Interfacing SPI Camera with Raspberry Pi Pico *For CROWD COUNTING

HASANTHI RAYABARAM ,CB.EN.U4CSE21651

3rd year, Computer Science and Engineering Amrita School of Computing, Amrita Vishwa Vidyapeetham Ettimadai, Coimbatore-641112

MEGHANA KURUSALA, CB.EN.U4CSE21669

3rd year, Computer Science and Engineering Amrita School of Computing, Amrita Vishwa Vidyapeetham Ettimadai, Coimbatore-641112

KUSUMA KAKOLLU, CB.EN.U4CSE21628

3rd year, Computer Science and Engineering Amrita School of Computing, Amrita Vishwa Vidyapeetham Ettimadai, Coimbatore-641112

G SAI SIDDHARTH, CB.EN.U4CSE21619

3rd year, Computer Science and Engineering Amrita School of Computing, Amrita Vishwa Vidyapeetham Ettimadai, Coimbatore-641112

TVS KRISHNA MURTHY, CB.EN.U4CSE21663

3rd year, Computer Science and Engineering Amrita School of Computing, Amrita Vishwa Vidyapeetham Ettimadai, Coimbatore-641112

Abstract—This project explores the integration of the ARD- UCAM Mega 5MP Autofocus SPI Camera, here on reffered as camera, with a Raspberry Pi Pico, focusing on data transfer via UART to an end system. The subsequent phase involves the connection of the Raspberry Pi Pico with vscode, a development platform for machine learning on edge devices. The objective is to train a crowd counting model, enabling real-time inferences for enhanced system capabilities.

I. Introduction

The integration of advanced imaging systems with microcontrollers has emerged as a crucial avenue, offering extensive possibilities for real-world applications. Our project focuses on interfacing the ARDUCAM Mega 5MP Autofocus SPI Camera with the Raspberry Pi Pico microcontroller. This collaboration streamlines data transfer from the Pico to an end system, employing UART communication. The camera's low-power consumption and instant-on operation, coupled with remarkable power efficiency, make it an optimal choice for this project.

Our primary objective extends to elevating the Raspberry Pi Pico's capabilities by integrating it with VSCode, a robust development environment. VSCode provides a versatile platform for machine learning and coding on embedded systems. Within this environment, we aim to explore the integration intricacies between the ARDUCAM and Raspberry Pi Pico, emphasizing UART data transfer methodologies.

Although initially considering Edge Impulse, a machine learning platform for edge devices, we've opted for VSCode. This shift allows us to leverage the flexibility of VSCode's extensions and libraries tailored for machine learning on embedded systems. By training an object detection model within VSCode, we aim to enable real-time inferences, enhancing the system's overall functionality.

This exploration endeavors to contribute significantly to the evolving domain of embedded systems, propelling advancements in imaging technology and edge-based machine learning. Subsequent sections will delve into technical details, covering the ARDUCAM and Raspberry Pi Pico integration, UART data transfer intricacies, and methodologies for training object detection models using VSCode.

Through this multidimensional approach, our goal remains steadfast—to pioneer the fusion of efficient image processing and machine learning, fostering innovation in applications where low-power and instant-on capabilities are imperative.

II. Primary Objectives

- Setup the PICO Hardware, Pico-sdk
- Display captured images on GUI.
- Connect VSCode to Raspberry Pi Pico for image

transfer to ML model.
- Populate VScode with labeled images for crowd counting
- Train an ML model in VSCode.
- Deploy the optimized model on Raspberry Pi Pico for

real-time object detection. • Evaluate performance.

III. Hardware Requirements

- Raspberry Pi Pico H
- ARDUCAM Mega 5MP Autofocus SPI Camera • Breadboard
- Male to Female Jumper Wires
- Micro USB Cable

IV. Software Requirements

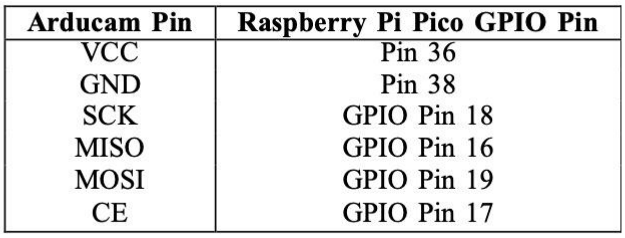
- Visual Studio
- Pico-sdk
- gcc-arm-none-eabi compiler
- CMake
- Code Editor (Visual Studio Code) • Arducam Mega GUI

V. METHODOLOGY FOR CAMERA INTERFACING

A. Hardware Connections

To integrate the Arducam Camera with the Raspberry Pi Pico, establish connections as per the GPIO Pins specified in Table 1.

Arducam Pins



Arducam Pin	Raspberry Pi Pico GPIO Pin
VCC	Pin 36
GND	Pin 38
SCK	GPIO Pin 18
MISO	GPIO Pin 16
MOSI	GPIO Pin 19
CE	GPIO Pin 17

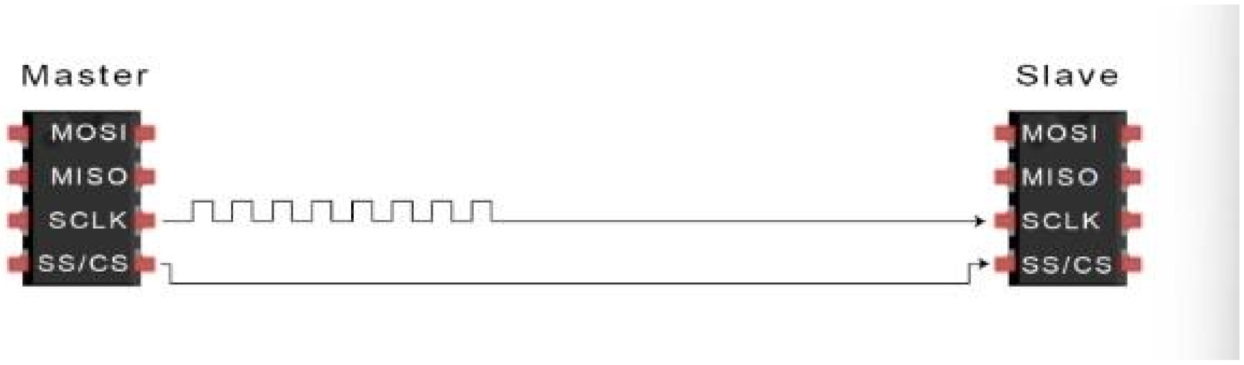


TABLE I: Arducam Camera GPIO Connections

Fig. 3: SPI pin configuration

data one bit at a time to the slave along the MOSI line. The slave reads the bits as they are received. 4.If

Fig. 4: SPI pin configuration

a response is needed, the slave returns data one bit at a time to the master along the MISO line. The master reads the bits as they are received.

Fig. 5: SPI pin configuration

· With I2C and UART, data is sent in packets, limited to a specific number of bits. Devices communicating via SPI are in a master-slave relationship.

· The Raspberry Pi Pico utilizes the UART protocol to establish serial communication with remote device for further analysis or display.

• In UART communication, both sides are set to a pre- configured baud rate that dictates the speed and timing of data transmission.

D. Code

• Command number 7 is sent from end device to pico via UART. Setup initialized.

• Command 8 sent from end device to pico via UART. To Initiate Arducam Setup

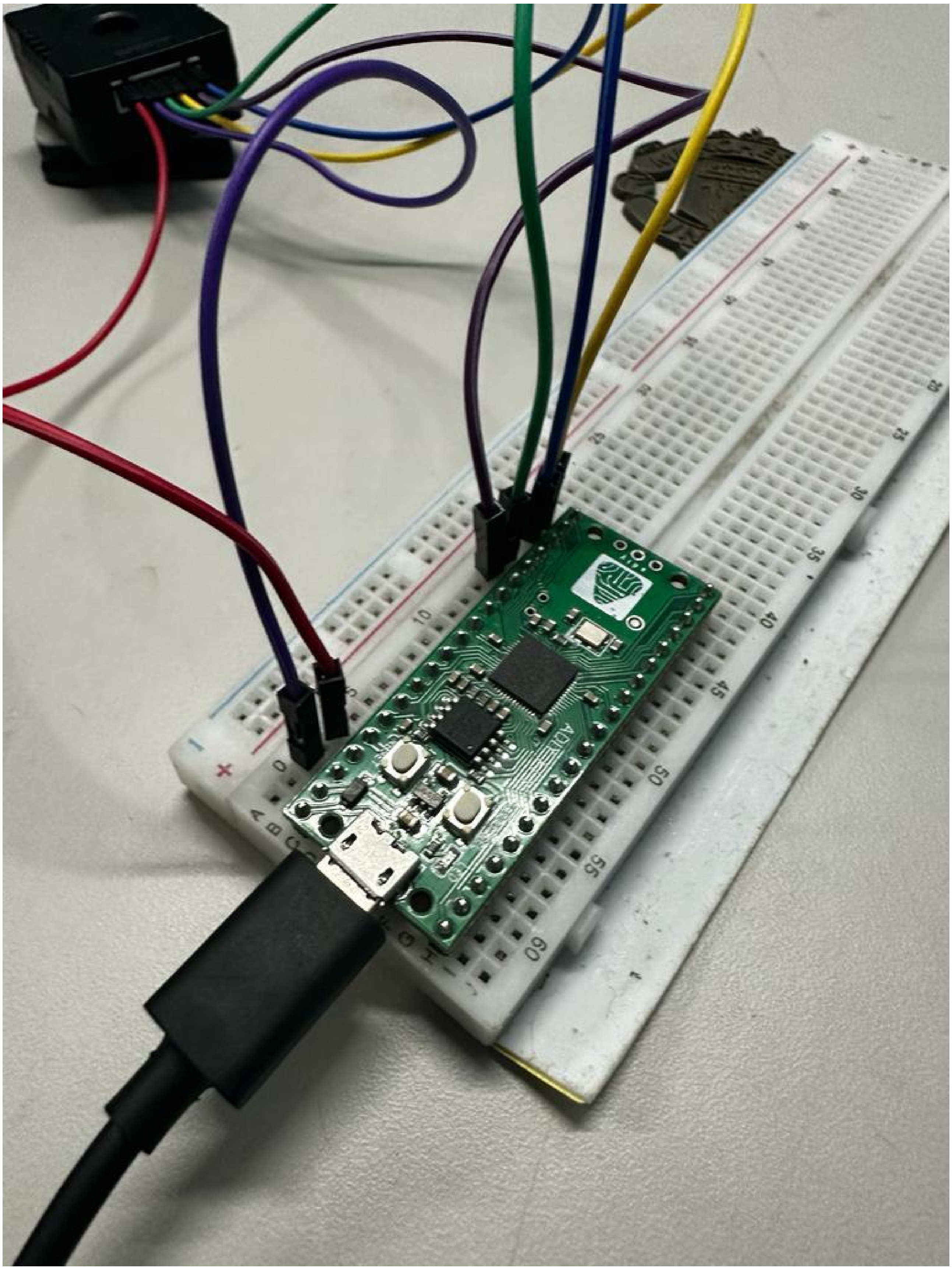
• When a new task is expected, the first value to be sent from the end device to the Pico is 0x55. This marks the begining of the further values to be sent

• It will continue accepting parameters and storing them in commandBuff (buffer).

• Data sent will be stored in commandBuff until 0xAA is passed which marks the end of the data to be sent.

• This buffer is then passed to the uartCommandProcess- ing() function where the request will be processed.

• Other Utility Functions include arducamUartAvailable(), sendDataPack(), arducamFlush().



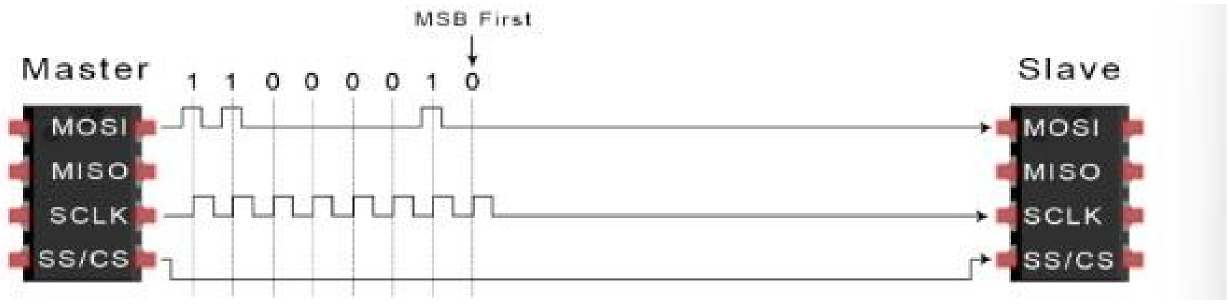


Fig. 1: Arducam Camera Pin Diagram

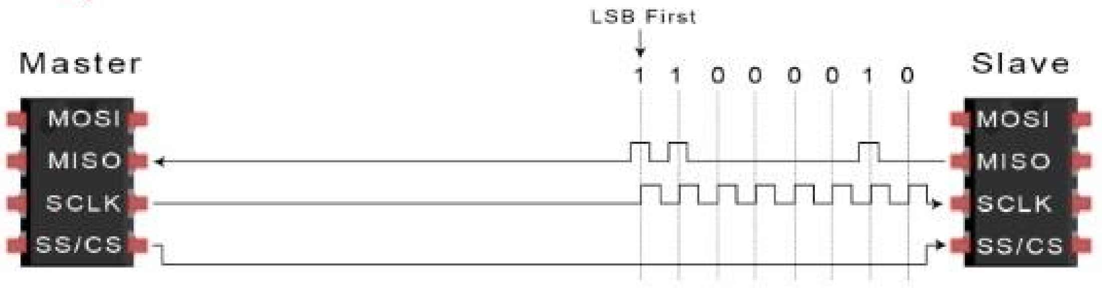
B. Communication Protocol

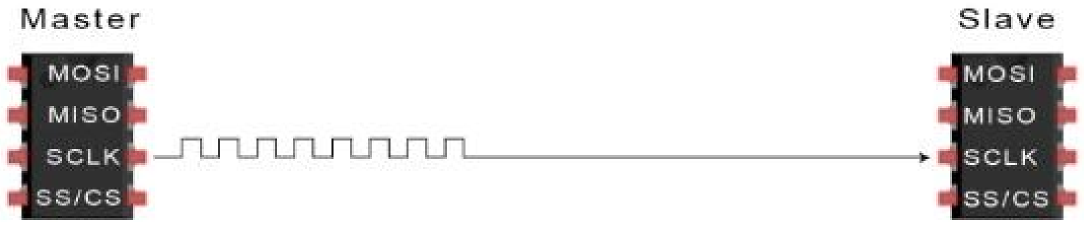
- The Arducam Mega SPI Camera module uses SPI (Serial Peripheral Interface) protocol with the Raspberry Pi Pico microcontroller for all communication.

- SPI is a full-duplex interface, both main and subnode can send data at the same time via the MOSI and MISO lines respectively.
  C. explanation

- Only 4 pins (GPIOs) are required excluding VCC GND are interfaced in SPI module i.e MOSI (Master Output/Slave Input) ,MISO (Master Input/Slave Out- put),SCLK (Clock),SS/CS (Slave Select/Chip Select).

- Steps for data transmission in spi module:
  1) T: he master can choose which slave it wants to talk to by setting the slave’s CS/SS line to a low voltage level. 1.SPI communication is always initiated by the master since the master configures and generates the clock signal. Usually CS is an active low signal; hence, the main must send a logic 0 on this signal to select the subnode.
  Fig. 2: SPI pin configuration
  2.The master switches the SS/CS pin to a low voltage state, which activates the slave. 3.The master sends the





Given below is the communication protocol between host and camera

Algorithm 1 Main Program

Start

0X55

TABLE II: Communication Protocol: Host to Camera

0: 0: 0: 0: 0: 0:

function MAIN
Command 7 - Pico Init Command 8 - ArduCam Init while true do

if MYUART.ARDUCAMUARTAVAILABLE then temp ← MYUART.ARDUCAMUARTREAD

if temp = 0x55 then
while MYUART.ARDUCAMUARTAVAILABLE do

0x02

0x10 0x21 0xFF 0x0F

Start streaming mode

Taking Pictures
Stop streaming mode Ensure proper connections Get basiccamera information

B. Model architecture and training

.

VIII. CONCLUSION

Our project accomplished the seamless integration of the ARDUCAM Mega 5MP Autofocus SPI Camera with the Raspberry Pi Pico, prioritizing efficient data transfer through UART. Departing from the initial plan, we chose to utilize VSCode directly for crowd counting through machine learning.

Key Achievements

Hardware Setup: Successfully linked the ARDUCAM camera with the Raspberry Pi Pico, ensuring optimal functionality.

SPI Communication Protocol Implementation: Implemented SPI communication protocols, enabling effective data exchange between the camera and the microcontroller.

Direct VSCode Machine Learning: Utilizing VSCode's comprehensive machine learning capabilities, we focused on crowd counting without relying on Edge Impulse, showcasing the potential for direct application development.

Acknowledgments

Our gratitude extends to the IEEE community, reviewers, colleagues, and advisors for their unwavering support and invaluable contributions to this project.

Contributions and Significance

By choosing to directly employ VSCode for machine learning and emphasizing crowd counting, our project underscores the convergence of image processing and real-time object detection on edge devices. This approach represents a deliberate effort to innovate applications requiring critical low-power and instant-on capabilities, showcasing the potential of direct VSCode utilization in this domain.

IX. ACKNOWLEDGEMENT

- We extend our thanks to the reviewers for their insightful feedback and constructive criticism, contributing to the refinement and enhancement of our work. Additionally, we acknowledge the mentorship and guidance received from our colleagues and advisors, whose expertise and encouragement significantly enriched the quality of our research. The collaborative environment fostered by IEEE has been instrumental in advancing our understanding of the subject matter and facilitating the sharing of knowledge within the global scientific community.

Command

Parameter Stop

CMD

parameter 0xAA

CMD

0x01

Effect

Set camera resolution

Parameter

0:

bit[6:4]: Camera data for- 0:

mat
- 1: Indicates JPEG reso-
lution
- 2: Indicates RGB565 0: resolution
- 3: Indicates YUV reso-
lution
[3:0]: Set resolution of 0: camera
- 0: 160x120
- 1: 320x240
- 2: 640x480 0:: - 3: 800x600
- 4: 1280x720
- 5: 1280x960
- 6: 1600x1200 0: - 7: 1920x1080
- 8: 2048x1536
- 9: 2592x1944

- 1: 320x240
- 2: 640x480

```
commandBuff[commandLength] ← myUart.arducamUartRead

0:

0: 0:

0: 0::

0: 0:

if commandBuff[commandLength] = 0xAA then break

end if

commandLength ← 0

myCAM.captureThread

end while end function=0

1

commandLength + uartCommandProcessing(commandBuff)

commandLength ← end while

end if
```

E. PseudoCode

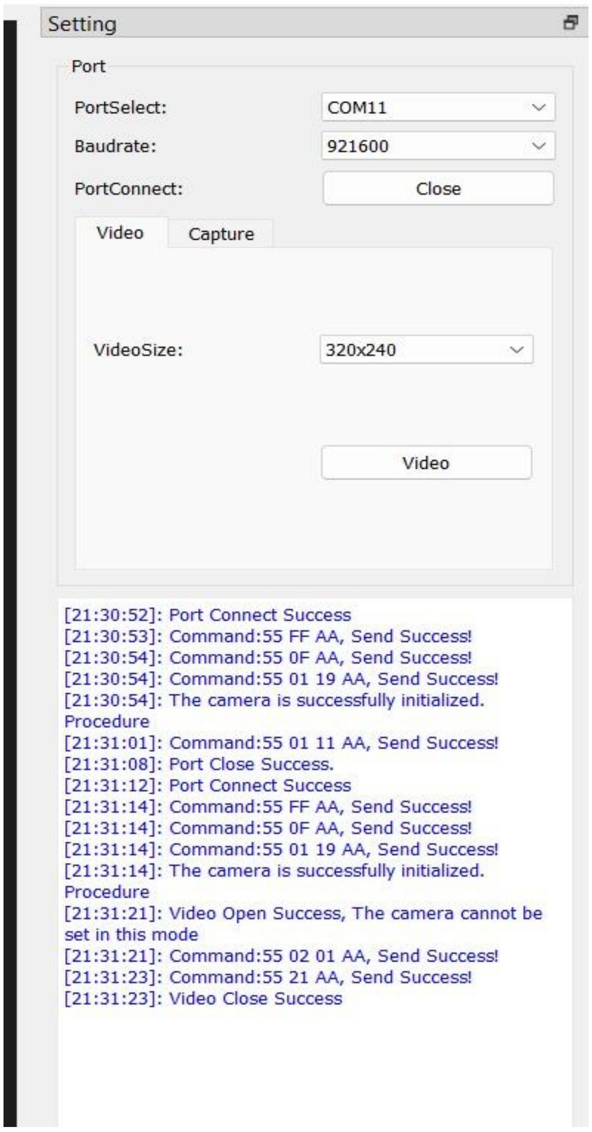
TABLE III: Command Table

VI. ARDUCAM GUI

- The Arducam Mega GUI is a useful tool to interact with the Arducam Mega.

- It allows us to set resolutions, choose input modes, set baud rate and multiple other features.

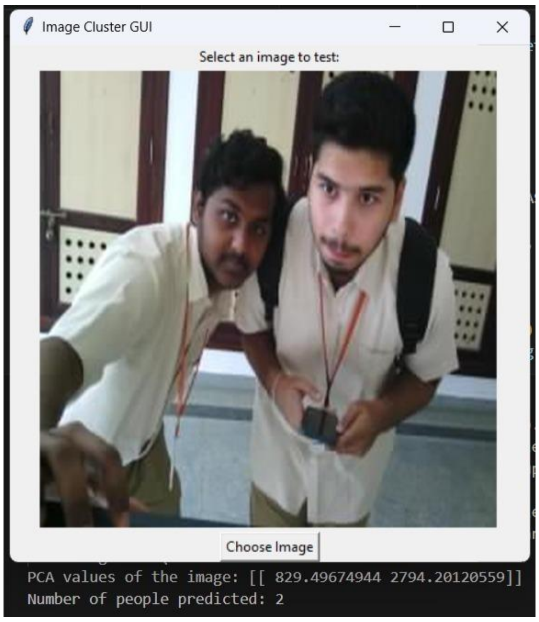
- The corresponding commands from Table 3 are sent for the selected requirements.
  The above diagram shows the stream of commands being sent on each action request.



PortSelect: COM11
Baudrate: 921600
PortConnect: Close
VideoSize: 320x240

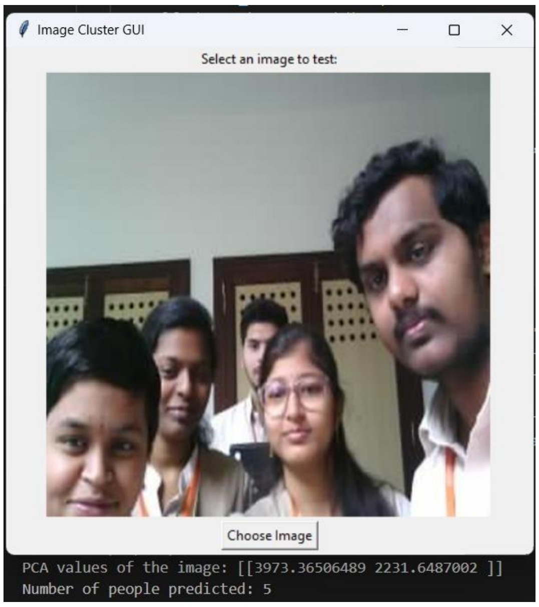
```
[21:30:52]: Port Connect Success
[21:30:53]: Command:55 FF AA, Send Success!
[21:30:54]: Command:55 0F AA, Send Success!
[21:30:54]: Command:55 01 19 AA, Send Success!
[21:30:54]: The camera is successfully initialized.
Procedure
[21:31:01]: Command:55 01 11 AA, Send Success!
[21:31:08]: Port Close Success.
[21:31:12]: Port Connect Success
[21:31:14]: Command:55 FF AA, Send Success!
[21:31:14]: Command:55 0F AA, Send Success!
[21:31:14]: Command:55 01 19 AA, Send Success!
[21:31:14]: The camera is successfully initialized.
Procedure
[21:31:21]: Video Open Success, The camera cannot be
set in this mode
[21:31:21]: Command:55 02 01 AA, Send Success!
[21:31:23]: Command:55 21 AA, Send Success!
[21:31:23]: Video Close Success
```

Fig. 7:Person_count



```
PCA values of the image: [[ 829.49674944 2794.20120559]]
Number of people predicted: 2
```

Fig. 8:Group image



```
PCA values of the image: [[3973.36506489 2231.6487002 ]]
Number of people predicted: 5
```

X. REFERENCES

- 1 Arshad, M. K.Md, U. Hashim, and Chew Ming Choo. 2006. “Characteristics of Serial Peripheral Interfaces (SPI) Timing Parameters for Optical Mouse Sensor.” IEEE International Conference on Semiconductor Elec- tronics, Proceedings, ICSE: 576–82.

- 2 Li, Li Li, Jing Yu He, Yong Peng Zhao, and Jian Hong Yang. 2014. “Design of Microcontroller Standard SPI Interface.” Applied Mechanics and Materials 618(March 2017): 563–68.

- 3 Kasenda, Sonny, Doostenreyk Kantohe, Maureen Langie, and Anthoinete Waroh. 2018. “Light Intensity Control Prototype Design Using Arduino Uno.” Proceedings - 2018 International Conference on Applied Science and Technology, iCAST 2018: 563–66.