Handwritten Digit Classification using Raspberry Pi Pico

Kodali Mohana Siva Ramakrishna
CSE 2021 - 2025
Amrita Vishwa Vidyapeetham
Coimbatore, India
Krishnakodali61@gmail.com

Prabhada Naga Srinivas
CSE 2021 - 2025
Amrita Vishwa Vidyapeetham
Coimbatore, India
srinivasprabhala18@gmail.com

Sri Venkatesh Lankalapalli
CSE 2021 - 2025
Amrita Vishwa Vidyapeetham
Coimbatore, India
srivenkatesh13007@gmail.com

Thota Yaswanth Avinash
CSE 2021 – 2025
Amrita Vishwa Vidyapeetham
Coimbatore, India
yaswanthavi99@gmail.com

***Abstract*— This report outlines the development of a handwritten digit classification system using a Raspberry Pi Pico, an OV7670 camera module, a 128x160 TFT LCD display, and machine learning techniques. The project's primary goal is to execute a machine learning model on the Raspberry Pi Pico to analyze images captured by the camera, inferring the handwritten digit present in the image. The report covers the hardware and software requirements, wiring connections, image processing, machine learning model training, and the process of exporting the model into a format compatible with the Raspberry Pi Pico. The challenges, future developments, and troubleshooting guidance are also discussed.**

***keywords*— PicoBoard,TFT LCD Display,OV7670 CAM Module**

I. Introduction

A. Problem Statement

The primary objective of this project is to develop a robust handwritten digit classification system for deployment on the Raspberry Pi Pico microcontroller. This entails designing and implementing an efficient machine learning model, striking a balance between accuracy and computational efficiency, with a focus on algorithms like Convolutional Neural Networks or Support Vector Machines. Integration with the Raspberry Pi Pico environment, including the use of CircuitPython, will be a key aspect. A user-friendly interface will be developed to facilitate interaction, allowing users to input handwritten digits, receive predictions, and visualize results.

B. Objective

The primary objective of this project is to implement a self-contained handwritten digit classification system on the Raspberry Pi Pico using machine learning techniques. The goals include developing an efficient machine learning model, optimized for the Raspberry Pi Pico's resource constraints, to accurately classify handwritten digits.Additionally, the project aims to integrate a user-friendly interface, possibly utilizing a TFT LCD display, to visualize and interpret the digit classification results in real-time.By achieving these objectives, the project seeks to demonstrate the feasibility of deploying machine learning applications on edge devices like the Raspberry Pi Pico while providing an educational and accessible platform for understanding the intersection of machine learning and embedded systems.

C. Scope of the Project

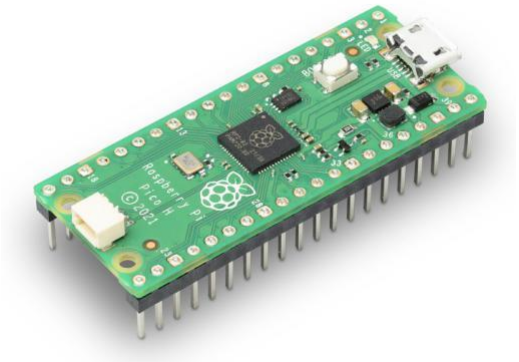
Through rigorous testing and documentation, the project seeks to provide valuable insights for the community interested in implementing machine learning models on microcontrollers. The ultimate goal is to showcase the practical application of embedded machine learning, specifically for handwritten digit recognition on the Raspberry Pi Pico.

.

II. Components used

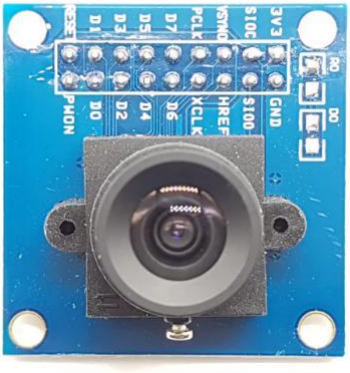
A. Raspberry Pi Pico:

The Raspberry Pi Pico is a microcontroller board based on the RP2040 chip. It's designed for low-cost and high-performance microcontroller applications. In this project, the Raspberry Pi Pico serves as the central processing unit, running the machine learning model and controlling the interactions between other components.



B. OV7670 Camera Module:

The OV7670 is a low-cost, small-sized camera module capable of capturing images. It plays a crucial role in this project by capturing handwritten digits. The images taken by the camera are processed and analyzed by the machine learning model running on the Raspberry Pi Pico.

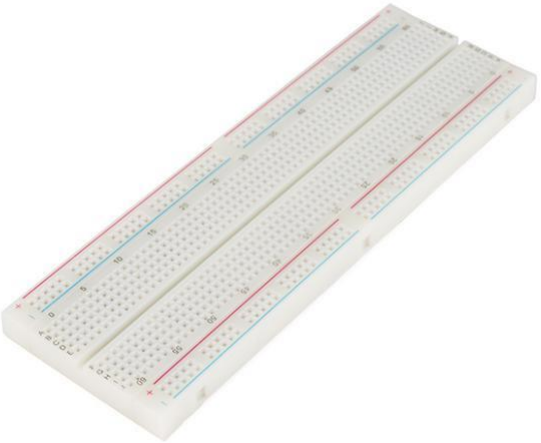


C. 128x160 TFT LCD Display:

The 128x160 TFT LCD display provides a visual output for the project. It allows users to see the captured images, processed results, and the predicted handwritten digit.
The display enhances the user experience by providing real-time feedback.

D. Full-sized Breadboard:

A full-sized breadboard is used for prototyping and creating temporary connections between various components. It simplifes the wiring process and allows for a more organized setup during the development phase.



E. Jumper Wires for Peripheral Connections:

Jumper wires play a crucial role in establishing reliable connections between the Raspberry Pi and various peripherals, including components such as the ST7735 TFT SPI display, USB microphone, and other hardware relevant to language translation applications. These wires are flexible connectors with male or female ends, enabling seamless linking of different elements within a project. Jumper wires are highly versatile, facilitating the creation of custom wiring configurations. Their flexibility allows users to adapt and customize the connections based on the specific requirements of the language translation system. This versatility is particularly advantageous in projects where space constraints or unique

Fig. 5. Jumper Wires

III. START UP PROTOCOL FOR RASPBERRY PI PICO OS

A. About the OS:

The Raspberry Pi Pico OS is designed to optimize the computational capabilities of the Raspberry Pi Pico microcontroller. It leverages a tailored strategy to enhance performance, with a focus on the unique features of the Pico platform. This OS is crafted to provide an efficient and seamless experience for developers and enthusiasts utilizing the Raspberry Pi Pico for their projects.

B. Empowering Users with Microcontroller Computing:

The Raspberry Pi Pico OS empowers users to unlock the full potential of the Raspberry Pi Pico microcontroller. Tailored to the specific architecture of the Pico, this OS delivers an optimized computing experience, ensuring that users can harness the capabilities of the microcontroller to their fullest extent. It offers a balance between innovation and compatibility within the Raspberry Pi ecosystem, showcasing the adaptability and versatility of the Raspberry Pi Pico OS.

C. Booting process:

The Raspberry Pi Pico OS supports flexible booting options, allowing users to initiate the system from various sources such as Flash, UF2, or a custom bootloader.

The startup process involves the following steps:

1. Insert Media: Insert the Flash, UF2, or custom bootloader into the Raspberry Pi Pico.
2. Power Connection: Connect the Raspberry Pi Pico to a power source.
3. Automatic Boot: The Raspberry Pi Pico will automatically boot from the inserted media.
4. Initialization: Once the boot process is complete, the system initializes, and users gain access to the Raspberry Pi Pico OS environment.

This streamlined startup protocol ensures a straightforward and efficient process for initializing the Raspberry Pi Pico OS, enabling users to quickly embark on their microcontroller projects.

IV. CONNECTION SETUP

A. Connection of Pi and ST7735 TFI SPI Display:

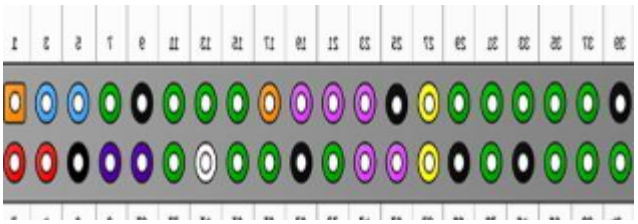


Fig. 7. Pin diagram of Raspberry pi 3

Fig. 8. Pin diagram of ST7735 TFI SPI Display

TABLE I

Raspberry Pi pico	ST7735 TFI SPI	Purpose	Connection
3.3V Power (36st Pin):*	LED (Light Emitting Diode)	Provides power to the LED backlight of the display.	Linked to the 3.3V power source on the Raspberry Pi (1st pin).
SPI Clock(Serial Peripheral Interface) (14th Pin)	SCL (Serial Clock)	Transmit clock signals for Aligned data transfer.	Wired to GPIO (General Purpose Input/Output) on the Raspberry Pi (23rd pin)
SPI MOSI(Master Out Slave In) (15th Pin)	SDA (Serial Data Line)	Transmits data from the Raspberry Pi to the display	Connected to GPIO IO (MOSI) on the Raspberry Pi (19th pin)
GP1018 (21st Pin)	AO(Analog 0)	Selects between command and data for the display.	Linked to GPIO 18 on the Raspberry Pi (12th pin).

Raspberry Pi	ST7735 TFI SPI	Purpose	Connection
GP1023 (22th Pin)	RESET	Resets the display to its default state.	Wired to GPIO 23 on the RaspberryPi (16th pin).
GP1024 (24th Pin)	CS(Chip Select)	Enables or disables the display for data communication.	Connected to GPIO 24 on the RaspberryPi (18th pin).
Ground (38th Pin)	GND (Ground)	Serves as the common ground reference.	Connected to the ground (GND) pin on the RaspberryPi (39th pin).
5V Power (39th Pin):	VCC (Voltage at the Common Collector)	Provides power to the display.	Wired to the 5V power source on the RaspberryPi (2nd pin).

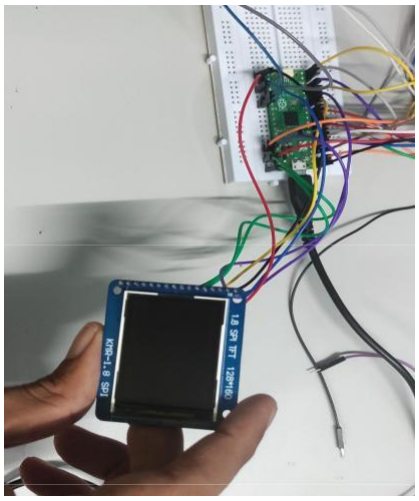


Fig. 9. Connection of Pi and ST7735 TFI SPI Display

B. Connection of Pi and OV7670:

Connecting the OV7670 image sensor module to the Raspberry Pi involves a careful hardware and software integration process. After soldering the pin header onto the OV7670 breakout board, the power pins (VCC and GND) are connected to corresponding GPIO pins on the Raspberry Pi.The Raspberry Pi camera interface is then configured to recognize the OV7670, and testing ensues to ensure proper functionality. Adjustments to script parameters optimize image quality, resolution, and other factors.

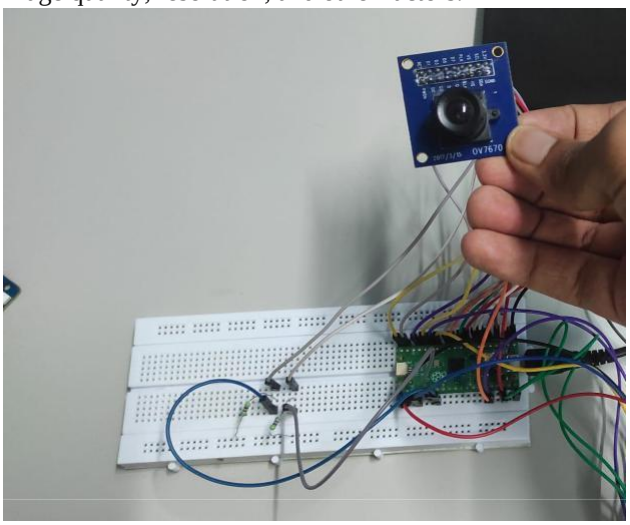


Fig. 10. Connection of Pi and OV7670

V. Methodology

A. User Input:

1. Users Will Write a number and it has to be shown to OV7670 camera
2. Detecting processes is initiated upon detecting user input.

B. Target Detection:

1. Machine Learning Model identifies the input language.
2. The detected Number will be displayed on the TFT LCD.

C. Target Prediction:

1. The number is predicted using SVM Machine Algorithm
2. System processing prepares for Prediction

D. Console Display:

The Predicted Number will be displayed on the console allowing for easy verification..

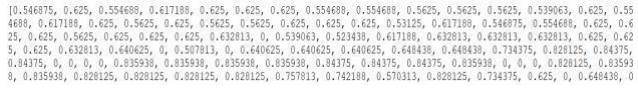


Fig. 12. Console output

output provides comprehensive information.

1. It Provides the image of that particular number that user has shown.
2. It will show the Prediction of that particular number.

The Predicted Model will not always give the expected output as the model is still learning.

E. ST7735 SPI Display:

Simultaneously, the translated output is sent to the ST7735 TFI SPI Display, providing a visual representation of the Predicted Number.

Fig. 13. ST7735 SPI Display output

VI. CONCLUSION

In conclusion, the Handwritten Digit Classification project utilizing the Raspberry Pi Pico and Machine Learning represents a significant exploration into the convergence of microcontroller technology and artificial intelligence. Our primary goal was to create a standalone system capable of analyzing handwritten digits captured by an OV7670 camera module and providing real-time predictions using a machine learning model.

The integration of a compact and affordable microcontroller like the Raspberry Pi Pico, coupled with an OV7670 camera module and a 128x160 TFT LCD display, showcased the feasibility of deploying machine learning on resource-constrained devices. The project's success in running a machine learning model entirely on the Raspberry Pi Pico emphasizes its adaptability and opens doors for similar applications in edge computing.

The meticulous wiring and hardware components, including the TFT LCD and OV7670 camera, played crucial roles in the system's functionality. The post-processing of camera images, training of the machine learning model using the scikit-learn library, and the subsequent export of the model to a Pico-friendly format added layers of complexity, demonstrating the interdisciplinary nature of the project.

While the project provides a functional handwritten digit classification system, there are avenues for future enhancements. Experimentation with larger image sizes, data obtained directly from the camera for training, and exploring different datasets (shapes, signs, patterns) are exciting directions for further development. Additionally, the potential conversion of the project to C/C++ using the Pi Pico SDK and TFLite micro offers opportunities for optimization and broader application.

1. In essence, the Handwritten Digit Classification project serves as a testament to the evolving landscape of embedded systems and machine learning integration. As technology advances, the synergy between microcontrollers and artificial intelligence opens up new possibilities for compact and intelligent edge devices. This project contributes to this narrative, showcasing the Raspberry Pi Pico's capabilities and its potential to drive innovation in the intersection of hardware and machine learning.

VII. FUTURE DEVELOPMENTS

In conclusion, the future holds promising avenues for the Handwritten Digit Classifcation project on the Raspberry Pi Pico. Potential developments include real-time training from camera data, memory optimization for larger image sizes, and exploration of diverse datasets for expanded classifcation capabilities. Transitioning to C/C++ using Pi Pico SDK

nd TensorFlow Lite, enhancing the user interface, and fostering community collaboration are also envisioned. These endeavors aim to elevate the project's functionality, efciency, and collaborative potential. Overall, the project exemplifes an innovative fusion of machine learning and microcontroller technology, inviting further exploration and advancements in the feld. The provided documentation and troubleshooting guidance ensure accessibility for enthusiasts and developers keen on similar pursuits.

[1] Programming the Raspberrypi Pico in CircuitPython by SteveCope

[2] RaspBerrypi Pico DIY Workshop by Sai Yamanoor (Author)

[3] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[4] https://ashishware.com/2022/09/03/pipico_digit_classification/#Trai