A

Project Report

on

"Design, Fabrication and Testing of Automatic Potato Sorting System"

Prepared by

Nitesh Silwal

Abstract

Manual sorting of potatoes in the agricultural and food processing industries is not only labor-intensive and time-consuming but also prone to human error, leading to inconsistent classification and operational inefficiencies. The high demand for precision in sorting, driven by varying consumer and industry preferences for potato size and quality, necessitates a more efficient solution. This project introduces an automated potato sorting system based on advanced image processing techniques to address these challenges. Leveraging the OpenCV library, the system captures high-resolution images of potatoes as they pass through a conveyor belt, analyzing these images to classify the potatoes by size, shape, and quality. The hardware setup integrates cameras, servo motors, and a conveyor mechanism to ensure seamless operation. Potatoes are automatically sorted into predefined categories, ensuring uniformity and reducing manual labor intervention. Compared to traditional manual sorting, the automated system offers significant advantages in terms of cost-effectiveness, accuracy, and reliability. It minimizes labor demands while ensuring consistent classification standards, reducing the likelihood of errors that can arise from fatigue or subjective judgment in manual sorting processes. The system is designed with flexibility in mind, allowing easy integration into existing production lines and scalability for larger operations. This approach provides a practical, efficient solution for optimizing potato sorting operations, aligning with modern agricultural practices and Industry 4.0 standards.

Keywords: *Potato sorting*, *image processing*, *automation*, *OpenCV*, *smart manufacturing technology.*

Table of Contents

1. Introduction

The automatic segregation of vegetables based on size plays a crucial role in the agricultural and food processing industries, significantly impacting market distribution and pricing strategies. In Nepal, there is a clear demand for size-specific potatoes: small to medium potatoes are preferred by hotels and restaurants, while larger potatoes are more suited for food processing industries. Traditionally, sorting potatoes by size has been done manually, a process that is not only labor-intensive but also prone to errors and inefficiencies.

To address these challenges, the "Automatic Potato Sorting System Using Image Processing" project was developed in response to a request from K-Mart, a local departmental vendor. The project aims to automate the sorting process by utilizing advanced computer vision, machine learning algorithms, and a robust integration of hardware and software systems. This automation significantly improves sorting speed, accuracy, and cost-effectiveness. The system's innovation was recognized when it received the Best Project Award at the 1st Thapathali Graduate Conference 2023, and it was supported by a NRs. 30,000 grants from K-Mart.

The project was developed in collaboration with Assistant Professor Sushant Raj Giri and a team of researchers, bringing Industry 4.0 principles into the agricultural sector. By automating the sorting process, the system dramatically reduces the need for manual labor, enhances efficiency, and allows for greater scalability. The system is tailored to meet the specific size requirements of various consumer groups, optimizing the potato supply chain and supporting the growth of Nepal's agricultural industry. This innovation represents a significant advancement in how agricultural products are processed and distributed in Nepal.

2. Objective

The major objectives and motivation for this project were:

- To design and develop an automated system that accurately identifies and classifies potatoes by size using computer vision algorithms and machine learning models, ensuring efficient and consistent sorting.

- To seamlessly integrate hardware components, including cameras for image capture, ultrasonic sensors for size measurement, and servo motors for precise control of the sorting mechanism, ensuring smooth and reliable operation.

3. Methodology

3.1 Dataset collection and preparation:

The system leverages the Fruits-360 dataset, containing over 90,483 images from more than 100 different vegetable categories. From this dataset, we selected and resized images of 14 types of vegetables commonly available locally. To ensure the model adapts to real-world scenarios, we also captured approximately 500 images of potatoes on a conveyor belt and processed them and mixed in the dataset from Fruits-360, enhancing the model's robustness and versatility. A summary of the data is provided below:

- Training set: 10,000 images
- Test set: 4,000 images
- Image size: 150 × 150 pixels

Figure 1. *Potato images captured under working condition*

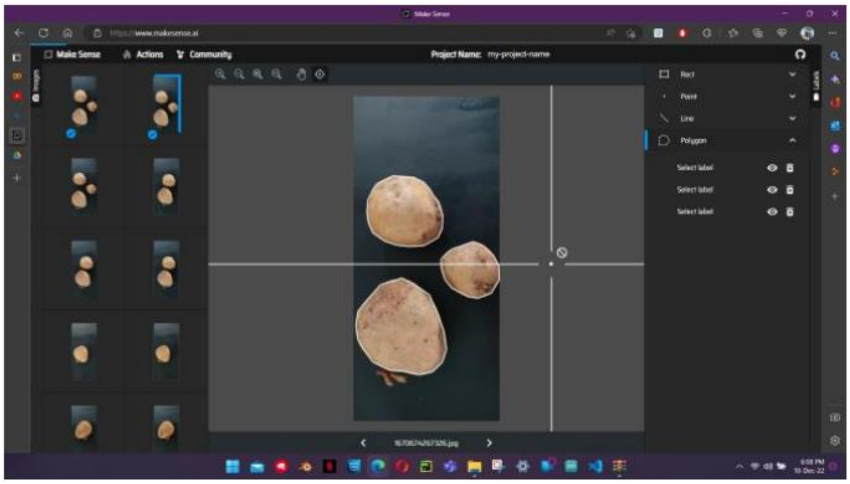

Figure 2. *Masking images and preparing custom image dataset*

3.2 Model development

A Convolutional Neural Network (CNN) was developed to classify whether the vegetable moving on conveyor belt is either a potato or not. The Convolutional Neural Network (CNN) model developed for the potato sorting system follows a multi-layer architecture designed using Python's TensorFlow and Keras libraries. The architecture consists of convolutional layers, activation functions, pooling layers, and fully connected layers:

- *Input Layer:* Receives input images of size 150x150 pixels.
- *Convolutional Layers*: Three convolutional layers with 32, 64, and 128 filters, using a 3x3 kernel.
- *Activation Functions:* ReLU activation is applied after each convolutional layer.
- *Pooling Layers:* MaxPooling layers are applied to reduce dimensionality.
- *Fully Connected Layers:* Two dense layers with 512 and 128 neurons, respectively.
- *Output Layer:* Softmax activation is used for final classification.
- *Optimizer:* Adam & *Loss function*: categorical cross-entropy

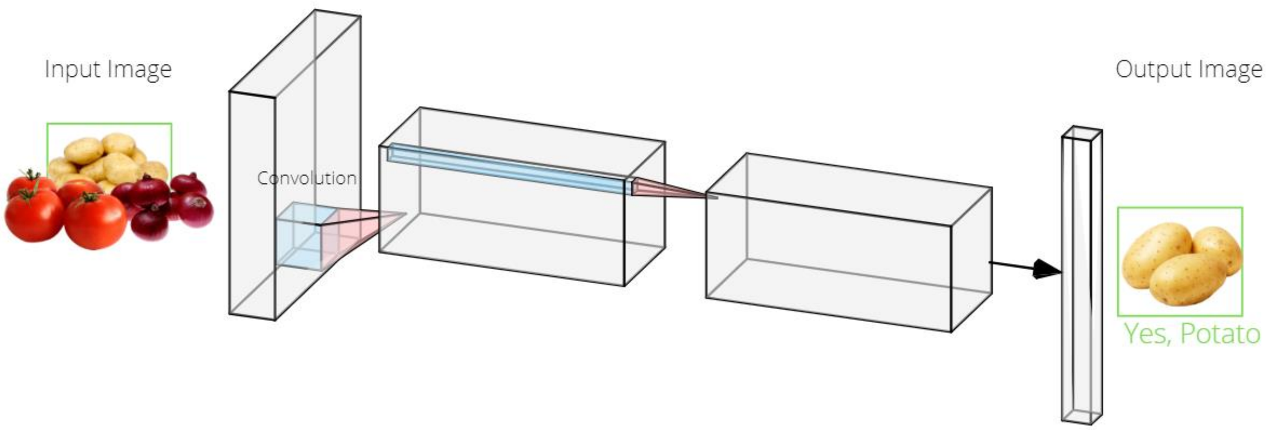

Figure 3. *Convolutional Neural Network AlexNet architecure and workflow*

The mathematical explanation focusing on each layer of CNN architecture is described below:

Input Layer

The input to the network consists of images of size 150×150×3(RGB format), where the input image ***X*** is represented as:

$$X \in \mathbb{R}^{150\times150\times3}$$

This input is then passed to the convolutional layers.

Convolutional Layers

The first three layers perform 2D convolution operations with filters of size 3×3 and increasing filter counts (32, 64, and 128 filters respectively). The convolution operation between X_{l-1}(the output from the previous layer) and filter F_l in layer l is defined as:

$$Y_l(i,j) = \sum_{m=1}^{3}\sum_{n=1}^{3}\sum_{c=1}^{C_{l-1}} X_{l-1}(i+m-1, j+n-1, c) \cdot F_l(m,n,c)$$

where:
- $Y_l(i,j)$ is the output feature map for the *l*-th layer.
- C_{l-1} is the number of channels in the input feature map X_{l-1} .
- F_l is the filter with size 3×3×C_{l-1}

After each convolution, a Rectified Linear Unit (ReLU) activation is applied:

$$ReLU(x) = max(0, x)$$

ReLU introduces non-linearity to the model, allowing it to learn more complex representations.

MaxPooling Layers
To reduce the spatial dimensions, MaxPooling is applied with a pool size of 2×2. The operation selects the maximum value from each 2×2 patch of the feature map:

$$Y_{pool}(i,j) = max\{Y_{conv}(2i+m, 2j+n) \mid m, n \in \{0,1\}$$

MaxPooling reduces the height and width of the feature maps, retaining only the most important features while reducing the computational load.

Flattening Layer
After passing through three convolutional layers and their associated pooling layers, the final feature maps are flattened into a 1D vector. If the spatial dimensions after the last pooling operation are $h \times w$ and the number of filters is C_3, the flattened vector size will be:

$$X_{flat} \in \mathbb{R}^{h\times w\times C_3}$$

This vector serves as the input to the fully connected layers.

Fully Connected Layers

Two fully connected layers are used, with 512 and 128 neurons, respectively. Each layer applies a linear transformation followed by a ReLU activation:

$$Z^{(l)} = W^{(l)} \cdot X^{(l-1)} + b^{(l)}$$

- $W^{(l)} \in \mathbb{R}^{k \times d}$ is the weight matrix, with *k* = 512 or 128 neurons.
- $b^{(l)} \in \mathbb{R}^{k}$ is the bias vector.
- $X^{(l-1)}$ is the input from the previous layer (the flattened vector or the output from the first dense layer).

The ReLU activation is then applied to the result:

$$A^{(l)} = ReLU(Z^{(l)})$$

Output Layer with Softmax Activation

The output layer is responsible for classifying the input image into one of 15 categories (vegetables). It outputs a probability distribution over the classes using the Softmax function:

$$P(y = i \mid x) = \frac{e^{Z_i}}{\sum_{j=1}^{15} e^{Z_j}}$$

where Z_i represents the score for class *i*, and the denominator is the sum of the exponentials of all class scores.

Loss Function (Categorical Cross-Entropy)

The model is trained using the categorical cross-entropy loss function, which measures the difference between the predicted probability distribution $\hat{P}$ and the true label *y*. The loss is calculated as:

$$L = -\sum_{i=1}^{15} y_i log(\hat{P}_i)$$

where y_i is 1 if class *i* is the true label and 0 otherwise, and $\hat{P}_i$ is the predicted probability for class *i*.

Optimizer (Adam)

The weights of the network are updated using the Adam optimizer. For each weight W_t at time step *t*, Adam computes updates using the first and second moment estimates m_t and v_t as:

$$W_{t+1} = W_t - \eta \cdot \frac{\widehat{m_t}}{\varepsilon + \sqrt{\widehat{v_t}}}$$

where η is the learning rate, $\widehat{m_t}$ and $\widehat{v_t}$ are bias-corrected estimates of the mean and variance, and ε is a small constant to prevent division by zero.

3.3 Contour area calculation

OpenCV- a python library for image processing and segmentation is utilized to monitor the movement of vegetables on a conveyor belt. It captures video and processes each frame in real time, creating contours around the moving potatoes.

- *Grayscale Conversion:* Simplifies the image to a single channel.
- *Noise Reduction:* Apply Gaussian blur to smooth the image.
- *Edge Detection:* Use Canny edge detection to highlight object boundaries.
- *Segmentation:* Apply background subtraction to separate potatoes from the background.
- *Contour Extraction:* Detect and extract contours using image processing techniques.

By calculating the area of these contours, OpenCV determines the size of each potato. This is done using Green's Theorem, which is a fundamental theorem in vector calculus. Green’s Theorem relates the double integral over a region to a line integral around the boundary of the region.

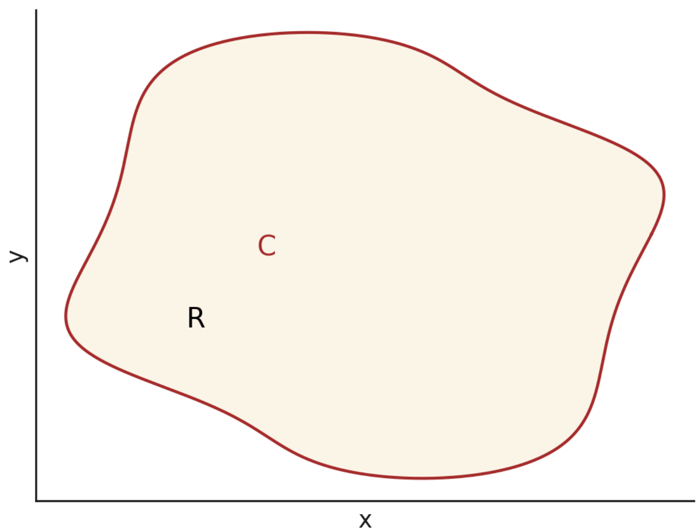

Figure 4. *Closed region for Green's Theorem application*

Mathematically, Green's Theorem is expressed as:

$$\oint_C (P\,dx + Q\,dy) = \iint_R \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y}\right) dA$$

Taking the right-hand side region, P $= -y$ and Q $= x$,

$$\oint_C (-y\,dx + x\,dy) = \iint_R \left(\frac{\partial x}{\partial x} - \frac{\partial(-y)}{\partial y}\right) dA$$

$$\oint_C (-y\,dx + x\,dy) = \iint_R (1 - (-1))\,dA$$

$$\oint_C (-y\,dx + x\,dy) = \iint_R 2\,dA$$

$$A = \left| \frac{1}{2} \oint_C (x\,dy - y\,dx) \right|$$

In this case, 'R' represents the region enclosed by a closed curve 'C'. The area of a simple, closed contour in an image can be calculated by converting the boundary into a sum of signed areas. OpenCV applies this method to determine the contour area in image processing.

The boundary *C* can be parameterized by $x = x(t)$ and $y = y(t)$, where $t \in [t_1, t_2]$. The differentials *dx* and *dy* can then be expressed as:

$$dx = \frac{dx}{dt}dt \text{ and } dy = \frac{dy}{dt}dt$$

$$A = \frac{1}{2}\int_{t_1}^{t_2} \left(x(t)\frac{dy}{dt} - y(t)\frac{dx}{dt} \right) dt$$

Using Green's Theorem simplifies the task of determining the area of irregular shapes like the potatoes in this project. The contour points obtained by OpenCV are used to compute the total area, which is then used to classify the potatoes as small, medium, or large.

- Large Potatoes: area > 50,000 pixels
- Medium Potatoes: 10,000 pixels < area ≤ 50,000 pixels
- Small Potatoes: 4,000 pixels ≤ area ≤ 10,000 pixels

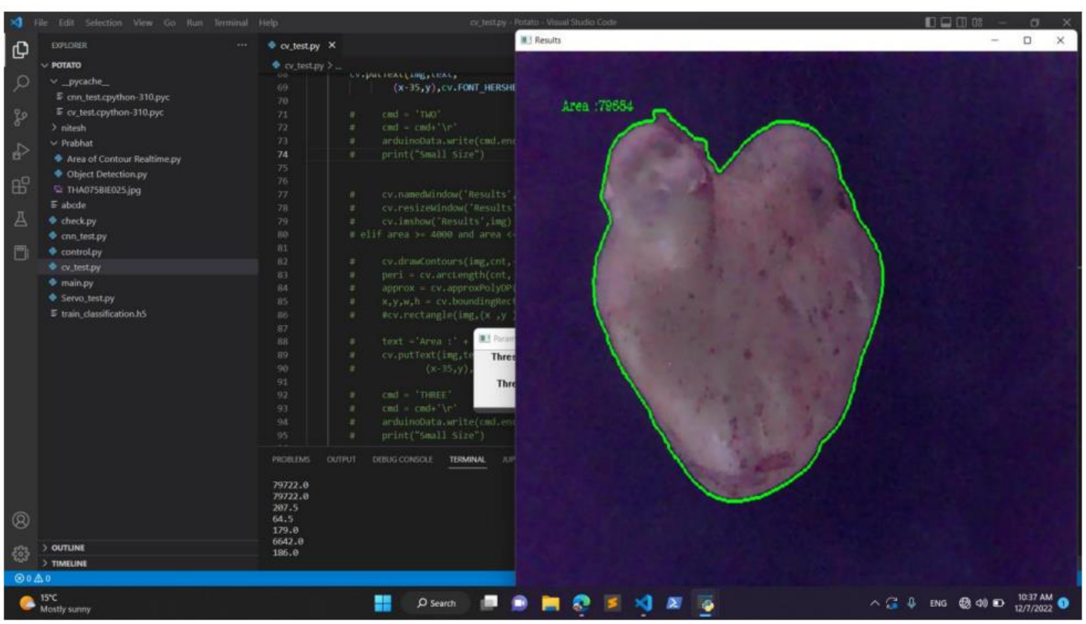

Figure 5. *Threshold adjustment for contour generation*

3.4 Software-hardware integration

The hardware components include a high-definition camera for real-time image capture, ultrasonic sensors to detect the proximity of potatoes, and servo motors controlled by an Arduino microcontroller. The communication between the hardware and software components is managed using the *Pyserial* library, which facilitates interaction between Python and Arduino.

Ultrasonic sensors are employed to detect when a potato approaches the servo arm. Once detected, the servo arms are activated to direct the potato into the correct container according to its size. Additionally, the motion of the conveyor belt is regulated by a wiper motor. The entire system is powered by a wiper motor that drives the conveyor belt.

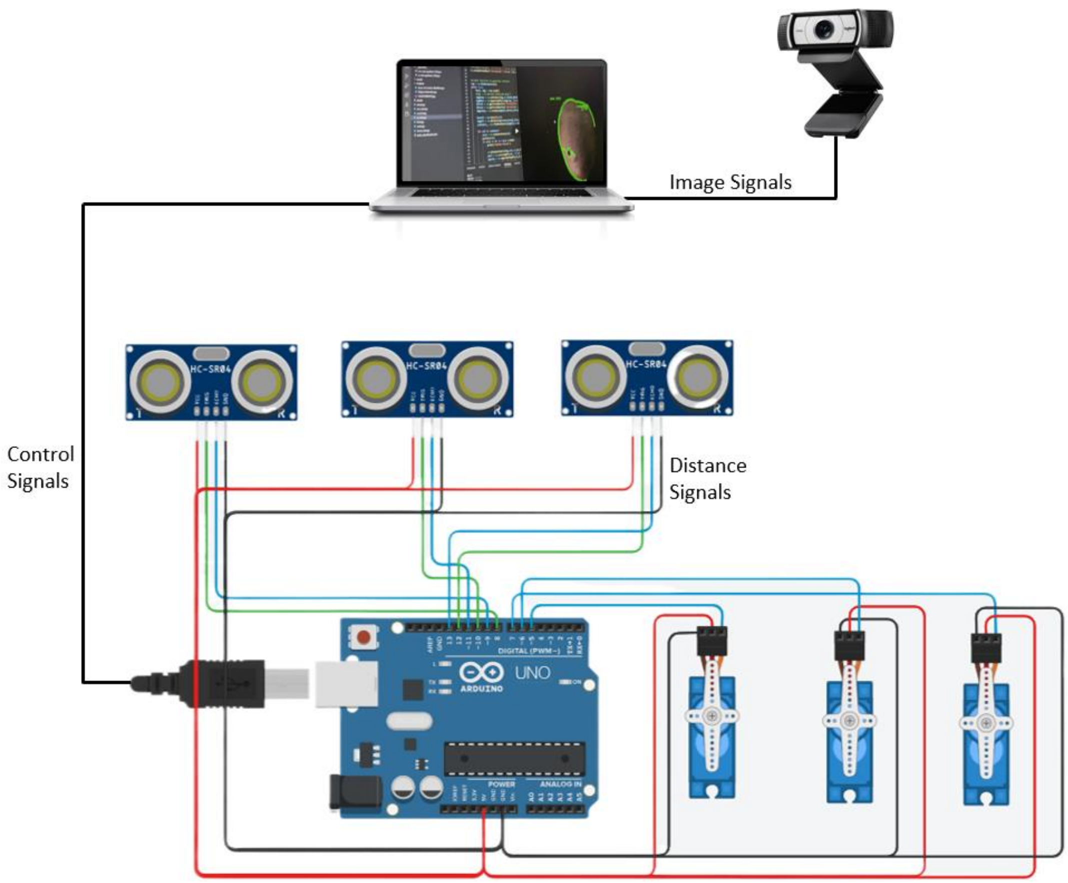

Figure 6. *System software-hardware integration*

3.5 System operation workflow

The process begins with a webcam that captures real-time images of vegetables as they are placed on a conveyor belt. These captured images are then processed using a Convolutional Neural Network (CNN), which is trained to detect and classify the vegetable type. The CNN algorithm identifies whether the detected vegetable is a potato. If the vegetable is not a potato, the system discards the object, bypassing further processing.

When a potato is detected, the system moves forward with two parallel operations. First, OpenCV is used to perform image processing and calculate the contour area of the potato, which helps determine its size. Simultaneously, ultrasonic sensors are employed to detect the presence and orientation of the potato for accurate sorting. The processed data from both these operations is fed into a servo motor system, which activates guiding mechanisms that sort the potatoes into different containers based on their size.

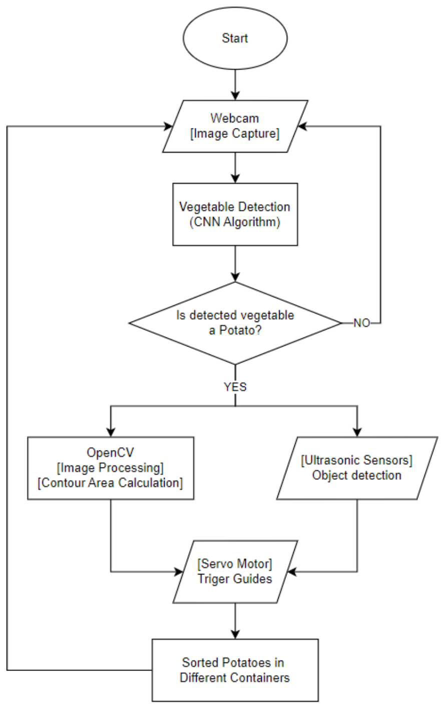

Figure 7. *Overall system workflow*

4. Results and Discussion

4.1 Sorting Efficiency Based on Potato Size

The system's sorting efficiency was measured using 50 samples for each size of potato. The results show that sorting speed decreases as potato size increases:

Table 1. *Rate of sorting potatoes per minute by size*

Potato Size	Avg. sorting time per potato (in seconds)	Avg. number of potatoes sorter per minute
Small	5	12
Medium	6.8	9
Large	8	7

- Small potatoes were sorted in an average of 5 seconds per potato, with a rate of 12 per minute.
- Medium potatoes took 6.8 seconds per potato, resulting in about 9 potatoes per minute
- Large potatoes took 8 seconds, with a rate of 7 potatoes per minute.

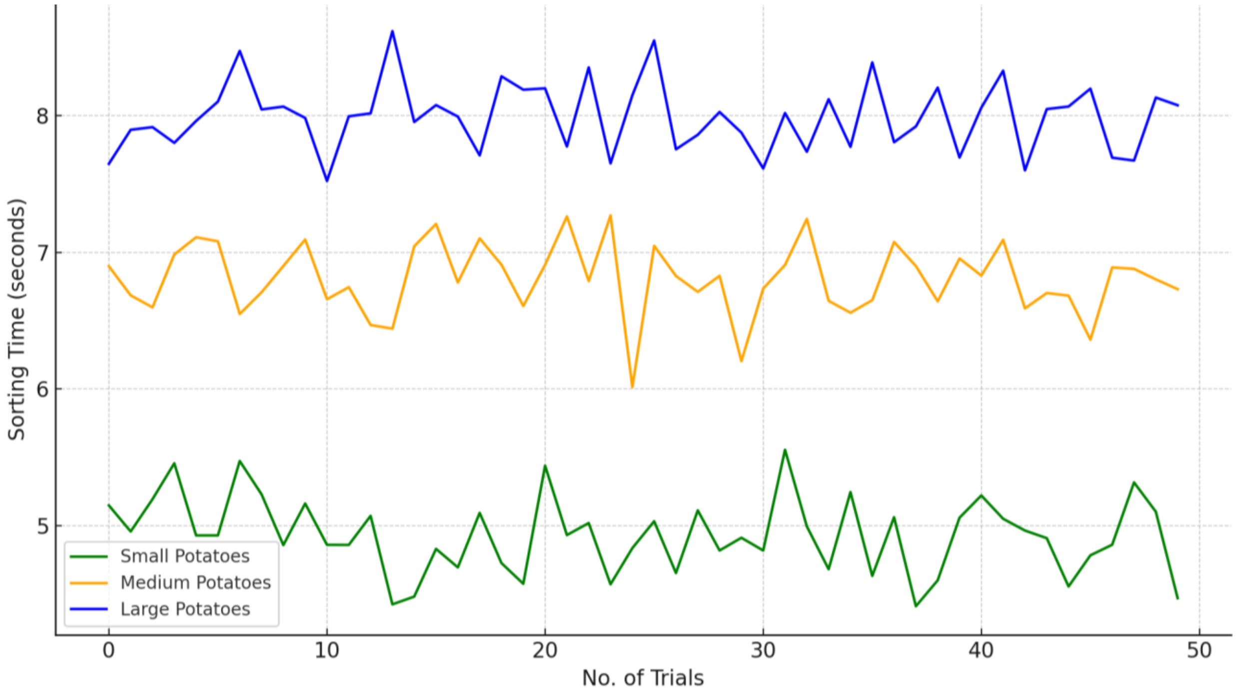

Figure 8. *Sorting time samples per size of potatoes*

The slower sorting rate for larger potatoes was due to the system sending only one potato at a time on the conveyor belt. This was necessary because a low-resolution camera caused frame lags, preventing simultaneous detection of multiple potatoes.

4.2 Economic Analysis

The automated system required an initial investment of Rs. 30,000, with annual maintenance and operating costs of Rs. 32,920. In contrast, manual labor would cost Rs. 216,000 annually. This results in annual savings of Rs. 183,080.

Table 2. *Cost comparison between automatic and manual sorting process*

Cost Item	**Amount (in NRs.)**	
	Automatic System	**Manual Labor**
Investment	30,000	-
Annual Maintenance Cost	10,000	-
Annual Operating Cost	1910 x 12 = 22,920	-
Total Annual Cost	32,920	18,000 x 12 = 216,000
Annual Savings	183,080	-

With these savings, the breakeven point for the system is reached in just two months, making the investment highly cost-effective in both the short and long term.

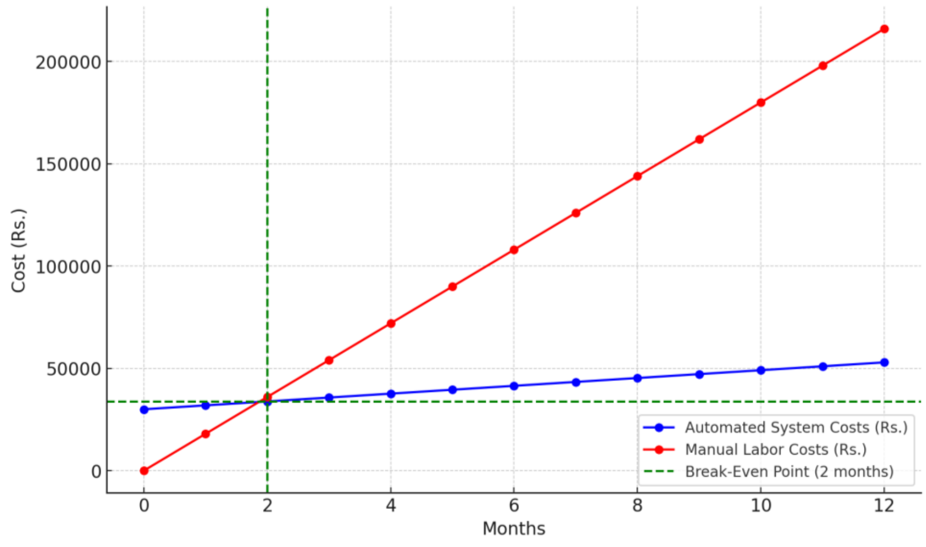

Figure 9. *Breakeven analysis of the project*

Despite slower sorting of larger potatoes due to camera limitations, the automated system proves to be far more economical than manual labor. Its ability to achieve breakeven within two months and deliver significant annual savings makes it a highly viable solution for large-scale operations. Upgrading the camera could further improve sorting rates, especially for larger potatoes.

4.3 Future Works

Future improvements for the system include:

- Installing higher-resolution cameras to improve image clarity and sorting precision.
- Refining the classification and image processing algorithms to increase both speed and accuracy.
- Upgrading the microcontrollers, motors, and sensors to boost overall system performance (currently in progress).
- Incorporating thermal sensors to identify and remove rotten potatoes from the sorting process.

5. Conclusion

The project successfully integrates smart manufacturing principles into Nepal's agricultural sector by automating the labor-intensive process of potato sorting. Utilizing computer vision and machine learning, the system improves sorting accuracy, speed, and scalability, addressing market demands for size-specific potatoes. Despite slower performance with larger potatoes, the system is highly cost-effective, reaching breakeven within two months and delivering substantial annual savings.

This project not only enhances efficiency but also aligns with Industry 4.0 concepts, optimizing supply chains and reducing manual labor. Future improvements, such as higher-resolution cameras and upgraded hardware, will further enhance performance, making it a key step toward modernizing Nepal's agricultural industry through automation and smart manufacturing.

References

[1] Abdullahi, H. S., Mahieddine, F., & Sheriff, R. (2017). Convolution neural network in precision agriculture for plant image recognition and classification. *Seventh International Conference on Innovative Computing Technology (INTECH)* (p. 256). New York: IEEE.

[2] Anderson, R., Hauagge, D. C., Wainer, J., & Goldenstein, S. (2010). Automatic fruit and vegetable classification from images. *Computers and Electronics in Agriculture* , 96-104.

[3] Green, G. (1828). Divergence Theorem. *An Essay on the Application of Mathematical Analysis to the Theories of Electricity and Magnetism*, 10-12.

[4] *Image Processing in OpenCV*. (n.d.). Retrieved from OpenCV: https://docs.opencv.org/4.x/d2/d96/tutorial_py_table_of_contents_imgproc.html

Annex A: Operating Cost and Break-Even Calculation

Power consumed by Servo motor & sensors: 15Watt (2.5 amps at 6 volts)

Power consumed by Wiper motor: 768Watt

Power consumed by Microcontroller: 0.29Watt (42mA at 7 volts)

Total Power Consumed = Power consumed by (Servo motor + Wiper motor + Microcontroller)

= (15 + 768 + 0.29) Watt

= 783 Watt/hrs.

With Total Operating Time: 10hr per day for 26 days/month

Total Power Consumed/day = Total power consumed × Total operating time

= 783 Watt × 10 hours

= 7830 Wh

= 7.83 kWh

Unit Cost of Electricity = Rs. 9.370 (For business purpose in Nepal)

Total Operating Cost/month = Total power consumed/month × Unit cost of electricity

= 7.83 × 26 × 9.3750

= Rs. 1910/month

= Rs. 22,920/year

Breakeven Point:

$$BEV = \frac{Annual\ Savings}{Investment}$$

$$= \frac{30{,}000}{183{,}080}$$

$$= 0.16\ yr$$

$$\approx \boldsymbol{2\ months}$$

Annex B: Design and Final Product

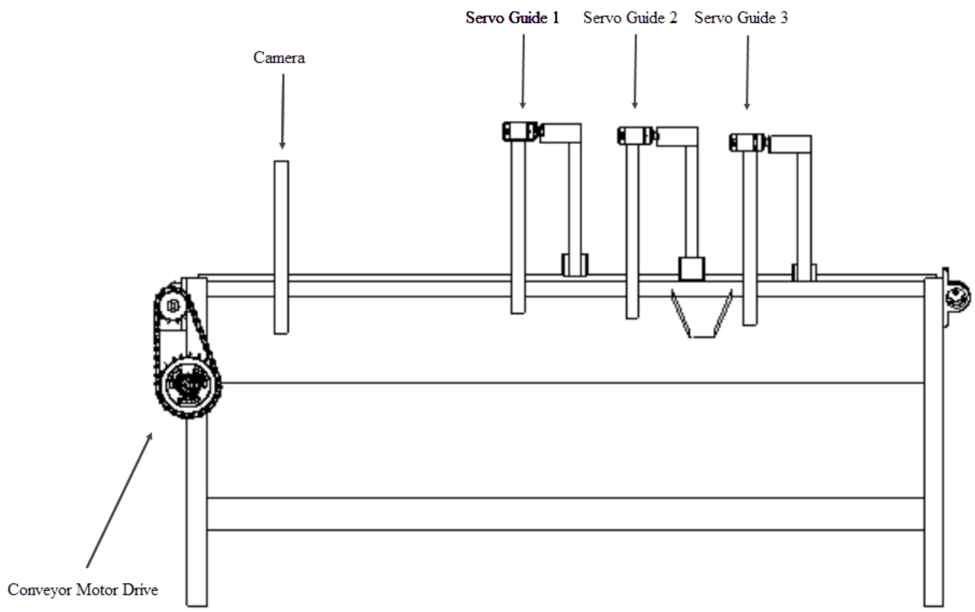

Figure 10. *Proposed Design*

Figure 11. *Final Product*