GERMAN - MONGOLIAN INSTITUTE
FOR RESOURCES AND TECHNOLOGY

The present work was submitted to the Faculty of Electrical and Mechanical Engineering.

Automated trash sorting system

First Supervisor: Myagmarjav. B

Second Supervisor: Narangarav.T

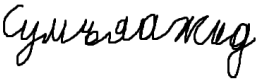
Student: Sumiyajid Ulziibayar

Ulaanbaatar/Nalaikh, 2025/11/14

Statutory Declaration

Sumiyajid Ulziibayar	B2000505
Last Name, First Name	Student ID Number

I hereby affirm an oath that I provided the submitted bachelor's thesis. I did not use any sources other than those stated. In case the work is additionally submitted on a data medium, I declare that the written and the electronic form are completely identical. The work was not submitted in the same or similar form to any examination authority.

Nalaikh, 2025 November 14	
Place, Date	Signature

Acknowledgments

The completion of this bachelor's thesis was a challenging journey that I would not have been able to complete without the help of these individuals, to whom I am deeply indebted.

Firstly, I would like to thank all the professors and teachers who have taught me a great deal about not only engineering and science, but also about life's ups and downs, social skills, ethics, and much more. From these well-respected individuals, I would like to specifically mention my dearest teacher, Mr. Myagmarjav, whose unwavering belief in me was the primary source of my motivation in completing not only this thesis but also the past few years. Thank you for your mannerisms towards me.

Secondly, the guidance and motivation of the staff of Academic Affairs and other staff members were invaluable to me. I have deeply enjoyed these last few years of this GMIT journey and hope there will be more to come in the future.

Finally, I am profoundly grateful to my family for their continuous patience, understanding, and love.



Table of Contents

Statutory Declaration 1

Acknowledgments 2

Table of Contents 3

List of abbreviations 5

List of figures 6

List of tables 6

Abstract 7

Introduction and Project Context 8

1.1 Global Waste Management Crisis 8

1.2 Waste Management in Mongolia 9

1.3 Motivation and Objectives 9

1.4 Thesis Scope and Structure 10

1.4.1 Scope of Thesis 10

1.4.2 Thesis Structure 11

Literature Review 13

2.1 Manual separation 13

2.2 Mechanical & Electrical separation techniques 14

2.2.1 Mechanical sorting 14

Air classification – 14

Mechanical screening and sieving – 14

Density-based separation – 14

2.2.2 Electrical sorting 15

The eddy current separation – 15

Magnetic separation – 15

Electrostatic separation 15

2.2.3 Sensor-based Algorithmic separation 16

2.3 IoT and AI-based separation 16

2.4 Theoretical background of small-scale AI models 18

2.4.1 Biological inspiration 18

2.4.2 Artificial Neurons 19

2.4.3 Neural networks and the training process 21
2.4.3.1 FNN 21
2.4.3.2 Training or learning process 22
2.4.4 CNN 24
The convolutional layer 25
The pooling layers 26
Methodology 28
3.1 System design and architecture 28
3.1.1 System requirements 29
1. Functional Requirements 29
2. Hardware Requirements 30
3. Software Requirements 30
4. Environmental Requirements 30
5. Performance Requirements 31
6. Non-Functional Requirements 31
3.1.2 Hardware components 31
Connection and Circuitry of the Hardware components 33
3.1.3 Power Management 35
3.2 Implementation, Testing, and Evaluation of the CNN 37
3.2.1 Data Collection, Augmentation, and Training 37
3.2.2 Implementation & Testing 38
3.2.3 Evaluation 38
3.2.4 Mongolian waste database 39
3.3 Integrated System Software and Operation 41
3.3.1 Communication between components 41
3.3.2 Web app integration 43
3.3.3 Database Structure 45
Discussion 48
Conclusion 50
References 51
Appendix 52

List of abbreviations

Abbreviation	Definition
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
FNN	FeedForward Neural Network
API	Application Programming Interface
LLM	Large Language Model
DC	Direct Current
AC	Alternating current
ESP32	Espressif Systems 32-bit Microcontroller
GPIO	General Purpose Input/Output
IoT	Internet of Things
IR	Infrared
NIR	Near-Infrared
NEMA	National Electrical Manufacturers Association
PCB	Printed Circuit Board
PWM	Pulse-Width Modulation
RGB	Red-Green-Blue
UART	Universal Asynchronous Receiver-Transmitter
MQTT	Message Queuing Telemetry Transport
JSON	JavaScript Object Notation
HTTP	Hypertext Transfer Protocol
A/D	Analog to digital



List of figures

1. Figure 1. Comparative visualization of a neuron and a perceptron.
2. Figure 2. Visualization of different functions.
3. Figure 3. The simplest version of a Neural Network.
4. Figure 4. Sample 3D projection of a Gradient descent landscape.
5. Figure 5. The gradient descent plane of an algorithmic trading bot's stop loss and take profit ratio.
6. Figure 6. Convolutional layer operation.
7. Figure 7. Pooling example.
8. Figure 8. Flowchart representation of a typical CNN model.
9. Figure 9. Example Raspberry pi connections.
10. Figure 10. Example ESP-32 connections.
11. Figure 11. Full Circuit diagram with Solar panels and Power supply line.
12. Figure 12. Gmit.life webpage.
13. Figure 13. Web Appliccation dashboard
14. Figure 14. Simplified Communication Flowsheet.

List of tables

Table 1. Main components order history.

Table 2. Comparison of different CNN models

Table 3. Database structure



Abstract

The proper sorting of household waste remains a significant challenge globally, often hampered by complex logistics and inconsistent public participation. This thesis presents the design and implementation of a cost-effective, automated trash sorting system specifically engineered to address the issue locally, in Ulaanbaatar city, at the request of Nalaikh District. The system aims to actively motivate and train the population, particularly children in educational settings, to sort recyclables responsibly.

The initial project scope, which involved sorting six distinct waste types using multiple sensor modalities, was narrowed due to practical and economic constraints, particularly the difficulties associated with safely storing glass and organic waste within a compact unit. We have decided to use an integrated embedded system built around a Raspberry Pi and an ESP32 microcontroller. This system utilizes a camera-based machine learning model for classification, categorizing inputs as Plastic Bottles, Metal Cans, or Others (namely, paper, cardboard, glass, and organic materials), combined with weight sensor data.

The mechanical separation is handled by a combination of a stepper motor and/or a servo motor integration to make the unit more scalable. One of the key elements of the project is the development of a server-based tracking and reward system, which leverages the motivation of group competition to promote consistent usage and habit formation among its target demographic. The system achieved a classification accuracy exceeding 90% in testing, demonstrating that a highly focused, integrated engineering approach can provide a scalable and financially viable tool for improving localized recycling compliance.



Introduction and Project Context

1.1 Global Waste Management Crisis

Human activities have resulted in the highest waste production levels throughout history. People produce billions of tons of waste annually, while the waste production rate continues to rise rapidly. The World Bank predicts that waste production will increase by 70% between 2025 and 2050 unless countries implement effective waste management solutions [1]. The expanding waste problem has multiple negative impacts that harm public health, environmental sustainability, and create social inequalities.

Multiple elements within society have contributed to the current waste management crisis. The increasing number of people in cities, combined with population growth, results in higher waste production at unprecedented levels. The growing demand for disposable plastics and packaged items worldwide continues to exacerbate the waste management problem. The World Bank (2025) reports that waste collection systems in low- and middle-income countries often fail to manage waste effectively, as 60% of garbage remains uncollected and substantial amounts end up in open dumps [2].

The current waste management practices lead to multiple severe environmental problems. The presence of trash in waterways and on streets creates damage to wildlife populations while simultaneously contaminating the environment. The practice of open waste burning produces hazardous atmospheric pollutants, which exacerbate the effects of climate change. People residing near uncontrolled waste disposal sites face multiple health threats, as their water supply, air quality, and soil health become contaminated (UNEP, 2024) [3].

Current waste management systems operate with outdated methods. Several nations, including Mongolia, lack effective recycling and composting programs, while their landfills operate without proper management. The recycling process becomes challenging because modern products contain a variety of materials, including electronics and mixed-material packaging, making it difficult to separate and recycle them effectively. The construction of improved waste management systems faces financial and governance obstacles, which primarily affect developing nations.



1.2 Waste Management in Mongolia

The worldwide waste problem continues to expand, but solutions exist to solve it. Countries can safeguard their environmental resources and public health by implementing waste reduction initiatives, establishing effective collection systems, and developing circular economic frameworks. The solution to waste management presents itself as both an environmental solution and a social and economic development opportunity.

Waste sorting isn't just a technical issue; it's a behavioral and cultural challenge. Countries like Germany and South Korea have succeeded in recycling because waste segregation is deeply integrated into everyday life. Germany's Pfand system, for example, allows consumers to reclaim deposits by returning recyclable containers, resulting in recycling rates above 67% [4]. Meanwhile, in Seoul, the implementation of the pay-as-you-throw (PAYT) system in 2013 led to a 10% decrease in household food waste within one year. Unfortunately, such systems cannot be directly transplanted into Mongolia without addressing differences in infrastructure, economic conditions, and public attitudes toward waste and politics.

Mongolia, like many developing countries, faces a growing waste management crisis. According to the Asian Development Bank, the country generated over 3 million tonnes of solid waste in 2019, with Ulaanbaatar alone contributing nearly 60% of it [5]. Shockingly, only about 7% of this waste is recycled, while the rest is transported to open dumpsites, such as the Nalaikh and Tsagaan Davaa landfills. These sites are nearing capacity and pose serious environmental and health risks due to leachate contamination and spontaneous combustion.

The Nalaikh district, where this project originated, represents both the challenge and opportunity of waste management reform. The district authorities have expressed growing concern about uncontrolled dumping and weak enforcement of sorting laws. While several local initiatives have encouraged community-based recycling, participation remains low due to a lack of incentives and sorting infrastructure.

The Nalaikh District Municipality proposed this project as part of a pilot effort to modernize community waste handling through technology and behavioral design. The system aims to motivate residents to sort recyclables properly by introducing a reward-based automated sorting bin.

1.3 Motivation and Objectives



The motivation for developing this system stemmed from two intertwined issues:

- The technical inefficiency of manual sorting, and
- The behavioral inertia prevents people from separating recyclables.

In Mongolia, waste segregation is often seen as a chore with little reward. However, studies have shown that when sorting behavior is reinforced through gamified incentives or visible social recognition, participation rates increase dramatically [6]. This insight inspired the design of an automated system that not only sorts waste but also engages users in the process, conditioning people's behavior to sort their waste responsibly through reward-based operant conditioning.

The primary objectives of this project are therefore:

- To design and implement an automated trash sorting bin capable of identifying and classifying waste materials using computer vision.
- To establish a behavioral incentive system, encourage consistent waste sorting through feedback and rewards.
- To evaluate the technical and social feasibility of deploying such systems in Mongolian schools and community centers.

Ultimately, this project aims to demonstrate how engineering, artificial intelligence, and human-centered design can converge to tackle one of Mongolia’s most pressing urban challenges.

1.4 Thesis Scope and Structure

1.4.1 Scope of Thesis

The primary objective of this thesis is to design, develop, and validate an Automated Trash Sorting System that is optimized for cost-efficiency and motivational impact within a community setting. The final system scope was streamlined to focus on the following key functions:

1. **Classification:** The system is designed to accurately classify input items into three or more distinct categories: Plastic Bottles, Metal Cans, and Other.



2. **Actuation:** A reliable actuation mechanism driven by a stepper motor and/or servo motors to physically separate the classified items into their respective bins.

3. **Data Integration:** Integration of a camera-based machine vision system with a high-precision weight sensor (load cell) to enhance identification accuracy and provide quantifiable metrics for the reward system.

4. **Reward System:** Development of a server-side framework for user tracking and the implementation of a group-based competition mechanism, specifically targeting children in the school environment of the Nalaikh district.

The project is constrained by a budget and the necessity of using readily available, modular components (Raspberry Pi, ESP32) to ensure the final product is **scalable and easily replicable** in similar community settings. Performance is measured by classification accuracy (must exceed 90%) and sorting throughput.

1.4.2 Thesis Structure

This document is organized into six main parts, detailing the transition from conceptual design to the final validation of the automated sorting system:

- **The Introduction** and Project Context section explains why automated sorting needs to be implemented while establishing the project objectives and determining the complete range of work to be performed.

- **The Literature Review** and Background provides a comprehensive overview of existing waste sorting technologies, with a specific focus on the principles of machine vision and AI (CNNs).

- **System Design and Architecture** details the hardware architecture, electrical connectivity, and justifies the selection of all major components.

- **Implementation, Testing, and Evaluation** explains the process of data acquisition and preprocessing, outlines the model benchmarking methodology, and details the implementation and final evaluation of the chosen machine learning architecture.

- **Integrated System Software and Operation** focuses on embedded software design, communication protocols between microcontrollers, sensor calibration procedures, and the specific implementation of the user reward system.



- **The Discussion and Conclusion sections** present the engineering trade-offs made during development and conclude with recommendations for future work.

Literature Review

This concern about waste management was not something that arose overnight; the problem had existed for years, and efforts to mitigate the threat had been developing in tandem with its growing concern. As humans, we have developed numerous technologies and methods to separate waste long before the AI boom. This part of the thesis aims to explore various techniques for achieving waste separation in greater detail.

Moreover, at the start of this project, in June and July, the project's scope was ambiguous. It was unclear whether the objective was to design a large-scale waste sorting system for the dumpsites in Nalaikh District, or to develop a smaller, IoT-enabled trash bin intended to enhance user convenience, and even what kinds of trash needed to be sorted. Consequently, portions of this literature review may address some topics that are not fully aligned with the finalized project focus.

2.1 Manual separation

Manual separation represents the most fundamental and historically prevalent method of waste sorting. This process relies entirely on human labor to identify, categorize, and segregate different types of waste materials. Typically conducted on picking belts at Material Recovery Facilities (MRFs), workers manually remove specific recyclables (like PET bottles, aluminum cans, and cardboard) from a moving stream of mixed waste.

The primary advantage of this method is its flexibility and low technological barrier. Human sorters can adapt to new or unexpected waste items and make nuanced decisions that may challenge rigid automated systems, such as identifying a soiled paper container that is no longer recyclable. This makes manual separation a crucial initial or supplementary step in many recycling facilities worldwide, particularly in regions with lower labor costs.

However, manual separation suffers from significant drawbacks that limit its efficiency and scalability. These include:

- **Low Throughput and High Cost:** The process is inherently slow and labor-intensive, resulting in high operational costs, particularly in developed countries. The sorting speed is limited by human physical capacity and concentration.



- **Inconsistency and Human Error**: Sorting accuracy is highly variable and can decline due to factors such as fatigue, boredom, or inadequate training. This inconsistency leads to high contamination rates in sorted material streams, reducing their market value.
- **Health and Safety Risks**: Workers are exposed to hazardous materials, sharp objects, biohazards, and dust, posing significant health risks. The repetitive nature of the work can also lead to musculoskeletal injuries.

Given these limitations, the global trend is toward automation to improve efficiency, safety, and sorting purity. The automated system proposed in this thesis seeks to address these very issues at a smaller, community-based scale.

2.2 Mechanical & Electrical separation techniques

2.2.1 Mechanical sorting

The oldest and most dependable waste separation method exists through mechanical sorting. The system utilizes physical characteristics, including size, shape, and density, to categorize materials into distinct groups. The initial stage of recycling facilities relies on this process because it reduces human involvement while increasing operational speed.

Air classification –

– separates light materials, including paper and plastics, from heavy materials, including glass and metal, through controlled air flow systems. The waste material moves through an air stream that rises or falls while the system separates materials through their aerodynamic characteristics. The system demonstrates superior performance when it separates lightweight packaging materials from various types of municipal waste.

Mechanical screening and sieving –

– uses perforated screens and rotating trommels to separate materials based on their particle size. The basic design of these devices enables them to remove large and small particles from the waste stream before the next stage of separation takes place. The screening process achieves its best results when operators maintain specific conditions for feed moisture levels, screen angles, and vibration rates.

Density-based separation –



– depends on flotation and sink-float principles to achieve its goals. The process of submerging materials in density-controlled fluids allows plastics to rise to the surface while metals and glass remain at the bottom. The technique proves most useful for plastic recycling because it allows the extraction of polyolefins from different types of plastic waste .

2.2.2 Electrical sorting

The electromagnetic characteristics of materials enable electrical sorting methods to perform separation operations. The systems excel at separating metallic from non-metallic materials because they deliver both high recovery performance and precise operation.

The **eddy current separation** –

– method stands as the leading electrical technique for waste separation. The process of electromagnetic induction allows non-ferrous metals to generate induced currents when they move through fast-spinning magnetic fields, which produce repulsive forces that push them out of the waste stream. The technique provides excellent results for extracting valuable non-ferrous metals from processed municipal and industrial waste materials.

Magnetic separation –

– serves as a fundamental electrical sorting method, extracting ferrous materials from waste streams that contain various materials. The selection of magnets depends on the material targets, particle dimensions, and desired separation outcomes.

Electrostatic separation

The recycling process of fine granular materials uses electrostatic separation as one of its methods. The technique depends on the material's electrical conductivity levels to achieve separation. The electric field causes conductive particles to lose their charge more quickly than non-conductive particles, which enables successful separation. The technique is especially used for separating plastic and metal from electronic waste. Moreover, the electrostatic method can be implemented in small-scale waste type determination by using a simple algorithmic code that determines if the material is a conductor(meaning metal), an insulator(meaning glass, plastic, etc), or by determining how much resistivity the item has to decide whether it is food waste or not.



2.2.3 Sensor-based Algorithmic separation

There are numerous types of sensors that can be implemented to determine and sort waste items. To utilize these sensors, there has to be a backbone of the system that operates as a brain to coordinate sensor input, process the collected signals, and execute sorting decisions.

This backbone is typically a central control unit programmed with a set of logical and mathematical algorithms designed to analyze sensor readings and make real-time decisions. Each sensor provides specific data; for example, optical sensors detect color and transparency, infrared sensors distinguish between organic and inorganic materials, inductive sensors identify metals, and ultrasonic sensors measure object size and density.

The algorithm processes this data by comparing the measured parameters to predefined thresholds or reference values stored in its database. Based on these comparisons, it classifies each item according to material type or category. For example, if the conductivity reading exceeds a set limit, the algorithm categorizes the object as a metal; if the infrared absorption pattern matches that of glass, it directs the item to the corresponding output stream.

Once classification is completed, the control unit sends commands to mechanical actuators or diverters that guide the waste item to the appropriate collection bin or conveyor belt. The algorithm continuously monitors sensor performance and signal consistency to ensure the accuracy and reliability of sorting operations.

This method offers a structured, rule-based approach to automated waste separation, ensuring consistent sorting outcomes, minimizing cross-contamination, and enhancing material recovery efficiency within waste management systems.[7]

2.3 IoT and AI-based separation

Modern waste management systems have undergone substantial changes through the implementation of IoT and AI technologies. The implementation of IoT and AI systems enables waste sorting operations to utilize intelligent, connected systems that perform real-time monitoring, automated decision-making, and operational optimization. The combination of IoT devices for data sensing and communication with AI algorithms for analysis and prediction enables waste sorting operations to become more efficient and flexible.



The IoT waste sorting system utilizes multiple smart nodes, which include sensors, cameras, and actuators that connect via Wi-Fi, MQTT, and HTTP wireless communication standards. The system nodes operate continuously to collect waste data through image capture, weight measurement, and material identification, before transmitting this information to a central processing unit or cloud platform. The main processing unit in this system operates through a Raspberry Pi single-board computer. The AI model, which operates as a CNN, runs on a single-board computer to identify waste items through visual inspection of their colors, textures, and shapes.

The AI model produces digital output signals that match waste categories after completing the identification process. The digital signal from the AI model reaches a microcontroller, which operates the actuators, motors, and servos to move waste items into their designated bins. The system stores classification results, user interactions, and environmental data on a web server or online database via IoT connectivity. The system uses this stored information to perform analysis, system monitoring, and performance assessment.

The system achieves better maintenance and tracking functionality because of IoT integration. The system tracks motor operation time, sensor responses, and power usage data, which gets transmitted to an online dashboard for predictive maintenance and diagnostic purposes. The system enables user engagement through its networked design, which includes a reward system accessible through mobile or web applications that tracks user and institutional waste sorting achievements. The system uses feedback mechanisms to boost user involvement while promoting sustainable waste management practices and environmentally friendly conduct. [7]

2.4 Theoretical background of small-scale AI models

The theoretical and mathematical representation of AI's operational methods through ANNs and CNNs will be discussed in this section of the thesis. We will be explaining how researchers use mammalian neuron structure as biological inspiration, the mathematical definition of artificial neurons, along with backpropagation learning methods and gradient descent optimization techniques.

2.4.1 Biological inspiration

The concept of ANNs arose thanks to Warren McCulloch and logician Walter Pitts in 1943, who created a simple logic gate design that might resemble the neurons in our brains. Although the processing power was limited and the model lacked learning capabilities, this inspired later generations to delve deeper into this field, thus marking the beginning of AI study. The foundation of these studies was the biological nervous system, specifically the neural networks inside our brains.

A typical neuron consists of 4 main parts:

- **Dendrites:** Tree-like structures that receive electrical signals from other neurons.
- **Soma** (The cell body): Integrates the incoming signals. If the cumulative signal exceeds a certain threshold, the neuron fires(generates sudden electrical pulses).
- **Axon**: A long, thin structure that transmits the output signal to other neurons.
- **Synapses**: The junctions between the axon of one neuron and the dendrites of another. The strength of the signal transmission across a synapse is modifiable, a phenomenon known as synaptic plasticity, which is considered the physiological basis for learning and memory.



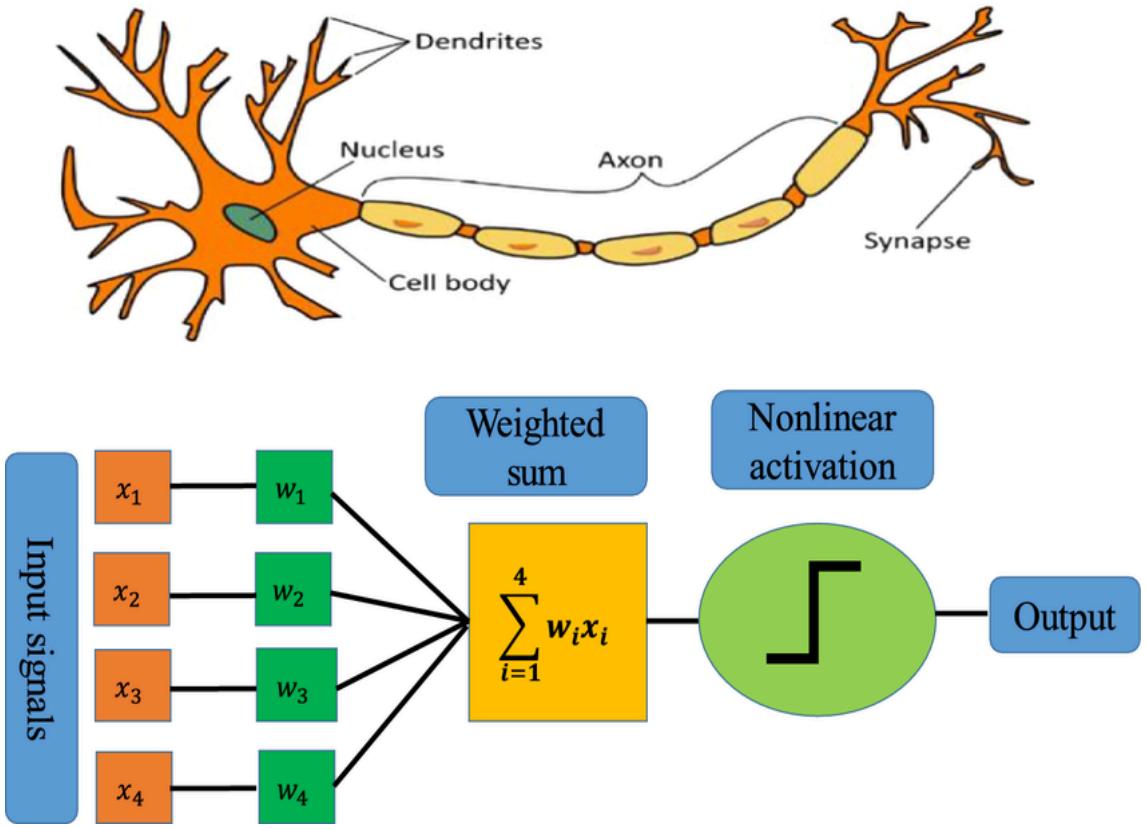


Figure 1. Comparative visualization of a neuron and a perceptron. [8]

As you can see from the figure, the architecture of an artificial neuron is remarkably similar to that of a biological one.

2.4.2 Artificial Neurons

An AN or a perceptron represents the simplest type of artificial neuron, a computational unit that transforms a vector of inputs into a single scalar output. What a perceptron represents is a single neuron from the nervous system.

Let the input vector be defined as:

Equation 1 : $x = [x1, x2, ..., xn]^T$

Where $xi \in R$ denotes the i-th input feature. The perceptron associates each input with a corresponding weight w_i, and collectively expresses it as:

Equation 2 : $w = [w1, w2, ..., wn]^T$

For an AN, the weight is the deciding factor of how important each input is.

The perceptron then computes a linear combination of its inputs, augmented by a bias term b, which allows the neuron to shift its activation threshold independent of the input, producing a scalar pre-activation value z.

Equation 3 : $z = \sum nw_i x_i + b = w^T x + b$

The weighted sum z is then transformed through a non-linear activation function f(z) to produce the final neuron output:

Equation 4 : $y = f(z)$

The activation function of a neuron can be defined as any function that can be linear, non-linear, or non-monotonic. However, because the weighing of inputs and bias additions are always linear, the activation function is almost always chosen to be a non-linear function to create flexibility so that it can approximate complex, non-linear mappings between inputs and outputs.

Some of the most popular ones used today are:

- Relu(Rectified Linear Unit): Most commonly used function due to its reliable and efficient computation, and usually used for hidden layers.

Equation 5 : $ReLU(z) = max(0, z)$

- Sigmoid(Logistic): Maps inputs to outputs of 0 or 1, making it useful for probabilistic interpretations and binary classification.

Equation 6 : $\sigma(z) = \frac{1}{1+e^z}$

- Tanh(Hyperbolic tangent): Produces outputs in the range (−1,1), effectively centering data and often leading to faster convergence in gradient-based optimization.

Equation 7 : $tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

- Softmax: Converts raw scores into a probability distribution across multiple classes. Outputs a range from 1 to 0.



Equation 8 : $\sigma(z_i) = \frac{exp(z_i)}{\Sigma exp(z)}$

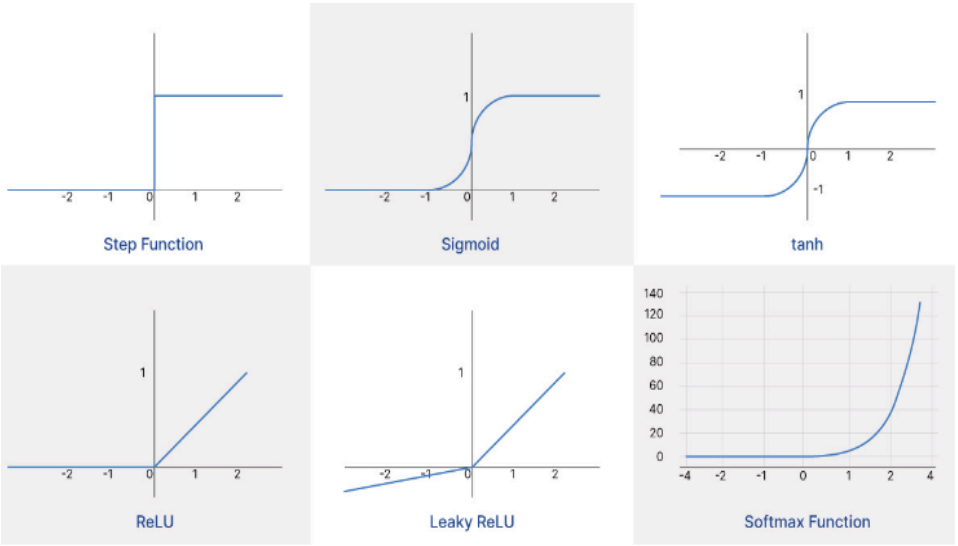


Figure 2. Visualization of different functions [9].

There are infinite amounts of functions that can be used as activation functions, but these are some of the functions that are most commonly used in AI development, presumably so because they are the most optimal choices we have today for various tasks. Each function is used in different projects and scenarios for its respective traits.

2.4.3 Neural networks and the training process

2.4.3.1 FNN

With a complete representation of a single neuron that can “fire” outputs similar to those of ours, the next step is to create a nervous system with a multitude of these neurons working together to simulate a nervous system. This type of system is referred to as an FNN(Feed-Forward Neural Network). It is composed of an input layer, one or more hidden layers, and an output layer, as illustrated in Figure 3.

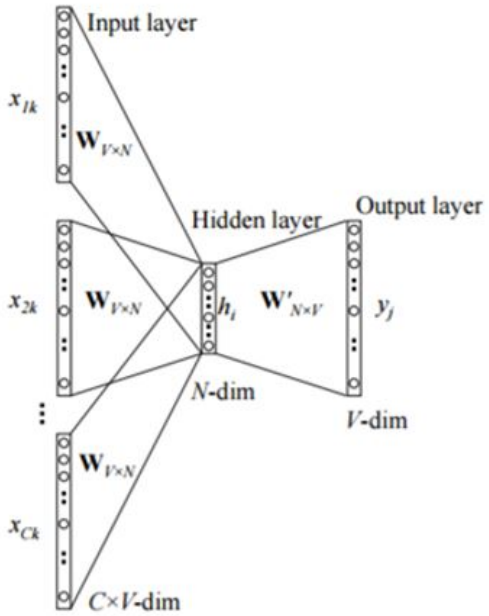


Figure 3. Neural Network [9]

In an FNN, data flows unidirectionally from the input layer through the hidden layers to the output layer, without cycles or feedback connections. The term deep learning refers to networks containing multiple hidden layers, each capable of learning increasingly abstract representations of the input data.

The mathematical representation of a layer l of a neural network with L layers:

Equation 9. $z^l = W^l a^{l-1} + b^l,\ a^l = f^l(z^l)$

Where:

- a^{l-1} is the activation vector from the previous layer.
- W^l and b^L are the weights and biases for layer L.
- f^l is the activation function applied element-wise.

The output of the final layer, $\hat{y} = a^L$, represents the model's prediction.

2.4.3.2 Training or learning process

The goal of training a neural network is to find the optimal set of parameters {W, b} that minimizes the discrepancy between predictions and true labels. This discrepancy is quantified by a loss function L(y,y').

For regression tasks, a commonly used loss is the Mean Squared Error (MSE):

Equation 10. $L(y,\hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$

Where:

- y_i is the true target value
- $\hat{y}_i$ is the model's predicted output
- N is the number of training samples.

Then, by using the chain rule of calculus, the partial derivatives of the loss with respect to each weight and bias are computed. These gradients represent how sensitive the loss is to changes in each parameter, allowing the network to attribute error signals backward through layers. This step of the system is called backpropagation.

$$\frac{dL}{dw_{ij}^{l}} \text{ and } \frac{dL}{db_{i}^{l}}$$

The parameters W and b are then updated using the gradient descent rule. The gradient descent algorithm iteratively adjusts model parameters in the direction of the steepest loss reduction. Each step moves opposite to the gradient vector, seeking a local (or global) minimum on the loss surface. The method is practical not just in AI training, but also in numerous other fields.

Equation 11. $w_{new}^{l} = w_{old}^{l} - \eta\frac{\delta L}{\delta w_{old}^{l}}$

Where:

- η is the learning rate, a hyperparameter controlling the magnitude of the update step
- $\frac{\delta L}{\delta w_{old}^{l}}$ is the computed gradient for the weight w.

Choosing the optimum learning rate for models can be difficult, as a large one can cause divergence, while a too small one may result in slow convergence.



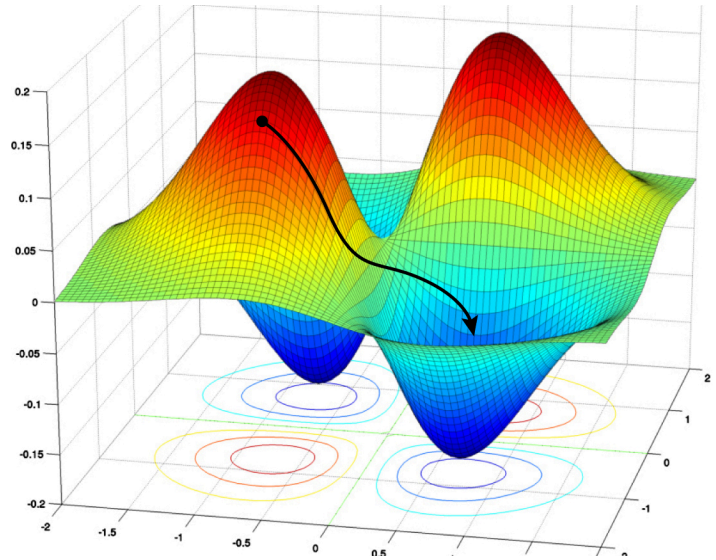


Figure 4. Sample 3D projection of a Gradient descent landscape [10].

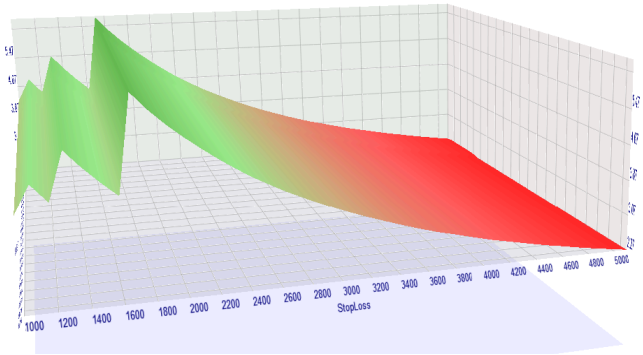


Figure 5. Gradient descent plane of an algorithmic trading bot's stop loss and take profit ratio.

2.4.4 CNN

Feedforward Neural Networks (FNNs) function as universal function approximators, yet they become inefficient when handling data with grid-like structures like images because of their dense network structure and absence of spatial understanding.

The architectural features of **Convolutional Neural Networks (CNNs)** solve these efficiency problems by utilizing data spatial structure through local connections and shared parameters, and progressive feature extraction. These operations are done by two specific layer types, called the convolutional layer and the pooling layer, that are introduced into a normal NN.

The **convolutional layer**

- serves as the core component that makes up CNNs. The convolutional layer uses learnable filters (kernels) to scan the input image, which generates output feature maps that detect spatial patterns, including edges, textures, and shapes. For a given input image I and the convolutional layer K, the output O is calculated as follows in 2D space.

Equation 12. $$O(i,j) = (I*K)(i,j) = \sum_m \sum_n I(i+m, j+n) \cdot K(m,n)$$

In convolutional layers, the connection pattern between neurons differs from that of fully connected layers because each neuron receives input only from a specific local region of the input data, known as its receptive field. This localized connectivity allows the network to capture spatial hierarchies and reduces the number of learnable parameters, resulting in improved computational efficiency and reduced risk of overfitting. Additionally, convolutional networks use the same set of filter weights across all spatial positions in the input, enabling weight sharing. This property allows the network to detect features such as edges and corners regardless of their position in the image, thereby achieving translation invariance in feature detection.

Additionally, in the convolutional layer, another set of dimensions can be added to make the model recognize colors through multiple input channels by adding additional dimensions to the convolutional layer, which processes RGB channels in color images. The convolutional filters analyze all three color channels (red, green, and blue) at each pixel location. The model uses filters with depths matching the number of input channels (3 for RGB) to extract color-dependent patterns and features from all channels. This would change the existing equation into:

Equation 13. $$O(i,j) = (I*K)(i,j,c) = \sum_c^C \sum_m \sum_n I(i+m, j+nc) \cdot K(m,n,c)$$



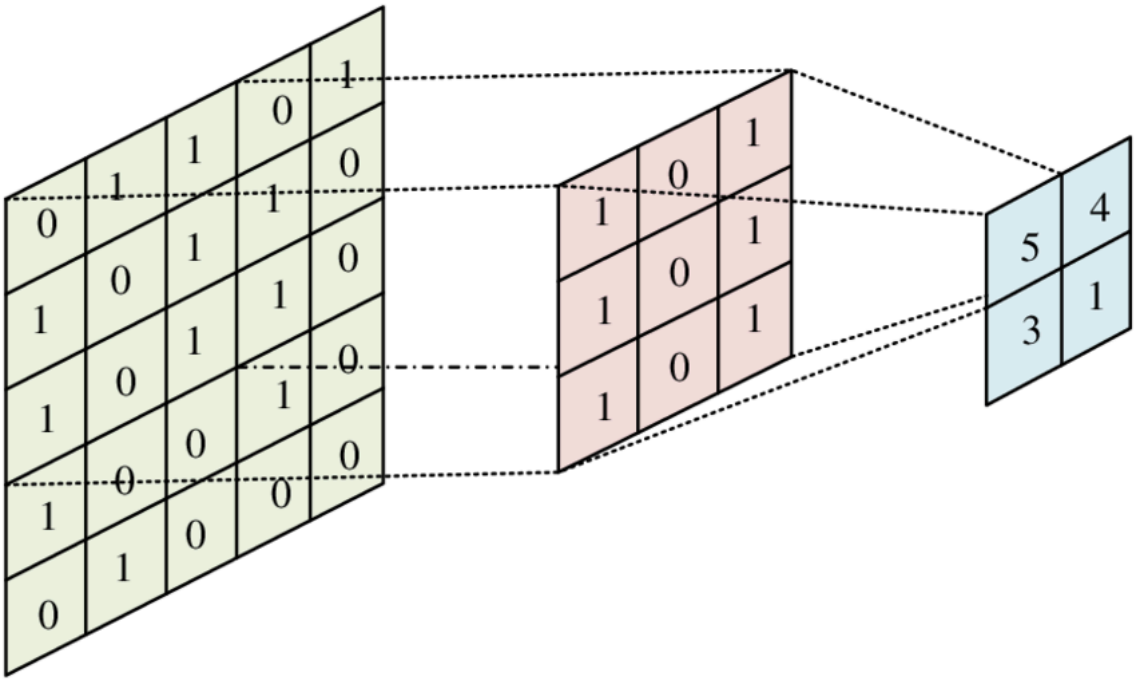


Figure 6. Convolutional layer operation. [12]

The **pooling layers**

- in the network perform down-sampling operations on feature maps to achieve two main goals: they reduce spatial dimensions and input sensitivity to small position changes. The pooling operation maintains essential features while discarding unimportant details, resulting in improved network resistance against various input transformations.

Equation 14. $O(i,j) = max\{I(2i,2j), I(2i+1,2j), I(2i,2j+1), I(2i+1,2j+1)\}$

While there are several different pooling functions, the most commonly used is “max” because of its ability to map features within an image effectively.

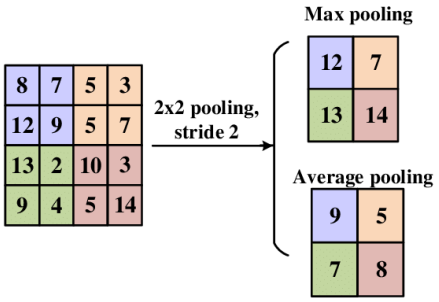


Figure 7. Pooling example. [13]

A typical CNN for image classification uses multiple sequential layers of convolutional and pooling operations before it reaches the fully connected layers, which make the final classification decisions. The network learns to detect features, ranging from basic edges to complex semantic elements, due to its structured design.

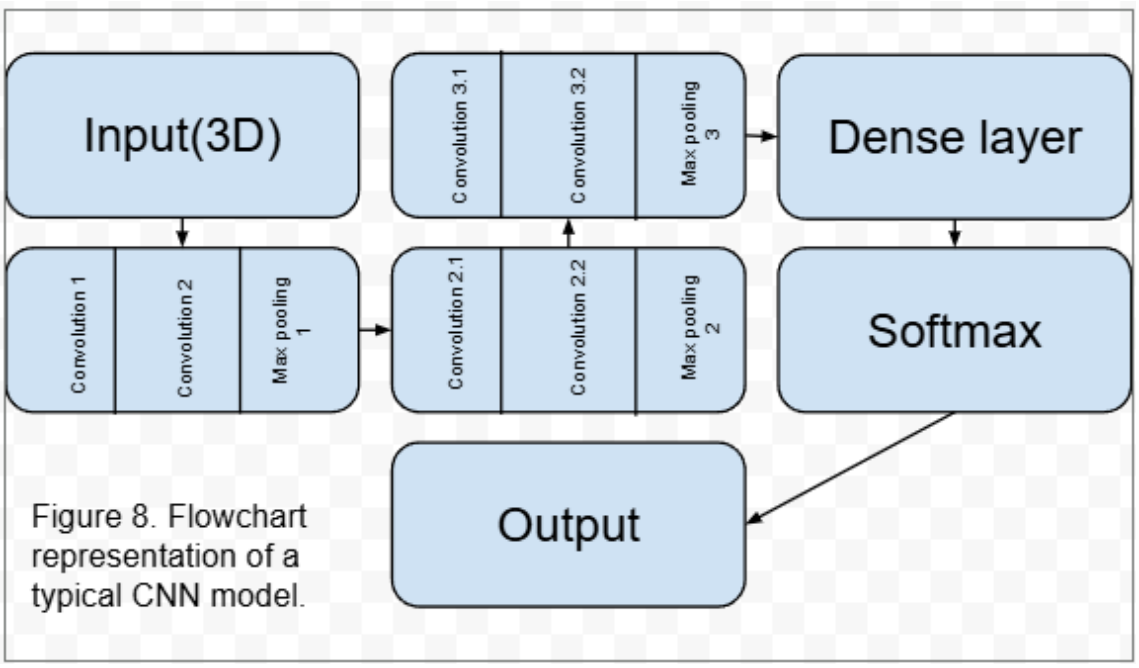


Figure 8. Flowchart representation of a typical CNN model.

Methodology

After extensive research on existing technologies for classifying and separating different types of trash for recycling, it was determined that an AI-based IoT system offers the best price-to-performance capabilities while being less bulky than a sensor-based algorithmic system.

3.1 System design and architecture

The system is most likely to be implemented in various shapes and sizes, and in different environments. The goal is to build the backbone of the final product that is flexible to use in any of these scenarios.

The garbage dump version stands as the largest and most difficult to construct because Mongolia's extreme climate conditions make it challenging to build. The system requires proper insulation for its electronic and moving parts due to Mongolia's long, cold winters, windy autumns, dusty springs, and heavy summer rains, which create conditions that could cause freezing, corrosion, and mechanical failures. The system requires a weatherproof case and strong materials, as well as thermal control systems, to operate reliably in varying environmental conditions. Additionally, the system would profit from solar power backup with battery storage to operate continuously when grid power becomes unstable during outdoor deployments.

Secondly, the system will operate in indoor spaces, which include offices, schools, and public facilities, as its main application area. The system needs to focus on user-friendly design and operational simplicity instead of durability in these specific environments. The system needs to be compact, quiet, and visually appealing to fit seamlessly into its surroundings. The indoor version needs to focus on user-friendly design and simple maintenance while providing automatic lid control with visual and/or audio feedback for waste sorting assistance. The system needs network capabilities to send performance data to a server for user reward management and performance tracking of waste sorting activities.

For the final scenario, the model will be targeting primary schools and kindergarten-age children for educational purposes, through its toy-like design. The system teaches children to care for the environment through interactive learning experiences. The system needs a compact design with safety features and user-friendly interface elements, which include visual indicators and audio-visual effects to help users perform



waste sorting tasks easily. The educational model would need to use basic camera hardware and a simplified classification system.

The system requires modular design principles for scalability because its different applications need distinct operational conditions and design requirements. The system needs replaceable modules for sensing units, control boards, and actuation mechanisms to enable various configurations without extensive redesign work. The system design allows for universal application through its core architecture, which can operate as an outdoor unit, office tool, or educational device for children. The system needs to achieve equilibrium between technical scalability and social adoption through a flexible design that maintains reliability and cost-effectiveness for various environmental settings.

3.1.1 System requirements

The system requirements define the hardware, software, environmental, and performance expectations necessary for the successful operation of the Automated Trash Sorting System.

1. **Functional Requirements**

These specify what the system must do:

- The system must automatically classify trash into three or more categories: plastic bottles, metal cans, and others (paper, glass, organic, cardboard).
- The system must use a camera module for visual classification and a load cell for weight measurement.
- The classification accuracy must exceed 90%.
- The actuation mechanism must direct each item into the correct bin within 2 seconds of classification.
- The system must log classification results and user data to a central server or web application.
- The system must connect to the IoT network for data transfer.
- The system must access the internet via cellular or Wi-Fi.



- The reward system must register user contributions and display group rankings on the web interface.
- The system must be able to operate autonomously for at least 24 hours without human intervention.

2. **Hardware Requirements**

- Main controller: Raspberry Pi 4 Model B (or equivalent, 4GB RAM minimum)
- Access to cellular network: a SIM module that can access a cellular connection
- Microcontroller: ESP32 for motor and sensor control
- Camera: Raspberry Pi Camera Module V2 or equivalent (1080p supported)
- Actuation: NEMA 17 stepper motor with driver (e.g., A4988). Metal drive servo motor
- Sensors: Load cell (1–5 kg range), IR sensor (for object detection and bin capacity)
- Power supply: 12V DC (regulated) with backup battery option
- Connectivity: Wi-Fi (2.4 GHz), serial communication between Raspberry Pi and ESP32
- Mechanical structure: 3D-printed or sheet-metal housing, weatherproof enclosure for the outdoor version

3. **Software Requirements**

- Operating System: Raspberry Pi OS / Linux-based environment
- Programming Languages: Python (for CNN inference), C++ (for ESP32 control)
- Communication Protocols: UART, HTTP, JSON, and MQTT (optional for IoT scalability)
- Database: MySQL or Firebase for storing classification data and user scores

4. **Environmental Requirements**

- Must operate within the temperature range of −30°C to +30°C.



- Must be dust- and moisture-resistant (IP54 minimum) for outdoor deployment.
- Must be insulated against freezing to prevent actuator or sensor malfunction in Mongolian winter conditions.

5. **Performance Requirements**

- Average inference time per image: < 1 second
- Classification throughput: at least 10 items per minute
- Power consumption: less than 50W during peak operation
- Data transmission delay: < 3 seconds to the server
- System uptime: > 95% reliability during continuous operation

6. **Non-Functional Requirements**

- Scalability: Must be adaptable for smaller (school) and larger (public bin) versions.
- Maintainability: Hardware modules must be modular and replaceable.
- Security: Data transfer between the device and the server must use encryption (HTTPS).
- Aesthetics: Must have a clean, inviting design suitable for educational and public environments.

3.1.2 Hardware components

Adhering to system requirements and ease of access, the hardware components were chosen as such:

#	Components	Name, Specification	Price	#	Price after amount
1	Main computer	Raspberry Pi 5, 8GB	355,000	1	355000
2	Cellular module	A7670E	97,700	1	97700

3	Weight sensor	2.5 KG Load cell	3,200	4	12800
4	Weight sensor amplifier	HX711	7,400	4	29600
5	Microcontroller	ESP-32	34,000	3	102000
6	Power supply	220-250V to 12V10A, 5V10A, 3.3V5A	80,000	2	160000
7	Keyboard/button	Metal frame 4X4 Numpad	150,000	1	150000
8	Stepper	NEMA-17	15,000	2	30000
10	Stepper framing	20X20 aluminum 1M length	8,000	5	40000
13	Driver stepper	A4988	2,500	4	10000
15	Display	10-inch IPS display	172,500	1	172500
22	Camera module	1080p USB camera with integrated LED	87,200	1	87200
24	IR sensor	Regular IR proximity sensor 10 pack	25,000	1	25000
25	Servo motor	SG995 servo 90 to 180 degrees	6,000	2	12000
2	Servo motor drive	PCA9685	8100	2	16200



6					
Table 1. Main components order history				Total	1300000

All of the components were bought from Hibox.mn. You can view the order receipt in PDF format in the Appendix. These are only the main components that have been selected and bought at the moment. Other miscellaneous components have not been mentioned here. Some components are not yet discussed or selected.

Connection and Circuitry of the Hardware components

The system's hardware components are interconnected through a structured circuitry design to ensure reliable communication and power distribution. The architecture is centralized around the Raspberry Pi, which acts as the main computing unit.

The Raspberry Pi is connected to several key peripherals:

- It interfaces with the 10-inch IPS display via HDMI for the user interface.
- A 4x4 metal-plated NumPad keyboard switches are connected for the user interface.
- The 1080p USB camera is connected directly to a USB port for image capture.
- The A7670E cellular module is connected via GPIO pins for internet connectivity in areas without Wi-Fi.
- A UART (Universal Asynchronous Receiver-Transmitter) serial connection is established with the ESP32 microcontroller using GPIO pins (Tx, Rx) for sending classification commands and receiving status updates.

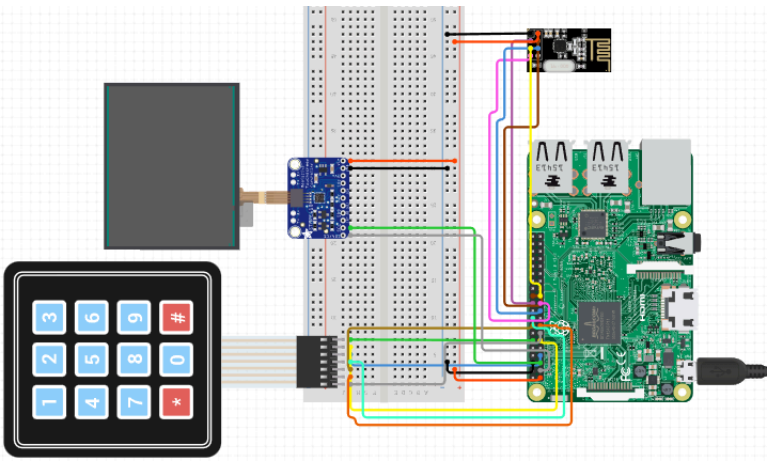


Figure 9. Example Raspberry pi connections

The ESP32 microcontroller functions as the subordinate control unit, managing all actuation and sensor reading:

- It is connected to the NEMA 17 stepper motor via an A4988 driver module, which controls the precise rotation of the sorting mechanism.
- Servo motors are connected to the ESP32's PWM (Pulse-Width Modulation) pin, through a PCA9685 servo driver, to operate trapdoors or diverters.
- The load cells are connected to the HX711 amplifier modules, which, in turn, communicate with the ESP32 via a digital serial interface at a 115200bps baud rate to provide weight data.
- IR proximity sensors are connected to the ESP32's digital input pins to detect the presence of an item.

At the time being, the system is to be powered by a central Power Supply Unit (PSU) that outputs 12V, 5V, and 3.3V rails.

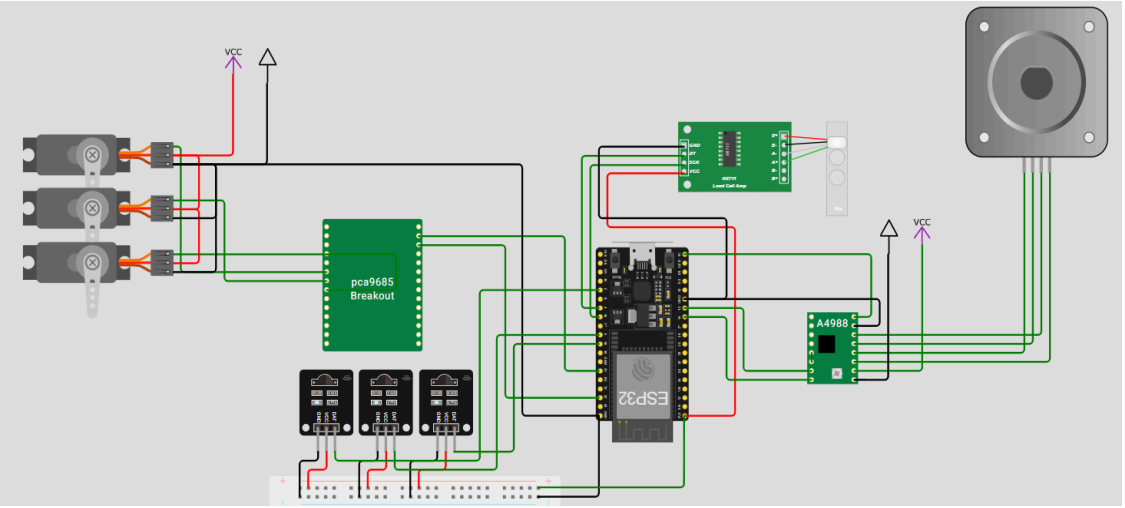


Figure 10. Example ESP-32 connections

3.1.3 Power Management

Because there are several different deployment scenarios, the system's power management had to adapt to these situations by having diverse methods of powering the bins. With this in mind, we have planned a few ways to power the system.

For the outdoor garbage dump version, we have decided to implement a solar panel with an integrated charge controller system (Usually an MPPT) with a detachable and swappable 12V lithium-ion battery pack. This was a necessary move because looking for the right spot that has a power source nearby would have been a hassle, and the bin will be easily transported and will not require additional work to connect to an electrical line.

Additionally, while a dedicated MPPT charge controller is required for efficient and safe solar power management, the system can utilize the ESP-32's ADC inputs to monitor the solar panel's voltage/current and battery levels. This data can be used for system diagnostics and to send alerts when power is low. A number of calculations were done to find the suitable requirements for this system, and we have concluded that a 200W-rated solar panel paired with a 30Ah battery would be sufficient.

For the indoor versions, the bins would likely have access to 220V power sockets, which meant having an AC/DC PSU was a feasible option. If that is not the case, the unit could implement rechargeable and hot-swappable battery packs, similar to the outdoor version.

Across all models, energy optimization and circuit safety are achieved by implementing polyfuses (5A for the controllers, 10A for the actuators) and dynamically disabling idle modules. For instance, the ESP32 enters deep-sleep mode when no sorting activity is detected, while the Raspberry Pi reduces CPU frequency to conserve power.

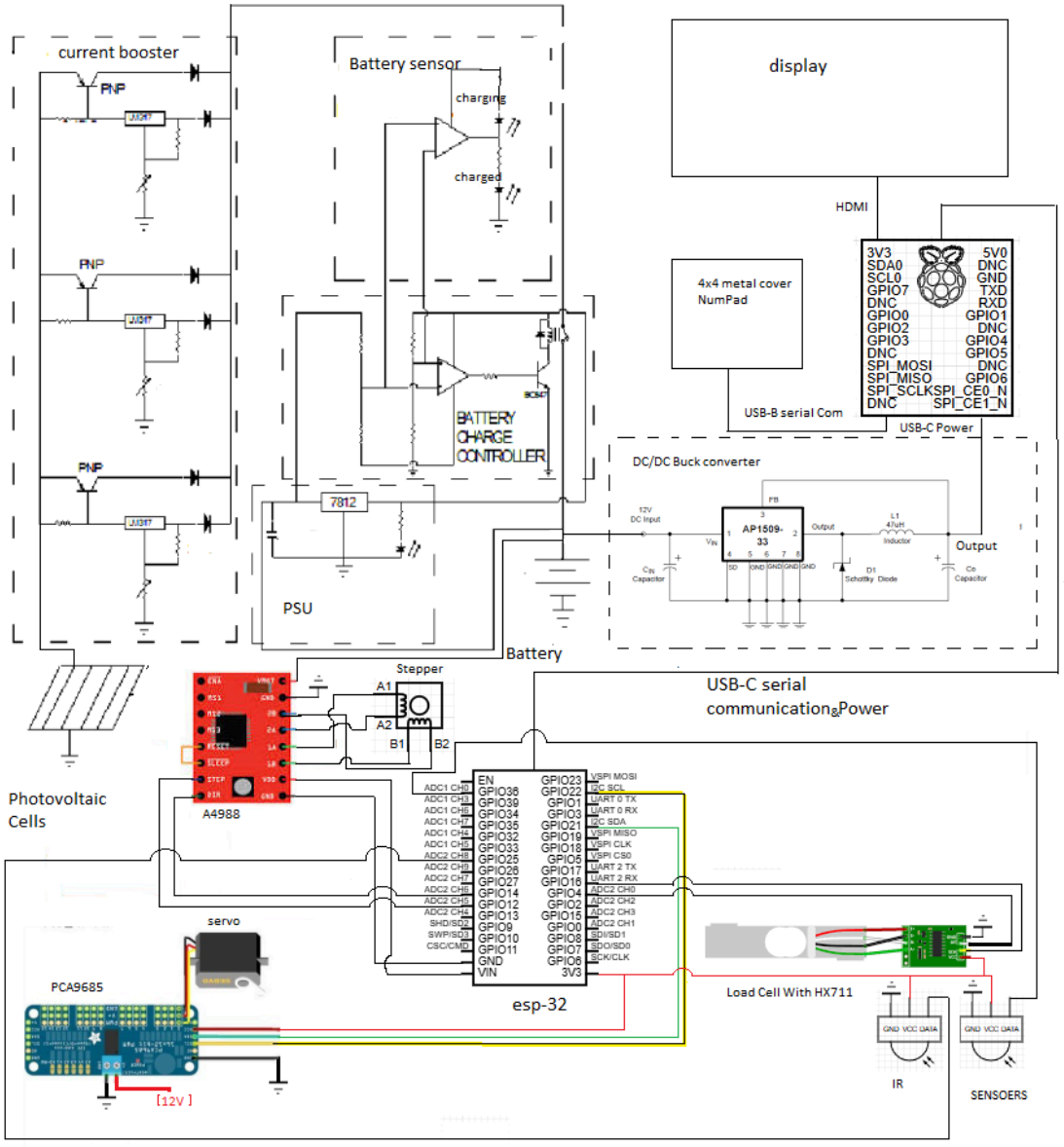


Figure 11. Full Circuit diagram with Solar panels and Power supply lines[15].

3.2 Implementation, Testing, and Evaluation of the CNN

To implement an AI model into the project, there were three different choices, each with its pros and cons. The first choice was to implement an already existing LLM model, such as ChatGPT, Deepseek, Claude, or Gemini, through APIs or as agents. These models are highly capable of categorizing not only trash but also proficient in various tasks. However, this was quickly dismissed as unnecessary because such computing power is likely overkill, and the monthly or yearly subscription fees would have been a nuisance.

The second one was to build an AI model from scratch, which was also written off. Since numerous types of open-source CNN models already existed, we decided that building another one for our use case was redundant. Instead, we searched for an AI model that somewhat aligned with our goal and tried to integrate it into our system. After searching GitHub repositories and collecting and testing a number of models, the following CNN model not only perfectly aligned with our task at hand but also showed promising performance with a minimal amount of computing power [14].

3.2.1 Data Collection, Augmentation, and Training

The model training process used data from the TrashNet dataset, which contains thousands of waste images that are labeled into six categories: cardboard, glass, metal, paper, plastic, and trash. The dataset underwent additional processing, which involved two steps: resizing to 224×224 pixels and normalization to bring pixel values between 0 and 1. The preprocessing method helped models achieve better convergence rates while decreasing their training time requirements.

The model received data augmentation treatment through random flipping, rotation, zoom, and brightness modification techniques. The transformations duplicated waste image variations that occur in real-world settings through changes in lighting and object positions, which made the model more resistant to environmental interference during deployment.

The CNN training process utilized transfer learning to train models based on EfficientNetB0, EfficientNetV2, ResNet50, and MobileNetV2 architectures. The models received their initial weights from ImageNet training, which enabled them to learn complex features from extensive datasets.

The deep learning framework TensorFlow/Keras served as the platform for executing the training process. The model used Adam optimizer with a learning rate of 0.001 and categorical cross-entropy as its loss function. The models underwent 15-30 training epochs with 32 samples per batch while using 80% of the data for training and 20% for validation. The model implemented early stopping to prevent overfitting by monitoring the validation loss.

3.2.2 Implementation & Testing

The model at hand was forked from the aforementioned GitHub repository and was launched on a private server after the training process. The integration of the model into the project's system was fairly smooth, as the creator had provided a detailed and comprehensive overview of the model and its building process. The model was then trained and tested on different lighting and background scenarios on various types of trash.

3.2.3 Evaluation

The evaluation of the CNN focused on three main metrics: accuracy, loss, and confusion matrix analysis. Among all benchmarked models, EfficientNetB0 achieved the highest classification accuracy of 91.97% on the test dataset, outperforming ResNet50 and MobileNetV2. EfficientNetV2 variants also achieved competitive performance, though with higher computational cost.

Model	Test Accuracy (%)	Validation Loss	Notes
EfficientNetB0	**91.97**	0.18	Best trade-off between speed and accuracy
ResNet50	89.62	0.22	High accuracy but slower inference
EfficientNetV2-S	90.13	0.2	Slightly better feature extraction, heavier model
MobileNetV2	19.7	1.85	Poor convergence, underfitting observed.

Table 2. Comparison of different CNN models [14].

The model achieved an F1-score of 0.91 and precision and recall values above 0.90 for the main categories, confirming consistent performance.

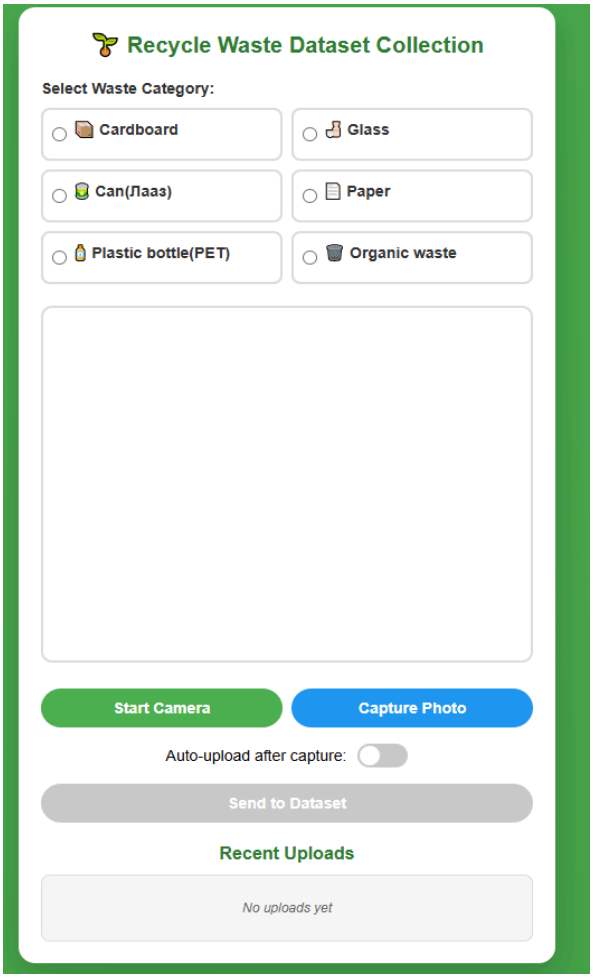
However, this finding was deemed unreliable when given images of Mongolian waste as input, because Mongolian waste differed drastically in shapes, colors, and texture patterns, thus giving mixed results on some occasions. This meant that the CNN was still raw for deployment and needed further training on Mongolian-specific datasets.

3.2.4 Mongolian waste database

Since the model required further training on Mongolian-specific waste data, finding suitable data was a crucial task. After a thorough search for such repositories, one with neatly labeled data was not found. The ones that were found had too generic labels or had mixed data with unwanted pictures for our use case. Therefore, we decided to create our own database on GitHub with the help of the public (mainly GMIT students, my friends, and family).



Now, with the help of a webpage anyone can send pictures of their trash in the correct folder. The page preprocesses the incoming images automatically before saving them, making the collected data ready for the training process. Of course, to mitigate human error, which is impossible to bypass completely, the dataset is closely monitored, and unwanted or incorrect images are manually deleted or relocated.



Currently, the page has been up and running for over a month and has collected around 400 pictures.

Figure 12. Gmit.life webpage.

3.3 Integrated System Software and Operation

3.3.1 Communication between components

Since the final architecture is not yet fully decided, with two possible outcomes, the communications and actuation step controls differ in various areas. However, in both scenarios, the Raspberry Pi serves as the main processor, which uses the ESP32 microcontroller to perform basic actuation and sensor operations. In this slave-master hierarchy, the two devices exchange data through serial UART protocol at 9600 bps baud rate by using lightweight JSON messages for structured and reliable data transfer.

In the first configuration, two 180-degree servo motors are used to control the physical sorting mechanism. The process sequence is as follows:

1. When an object is detected entering the input chute, the Raspberry Pi sends a pulse signal to the ESP32, prompting it to wake from idle mode.
2. The Raspberry Pi then measures the weight of the item using the load cell and captures an image using the connected camera module.
3. The captured image is transmitted to the remote CNN model hosted on the server for classification.
4. Once the server returns the classification result in JSON format (e.g., {"type": "plastic"}), the Raspberry Pi forwards this information to the ESP32 via serial communication.
5. Upon receiving the command, the ESP32 actuates the two servo motors according to predefined sequences:
    - For plastic items, Servo #1 rotates to 180°, followed by Servo #2 after a 0.2-second delay.
    - For metal items, Servo #2 rotates to 0°, followed by Servo #1 after a 0.2-second delay.
    - The default (idle) position for both servos is 90°.

This method offers a fast and straightforward actuation scheme, making it suitable for compact prototypes.

1.

In the second configuration, a NEMA 17 stepper motor is introduced to improve mechanical precision and scalability. The workflow proceeds as follows:

1. Upon detecting an item, the Raspberry Pi activates the weighing system and captures an image for CNN-based classification.
2. The image and weight data are sent to the server, which processes and returns the result in JSON format (e.g., {"cmd":"type":"metal"}).
3. The Raspberry Pi then relays this result to the ESP32, which controls both the stepper motor and the servo mechanisms:
    - If the command is {"cmd":"type":"metal"}, the ESP32 rotates the stepper by 1 unit (a pre-defined angular displacement) and actuates Servo #1 to open the designated bin lid, before returning the stepper to its home position.
    - If the command is {"cmd":"type":"plastic"}, the ESP32 rotates the stepper by two units and triggers Servo #2 accordingly, then resets the stepper position.
    - The logic can be extended to additional categories by mapping stepper rotations to corresponding bin locations.

This method enables more precise positioning and supports multi-bin configurations, making it ideal for large-scale or outdoor implementations.

As mentioned before, the communication of the two devices is done by serial coms through structured text in JSON format, and will be seen as follows:

esp→ {"event":"booted"}
Pi → {"cmd":"start"}
esp → {"event":"ack_start"}
esp → {"event":"ready"}
Pi → {"cmd":"type":"metal"}
Pi → {"cmd":"reset"} <-- arrives mid-sort
esp → {"event":"processing"}
esp → {"event":"done","bin":"metal"}
esp → {"event":"idle"} <-- queued reset executes next

The structured text format was chosen to be easily readable to us so that a failure in coms could be easily detected. Moreover, the system employs acknowledgment (ACK/NACK) messages and watchdog timers to prevent communication failures. For example, if the ESP32 does not respond within a predefined time, the Raspberry Pi initiates a retry sequence or reboots the peripheral subsystem. Similarly, communication integrity between the Raspberry Pi and the cloud server is maintained through secure HTTPS encryption and retry-on-failure mechanisms.



3.3.2 Web app integration

Developing an intuitive and accessible web application was a critical step in the project, as it serves as the main interface between the users and the system. This part of the project, developing a web app, was solely done by the thesis supervisor, Mr. Myagmarjav.

When the Raspberry Pi is in idle mode, it displays a QR code on its screen. Upon scanning this code, the Raspberry Pi activates and prepares itself, along with the connected components, for the waste processing operation. The accompanying web application was designed primarily to support a reward system that incentivizes user participation.

Initially, the reward mechanism was conceptualized as a monetary system using e-banking services such as HiPay, SocialPay, and QPay. However, determining an appropriate conversion rate between the weight of the recycled materials and the monetary reward proved challenging. If the reward was set too low, users would lack motivation; if set too high, it could be exploited and result in financial losses for the operation. Furthermore, it was deemed unrealistic to expect adults, who most likely have not practiced waste sorting throughout their lives, to suddenly adopt the behavior for a small financial incentive.

This realization led to the insight that the primary users of the system would likely be children and, to some extent, homeless individuals. Focusing on children, however, inspired a new direction for the project: implementing the smart waste bins in school zones to encourage students to compete for rewards or prizes. Targeting children was considered a more effective long-term strategy, as their brains are still developing with huge learning potential for habit formation.

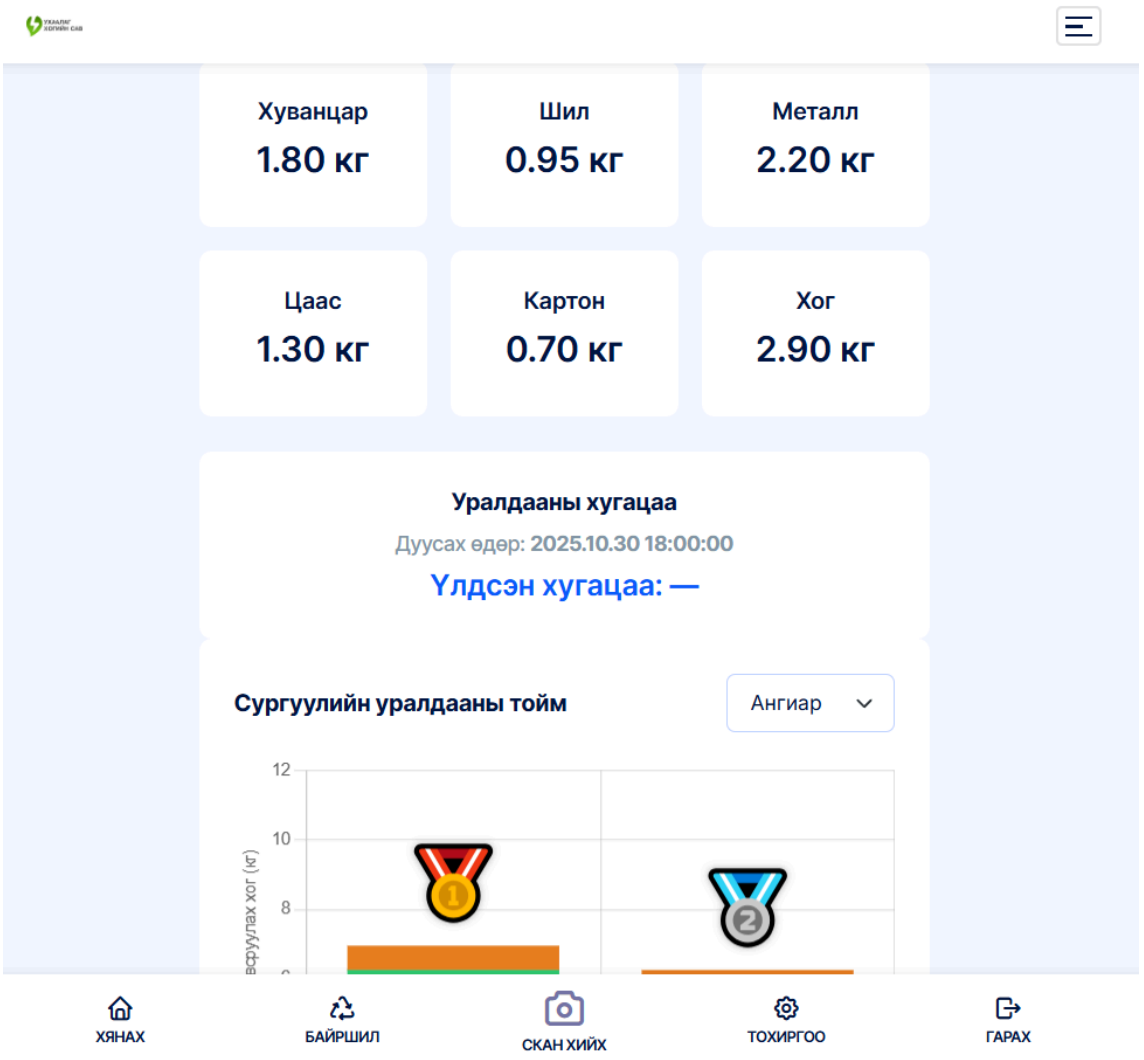


Figure 13. Web Appliccation dashboard

The web application was therefore designed with this goal in mind. As seen on the Figure 13. Its current version includes a map showing the locations and serial numbers of all smart bins, a user dashboard displaying the individual's contributions, and a leaderboard ranking classrooms or schools based on their collective recycling performance. Additionally, it features a camera activation button that allows users to scan the QR codes displayed on the bins directly through the web interface.

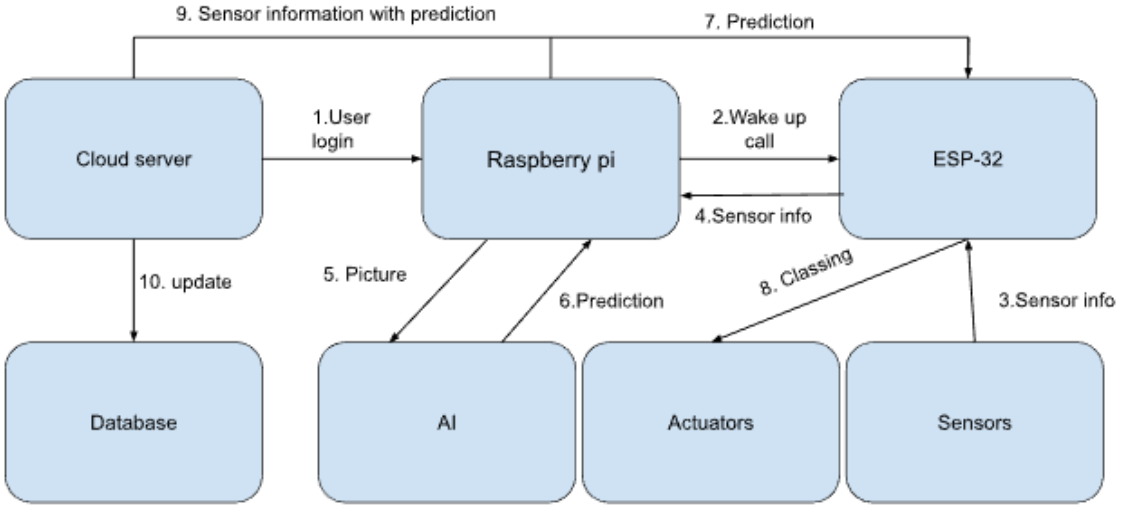


Figure 14. Simplified Flow Sheet Diagram

3.3.3 Database Structure

The Smart Bin system operates through MySQL database which contains six essential tables The Classrooms table contains all participating classrooms together with their total recycling amounts which serve to generate leaderboards and classroom-specific data analysis. The Users table maintains records of all participants who contribute to recycling while allowing users to view their individual statistics and performance data. The Devices table contains information about each Smart Bin hardware device along with its operational status and configuration details for device health monitoring. The Bins table contains information about waste-bin compartments which link to devices through their type and specific characteristics. The Reports table contains all waste-deposit events from devices which enables system analytics and classroom total calculations and leaderboard generation. The Misclassifications table stores human-made corrections of classification mistakes which enhances data precision and enables model assessment and improvement. The indebt description of each of the tables are shown as follows:

Table 3. Database structure			
	Field	**Type**	**Description**
classrooms	id	INT (PK)	Unique identifier for each classroom.
	name	VARCHAR(255)	Classroom name (e.g., “Class 1A”).
	total_kg	DOUBLE	Total weight collected by the classroom.
	created_at	DATETIME	Record creation timestamp.

users	id	INT (PK)	Unique identifier for each user.
	name	VARCHAR(255)	User’s full name.
	email	VARCHAR(255) UNIQUE	Login email for the user.
	classroom_id	INT (FK → classrooms.id)	Classroom the user belongs to.
	total_kg	DOUBLE	Total weight contributed by this user.
	created_at	DATETIME	Record creation timestamp.

devices	id	INT (PK)	Internal ID of the device.
	device_id	VARCHAR(128) UNIQUE	Unique device identifier used for reporting.
	location	VARCHAR(255)	Physical location of the device.
	serial	VARCHAR(255)	Serial number of the hardware.
	last_seen	DATETIME	Last time the device connected to the system.
	created_at	DATETIME	Record creation timestamp.



bins	id	INT (PK)	Unique identifier for each bin.
	device_id	VARCHAR(128)	Device associated with this bin.
	bin_index	INT	Bin number on a multi-bin device.
	label	VARCHAR(50)	Human-readable bin name.
	capacity_kg	DOUBLE	Maximum weight capacity of the bin.
	created_at	DATETIME	Record creation timestamp.

reports	id	INT (PK)	Unique disposal event record.
	device_id	VARCHAR(128)	Device that submitted the report.
	classification	VARCHAR(50)	Classified waste category (plastic/metal/other).
	weight_kg	DOUBLE	Measured weight of the waste.
	image_url	VARCHAR(1024)	URL of captured image (optional).
	user_id	INT (FK → users.id)	User who deposited the waste.
	created_at	DATETIME	Time the report was created.
misclassifications	id	INT (PK)	Unique correction ID.
	report_id	INT (FK → reports.id)	Report that was misclassified.
	corrected_class	VARCHAR(50)	Correct waste classification.
	corrected_by	INT (FK → users.id)	Administrator who made the correction.
	created_at	DATETIME	Timestamp of correction.



Discussion

The development of the Automated Trash Sorting System unfolded as both an engineering challenge and a learning process, and working through the project as a team made the experience even more meaningful. What initially seemed like a straightforward integration of AI, embedded hardware, and a web application quickly revealed layers of complexity once each subsystem was placed under real-world constraints. Our modular approach, developing and testing components separately before attempting full integration, proved to be one of the most valuable decisions. It allowed us to deeply understand how each module behaved under different conditions, from the timing sensitivities of the ESP32 to the quirks of serial communication and the unpredictable performance of machine learning models when exposed to new, local data. This systematic breakdown of tasks not only prevented the project from ballooning out of control but also gave us time to reflect critically on design decisions as we progressed.

When interpreting the results from individual components, especially the AI-based classifier, we realized just how closely they aligned with findings in the literature. Many studies emphasize that CNNs trained on curated datasets tend to perform exceptionally well in controlled environments, but their accuracy drops when deployed in novel contexts with different lighting or object variations. Our experience mirrored this exactly. This confirmed existing academic observations and reinforced the argument that localized datasets are crucial for real-world AI deployments. The web application also behaved in ways consistent with documented IoT system behavior: isolated modules typically perform smoothly, while integration challenges emerge only when communication, load, and timing dependencies are introduced. This pattern was evident in our system, where backend–frontend communication was stable, but synchronizing live device actions with server data introduced delays and occasional inconsistencies.

The engineering implications of this project became clearer as we advanced. The master-slave architecture, where the Raspberry Pi handles computation and the ESP32 manages actuation, proved both scalable and efficient. It affirmed the value of offloading lightweight tasks to microcontrollers to preserve resources on the main processor. However, we also encountered the practical limits of this design. Hardware integration became the bottleneck, not computation, especially when component deliveries were delayed by international shipping. This real-world obstacle reminded us



that engineering systems, no matter how well designed digitally, ultimately depend on the physical world and supply chains. Debugging mechanical–electrical interactions further highlighted the need for enclosure refinement and parameter tuning, which are often overlooked in early design stages.

We also reflected on the behavioral aspect of the system, something not commonly addressed in technical studies. While our technological components were advancing, the realization that children, not adults, would be the primary users reshaped the project's direction. This stood in contrast to much of the literature, where waste sorting automation is typically directed toward industrial use or municipal facilities. Our approach positions the technology not just as a functional tool but as an educational intervention one as well, aligning more with behavioral studies on gamification and habit formation. It confirmed that effective waste management is as much about human motivation as about engineering solutions.

At the same time, the limitations of the current prototype became increasingly evident. The system remains partially integrated, with functional modules that have not yet been assembled into a unified physical device. Mechanical construction, enclosure manufacturing, and full operational testing are pending due to time constraints caused by late hardware arrival. Furthermore, the lack of a robust Mongolian dataset restricts the CNN's reliability under real operating conditions. These limitations, however, are not failures but natural parts of an iterative engineering cycle. They highlight areas where the next phase of work must focus—completing integration, refining mechanical design, retraining the classifier, and validating system performance over longer periods.

Looking ahead, our future work is clear and well-defined. The immediate priority is expanding the Mongolian waste image database to enable model retraining with context-specific data. Enhancing the server-side database to capture misclassified images will create a valuable feedback loop for continuous AI model improvement. Full physical assembly of all hardware components into a cohesive prototype is planned, followed by a pilot deployment at GMIT to observe system behavior, user interaction patterns, and long-term data consistency. Beyond this pilot, optimization efforts will target communication stability, power efficiency, and cloud-based scalability to support deployment across schools and community centers. These steps form the natural continuation of the work and reflect a commitment to evolving the system into a fully operational, reliable, and educationally impactful tool.



Conclusion

This thesis set out to design and develop an Automated Trash Sorting System that integrates computer vision, embedded hardware, and a user-engagement platform to improve waste sorting behavior within the Mongolian context. Using a Raspberry Pi as the primary computing unit, an ESP32 microcontroller for actuation and sensing, and a CNN model for classification, we built a modular framework that reflects both practical engineering considerations and the social motivation behind the project. Through systematic testing of individual modules, from AI accuracy benchmarking to motor control and database synchronization, we demonstrated that each subsystem performs reliably in isolation, even though full integration remains a work in progress.

The key outcomes of this work include the successful implementation of a CNN achieving over 90% accuracy on standard datasets, the establishment of a functioning backend–frontend web application with real-time data handling, and the development of a communication architecture that can support scalable, distributed IoT systems. Perhaps most importantly, the project generated the first public Mongolian waste image database, with labeled structure, laying the groundwork for future AI advancements tailored to local conditions. By merging engineering design with behavioral insights, the project contributes both technically and socially, offering a blueprint for waste-sorting technology that resonates with children and community institutions.

Naturally, the work carries several limitations. The prototype is not yet fully assembled, the CNN model requires retraining on localized data, and long-term system validation has not yet been conducted. Yet these limitations do not diminish the progress made, but rather they simply point toward the next stage of development, which is already clearly defined.

In the end, this project demonstrates that even within tight constraints, an iterative engineering process grounded in modular design and real-world relevance can produce meaningful foundations for scalable solutions. The Automated Trash Sorting System stands not only as a technical prototype but also as a stepping stone toward building a culture of responsible waste handling in Mongolia.



References

1. Global Waste to Grow by 70 Percent by 2050 Unless Urgent Action is Taken: World Bank Report
2. How the World Bank is tackling the growing global waste crisis
3. World must move beyond waste era and turn rubbish into resource: UN Report
4. Waste management - German Federal Statistical Office
5. Mongolia Is Introducing Effective and Innovative Solid Waste Management with ADB's Support – Part 2 | Asian Development Bank
6. Does Gamification Work? -- A Literature Review of Empirical Studies on Gamification | IEEE Conference Publication | IEEE Xplore
7. TUBAF_2024_ Bachelorarbeit_Byambadorj_Enkhbold.pdf
8. Artificial neuron versus biological neuron. The artificial neuron... | Download Scientific Diagram
9. Activation function
10. Researchers Build AI That Builds AI | Quanta Magazine
11. Deep learning for algorithmic trading: A systematic review of predictive models and optimization strategies - ScienceDirect
12. Convolution operation diagram
13. Pooling layer operation oproaches 1) Pooling layers: For the function... | Download Scientific Diagram
14. https://github.com/akathedeveloper/Trash-Classifier/tree/main?tab=readme-ov-file
15. DESIGNANDIMPLEMENTATIONOFASOLARCHARGECONTROLLERWITHVARIABLEOUTPUT.pdf

Appendix

1. Initially bought components:
    - Монгол-Герман сургууль 10-23 279915_by=order&stamp=true&vat=true (1).pdf
    - Монгол-Герман сургууль зарлага 279940_by=order&type=expense&stamp=true&vat=true (1).pdf
2. Initial training dataset: TrashNet
3. Dataset collection webpage: Recycle Waste Dataset Collection
4. User interface web APP: https://garbage.onistech.mn
5. Source code of ESP-32: Esp32.md
6. Source code for webpage: Untitled-1.html
7. Source code for the training and testing: app.py
8. Source code for the CNN model: CNN.py
9. Source codes for web app: https://gitlab.com/swermak/garbage-sort (A private gitlab project under the user SWERMAK, AKA Mr. Myagmarsuren)