

CS437 - Internet of Things:

USPS Mail Casing Automation Proposal

Name: David Orona

NetIDs (include NetID of all group members): orona2

Late days used: 0

Motivation

The USPS mail casing process, while functional, can be significantly optimized to improve delivery speed, accuracy, and ease. Postal workers handle three types of mail during their daily start: DPS (Delivery Point Sequence), raw mail, and flats. DPS is already sorted according to routing sequence, requiring minimal handling besides separation by address. Raw mail and flats, on the other hand, remain unsorted and often include exceptional pieces such as holds, forwards, and/or missed deliveries from the previous day.

After sorting all three types of mail, combining them into a single stack and separating by address remains an even bigger hurdle. This procedure does not even account for box fillers—mail items with no address that require deliverance to each residence. These issues collectively lead to increased time in the office, delayed deliveries, and overall higher risks of misrouted or missent mail.

This proposal introduces an automated scanning and sorting system designed to address these inefficiencies. In essence, the primary purpose is to sort a route's mail into single-address bundles that can be quickly grabbed and delivered, thereby streamlining the entire delivery process.

Project Scope

The goal is to design and implement an automated mail scanning and virtual sorting system capable of processing DPS, raw mail, and flats efficiently. The system will:

- Scan mail items using an integrated scanner.
- Extract address text using advanced text recognition APIs (e.g., Google Cloud Vision API, OpenAI's GPT models, etc.)

- Compare extracted data to a USPS relational dataset for address verification and sequencing.
- *Virtually* sort mail items into single-address bundles for rapid delivery.

Stretch Goal

A hardware-based, physical implementation that directs mail items into corresponding bins using servo controllers.

Timeline

Week 1-2: Research and Planning

- Finalize component list: scanner, microcontroller, database interface.

Week 3-4: Hardware Assembly

- Assemble gravity-fed mail feeder.
- Install scanning module and integrate with microcontroller.

Week 5-6: Software Development

- Develop software for text extraction and interpretation using Cloud Vision API or equivalent.
- Implement database comparison logic for address verification.
- Create virtual sorting interface to display sorted output in single-address bundles.

Week 7-8: Virtual Sorting Optimization

- Refine virtual sorting algorithm based on testing results.
- Implement logic to recognize and allocate special cases (ex. box fillers) appropriately.

Week 9: Testing and Evaluation

- Conduct performance tests with varied mail samples.
- Optimize scanning speed, address recognition accuracy, and bundling efficiency.

Week 10: Final Presentation and Report

- Prepare demonstration video showcasing system functionality.
- Compile final report detailing technical approach, implementation, and results.

Contribution

David Orona (Sole Contributor):

- Responsible for all aspects of the project, including:
 - Software architecture and API integration
 - Hardware design and assembly
 - Microcontroller programming
 - Performance testing

Technical Approach

The system will consist of a mail feeder, which guides items into a high-speed scanner. The scanner triggers optical character recognition (OCR) via APIs such as Google Cloud Vision. Extracted address data is cross-referenced with a mock USPS database to determine order. Mail items will be virtually sorted into single-address bundles based on the verified sequence, including box fillers will be identified and grouped with each corresponding address bundle. If the stretch goal is achieved, servo motors will direct each mail item into corresponding bins based on this information.

Implementation Details

Hardware:

- Gravity-fed mail feeder.
- High-speed scanner (capable of 30+ pages per minute?).
- Raspberry Pi or Arduino for microcontroller tasks.

Software:

- Python for API integration and data processing.
- Google Cloud Vision API for OCR.
- Relational database querying for USPS customer data.
- Virtual sorting visualization for output display.
- Bundling algorithm for single-address mail grouping.

Results

Upon completion, the prototype will:

- Accurately identify and virtually sort 95%+ of mail items into single-address bundles.
- Reduce in-office sorting time by 40%.
- Identify misrouted mail and forward-address issues in real time.
- Assign one box fillers for each corresponding address bundle.

Performance testing will measure scanning speed, accuracy, and reliability. Likewise, lessons learned will include hardware limitations, software optimization challenges, and potential scalability. If the stretch goal is reached, physical sorting capabilities will also be demonstrated.

Conclusion

This automated mail scanning and single-address bundling system offers a significant improvement over manual processes. By leveraging advanced OCR, robust database integration, and scalable virtual sorting outputs, USPS can enhance operational efficiency, reduce sorting errors, and improve delivery times. The project's completion will demonstrate practical IoT applications in postal services, with future potential for full hardware-based automation and physical bundling.