Affective and Pleasurable Design, Vol. 164, 2025, 42–52
https://doi.org/10.54941/ahfe1006059

Real-Time Adaptive Gripping Mechanism Using Object Classification and Feedback Control

Diptesh Kumar Mandal[1], Kazunori Kaede[1,2], and Keiichi Watanuki[1,2]

[1]Graduate School of Science and Engineering, Saitama University, 255 Shimo-Okubo, Sakura-ku, Saitama-shi, Saitama, 338-8570, Japan
[2]Advanced Institute of Innovative Technology, Saitama University, 255 Shimo-Okubo, Sakura-ku, Saitama-shi, Saitama, 338-8570, Japan

ABSTRACT

This paper presents a novel approach to adaptive robotic gripping by integrating force-sensitive resistors (FSRs) into a three-fingered robotic gripper fabricated using TPU 95A and PLA. The proposed methodology extends prior research on vision-based object classification and incorporates force-sensitive feedback for dynamic grip adjustment. By calibrating FSR strips and deriving force-voltage equations, we have developed a gripping mechanism that dynamically adjusts its pressure based on object characteristics. The force thresholds and minimum and maximum gripper pressures have been empirically determined through experimentation. This research provides a practical and scalable solution for adaptive robotic gripping, with applications in industrial automation and assistive robotics.

Keywords: Adaptive gripper, Force-sensitive resistors, TPU 95A, Machine learning, Robotic manipulation, Dynamic gripping

INTRODUCTION

Robotic grippers play a crucial role in industrial and assistive robotics, enabling precise object manipulation. Traditional gripping mechanisms rely on pre-set parameters or static force application, which limits adaptability in unstructured environments. Recent advancements in machine learning have allowed robots to classify objects visually, but real-time grip adjustment remains an open challenge.

Building on prior research presented at AHFE 2024 Hawaii, which utilized ResNet50 for vision-based classification of hard and soft objects, this study introduces a force-sensitive gripping system. The three-finger robotic gripper, fabricated using TPU 95A for flexibility and PLA for structural rigidity, is integrated with FSR strips to enable dynamic pressure adjustment. The objective is to enhance grasp stability by using feedback from the FSRs to control grip strength.

The core contribution of this work lies in integrating Force Sensitive Resistors (FSRs) into the robotic gripper design to facilitate real-time pressure sensing and generate feedback. These sensors are strategically placed on the gripper's contact surfaces to capture precise pressure data during gripping. This feedback, combined with the classification result, ensures the gripper dynamically adjusts its force in real time to handle a wide range of objects, including fragile items such as glass, compressible objects like cushions and rigid heavy objects like steel mugs.

This research bridges the gap between machine learning models and real-world robotic applications, demonstrating how object classification can cause dynamic interactions with physical environment.

BACKGROUND OF THE RESEARCH AND RELATED WORKS

Previously we have developed a TensorFlow based deep learning model that has been trained using the ResNet50 architecture (a pre-trained model which has been trained on the ImageNet dataset). This model was used on the CIFAR 100 dataset which was divided into 50,000 training images and 10,000 validation or test images. The images obtained from CIFAR 100 were normalised to aid in speeding up the convergence of the model training and this also helped improve the training and validation accuracy.

The TensorFlow model was able to classify the images of the CIFAR 100 dataset into hard and soft objects with an impressive accuracy of 83% for the training dataset and 80.25% for the validation dataset. The details of the research are available on the paper by Diptesh et al. (2024) titled, "Advancing Vision-based Adaptive Gripping Technology with Machine Learning: Leveraging Pre-trained Models for Enhanced Object Classification".

Similar works on the topic related to robotic gripping have been carried out by Calandra et al. (2018) who developed a deep learning model that utilised tactile data to predict grasp outcomes which in turn enabled a robotic hand to identify objects and adjust its grip. Li et al. (2020) in their paper "Design and performance characterization of a soft robot hand with fingertip haptic feedback for teleoperation" emphasised on fingertip haptic feedback thus relying extensively on tactile or sensor output only. While these studies have been able to innovatively capture the idea of adaptive gripping on one hand, it also brings us to a critical conclusion that these ideas are filled with sensor intensive operations that leads to complexities in build up and operation often limiting the use of such robots in scenarios where sensory input is restricted.

This study stands out by emphasizing the use of visual data for object classification. Unlike tactile sensors, visual data enables information gathering from a distance, simplifies hardware requirements, and seamlessly integrates with existing computer vision systems. Several studies have investigated the application of FSR sensors in robotics. However, few studies have implemented real-time dynamic adjustments based on measured force thresholds. This paper builds upon prior research by proposing a novel approach that combines FSR calibration, adaptive control, and real-time feedback to improve robotic gripping performance.

The ability of robotic systems to autonomously adjust their grip without continuous human intervention or extensive reprogramming marks a significant advancement. This research demonstrates how visual data and machine learning algorithms can enhance robotic autonomy and versatility, reducing reliance on human operators while improving efficiency. The proposed approach offers a scalable solution applicable across multiple industries, contributing to safer, more reliable, and highly effective robotic systems in diverse operational environments.

METHODOLOGY

Building a 3 Fingered Robotic Gripper

The development of a three-fingered robotic gripper involved designing a precise CAD model using Autodesk Fusion 360, ensuring an optimal balance between mechanical strength and flexibility. The gripper comprises a rigid wrist structure fabricated from PLA, providing a stable base for mounting and actuation, while the fingers are printed using TPU 95A to allow controlled compliance and deformation for adaptive gripping.

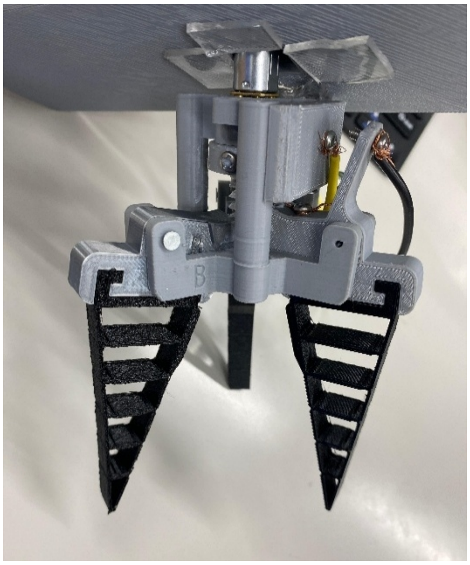


Figure 1: The robotic gripper (fingers made of TPU & parts of the wrist made of PLA).

The CAD model was exported in STL format and processed in IdeaMaker; the slicing software optimized for the Raise3D Pro 2 printer. Key slicing parameters for PLA included a 0.2mm layer height, 20% infill, and support structures, ensuring structural integrity. For TPU 95A, special considerations were made, including slower print speeds (20 mm/s), retraction disabled, and travel optimization, preventing extrusion issues common with flexible filaments.

Printing was conducted separately for the wrist and fingers, ensuring accurate dimensional tolerances for seamless assembly. A N95 DC geared motor was suitably placed at the intersection of the 3 fingers which is being used to control the opening and closing of the gripper fingers.

Force Sensitive Resistor (FSR) Calibration Process

The accuracy of force measurements using Force Sensitive Resistor (FSR) sensors is highly dependent on proper calibration. Calibration ensures that the sensor provides reliable and repeatable measurements, allowing for precise force estimation during gripping operations.

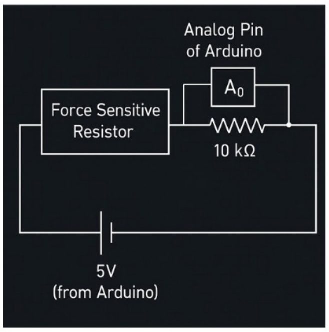


Figure 2: Voltage divider circuit used for calibration of the force sensitive resistor.

To achieve this, the FSR sensors were calibrated using a Weight vs. Voltage dataset, to establish a direct relationship between applied force and sensor output.

Calibration Procedure

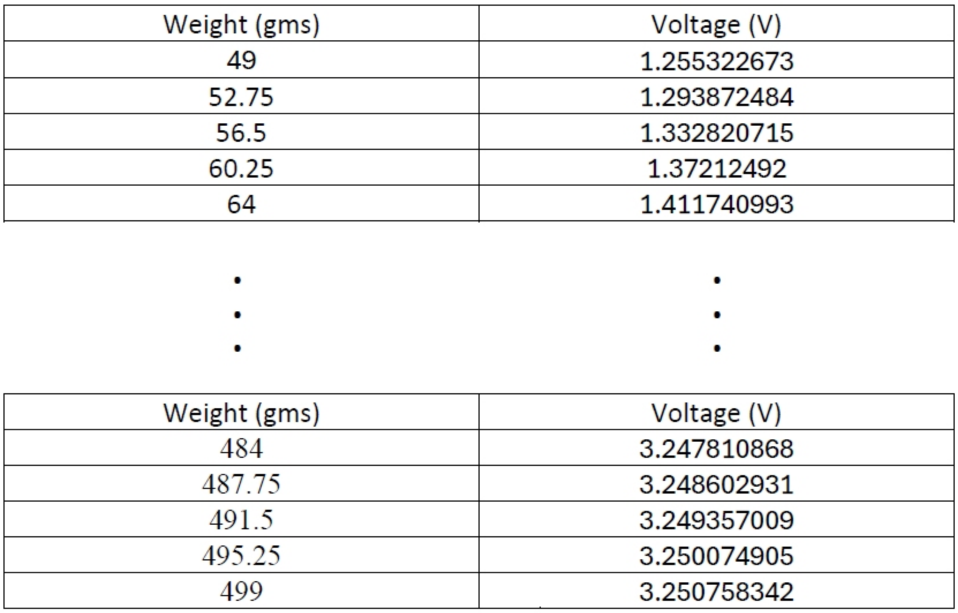
1. Experimental Setup:
    a. The FSR sensors were mounted on a stable, flat surface to ensure consistent contact with applied weights.
    b. A microcontroller (Arduino UNO) was used to supply a regulated 5V input to the sensor circuit and read the corresponding analog voltage output.
2. Weight Application and Data Collection:
    a. A series of known weights, ranging from light to heavy, were placed incrementally on the sensor surface.
    b. The voltage output corresponding to each weight was recorded in real time.

Curve Fitting and Derivation of Equation for Force vs Voltage

The voltage output from the FSR corresponding to the application of the known weights was carefully observed. The trend showed saturation after a certain volatge level was reached corresponding to application of weights. We used a polynomial regression or power-law curve fitting approach to develop an empirical equation that accurately models the force response of the sensor.

The values obtained from our experiment cited in Figure 3 below were used to calculate the relationship between Force and Voltage by fitting a third order polynomial of the nature:

$$F(V) = AV^3 + BV^2 + CV + D$$



Weight (gms)	Voltage (V)
49	1.255322673
52.75	1.293872484
56.5	1.332820715
60.25	1.37212492
64	1.411740993

Weight (gms)	Voltage (V)
484	3.247810868
487.75	3.248602931
491.5	3.249357009
495.25	3.250074905
499	3.250758342

Figure 3: Table showing relationship between weight and voltage for a part of the values (first 5 and last 5) obtained from the experiment.

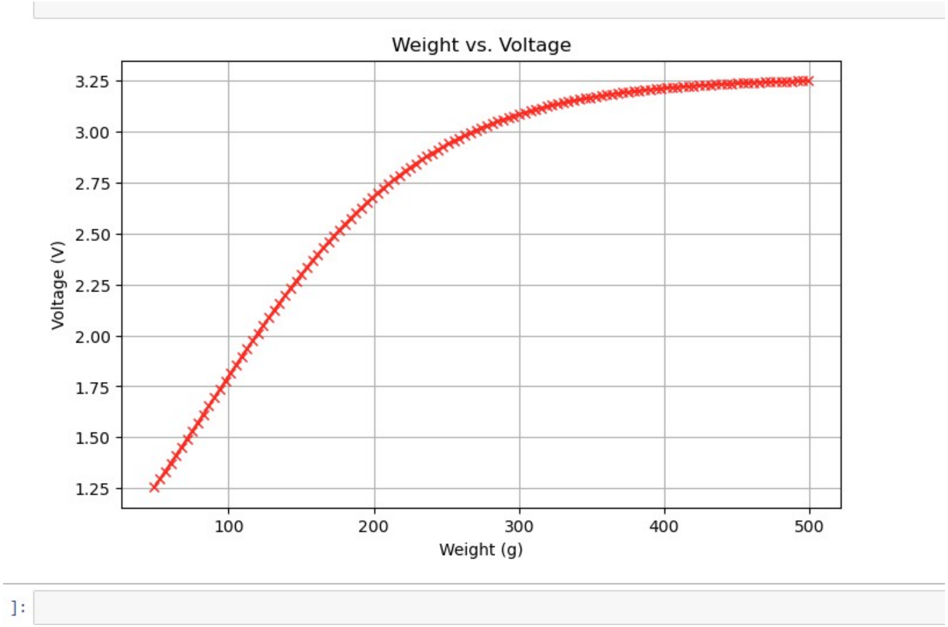
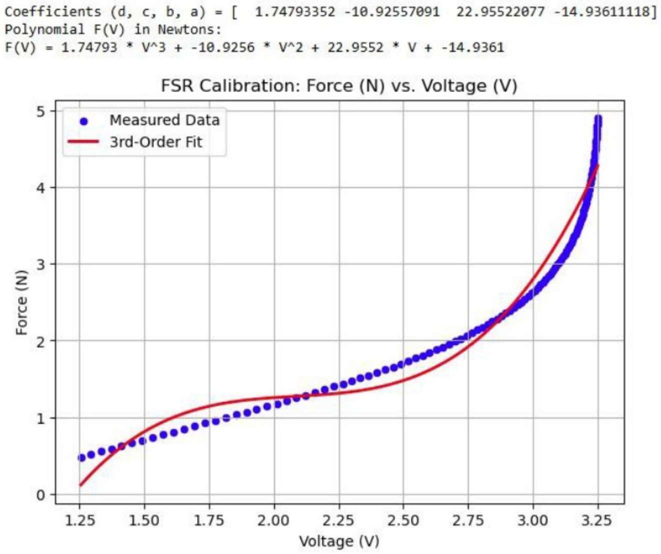


Figure 4: Weight vs voltage plot as read from the experiment for calibration of FSR.

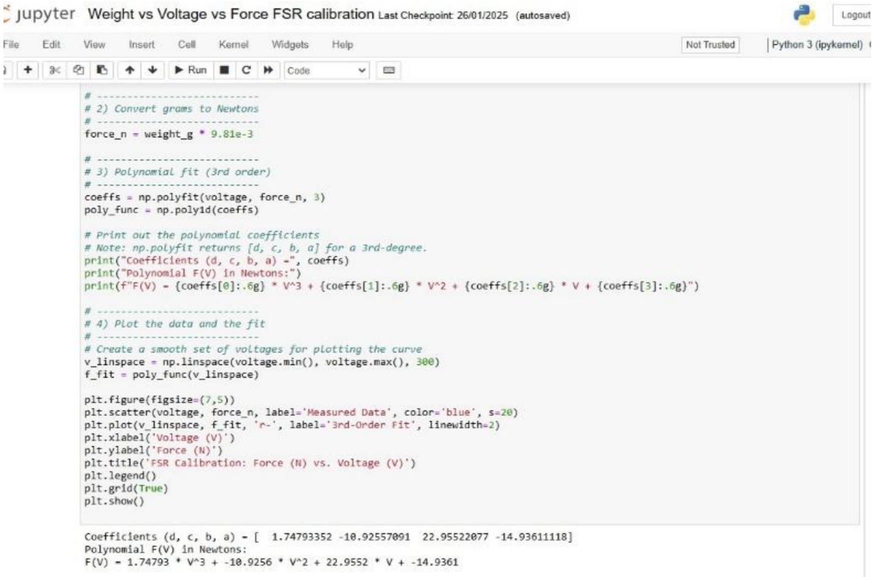
The calculation was done using python coding and the resulting polynomial expression is expressed as:

$$F(V) = 1.74793V^3 - 10.9256V^2 + 22.9552V - 14.9361$$



```
Coefficients (d, c, b, a) = [  1.74793352 -10.92557091  22.95522077 -14.93611118]
Polynomial F(V) in Newtons:
F(V) = 1.74793 * V^3 + -10.9256 * V^2 + 22.9552 * V + -14.9361
```

Figure 5: Graphical representation of the 3rd order polynomial equation for force vs voltage.



```
# --------------------------
# 2) Convert grams to Newtons
# --------------------------
force_n = weight_g * 9.81e-3

# --------------------------
# 3) Polynomial fit (3rd order)
# --------------------------
coeffs = np.polyfit(voltage, force_n, 3)
poly_func = np.poly1d(coeffs)

# Print out the polynomial coefficients
# Note: np.polyfit returns [d, c, b, a] for a 3rd-degree.
print("Coefficients (d, c, b, a) =", coeffs)
print("Polynomial F(V) in Newtons:")
print(f"F(V) = {coeffs[0]:.6g} * V^3 + {coeffs[1]:.6g} * V^2 + {coeffs[2]:.6g} * V + {coeffs[3]:.6g}")

# --------------------------
# 4) Plot the data and the fit
# --------------------------
# Create a smooth set of voltages for plotting the curve
v_linspace = np.linspace(voltage.min(), voltage.max(), 300)
f_fit = poly_func(v_linspace)

plt.figure(figsize=(7,5))
plt.scatter(voltage, force_n, label='Measured Data', color='blue', s=20)
plt.plot(v_linspace, f_fit, 'r-', label='3rd-Order Fit', linewidth=2)
plt.xlabel('Voltage (V)')
plt.ylabel('Force (N)')
plt.title('FSR Calibration: Force (N) vs. Voltage (V)')
plt.legend()
plt.grid(True)
plt.show()
```

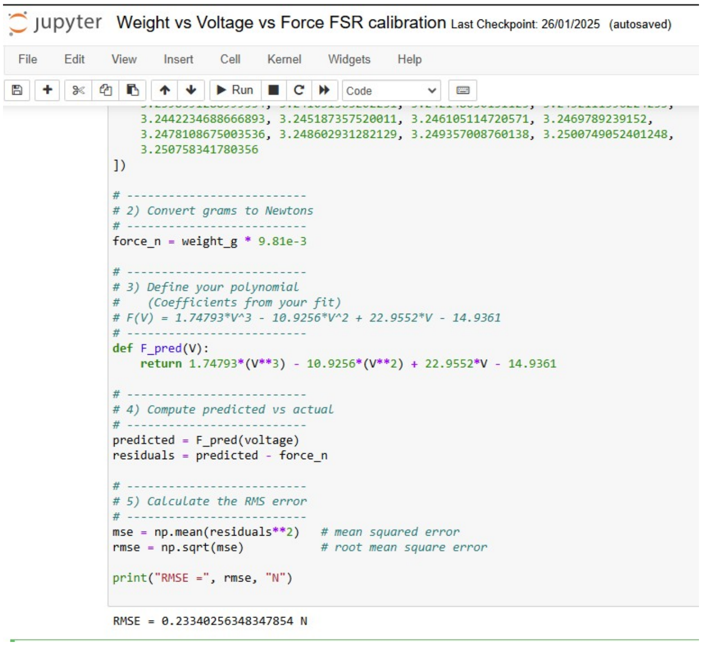
```
Coefficients (d, c, b, a) = [  1.74793352 -10.92557091  22.95522077 -14.93611118]
Polynomial F(V) in Newtons:
F(V) = 1.74793 * V^3 + -10.9256 * V^2 + 22.9552 * V + -14.9361
```

Figure 6: Python code snippet for calculation of relation between force and voltage.

This polynomial equation allows force values to be estimated from voltage readings, ensuring accurate force measurement during gripping operations. Multiple trials were conducted to validate the consistency of the calibration equation and minimize sensor drift errors.

Validation and Error Analysis

We ran a Root Mean Square (RMS) error check operation on the predicted values of the force versus the actual values of force applied on the Force sensitive resistor (FSR). RMS error tells us, on an average how many Newtons away is our 3rd order polynomial prediction from the actual measured forces. It is widely accepted that a RMS error anywhere below ~0.25 N is typically good for an FSR based fit over a broad force range.



```
    3.2442234688666893, 3.245187357520011, 3.246105114720571, 3.2469789239152,
    3.2478108675003536, 3.248602931282129, 3.249357008760138, 3.2500749052401248,
    3.250758341780356
])

# -------------------------
# 2) Convert grams to Newtons
# -------------------------
force_n = weight_g * 9.81e-3

# -------------------------
# 3) Define your polynomial
#    (Coefficients from your fit)
# F(V) = 1.74793*V^3 - 10.9256*V^2 + 22.9552*V - 14.9361
# -------------------------
def F_pred(V):
    return 1.74793*(V**3) - 10.9256*(V**2) + 22.9552*V - 14.9361

# -------------------------
# 4) Compute predicted vs actual
# -------------------------
predicted = F_pred(voltage)
residuals = predicted - force_n

# -------------------------
# 5) Calculate the RMS error
# -------------------------
mse = np.mean(residuals**2)   # mean squared error
rmse = np.sqrt(mse)           # root mean square error

print("RMSE =", rmse, "N")
```

```
RMSE = 0.23340256348347854 N
```

Figure 7: RMS error as obtained from our python code snippet is 0.2334.

The RMS error between the 3rd order polynomial fit and the actual values of forces is 0.2334 which is well within the accepted range of 0.25.

Experimental Observations and Force Threshold Determination

During the experimentation phase, it was observed that the Force Sensitive Resistor (FSR) strips exhibit a non-linear response to applied force, particularly at very low and very high, pressure levels. The FSRs do not generate an immediate voltage output upon initial contact but require a minimum threshold force before registering a measurable voltage response. This characteristic was carefully considered when determining the lower threshold pressure required for object gripping.

Given that the FSRs are mounted on the fingertips of the robotic gripper fabricated using TPU 95A premium material, an inherent material deformation occurs prior to force transmission to the FSRs. TPU 95A, being a flexible and compliant material, undergoes initial deformation under applied force, resulting in a delay before sufficient force is exerted on the FSR to generate a detectable voltage reading on the Arduino serial monitor. Consequently, the point at which the first measurable voltage is recorded during the gripping process is identified as the lower benchmark pressure threshold. This value represents the minimum amount of pressure required to establish initial contact with an object but does not necessarily ensure a stable grasp.

Following the initial detection of a voltage response, the gripping process continues with increasing pressure exerted by the N95 DC geared motor driving the gripper fingers. As the applied normal force increases, the corresponding voltage readings on the serial monitor also increase, reflecting the rising reaction force experienced by the FSR. This trend continues until the voltage readings exhibit saturation, indicating that further increments in gripping force yield negligible increases in voltage output. This saturation point is considered the maximum pressure threshold, representing the upper limit of the gripping force necessary to secure an object firmly.

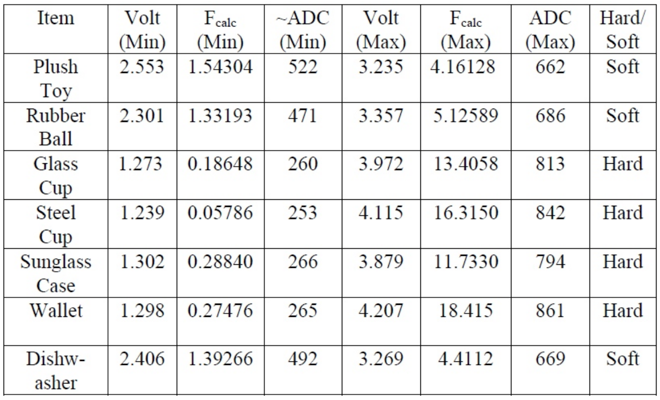
The force corresponding to both the lower and upper pressure thresholds has been quantified using the previously derived mathematical equation that establishes the relationship between voltage and force. These experimentally determined thresholds provide critical insights into the dynamic gripping behaviour of the robotic gripper and ensure optimal force regulation for different object categories. By leveraging this information, the gripper can adaptively modulate its gripping force to prevent excessive deformation of soft objects while ensuring secure handling of rigid objects.

RESULTS AND DISCUSSION

We put to test over 70 small objects that we encounter in our daily lives and noted down the Voltage values as observed on the Arduino serial monitor. The method of experimentation was the same as has been described above in the previous section before. It is worthwhile to mention here that since the power source is 5 Volts and we are using a 10-bit ADC (it has 1024 levels from 0 to 1023) the ADC value has been arrived at by using the equation below:

$$\text{ADC Value } = \left(\frac{\text{Voltage Input}}{\text{Reference Voltage}}\right) \times 1023$$

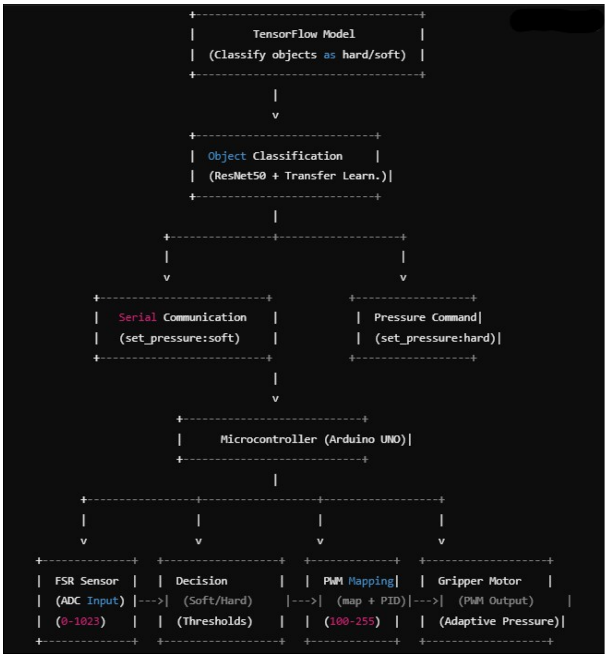
We have mentioned only a few items in the table above due to paucity of space although the representative categories have been covered.



Item	Volt (Min)	F_{calc} (Min)	~ADC (Min)	Volt (Max)	F_{calc} (Max)	ADC (Max)	Hard/ Soft
Plush Toy	2.553	1.54304	522	3.235	4.16128	662	Soft
Rubber Ball	2.301	1.33193	471	3.357	5.12589	686	Soft
Glass Cup	1.273	0.18648	260	3.972	13.4058	813	Hard
Steel Cup	1.239	0.05786	253	4.115	16.3150	842	Hard
Sunglass Case	1.302	0.28840	266	3.879	11.7330	794	Hard
Wallet	1.298	0.27476	265	4.207	18.415	861	Hard
Dishw-asher	2.406	1.39266	492	3.269	4.4112	669	Soft

Figure 8: Experimentation results for few of the objects which were put to test (values of F_{calc} are per finger and in Newton).

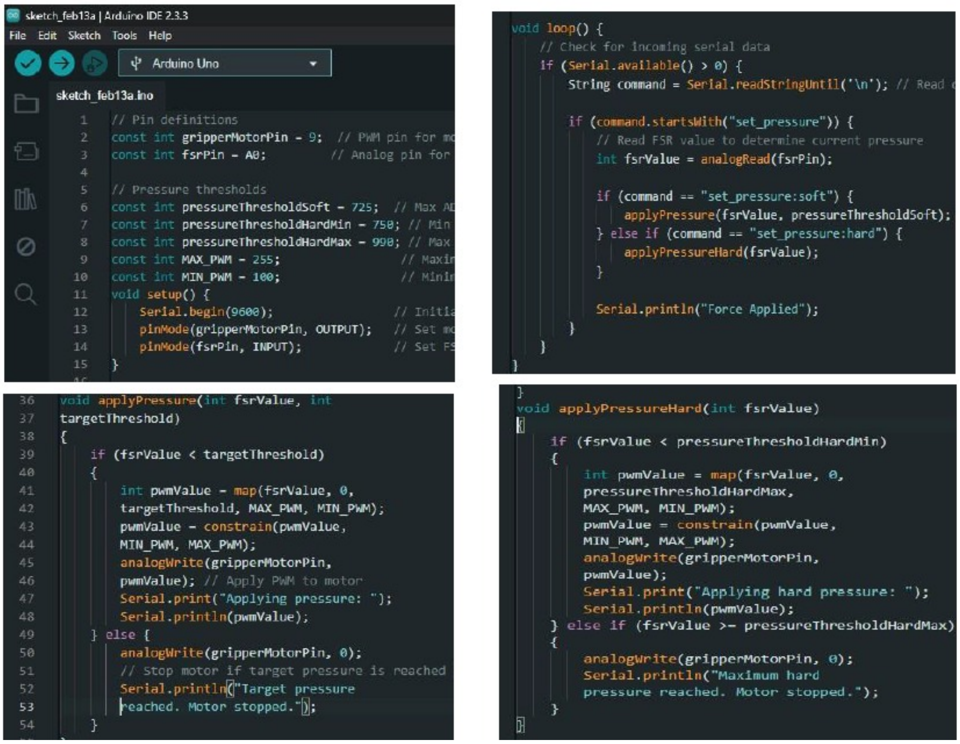
It is clearly observed that the values for maximum threshold force being applied in case of hard and soft objects follow an observable trend with the $\mathrm{ADC}_{\mathrm{max}}$ values being on the higher side for hard objects as compared to that of the soft objects. However, when we see the $\mathrm{ADC}_{\mathrm{min}}$ values we observe that the trend is opposite with the soft objects exhibiting a higher $\mathrm{ADC}_{\mathrm{min}}$ as compared to the hard objects. On detailed inspection it is also noted that the $\mathrm{ADC}_{\mathrm{min}}$ values for the plush toy and dishwasher sponge are higher than that of the rubber ball. It is hereby concluded that the objects like plush toy or the sponge are extremely compressible in nature and hence require a substantial amount of gripping force before the FSRs mounted on the fingers can sense the normal reaction force and send voltage readings to the Arduino serial monitor. This is not the case with incompressible harder objects where the normal reaction force is easily sensed by the FSRs. We also conclude that the values of $\mathrm{ADC}_{\mathrm{max}}$ serve as good benchmarks to implement the concept of adaptive gripping by classifying objects as hard and soft. We have observed that the $\mathrm{ADC}_{\mathrm{max}}$ values for soft objects are less than 725 while that of hard objects are greater than 750. This logic has been drafted into an Arduino algorithm combined with the algorithm from the previous research paper by Diptesh et al. (2024) titled, “Advancing Vision-based Adaptive Gripping Technology with Machine Learning: Leveraging Pre-trained Models for Enhanced Object Classification” to implement the concept of adaptive gripping the model architecture for which has been laid out in Figure 9.



[redacted]

```
                       +-----------------------------------+
                       |         TensorFlow Model          |
                       |  (Classify objects as hard/soft)  |
                       +-----------------------------------+
                                    |
                                    v
                       +----------------------------+
                       |  Object Classification     |
                       |  (ResNet50 + Transfer Learn.)|
                       +----------------------------+
                                    |
                    +---------------+-------------------+
                    |                                   |
                    v                                   v
         +-------------------------+           +------------------+
         |   Serial Communication  |           |  Pressure Command|
         |   (set_pressure:soft)   |           |  (set_pressure:hard)|
         +-------------------------+           +------------------+
                                    |
                                    v
                      +---------------------------+
                      |      Microcontroller (Arduino UNO)|
                      +---------------------------+
                                    |
       +----------------+-----------------+-----------------+
       |                |                 |                 |
       v                v                 v                 v
+-------------+  +-----------------+  +------------+  +------------------+
|  FSR Sensor |  |  Decision       |  |  PWM Mapping|  |  Gripper Motor   |
|  (ADC Input) |--->|  (Soft/Hard)     |--->|  (map + PID)|--->|  (PWM Output)     |
|  (0-1023)   |  |  (Thresholds)   |  |  (100-255)  |  |  (Adaptive Pressure)|
+-------------+  +-----------------+  +------------+  +------------------+
```

Figure 9: Final model architecture combining the TensorFlow model and the Arduino bridge code.



```
// Pin definitions
const int gripperMotorPin = 9;  // PWM pin for mo
const int fsrPin = A0;         // Analog pin for

// Pressure thresholds
const int pressureThresholdSoft = 725;  // Max AD
const int pressureThresholdHardMin = 750; // Min
const int pressureThresholdHardMax = 990; // Max
const int MAX_PWM = 255;                   // Maxin
const int MIN_PWM = 100;                   // Minin
void setup() {
    Serial.begin(9600);                  // Initia
    pinMode(gripperMotorPin, OUTPUT);    // Set mo
    pinMode(fsrPin, INPUT);              // Set FS
}
```

```
void loop() {
    // Check for incoming serial data
    if (Serial.available() > 0) {
        String command = Serial.readStringUntil('\n'); // Read

        if (command.startsWith("set_pressure")) {
            // Read FSR value to determine current pressure
            int fsrValue = analogRead(fsrPin);

            if (command == "set_pressure:soft") {
                applyPressure(fsrValue, pressureThresholdSoft);
            } else if (command == "set_pressure:hard") {
                applyPressureHard(fsrValue);
            }

            Serial.println("Force Applied");
        }
    }
}
```

```
void applyPressure(int fsrValue, int
targetThreshold)
{
    if (fsrValue < targetThreshold)
    {
        int pwmValue = map(fsrValue, 0,
        targetThreshold, MAX_PWM, MIN_PWM);
        pwmValue = constrain(pwmValue,
        MIN_PWM, MAX_PWM);
        analogWrite(gripperMotorPin,
        pwmValue); // Apply PWM to motor
        Serial.print("Applying pressure: ");
        Serial.println(pwmValue);
    } else {
        analogWrite(gripperMotorPin, 0);
        // Stop motor if target pressure is reached
        Serial.println("Target pressure
        reached. Motor stopped.");
    }
```

```
}
void applyPressureHard(int fsrValue)
{
    if (fsrValue < pressureThresholdHardMin)
    {
        int pwmValue = map(fsrValue, 0,
        pressureThresholdHardMax,
        MAX_PWM, MIN_PWM);
        pwmValue = constrain(pwmValue,
        MIN_PWM, MAX_PWM);
        analogWrite(gripperMotorPin,
        pwmValue);
        Serial.print("Applying hard pressure: ");
        Serial.println(pwmValue);
    } else if (fsrValue >= pressureThresholdHardMax)
    {
        analogWrite(gripperMotorPin, 0);
        Serial.println("Maximum hard
        pressure reached. Motor stopped.");
    }
}
```

Figure 10: Arduino IDE code to implement adaptive gripping.

CONCLUSION

The results of this study highlight the potential of this approach in improving robotic gripping strategies for a range of application. Our study has been limited to distinguishing between hard and soft objects only but there lies a huge scope in the future to extend this study to moderately soft, fragile, incompressible and semi-hard objects by further categorising the ADC values obtained from the voltage readings of the Force Sensitive Resistors (FSRs). The findings of this study contribute to the ongoing advancement of adaptive robotic manipulation, providing a foundation for further innovations in robotic grasping technology.

REFERENCES

Calandra, R., Owens, A., Jayaraman, D., Lin, J., Yuan, W., Malik, J. and Adelson, E. (2018) 'The feeling of success: Does touch sensing help predict grasp outcomes?', *IEEE Transactions on Robotics*, 34(6), pp. 1496–1505.

Chi, Y., Zhao, Y., Hong, Y., Li, Y. and Yin, J. (2023) 'A perspective on miniature soft robotics: Actuation, fabrication, control, and applications'. Available at: https://doi.org/10.1002/aisy.202300063.

Gjerstad, T. B., Lien, T. K. and Buljo, J. O. (2006) 'Handle of non-rigid products using a compact needle gripper', *Proceedings of the 39th CIRP International Seminar on Manufacturing Systems*, pp. 145–151.

Li, M., Zhuo, Y., Chen, J., He, B., Xu, G., Xie, J. and Yao, W. (2020) 'Design and performance characterization of a soft robot hand with fingertip haptic feedback for teleoperation', *Advanced Robotics*, 34(23), pp. 1491–1505. Available at: https://doi.org/10.1080/01691864.2020.1822913.

Liu, Y., Hou, J., Li, C. and Wang, X. (2023) 'Intelligent soft robotic grippers for agricultural and food product handling: A brief review with a focus on design and control'. Available at: https://doi.org/10.1002/aisy.202300233.