calibrated according to the feeding regimes of the cats. Also, time calibration will be done for the light bulb. However, at some nodes, the long cabling might create

Insufficient description of the mechanics
lacks many parts

7 Mechanical Design

In this module, the following sections will be discussed.

- Subsystem Description
- Tests and Results, Future Test Plans
- Modifications
- Compatibility Analysis of Mechanics

a) Subsystem Description

Mechanical design is composed of two parts namely:

- Exterior design
- Interior design

The exterior design is composed of the outer design of the box. It should satisfy the upper conditions. The expected weight of the food reservoir and the electronic components respectively is 4 kg and 3 kg, which add up to 7 kg. An average person can lift %15 of their body weight which corresponds to 11 kg for a 70 kg person without giving any harm to their body [1]. Therefore, an ordinary person can easily carry our system. The expected size of the box is 40X30 cm^2 base and 53 cm height which its volume is 63 liter. The technical drawing can be seen in Fig 27.

From the figures, it can be seen that there exists a handle at the upper side for lifting. In addition to that, the half circle is being cut in order for food to flow out. This is the place where food will leave the box.

Interior design is mainly the design of the inner box. Inner box is composed of three sub-parts namely:

- Connection of electronic equipment.
- Food mechanism.
- Camera unit.

1. Connection between electronic devices

The unplanned connection between the electronic components results with a messy looked box, therefore the inner connections are arranged such that one can easily see the connections and reach the problematic area when there is an issue. In order to satisfy this condition, the electronic devices that will be work together are grouped

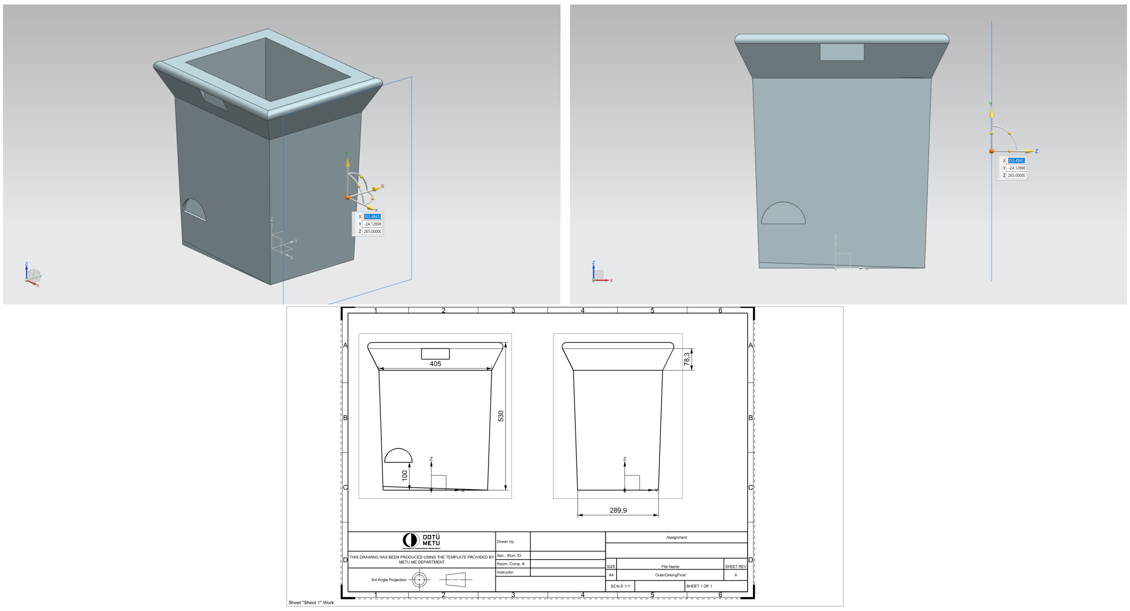


Figure 27: The Technical Drawing of the Outer Design

and put together on the solid surface. When there exists a problem in one specific sub-unit, the technician can reach that specific location and fix the problem easily. The board design can be seen on Fig 28. The details of the electronic components can be found in electronics part of this report.

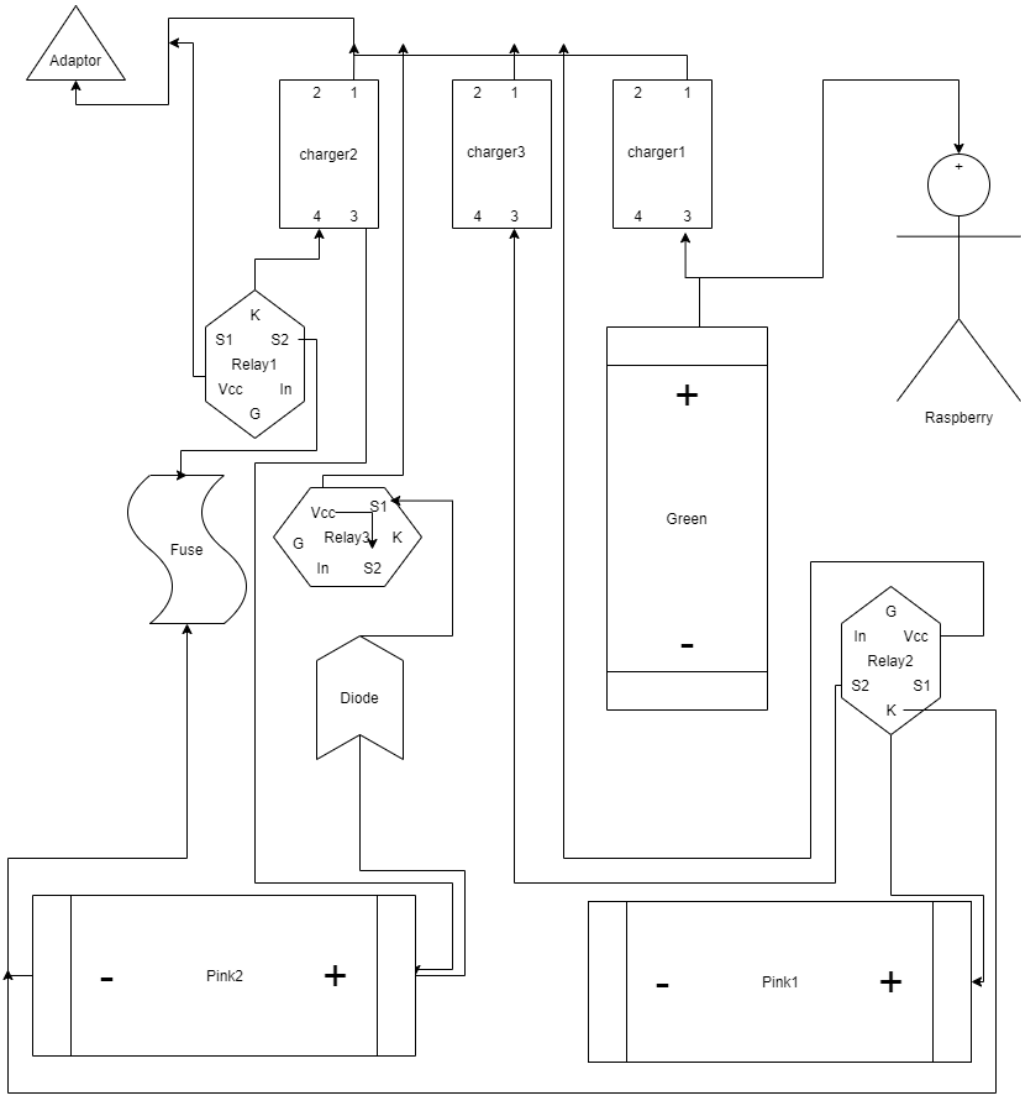


Figure 28: Electronic Board Design

2. Food Mechanism

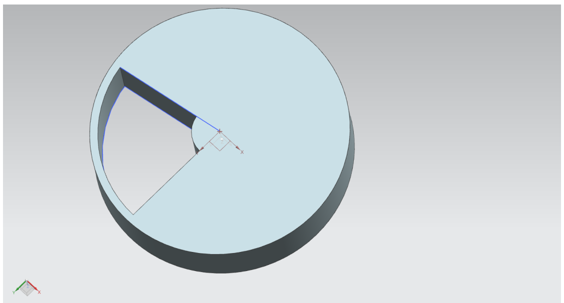
Food mechanism is designed such that it will provide minimum 100 meals of food for an ordinary cat and the food flow should be controllable.

An ordinary cat which weight 4 kg is advised to eat 60 gr of food per day [2], assuming that cat eats twice in a day, for one meal the amount is 30 gr. Therefore the food reservoir should sustain:

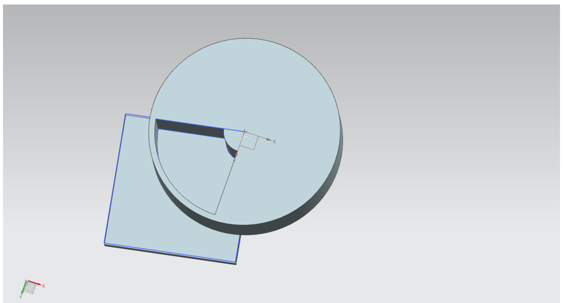
$$100x30 = 3000gr = 3kgfood \tag{2}$$

Our engineers had selected food reservoir as 5 liter, that can sustain 4 kg of food at full capacity.

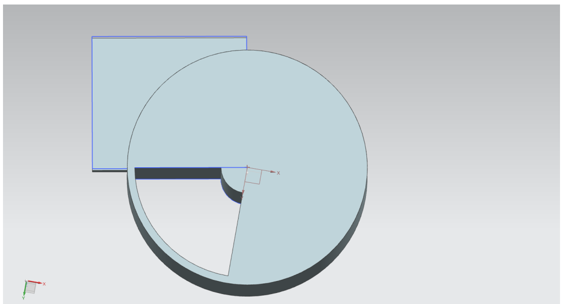
The main objectives of the food mechanism is to control the given food amount, to satisfy this condition a gate is designed. The below figures represents the gate.



(a) Cylinder



(b) Cylinder in Off state



(c) Cylinder in On state

Figure 29: The representation of the disk

The Fig 29 represents the states of the food mechanism. Food reservoir will be placed 1 cm upward of the cylinder shown in the Fig 29. Food reservoir's bottom will also be cut just like cylinder. By arranging the placement of the empty regions of cylinder and food reservoir one will change the status of the food mechanism. The gate has two states namely OFF and ON State. In the ON state food is given whereas in the OFF state food flow is stopped. When the state is OFF the empty side of the reservoir coincides with the empty side of the cylinder and food will stop the emptiness between the reservoir and cylinder (Fig 29b). Since there exists a blockage on under the cylinder, the food will not flow downward but it will be contaminated in the reserved area. The food flow toward the empty region will continue until it is fully filled.

The On state corresponds to the food given to the cat. When a cat arrives the servo motor rotates the cylinder shown in Fig 29c, then the empty side of the cylinder becomes free from the blockage of the barrier under it. In this way, the quantized food inside the reserved area will be given to the cat. While the reserved food is pouring the food path, the filled region of the cylinder will block the food flow from the reservoir to the cylinder, by this way only a quantized amount of food will be given to the cat at each cycle. The test results for this design can be seen at the Tests and Results, Future Test Plans part of Mechanical Design part of this report.

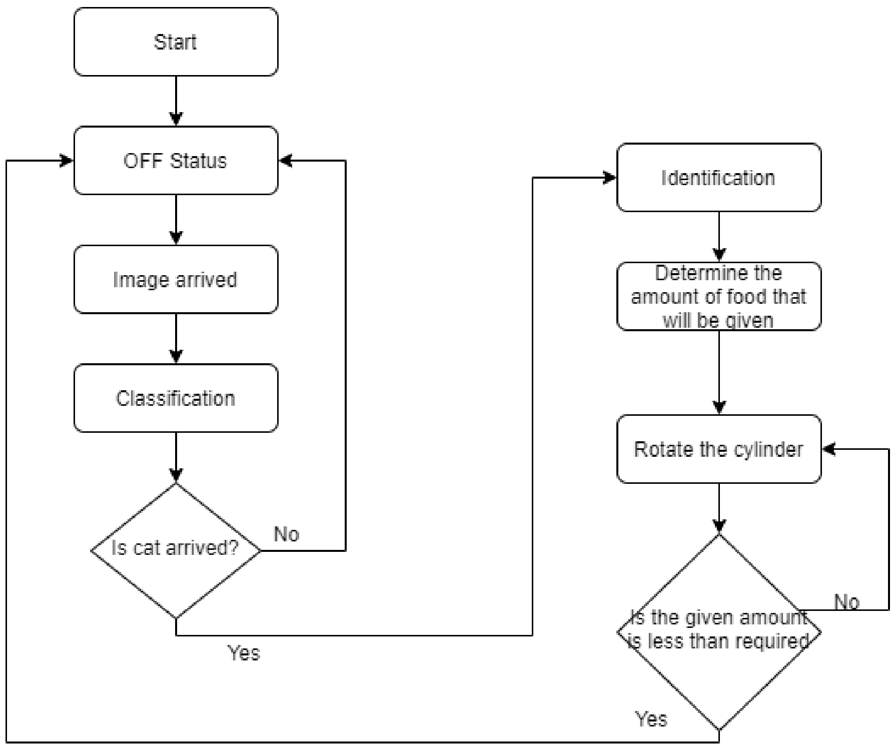


Figure 30: Working principle of Food Mechanism

In the Fig 30 one can observe the flow chart that represents the operation principle of the Food mechanism. Our mechanism is capable of feeding at the required amounts. By processing the incoming image, the food amount that should be given will be determined. The food will be given according to the amount determined by the identification of the cat. At each cycle of rotation, the same amount of food will be given, therefore by rotating the food mechanism at the necessary amount, one can control the given food amount.

b) Requirements

Mechanical part of the project should satisfy necessary conditions, which will be mentioned below

- Total system should be lifted by an ordinary person.
- Volume of the box should be enough for all electronic devices to fit in.
- Box should be durable for harsh outdoor conditions.
- Food reservoir should be provide at least 100 meals for an ordinary cat.
- There should be maximum %5 waste of food.
- Amount of food remaining, battery chargers while loading and in operation should be visible.
- Angle and the position of the camera must be adjusted properly in order to identify cats and dogs from 1.5 meter.
- There must be no obstacle in front of the camera unit, so that it can capture accurate photographs.
- Camera unit should be adjusted properly so that it should be protected from animal attacks.

c) Tests and Results, Future Test Plans

Both inner and outer designs will be tested in order to check the systems capability. The test procedures and results can be seen below.

1. Food Mechanism

Description of the Test for Food Mechanism Test was conducted in order to observe the behaviour of the food flow. The purpose of this test is:

- Determine the quantity of the average amount of the food poured.
- Determine if there any food is jammed between reservoir and the cylinder. If there exist some, determine if they will harm to rotation or not.

- Determine if motor is capable of rotating without difficulty.

Procedure can be followed below.

- Fully fill the reservoir with food.
- Observe the food flow in reservoir in off condition.
- Rotate the cylinder and observe the amount of food flow.
- While rotating observe the food flow from reservoir to the solid part of the cylinder. Observe if food is flowed from the solid part of the cylinder.
- Observe if there exist any food jammed between the solid part of the cylinder and the reservoir.
- Return the cylinder to its old position.
- Observe if the flow of the food is completely stopped.
- Turn the cylinder again and observe the food flow again. Compare with the previous amount of food flowed.
- Repeat this process until reservoir is fully empty.

The expected quantitative test results from the test is:

- Food should flow through the inside of the empty region of the cylinder only. There shouldn't be any food flowing out of the cylinder.
- The cylinder should completely cut the food flow from reservoir when system is in ON state.
- The jamming between cylinder and reservoir is expected however, the motor should be able to rotate the cylinder without any difficulty.

Test Results for Food Mechanism The test mentioned above is conducted and results can be seen below.

- It is observed that the average poured food per rotation is 18 gr.
- Although food is jammed between the reservoir and cylinder, it did not affect the performance of the overall system. The rotation continued without any difficulty.
- The jammed food also poured out of the box at following rotations, therefore it is observed that there can be an underestimation on the amount of the food given if one only considers the poured food is only due to the food that was

stored in the cylinder. However, the amount of the food poured which was the resultant of the jammed food is %7 on average of the total poured food.

- The variation on the amount of the food flow between rotations did not differ much. At maximum case, the amount of the food flow is 60 gr. Generally, the mass of the poured food is between 10 gr to 20 gr.
- At 86 rotations the poured food is between 10 gr to 20 gr.
- At 13 rotations the poured food is between is less than 10 gr but not zero.
- At 18 rotations the poured food is between 20 gr to 60 gr.
- The food flow is completely stopped when system is in OFF state.
- It is observed that when system switches from the ON state to OFF state, there exist an undesirable food flow resulting from the previously jammed food.
- It is observed that the food in the reservoir is not homogeneously poured but the food at the open side of the reservoir finishes more quickly. This creates a mountain-shaped distribution which results in an accumulation of food at one side in the reservoir.

The following bar graph (Figure 31) shows the amounts of the food poured. Where

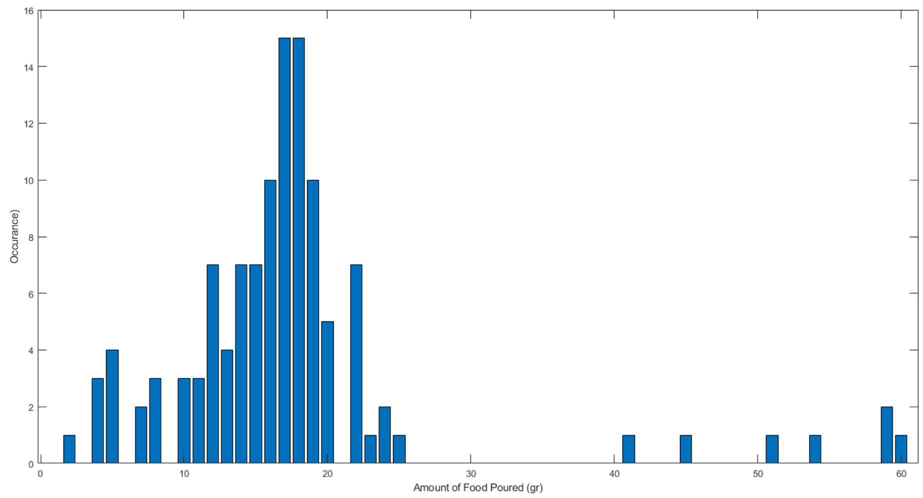


Figure 31: The Amount of Food Poured for Different Trials

x label shows the amount of the food poured and y label shows the trial number which corresponds to that poured mount. It can be seen that graph peaks at 18-19 gr food amount and this amount can be changed by varying the rotation angle.

From previous test, which was mentioned in conceptual design report was resulted with an undesired results. According to this results the given food amount is not controllable, in addition to that there exist lots of jamming of food between reservoir and cylinder which prevented motor to turn easily. These problems are solved by rearranging the placements of cylinder and reservoir. Now they are more closely placed that jamming is minimized. In addition to that, the motor was poperly fixed to its place such that when it rotates, it does not wobble which now the food flow has become controllable. The results of the new experiment shows that food system is reliable but has some errors.

From the result of the experiment it can be said that food mechanism works properly for but there exists some errors. In two trials poured food is 59 gr which is happened at the first trials of the experiment. It shows us that system needs to be settle down before it operates properly.

The experiment also shows that when the food amount is lowered in the reservoir, the poured food also decreases, since the food flow from the reservoir to the cylinder is not homogeneous.

2. Exterior Design

The purpose of the tests is to determine the reliability of the box.

- Apply force on the box and check if there exist any considerable damage seen at the outside of the box. If not continue.
- Place electronic devices and the food mechanism inside the box and apply force on the box. Check if electronic devices, camera and food mechanism works properly. Check the angle of the camera. Observe if it works functional. Repeat this process 5 times.
- Raise the box to half a meter height and release it to the tile surface.
- Observe if there exist any considerable damage seen at the outside of the box. If not continue.
- Check if electronic devices, camera and food mechanism works properly. Check the angle of the camera. Observe if it works functional. Repeat this process 5 times.

The measure of success for this experiment is having non-considerable damage on the box and it is desired that all electronic devices and the food mechanism work properly, meaning that all electronic devices and food mechanism works as if before. Moreover, the camera can move a little but it is expected to capture photographs that is useful for identifying cats and dogs.

d) Modifications

1. Food Mechanism

According to the test conducted on the food mechanism, it is seen that the food given in each cycle is close to each other. It is reliable, however there exist an undesired situations such as the food flow between the switching states. In order to solve this problem, first the food squeze between reservoir and cylinder should be minimized. Therefore more careful placement of the cylinder is required. In addition to that adding a food path between cylinder and reservoir can solve this problem.

It is observed that food flow from the reservoir is not homogeneous, moreover the food at the non-empty side of the reservoir is not poured through the cylinder. In order to solve this problem, our engineers will develop a ramp which will force food to flow through the empty region of the reservoir.

Test results show that, in first two or three trials after food is placed into the reservoir system poured more than expected. To solve this problem another food path will be developed and will be placed inside the reservoir, which will homogeneously fill the reservoir. By this way the first poured food into reservoir will not directly stuck in between food mechanism and reservoir but it will be placed into the empty region in Fig 29.

e) Compatibility Analysis of Mechanics

It is observed that the food system is reliable when it works alone, but what if all circuit operates together. Previous tests showed that Raspberry Pi zero creates a soft pwm that suffers from glitches when the camera unit and food mechanism operates together. Therefore, in order to drive food mechanism, another microcontroller will be used and its pwm can be seen in Fig 26 in the electronics of this report

In addition to that, food system is ~~reliable in~~ terms of the amount of the poured food. In each cycle the poured food is similar. The servo motor did not face any hard time while rotating the cylinder ~~although~~ there is a jamming between reservoir and cylinder. The torque of the servo motor is capable for this project.

vague statements

this is not a compatibility analysi

as required.

8 Power Analysis

Component	Average Idle Current (A)	Average Idle Power (W)	Max Current (A)	Max Power (W)
Raspberry Pi Zero WH	0.18	0.9	0.42	2.1
Arduino Uno R3	0.04	0.17	0.07	0.297
Servo Motor	0.03	0.1275	0.22	0.935

Table 5: Power Analysis of the Project

The Table 5 shows the power consumption under idle and maximum cases. The value of the idle case is the average over 20 minute measurement. The maximum case occurs while the camera is working. Power distribution diagram is shown in Figure 32.

Weight sensor and sonar sensors and analog to digital converter are supplied from the raspberry pi since they do not require high current.

Battery unit one shown in Figure 12 contains two Li-Ion batteries connected in series, which supplies 7.4 V to the servo motor. The idle current of this unit is 0.03 A, and the maximum current that occurs during the rotation is 0.22 A. The maximum current occurs for 0.2s, and then it returns to its idle case. The lithium-ion batteries used in this unit have 2000mAh capacity. This unit can operate for 90 hours on the batteries at its maximum power-consuming operation according to mathematical calculations. However, this time is not tested in the system, and it will be less than 90 hours due to the efficiency of the batteries. But, the real-time will be higher than 5 hours. Thus the requirement about the operating time of 5 hours is satisfied.

Battery unit two shown in Figure 12 contains two Li-Ion Batteries that have 2650mAh capacity connected in parallel. Hence, the overall battery capacity is 5300mAh, and the unit supplies both Raspberry Pi and Arduino Uno. Raspberry pi draws 0.18 A average on the idle case and 0.42 A on the maximum case. The maximum case occurs when the camera module is activated. Also, Arduino draws 0.04 amperes average on the idle case and 0.07 A on the maximum case. Hence, the maximum current driven from the supply is 0.49A, and the system is able to operate on the batteries for at least 10 hours. Therefore, the requirement for operation time is satisfied.

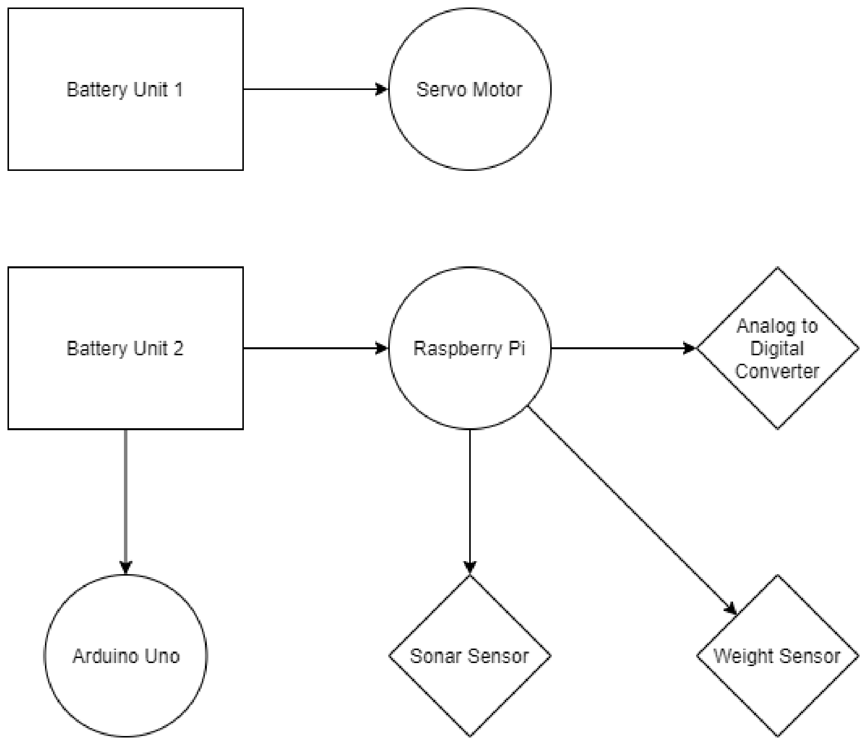


Figure 32: Power Distribution Diagram of the Project

9 Cost Analysis

After choosing the components used in the design, a price comparison for each component has been made to minimize the cost without performance loss. The components are shown in the Table 6.

For microcomputer unit, we bought Raspberry Pi Zero WH which is the least expensive Raspberry containing a wireless module inside it.

For camera unit, three different options are compared in terms of cost and performance, which are Raspberry Pi camera module v1, Raspberry Pi camera module v2, and a webcam. The performance of the webcam does not satisfy our image quality requirements, and the camera module v2 is expensive, Raspberry Pi camera module v1 is selected as the camera.

For motor unit, servo motors MG996R, SG90 are compared. Although the SG90 motor is cheaper than MG996R, its output torque does not satisfy the required rotation for the mechanical part. Hence, MG996 motor is selected.

For batteries, Li-Ion, Li-Po, and power bank options are compared in terms of the performance, capacity, and price, and 18650 Li-Ion batteries are selected.

For distance measurement unit, ultrasonic sonar sensor HC-SR04 and infrared sensor GP2Y0A710K0F are compared in terms of performance and price. The ultrasonic sensor is selected due to its lower price.

For cables, regulators, relays, and other components, unit prices are similar compared to different options, so the selection is made with respect to the performance.

Unnecessary space

Item	Quantity	Price ($)	Total ($)
Raspberry Pi Zero WH	1	20	20
Raspberry Pi Camera V1	1	16	16
Raspberry Pi Camera Cable	1	3.5	3.5
Raspberry Pi Camera Case	1	4.5	4.5
16 gb MicroSD Card	1	5	5
2650 mAh Li-Ion Battery	4	5	20
1A Lithium Battery Charger	3	1	3
Micro USB Adapter	1	4	4
MicroUSB-USB Cable	1	4	4
5V Voltage Regulator	1	2	2
Ultrasonic Sonar Sensor HC-SR04	1	1	1
Weight Sensor	1	1.5	1.5
Servo Motor MG996R	1	1	5
5V Relay	4	0.8	3.2
Arduino Uno R3	1	6.5	6.5
Mechanical Components	-	-	20
Outer Box	-	-	20
3D Printed Components	-	-	13
		Total Cost	153.2

Table 6: Estimated Cost Analysis for the Project

10 Deliverables

There will be five deliverable that Felerest will provide to users.

The products and services that will be offered are listed below.

- The box that only feeds cats,
- A mobile/web application that will show food, water and battery level.
- Customer oriented additional features to the mobile/web applications.
- A thorough user manual for the system setup, reservoir refilling process, and the mobile/web application,
- 3-year warranty that excludes user faults,
- Charger for the battery.

11 Conclusion

Looking at the environment one can see the food leftovers that cause visual pollution which also leads to health and hygiene problems. Felerest are determined to solve these problems while protecting our beloved friends. To do so, Felerest engineers are determined to create novel ideas which will lead the field of automatic cat feeding systems. Our goal is to improve the life standard of the cats and humans by using the blessings of technology.

This report's aim was to showcase the final design that was preferred over other choices in the Conceptual Design Report. All modules were thoroughly explained and backed up with related test results.

The purpose of this project is not to only feed cats, but also to start a new product that considers cat needs. Felerest also offer features for extensive cat feeding such that it respects the cats for the required amount of food while also determining their eating behaviours, without any human interaction. This project is useful for the owners which can not leave home due to their cats and for those cats who lives in the streets or in sanctuaries who are in need of human care. Happiness and healthy care of cats to get customer satisfaction is the main purpose of Felerest team. Every module's test results, planned future test ideas, the foreseen risks and disadvantages were discussed thoroughly. Furthermore, each engineer's following responsibilities and the role division was separated reasonably. This report includes flow charts that explain the algorithms implemented in the computer vision part, 3D designs of the mechanical structures, pictures of test results for each subsystem, compatibility analysis of overall system, expected power consumption, and cost analysis. This report was written in order to show a clear picture of the project for customers and members of the company. We believe that customers should always have the chance to personalize their products. This is why Felerest provides customer oriented additional features to the mobile/web applications.

With this report, Felerest team shows that it is capable of taking this project to the next level and mass produce autonomous cat feeding systems. All sub-modules are up to par and ready to be a part of the end result. This is demonstrated by test results and relevant pictures.

Our vision for the future is to market these feeding systems commercially available to everyone around the world. This way our furry little companions are never hungry and are always well taken care of. This will surely be an important step for the pet society, making it the first ever connected, smart, cat feeder.

References

[1] Cats: most interesting facts about common domestic pets. https://web.archive.org/web/20141006105806/http://english.pravda.ru/society/family/09-01-2006/9478-cats-0/.

[2] How much food you must give to your cat.

[3] Human interaction with cats. https://en.wikipedia.org/wiki/Human_interaction_with_cats#cite_note-3. Accessed: 2019-02-25.

[4] Si110: Digital cryptography.

[5] Simple image classification using convolutional neural network — deep learning in python.

[6] What do cats think about us? you may be surprised. https://www.nationalgeographic.com/news/2014/1/140127-cats-pets-animals-nation-dogs-people-science/. Accessed: 2019-1-25.

Appendices

Why here ?

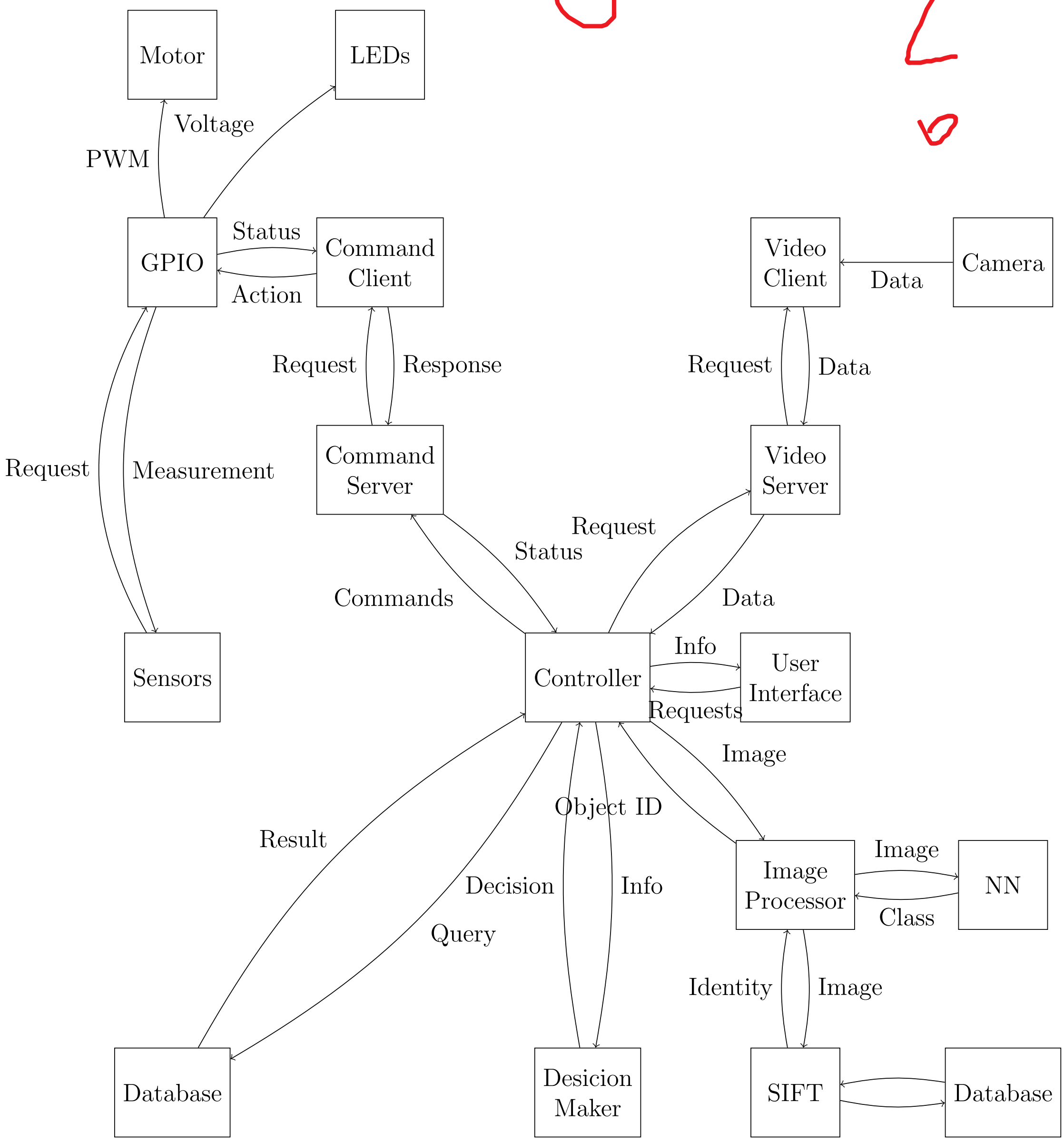


Figure 33: Overall System Diagram

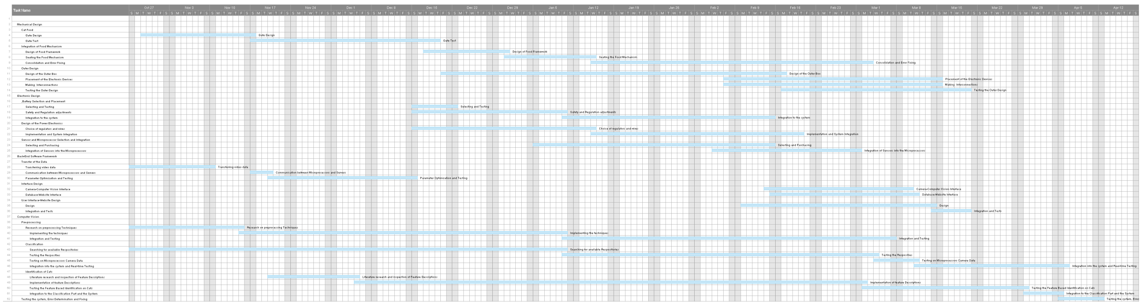


Figure 34: Gantt Chart for All Dates

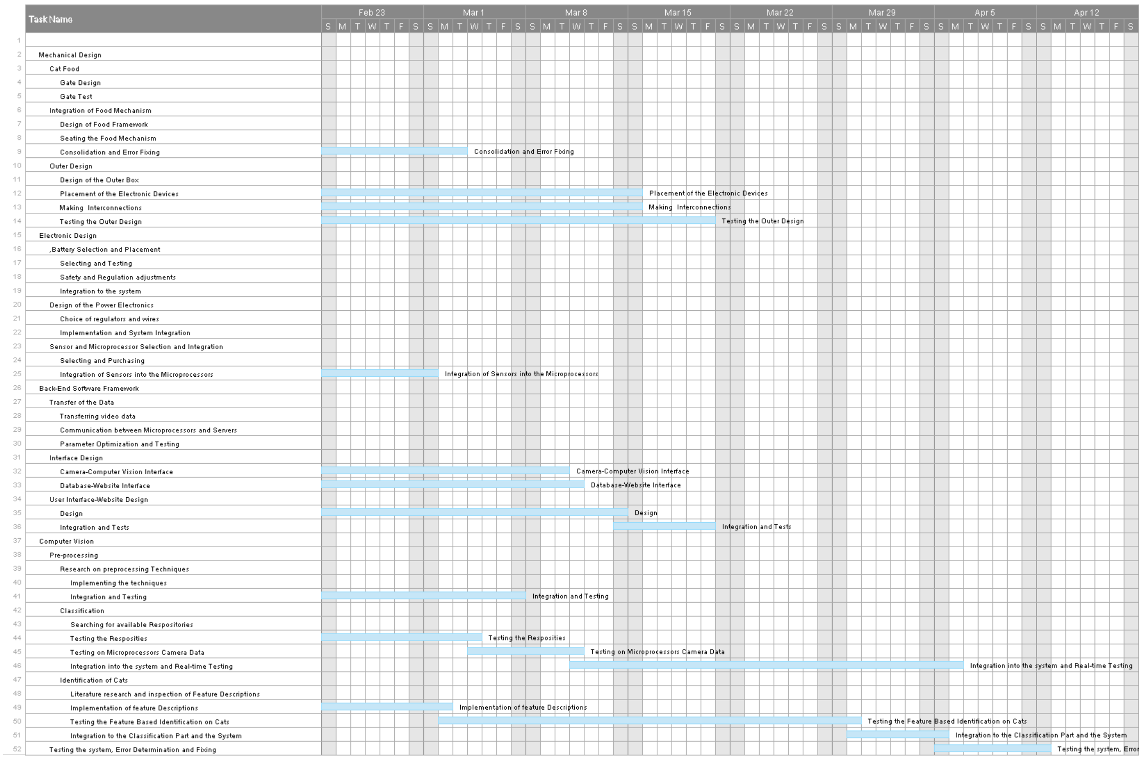


Figure 35: Gantt Chart Starting from February 23