*Comparing Performance Metrics Across Raspberry Pis*

1st Joshua Castillo   
Electrical Engineering  
California Polytechnic State UniversityPomona, USA  
castillo1@cpp.edu

4th Ignacio Velasco  
Computer Engineering  
California Polytechnic State UniversityPomona, USA  
idvelasco@cpp.edu2nd Yeachan Okh  
Electrical Engineering  
California Polytechnic State UniversityPomona, USA  
yokh@cpp.edu

3rd Vincent Seymour  
Electrical Engineering  
California Polytechnic State UniversityPomona, USA  
vsseymour@cpp.edu

***Abstract*— This paper presents the design and implementation of a basic image processor capable of performing six different operations: none, remove red, remove green, remove blue, grayscale, and negative. Designed for live video streams, the processor enables real-time image processing with potential applications in forensic investigations, medical imaging, industrial quality control, object detection, artistic design, and image compression. The project demonstrates the system architecture, design methodology, and implementation, while also addressing its contributions to diverse fields. Performance metrics were gathered on both a Raspberry Pi 5 and Raspberry Pi 4 Model B, employing both Python and C++ with Visual Studio Code serving as the main Integrated Development Environment, to analyze execution time and thermal efficiency. The findings demonstrate the system's viability for real-world image processing tasks and its adaptability to varied platforms.**

***Keywords—image processing, live video stream, grayscale, negative, medical imaging, forensic investigations***

# Introduction

Image processing is a critical area in modern computing, with applications spanning across diverse fields such as forensic investigations, medical imaging, industrial quality control, computer vision, and art design. Real-time image processing has gathered significant attention due to its ability to analyze and modify video streams. This paper presents the design and implementation of a basic image processor capable of performing six distinct functions—none, remove red, remove green, remove blue, grayscale, and negative—on live video streams using two distinct Raspberry Pi’s.

To evaluate the performance of the system, a comparative analysis of the Raspberry Pi 5 and Raspberry Pi 4 Model B is conducted, employing C++ and Python as programming languages. The experiments aimed to measure execution times, thermal efficiency, and frames per second (FPS) across the two platforms and languages all in one integrated development environment (IDE). OpenCV was chosen as the foundation for the image processing tasks. The findings of this work aim to inform the selection of hardware and software for similar real-world applications, contributing to advancements in the fields of image processing and single-board computing.

# Design and methodology

The team sought to evaluate the computational capabilities of a Raspberry Pi 5 against its predecessor, a Raspberry Pi 4 Model B. The team designed a series of performance tests to analyze hardware performance and software performance. By implementing languages such as C Plus Plus (C++) and Python, the team can run different tests across both languages. The team seeks to determine if which coding language will perform better and which Pi will perform better, as well as any hidden parameters such as which language is better optimized to run on each hardware.

The team will use Visual Studio Code (VS Code) as its Integrated Development Environment (IDE) for both pieces of hardware and software. Using the same IDE could ensure that both languages and hardware get a fair comparison.

## Raspberry Pi 5 vs Raspberry Pi 4 Model B

Evaluating two different Pi models stems from the team looking to find the best Single-board computer (SBC) available for students to use. The team implements a Raspberry Pi 5 with an ARM Cortex A76 Quad Core Processor with 8 GB of LPDDR4X RAM, and a Raspberry Pi 4 Model B with a Broadcom BCM2711 Quad core Cortex-A72 (ARM v8) 64-bit SoC with 8 GB of DDR4 RAM. Both Pi’s also ran the 64-bit Raspberry Pi Operating System (Pi OS) and both SBC’s used a 128-GB microSD.

## C++ vs Python

Alongside comparing the two Raspberry Pi’s the team intends to evaluate the efficiency of two coding Languages. C++ and Python were elected for comparison as these languages allow for ease of use with OpenCV. The team ensured the use of OpenCV 4.10.0 across both languages and that both languages were the most up to date versions as well.

* 1. *Integrated Development Environment and Laptop Operating System*

The implementation of Visual Studio Code (VS Code) was chosen as the IDE of choice for this project since it could host both languages and ensure the team could start on experimentation as soon as setup was completed.

Originally the team employed a laptop with Windows 11 as its main choice to hold all the codes, yet issues with OpenCV made it so the team shifted instead to a Laptop with Ubuntu 22.04, a Linux-based operating system, as a dedicated coding machine. This setup enabled the team to install any missing dependencies via terminal, instead of navigating websites for similar tools.

# Experimental Setup

The methodology for this experiment included using the laptop running Ubuntu to ssh into each Pi and have them run the code, Camera\_Server.py which is the experiment's camera server on each hardware. Once the camera is operational the experimenter will run the appropriate code per Pi, as there are some logical issues from device to device when attempting to measure temperature or Frames Per Second (FPS). Then the experimentation begins and following this sequence, the Pi being tested will be allowed to turn on and run for 2 minutes, then it begins. A test consisting of one minute and 45 seconds will commence, testing each mode of operation such as None, Remove Red, Remove Green, Remove Blue, Grayscale, and Negative. Fifteen seconds is allotted to each operation before it is switched to the next one, and after the Negative mode a second iteration of None will run for the allotted time before the code is terminated by the user by pressing “Q” on the laptop. This will then print and open a text file (txt) containing all the information gathered below.

Each Pi will follow this procedure strictly with a timer being used to change the modes. The first hardware tested is the Pi5 with C++ being the first language to run through the experimental setup. Immediately upon completion the user will switch to the Python code and run it in the exact same way. Upon completion the data will be collected and the Pi shut-off and given time to cool down before the second leg of the experiment commences. The process will be repeated with Python being executed before C++ this time, essentially capturing which language executes quicker, and with the least amount of thermal output.

These tests are run across both coding languages and both pieces of hardware the exact same way to ensure accurate readings. Upon completion the camera server will be shut down and documentation and evaluation of data gathered can commence.

# Results and discussion

## Observations and Findings

Experiments were conducted on both Raspberry Pi 5 and Raspberry Pi 4 Model B to compare the performance of C++ and Python in executing image processing functions across different modes.

For the **Raspberry Pi 5**, the results of **Experiment 1.1** (Table 1) show that Python consistently achieved significantly faster execution times, averaging ~14.8 ms in the "None" mode, compared to C++'s ~63.9 ms. Similarly, in **Experiment 1.2** (Table 2), Python maintained its speed advantage, with execution times averaging ~13.2 ms versus C++'s ~64.1 ms. Despite its slower execution, C++ demonstrated better thermal efficiency, with lower CPU temperatures in both experiments.

Table 1: Experiment 1.1: C++ Runs First Then Python (PI 5)

| Metrics | FPS | Execution Time | Pi Temp | FPS | Execution Time | Pi Temp |
| --- | --- | --- | --- | --- | --- | --- |
| Mode: None | 15 | 63.250 ms | 44.4 ℃ | 14 | 15.100 ms | 47.4 ℃ |
| 15 | 59.613 ms | 43.3 ℃ | 14 | 12.800 ms | 48.8 ℃ |
| 14 | 68.332 ms | 43.9 ℃ | 15 | 13.600 ms | 48.8 ℃ |
| 14 | 66.844 ms | 44.4 ℃ | 14 | 15.600 ms | 47.2 ℃ |
| 15 | 61.572 ms | 44.4 ℃ | 14 | 14.900 ms | 48.8 ℃ |
| Mode: Remove Red | 13 | 62.992 ms | 46.1 ℃ | 14 | 14.500 ms | 48.3 ℃ |
| 13 | 67.154 ms | 47.2 ℃ | 14 | 14.400 ms | 47.7 ℃ |
| 16 | 57.531 ms | 47.2 ℃ | 13 | 15.400 ms | 49.4 ℃ |
| 13 | 65.264 ms | 47.2 ℃ | 14 | 6.100 ms | 47.7 ℃ |
| 15 | 66.791 ms | 46.1 ℃ | 15 | 13.200 ms | 48.3 ℃ |
| Mode: Remove Green | 16 | 68.728 ms | 45.0 ℃ | 14 | 14.300 ms | 48.8 ℃ |
| 15 | 68.077 ms | 43.3 ℃ | 15 | 14.300 ms | 47.7 ℃ |
| 15 | 58.692 ms | 43.3 ℃ | 15 | 17.700 ms | 48.8 ℃ |
| 16 | 68.539 ms | 44.4 ℃ | 14 | 13.900 ms | 49.9 ℃ |
| 14 | 59.977 ms | 43.9 ℃ | 14 | 14.200 ms | 49.4 ℃ |
| Mode: Remove Blue | 15 | 55.787 ms | 45.0 ℃ | 16 | 19.300 ms | 47.7 ℃ |
| 14 | 59.241 ms | 45.5 ℃ | 14 | 13.500 ms | 49.4 ℃ |
| 14 | 62.529 ms | 44.4 ℃ | 14 | 13.300 ms | 49.4 ℃ |
| 16 | 56.671 ms | 45.5 ℃ | 14 | 15.400 ms | 49.4 ℃ |
| 14 | 64.786 ms | 45.5 ℃ | 15 | 13.800 ms | 50.5 ℃ |
| Mode: Grayscale | 14 | 59.987 ms | 44.4 ℃ | 14 | 13.300 ms | 49.4 ℃ |
| 15 | 65.219 ms | 44.4 ℃ | 15 | 13.200 ms | 49.4 ℃ |
| 14 | 66.331 ms | 45.0 ℃ | 14 | 6.800 ms | 49.9 ℃ |
| 14 | 59.004 ms | 45.0 ℃ | 15 | 14.000 ms | 48.3 ℃ |
| 16 | 69.288 ms | 44.4 ℃ | 15 | 13.300 ms | 49.9 ℃ |
| Mode: Negative | 14 | 65.027 ms | 46.1 ℃ | 14 | 14.400 ms | 48.3 ℃ |
| 14 | 62.622 ms | 46.1 ℃ | 14 | 13.800 ms | 49.4 ℃ |
| 15 | 68.406 ms | 46.1 ℃ | 15 | 21.600 ms | 49.4 ℃ |
| 14 | 64.924 ms | 46.1 ℃ | 14 | 13.300 ms | 48.8 ℃ |
| 15 | 56.162 ms | 45.0 ℃ | 14 | 13.600 ms | 48.8 ℃ |
| Mode: None | 14 | 65.052 ms | 46.1 ℃ | 14 | 11.100 ms | 49.4 ℃ |
| 14 | 63.835 ms | 46.6 ℃ | 14 | 13.200 ms | 49.4 ℃ |
| 16 | 61.932 ms | 45.5 ℃ | 15 | 13.100 ms | 49.9 ℃ |
| 14 | 65.598 ms | 46.6 ℃ | 14 | 15.700 ms | 49.4 ℃ |
| 14 | 62.069 ms | 46.6 ℃ | 14 | 15.200 ms | 49.9 ℃ |

Table 2: Experiment 1.2: Python Runs First Then C++ (PI 5)

| Metrics | FPS | Execution Time | Pi Temp | FPS | Execution Time | Pi Temp |
| --- | --- | --- | --- | --- | --- | --- |
| Mode: None | 14 | 66.227 ms | 33.4 ℃ | 14 | 13.000 ms | 30.1 ℃ |
| 15 | 61.695 ms | 34.0 ℃ | 14 | 15.300 ms | 30.7 ℃ |
| 14 | 64.515 ms | 34.0 ℃ | 14 | 11.400 ms | 29.0 ℃ |
| 15 | 62.004 ms | 34.0 ℃ | 14 | 5.700 ms | 30.7 ℃ |
| 14 | 66.386 ms | 34.5 ℃ | 15 | 12.800 ms | 29.0 ℃ |
| Mode: Remove Red | 15 | 61.670 ms | 35.1 ℃ | 15 | 25.600 ms | 30.1 ℃ |
| 15 | 63.163 ms | 34.5 ℃ | 14 | 13.900 ms | 30.7 ℃ |
| 15 | 61.583 ms | 34.5 ℃ | 13 | 13.200 ms | 30.7 ℃ |
| 14 | 66.657 ms | 34.5 ℃ | 15 | 13.900 ms | 30.7 ℃ |
| 15 | 63.640 ms | 36.2 ℃ | 13 | 14.000 ms | 29.6 ℃ |
| Mode: Remove Green | 14 | 67.148 ms | 35.6 ℃ | 14 | 15.100 ms | 31.2 ℃ |
| 16 | 56.257 ms | 35.6 ℃ | 15 | 15.200 ms | 30.7 ℃ |
| 15 | 60.912 ms | 36.2 ℃ | 14 | 14.100 ms | 30.1 ℃ |
| 14 | 62.432 ms | 34.5 ℃ | 14 | 13.800 ms | 30.1 ℃ |
| 14 | 67.457 ms | 35.6 ℃ | 18 | 13.000 ms | 31.2 ℃ |
| Mode: Remove Blue | 15 | 60.747 ms | 36.7 ℃ | 15 | 14.400 ms | 31.8 ℃ |
| 13 | 70.905 ms | 35.6 ℃ | 14 | 13.400 ms | 31.8 ℃ |
| 16 | 59.808 ms | 35.6 ℃ | 15 | 13.200 ms | 32.3 ℃ |
| 15 | 62.454 ms | 35.6 ℃ | 14 | 9.200 ms | 31.2 ℃ |
| 14 | 66.526 ms | 36.7 ℃ | 14 | 14.600 ms | 31.2 ℃ |
| Mode: Grayscale | 15 | 61.458 ms | 37.8 ℃ | 15 | 13.200 ms | 32.9 ℃ |
| 15 | 61.621 ms | 37.3 ℃ | 13 | 12.100 ms | 32.9 ℃ |
| 13 | 70.244 ms | 36.2 ℃ | 17 | 12.700 ms | 33.4 ℃ |
| 15 | 57.418 ms | 36.2 ℃ | 15 | 13.700 ms | 32.9 ℃ |
| 14 | 69.173 ms | 37.3 ℃ | 14 | 14.500 ms | 31.2 ℃ |
| Mode: Negative | 11 | 87.877 ms | 37.3 ℃ | 14 | 15.900 ms | 32.3 ℃ |
| 15 | 63.665 ms | 36.7 ℃ | 14 | 12.900 ms | 33.4 ℃ |
| 15 | 64.054 ms | 36.7 ℃ | 14 | 13.600 ms | 32.3 ℃ |
| 14 | 67.038 ms | 37.8 ℃ | 16 | 13.400 ms | 33.4 ℃ |
| 15 | 61.067 ms | 37.8 ℃ | 14 | 13.700 ms | 32.9 ℃ |
| Mode: None | 14 | 66.785 ms | 37.8 ℃ | 14 | 13.200 ms | 34.0 ℃ |
| 15 | 62.258 ms | 37.8 ℃ | 15 | 6.900 ms | 34.0 ℃ |
| 14 | 66.351 ms | 37.8 ℃ | 14 | 13.300 ms | 34.0 ℃ |
| 14 | 67.289 ms | 37.8 ℃ | 14 | 15.000 ms | 32.9 ℃ |
| 16 | 56.734 ms | 37.3 ℃ | 15 | 14.700 ms | 34.0 ℃ |

In the **Raspberry Pi 4 Model B**, the results of **Experiment 2.1** (Table 3), where C++ runs first, show that Python consistently outperformed C++ in terms of execution time. For example, in the "None" mode, Python's execution time was 1.131 ms compared to C++'s 46.222 ms. In contrast, when Python ran first in **Experiment 2.2** (Table 4), Python's execution time averaged 0.493 ms, significantly outperforming C++'s execution time of 62.673 ms. These experiments also highlighted the thermal behavior, with the Pi 4 temperatures remaining between 38.9°C and 45.8°C across various modes, although C++ still showed higher temperatures due to its longer execution times.

Table 3: Experiment 2.1: C++ Runs first then Python (Pi 4 Model B)

| Metrics | FPS | Execution Time | Pi Temp | FPS | Execution Time | Pi Temp |
| --- | --- | --- | --- | --- | --- | --- |
| Mode: None | 19 | 46.222 ms | 43.3 ℃ | 14 | 11.131 ms | 39.4 ℃ |
| 14 | 64.438 ms | 42.8 ℃ | 15 | 10.836 ms | 40.9 ℃ |
| 13 | 72.643 ms | 43.8 ℃ | 15 | 10.454 ms | 40.9 ℃ |
| 16 | 56.018 ms | 43.3 ℃ | 15 | 10.422 ms | 40.9 ℃ |
| 15 | 61.590 ms | 43.8 ℃ | 15 | 10.737 ms | 40.9 ℃ |
| Mode: Remove Red | 15 | 63.311 ms | 44.8 ℃ | 13 | 11.021 ms | 41.4 ℃ |
| 15 | 62.753 ms | 44.3 ℃ | 15 | 11.268 ms | 41.4 ℃ |
| 15 | 60.031 ms | 44.3 ℃ | 13 | 11.156 ms | 41.9 ℃ |
| 14 | 60.868 ms | 43.3 ℃ | 16 | 11.048 ms | 42.4 ℃ |
| 15 | 59.891 ms | 43.3 ℃ | 15 | 11.451 ms | 41.9 ℃ |
| Mode: Remove Green | 15 | 57.959 ms | 44.3 ℃ | 13 | 10.679 ms | 42.4 ℃ |
| 14 | 64.932 ms | 43.3 ℃ | 17 | 10.479 ms | 42.8 ℃ |
| 17 | 55.887 ms | 43.3 ℃ | 33 | 11.182 ms | 42.4 ℃ |
| 4 | 198.837 ms | 43.3 ℃ | 4 | 11.172 ms | 42.4 ℃ |
| 15 | 57.389 ms | 42.8 ℃ | 14 | 11.173 ms | 42.4 ℃ |
| Mode: Remove Blue | 15 | 59.169 ms | 44.3 ℃ | 14 | 11.053 ms | 42.8 ℃ |
| 15 | 63.114 ms | 44.3 ℃ | 14 | 11.276 ms | 43.8 ℃ |
| 15 | 63.654 ms | 44.3 ℃ | 14 | 11.106 ms | 41.9 ℃ |
| 14 | 66.713 ms | 43.8 ℃ | 13 | 11.377 ms | 42.8 ℃ |
| 15 | 60.762 ms | 43.8 ℃ | 14 | 11.453 ms | 42.8 ℃ |
| Mode: Grayscale | 14 | 65.380 ms | 44.8 ℃ | 15 | 11.473 ms | 41.9 ℃ |
| 14 | 66.555 ms | 44.8 ℃ | 14 | 11.076 ms | 43.8 ℃ |
| 14 | 67.932 ms | 44.8 ℃ | 12 | 10.366 ms | 43.3 ℃ |
| 14 | 65.493 ms | 43.8 ℃ | 16 | 11.088 ms | 42.8 ℃ |
| 13 | 70.623 ms | 44.8 ℃ | 14 | 11.631 ms | 42.8 ℃ |
| Mode: Negative | 15 | 60.033 ms | 43.8 ℃ | 13 | 11.076 ms | 42.4 ℃ |
| 16 | 57.888 ms | 44.8 ℃ | 15 | 11.061 ms | 42.8 ℃ |
| 18 | 49.101 ms | 44.8 ℃ | 13 | 10.398 ms | 44.3 ℃ |
| 15 | 62.391 ms | 45.2 ℃ | 13 | 10.944 ms | 43.8 ℃ |
| 14 | 64.824 ms | 44.8 ℃ | 13 | 11.060 ms | 44.3 ℃ |
| Mode: None | 16 | 57.561 ms | 45.2 ℃ | 17 | 10.974 ms | 42.8 ℃ |
| 13 | 66.012 ms | 44.3 ℃ | 16 | 10.819 ms | 45.3 ℃ |
| 15 | 61.191 ms | 43.8 ℃ | 3 | 10.869 ms | 43.8 ℃ |
| 6 | 159.155 ms | 45.2 ℃ | 14 | 10.911 ms | 43.8 ℃ |
| 15 | 63.980 ms | 45.2 ℃ | 15 | 10.816 ms | 43.8 ℃ |

Table 4: Experiment 2.2: Python Runs First then C++ (Pi 4 Model B)

| Metrics | FPS | Execution Time | Pi Temp | FPS | Execution Time | Pi Temp |
| --- | --- | --- | --- | --- | --- | --- |
| Mode: None | 14 | 62.673 ms | 38.9 ℃ | 14 | 10.493 ms | 44.3 ℃ |
| 15 | 61.638 ms | 39.9 ℃ | 12 | 10.444 ms | 43.3 ℃ |
| 13 | 69.669 ms | 38.9 ℃ | 16 | 10.847 ms | 43.8 ℃ |
| 15 | 62.916 ms | 39.9 ℃ | 15 | 10.390 ms | 43.8 ℃ |
| 16 | 55.137 ms | 39.9 ℃ | 13 | 10.873 ms | 44.3 ℃ |
| Mode: Remove Red | 14 | 67.193 ms | 39.9 ℃ | 16 | 11.197 ms | 45.3 ℃ |
| 13 | 69.761 ms | 39.9 ℃ | 14 | 11.112 ms | 44.3 ℃ |
| 15 | 64.074 ms | 40.9 ℃ | 13 | 11.338 ms | 45.3 ℃ |
| 15 | 58.473 ms | 38.9 ℃ | 15 | 11.222 ms | 44.3 ℃ |
| 16 | 56.862 ms | 41.3 ℃ | 14 | 11.224 ms | 43.8 ℃ |
| Mode: Remove Green | 14 | 67.647 ms | 41.3 ℃ | 14 | 11.288 ms | 45.8 ℃ |
| 15 | 59.589 ms | 41.3 ℃ | 13 | 11.269 ms | 44.8 ℃ |
| 14 | 64.809 ms | 40.9 ℃ | 14 | 11.046 ms | 46.7 ℃ |
| 4 | 209.064 ms | 40.9 ℃ | 9 | 10.513 ms | 46.3 ℃ |
| 1 | 581.225 ms | 40.9 ℃ | 2 | 11.067 ms | 45.3 ℃ |
| Mode: Remove Blue | 13 | 73.171 ms | 40.4 ℃ | 16 | 10.775 ms | 44.3 ℃ |
| 19 | 45.823 ms | 41.3 ℃ | 14 | 11.212 ms | 45.3 ℃ |
| 13 | 67.079 ms | 41.3 ℃ | 15 | 11.230 ms | 45.8 ℃ |
| 15 | 62.287 ms | 41.3 ℃ | 13 | 11.207 ms | 44.8 ℃ |
| 16 | 57.726 ms | 41.8 ℃ | 15 | 11.442 ms | 45.3 ℃ |
| Mode: Grayscale | 18 | 52.671 ms | 41.8 ℃ | 13 | 10.952 ms | 45.8 ℃ |
| 14 | 64.710 ms | 41.8 ℃ | 14 | 10.917 ms | 46.3 ℃ |
| 14 | 67.627 ms | 42.3 ℃ | 13 | 10.984 ms | 44.8 ℃ |
| 15 | 61.289 ms | 41.8 ℃ | 13 | 11.192 ms | 45.8 ℃ |
| 14 | 67.020 ms | 42.3 ℃ | 15 | 11.187 ms | 45.8 ℃ |
| Mode: Negative | 17 | 52.542 ms | 42.3 ℃ | 15 | 11.536 ms | 44.8 ℃ |
| 16 | 55.514 ms | 41.3 ℃ | 14 | 10.298 ms | 45.3 ℃ |
| 14 | 64.210 ms | 42.3 ℃ | 15 | 10.387 ms | 47.2 ℃ |
| 16 | 54.899 ms | 42.3 ℃ | 14 | 11.024 ms | 45.8 ℃ |
| 13 | 71.113 ms | 42.3 ℃ | 13 | 11.077 ms | 45.3 ℃ |
| Mode: None | 16 | 57.202 ms | 43.3 ℃ | 14 | 11.271 ms | 46.3 ℃ |
| 17 | 56.580 ms | 42.3 ℃ | 14 | 10.978 ms | 45.3 ℃ |
| 15 | 61.055 ms | 42.3 ℃ | 15 | 10.899 ms | 44.8 ℃ |
| 5 | 190.725 ms | 42.8 ℃ | 16 | 10.924 ms | 45.8 ℃ |
| 5 | 180.460 ms | 42.8 ℃ | 4 | 10.861 ms | 45.3 ℃ |

## Moreover, both **Experiment 2.1** and **Experiment 2.2** (Tables 3 and 4) demonstrate how varying the execution order affects performance, with Python running first resulting in the best overall performance for both execution time and frame rate. Computationally intensive modes like "Remove Red" and "Negative" continued to show higher execution times and CPU temperatures for both languages, as detailed in the tables.

## Comparative Analysis

The results from StereoPi.com indicate that, “at a resolution of 1280x480, C++ achieves an average frame rate of 40 FPS, while Python achieves 15 FPS. At a resolution of 640x240, the reported frame rates are 90 FPS for C++ and 20 FPS for Python” [1]. For our demo, the resolution is 640x480, and comparable metrics for this configuration were not directly available. To validate the performance of the Frames Per Second (FPS) measurements obtained in this study, the results were compared with those reported in [1]. Although the resolutions differ, the trends observed in the StereoPi data provide insights into the group’s results, C++’s average FPS was higher than those of the Python code's results, which seems to be inline with the results obtained in [1].

The results from Programming-Language-Benchmarks [2] compare Python and C++ across various computational tasks. On average, C++ consistently outperforms Python, however, the results of our project reveal that Python achieved faster processing times compared to C++ when using OpenCV. This discrepancy stems from how Visual Studio Code interacts with OpenCV for Python and C++, as shown by the disparaging results between execution times. Thus highlighting the importance of the development environment and library optimizations in performance-critical applications.

# Conclusion

This project demonstrates the performance of using Raspberry Pi computers for real-time image processing applications. The results revealed notable differences in performance between the Raspberry Pi 5 and Raspberry Pi 4 Model B, as well as between C++ and Python. Python consistently outperformed C++ in execution speed across all image processing modes, primarily due to how Visual Studio Code works with OpenCV on C++, highlighting its optimization for rapid development and execution in OpenCV-based tasks. Conversely, C++ exhibited superior thermal efficiency, making it more suitable for scenarios where hardware temperature is a critical concern.

The comparative analysis underscores the strengths and limitations of each platform and programming language, providing insights for developers and researchers. These findings have practical implications for applications in forensic investigations, medical imaging, and industrial quality control, among others, where selecting the right combination of hardware and software can make a difference. Future work could expand this research by exploring additional image processing modes and integrating machine learning algorithms to enhance functionality.

##### References

1. StereoPi, "OpenCV: Comparing speed of C++ and Python code on Raspberry Pi for stereo vision," StereoPi Blog, Feb. 2020. [Online]. Available: https://stereopi.com/blog/opencv-comparing-speed-c-and-python-code-raspberry-pi-stereo-vision. [Accessed: Dec. 7, 2024].J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
2. Programming Language Benchmarks, "Comparing Python vs C++ performance," [Online]. Available: https://programming-language-benchmarks.vercel.app/cpp-vs-python. [Accessed: Dec. 7, 2024].