Handwritten Digit Classification using Raspberry Pi Pico and Machine Learning

**Introduction:**This report elaborates on a project that combines hardware expertise and machine learning to achieve handwritten digit classification. Utilizing the Raspberry Pi Pico as the central unit, the project incorporates an OV7670 camera module for image capture and a TFT LCD display for real-time output. The system aims to accurately classify digits from 0 to 9, processed and predicted through a machine learning model implemented in a constrained hardware environment.

**Hardware and Software Specifications:**- Raspberry Pi Pico: A microcontroller board used as the project's central processing unit, handling image data processing and machine learning model execution.  
- OV7670 Camera Module: A cost-effective image sensor for capturing real-time images of handwritten digits. It communicates with the Raspberry Pi Pico and provides raw image data.  
- 128x160 TFT LCD Display: A small display that shows the camera's field of view and the digit classification results, assisting in aligning the digits for the camera.  
- Software Tools: The project leverages Python for scripting, CircuitPython for microcontroller programming, and scikit-learn for machine learning model training. Additional tools include m2cgen for converting the model into a format compatible with CircuitPython and python-minimizer for optimizing the model size to fit the limited memory of Raspberry Pi Pico.

**System Design and Implementation:**- Image Capture and Processing: Utilizing the OV7670's RGB565\_SWAPPED format, images are first captured, then converted to grayscale, and resized to 12x12 pixels, matching the input requirement of the machine learning model.  
- Machine Learning Model: A Linear Support Vector Machine (SVM) model is employed, trained on the widely-used MNIST handwritten digit dataset. This model choice balances computational complexity and performance for the limited processing power of the Raspberry Pi Pico.  
- Model Optimization: Significant effort went into optimizing the model to fit within the Raspberry Pi Pico's memory constraints, ensuring both efficiency and accuracy.  
- Model Conversion and Minimization: The trained SVM model was converted to CircuitPython-compatible code using m2cgen. To further accommodate the Raspberry Pi Pico's limited memory, the code was minimized using python-minimizer, reducing its footprint significantly.

**Setup and Execution:**- Assembly: The project required meticulous wiring and connections between the Raspberry Pi Pico, the OV7670 camera module, and the TFT LCD. Detailed instructions are provided for replicating the setup.  
- Execution: The final script, running entirely on the Raspberry Pi Pico, handles image capturing, processing, digit classification, and displaying the results on the LCD. This standalone operation showcases the Pico's capability to run complex tasks.

**Challenges and Troubleshooting:**- Calibration: Precise alignment of the handwritten digits within the camera's frame was crucial. Proper lighting conditions and the size of the digits were key factors in achieving accurate classification results.  
- Debugging: The project includes detailed debugging features, such as visualizing the processed image matrix, to assist in fine-tuning the model's predictions and handling any operational issues effectively.

**Future Enhancements:**- Data Acquisition: Future iterations of the project may include training the model with data captured directly from the OV7670 camera, enhancing the model's accuracy and robustness.  
- Advanced Applications: There is potential to expand this project to recognize other patterns, shapes, or symbols, showcasing the versatility of the Raspberry Pi Pico in handling various machine learning tasks.

**Conclusion:**The project demonstrates the capability of a microcontroller-based system to perform complex tasks like image processing and machine learning classification. It serves as a foundation for future explorations in embedded machine learning applications, highlighting the potential of low-cost, low-power hardware in implementing sophisticated computational tasks.