An event-based smart vision node for ultra-low-latency motion detection

By

Yueyao Si [yueyaos2]

Luying Wang [luyingw2]

Shuke Wang [shukew2]

Yaxing Zhang [yaxingz2]

Final Report for ECE445, Senior Design, Fall 2026

TA: Sanhe Fu

May 15, 2026

Project No. 5

Abstract

This project presents a low-cost FPGA-based smart vision system for real-time package inspection and sorting. Instead of relying on an external computer for image processing, the system uses an OV7670 camera connected to an FPGA board to capture package images and process them directly in hardware. The video stream is displayed through VGA and can also be converted into a grayscale edge-detection output to highlight useful package features. Based on the edge information within the detection region, the system determines whether a package is present and whether it is likely to be damaged. The classification result is then sent to a mechanical arm, which performs the corresponding sorting action. A conveyor belt, normal packages, and damaged-package samples are used to test the complete sensing-to-actuation pipeline. The final system demonstrates an embedded workflow from camera input and FPGA image processing to package damage classification and physical sorting, providing a practical prototype for real-time automated package inspection.

Contents

1 Introduction

1.1 Problem Statement

Traditional package inspection and sorting systems often rely on frame-based cameras and external computers for image processing and classification. Although these systems can provide acceptable results, they usually require high data bandwidth, significant computation, and a complicated setup. This makes them less suitable for embedded real-time applications that require compact size, low cost, and standalone operation. In many existing demonstrations, the result is mainly a classification output shown on a screen, rather than a complete physical sensing-to-actuation loop for real package handling.

In practical sorting scenarios, packaged items may be damaged during transportation or handling. If damaged packages are not detected in time, they may continue through the logistics process, reduce product quality, and increase labor cost for later inspection. Therefore, there is a need for a low-cost embedded system that can automatically inspect passing packages in real time and separate acceptable packages from damaged ones.

Our project addresses this problem by building a standalone smart sorting system based on a Dynamic Vision Sensor (DVS), FPGA-based event processing, and a physical output mechanism. As packages move on a conveyor under the DVS camera, the sensor generates asynchronous event data from brightness changes around the package surface and edges. The FPGA processes this event stream in real time to determine whether the observed package is acceptable or damaged. The final decision is then converted into a visible and physical response, such as LED indication and actuator-based sorting. In this way, the project demonstrates not only package inspection, but also a complete embedded hardware pipeline from sensing to decision to actuation.

Another important goal of this project is to make the system operate as a true embedded device rather than a software-assisted lab setup. After power is applied, the system should automatically start sensing, processing, and responding without requiring a separate computer for normal operation. This design improves practicality and better matches the goal of building a low-cost and potentially marketable embedded package inspection and sorting platform.

1.2 Solution Overview & Visual Aid

The proposed solution is a low-cost standalone smart sorting system composed of five major parts: a DVS camera, a conveyor or motion platform, an FPGA processing board, an output subsystem, and a power/system integration module. During operation, packaged items move through the field of view of the overhead DVS camera on a conveyor. When a package passes through the sensing region, the DVS generates asynchronous event data corresponding to brightness changes caused by the package shape, edges, and possible surface defects. These events are transmitted to the FPGA through a high-speed interface.

Inside the FPGA, the event stream passes through several processing stages, including event parsing, buffering, preprocessing, feature extraction, and decision logic. Based on the detected event pattern, the FPGA determines whether the package is acceptable or damaged. The result is then sent to the output subsystem, which provides a clear physical response. For example, an acceptable package may trigger a green LED and continue along the default path, while a damaged package may trigger a red LED and activate a diverter to send it to the reject path. This end-to-end workflow allows the audience to directly observe package movement, event sensing, FPGA-based real-time processing, and physical accept/reject behavior in a single embedded platform.

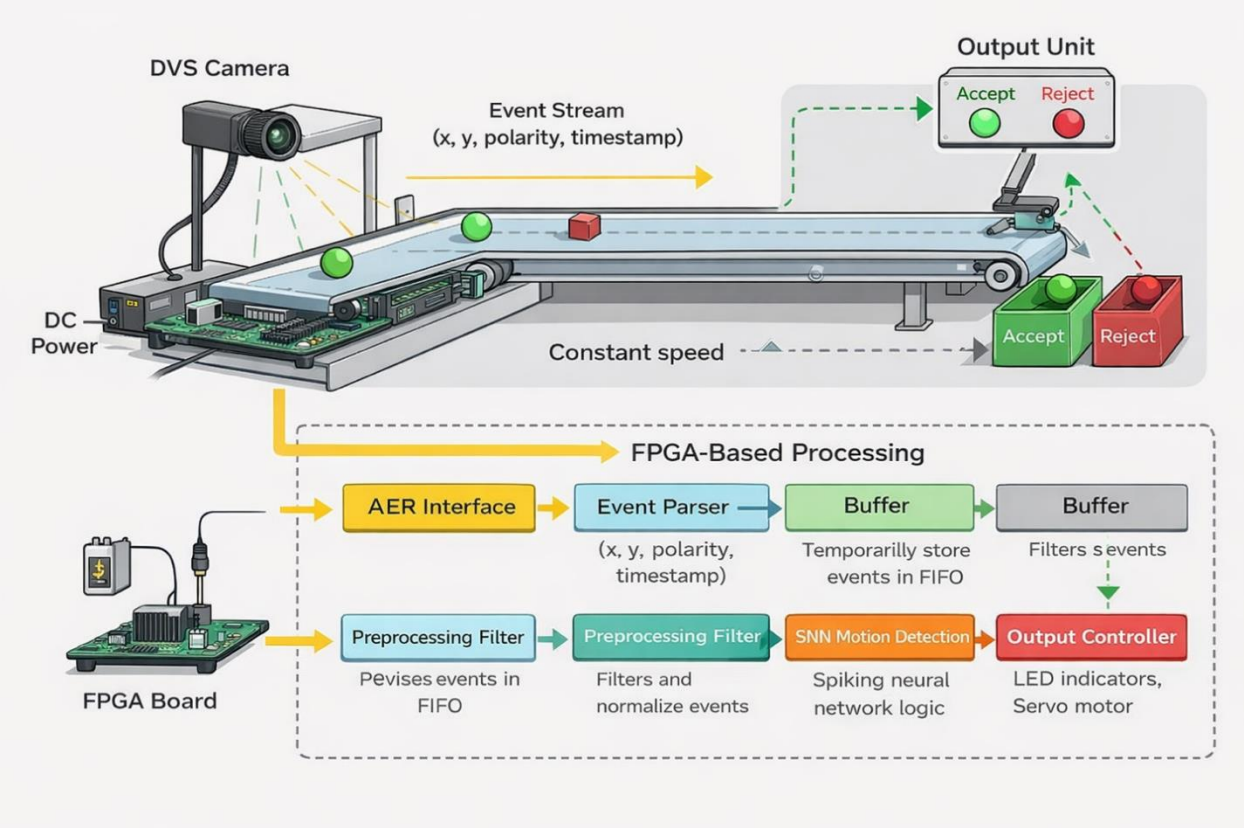

Figure 1. Visual overview of the proposed smart package inspection and sorting system.

Packaged items move on a conveyor under a DVS camera. The event stream is processed by the FPGA in real time, and the classification result drives an LED indicator and a sorting actuator to separate acceptable packages from damaged packages.

1.3 High-level Requirements List

The smart sorting system shall meet the following high-level requirements:

1.3.1 Real-Time Detection Requirement:

The system shall correctly classify passing packaged items as acceptable or damaged with at least 90% accuracy under the defined demonstration conditions.

1.3.2 End-to-End Response Requirement:

The system shall maintain an end-to-end latency below 50 ms from event sensing to output actuation.

1.3.3 Standalone Operation Requirement:

The system shall operate as a standalone embedded platform without requiring an external computer during normal demonstration, and it shall support one-button startup after power is applied.

1.3.4 Conveyor Stability Requirement:

The motion platform shall move objects through the sensing region with speed variation within ±10% of the target speed.

1.3.5 Output Response Requirement:

The output subsystem shall correctly route packages to the corresponding accept or reject path in at least 90% of test cases.

2 Design

2.1 Overall System Architecture

The proposed system is a standalone event-based smart sorting platform that performs real-time package inspection and physical sorting based on visual event data. The overall system is organized into five major subsystems: the Sensing Module, Motion Platform Module, Processing and Decision Module, Output / Sorting Module, and Power and Standalone Integration Module. Each subsystem is responsible for a specific stage in the sensing-to-actuation pipeline, and together they form a continuous real-time processing loop.

In operation, objects are transported by a conveyor belt through the field of view of an event-based camera. The sensor captures asynchronous brightness-change events and transmits them to the FPGA-based processing system. The FPGA performs event parsing, filtering, and SNN-based damage detection to determine whether the package is intact or damaged. Based on this decision, a control signal is generated to drive the sorting actuator, which physically directs the object to either the accept or reject path. At the same time, LED indicators provide visible feedback of system decisions.

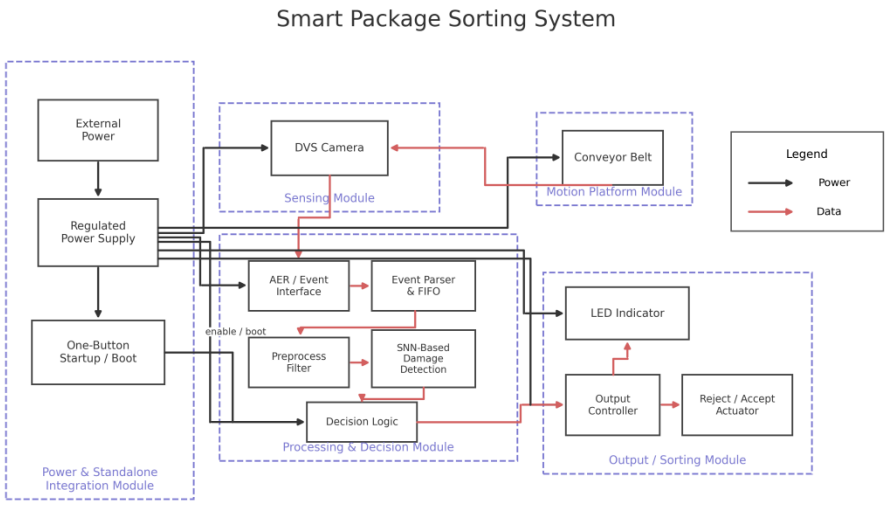

Figure 2. The workflow for the whole Smart Package Sorting System.

2.2 Camera Sensing Module

2.2.1 DVS Sensor Front-End

The sensing module is responsible for capturing visual information of moving objects in the form of asynchronous event streams. The system adopts a Dynamic Vision Sensor (DVS), which outputs

events only when brightness changes occur at the pixel level, significantly reducing redundant data compared to conventional frame-based cameras.

Hardware: DAVIS 346 Event Camera

Specifications

DVS Resolution	346 x 260 pixels
Frame Resolution	346 x 260 pixels, Grayscale, simultaneous output with DVS
DVS Dynamic range	120 dB
APS Dynamic range	56.7 dB
Min. latency	~ 20 us
Lens mount	CS-mount
Connectors / Power	USB 3.0 micro
Bandwidth	12 MEvents / second
Software	DV-Platform
Power consumption	< 180mA @ 5V DC
Dimensions	H 40 x W 60 x D 25 [mm]
Weight	100g (without lens)
Hardware multi-camera sync	Supported (HiRose Connector)
IMU	6-Axis Built-in
Case	Anodized aluminum, 4 mounting points
Tripod mount	Whitworth ¼" female
APS Frame Shutter	Configurable, Global or Rolling Shutter
CMOS Technology	0.18 um 1P6M MIM CIS
Chip size	8 x 6 [mm]
Pixel size	18.5 x 18.5 [um]
Array size	6.4 x 4.8 [mm]
Fill factor	22 %
Pixel complexity	48 transistors, 2 capacitors, 1 photodiode with micro-lens
Chip voltages	1.8 V and 3.3 V
Chip power consumption	DVS: 10-30mW (activity dependent) APS: 140mW
APS dark signal	18000 e^-/s
APS readout noise	55 e^-

Figure 3. The picture and specifications for DVS camera.

The DAVIS 346 provides pixel-level event outputs with microsecond-level latency, making it suitable for detecting motion and structural changes in packages. Its event-driven nature enables efficient processing on FPGA and supports low-latency system response.

2.2.2 Sensing Condition and Event Generation

The quality of the event stream depends on the physical sensing conditions, including object motion, lighting, and surface texture. In this system, events are primarily generated when packages move through the camera's field of view on the conveyor.

Key factors affecting event generation include:

- **Object motion speed:** Faster motion generates higher event rates, increasing detection signal strength but also raising the risk of burst congestion.
- **Lighting conditions:** Stable illumination ensures consistent event generation, while flickering or noise may introduce spurious events.
- **Surface characteristics:** Damaged packaging (e.g., cracks or deformation) produces distinct event patterns compared to intact surfaces.

The sensing module is designed to operate under controlled laboratory lighting and a bounded conveyor speed range to ensure stable and repeatable event generation.

2.3 Processing and Decision Module

2.3.1 Event Interface and Synchronization

The event interface receives asynchronous event data from the DVS sensor and converts it into a format compatible with the FPGA clock domain. This stage ensures reliable communication between external sensing hardware and internal digital logic.

2.3.2 Event Parsing and Buffering

Incoming event data is decoded into structured information, including pixel coordinates and polarity. A FIFO buffer is used to temporarily store events and smooth variations in event arrival rate.

This buffering mechanism is essential to prevent data loss during burst events and maintain stable pipeline operation.

2.3.3 Event Preprocessing

Before detection, the event stream undergoes preprocessing to improve data quality. This includes:

- Noise filtering (removal of isolated events)
- Temporal smoothing
- Basic spatial filtering

The goal of this stage is to suppress noise while preserving meaningful motion-related signals.

2.3.4 SNN-Based Package Damage Detection

This module performs the core detection task. Instead of traditional frame-based classification, the system uses an event-driven approach inspired by Spiking Neural Networks (SNN).

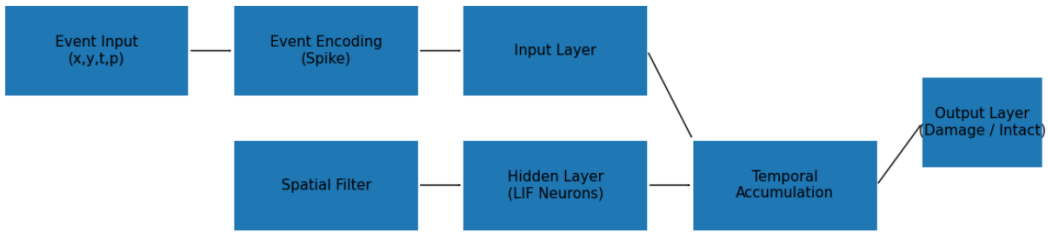

Figure 4. The block diagram for SNN Module.

We plan to experience event input, encoding, filter layer, hidden layer and accumulation to obtain a comprehensive output.

Events are accumulated over a short time window, and the system evaluates whether the activity pattern corresponds to damaged packaging. The detection mechanism relies on threshold-based activation, where significant deviations in event patterns indicate structural irregularities.

This design enables low-latency and hardware-efficient implementation on FPGA.

2.3.5 FPGA Resource and Timing Management

The processing module is implemented on an FPGA platform:

Hardware: Terasic DE10 FPGA Board

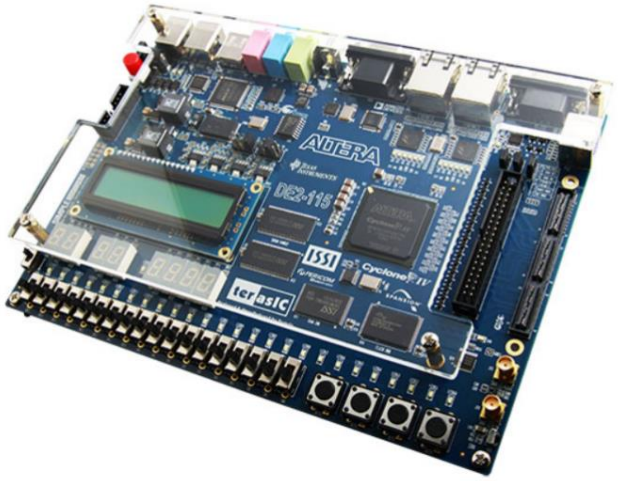

Figure 5. The FPGA Board DE2-115.

Table 1. The category and specifications of DE2-115 Board.

Category	Specification
FPGA	Cyclone IV EP4CE115, 114,480 LEs, 3,888 Kbits memory, 266 multipliers, 4 PLLs, 528 I/Os
Memory	128 MB SDRAM, 2 MB SRAM, 8 MB Flash, 32 Kbit EEPROM
Inputs	18 switches, 4 push-buttons
Outputs	18 red LEDs, 9 green LEDs, 8 seven-segment displays, 16×2 LCD
Audio	24-bit CODEC, line-in, line-out, microphone-in
Clocking	Three 50 MHz clocks, SMA clock input/output
Communication	2 Gigabit Ethernet ports, RS232, PS/2, IR receiver
Expansion	172-pin HSMC, 40-pin expansion port, SD card socket
USB	USB Type A/B, USB 2.0 host/device support
Video	VGA-out, TV-in (NTSC/PAL/SECAM)
Power	Desktop DC input, LM3150MH regulators

The FPGA is responsible for executing all real-time processing tasks, including event handling and detection logic. Resource allocation (logic, memory, and timing) is carefully managed to ensure continuous operation without overflow or delay violations.

2.4 Motion Platform Module

2.4.1 Conveyor Transport Mechanism

The motion platform provides controlled movement of packages through the sensing region. A conveyor system driven by a DC motor is used to transport objects at a relatively constant speed.

Hardware: DC Motor Conveyor System

Figure 6. The picture of DC Motor Conveyor.

The conveyor ensures repeatable motion conditions and serves as the physical input generator for the entire system

2.4.2 Conveyor Speed and Detection Reliability

The conveyor speed plays a critical role in system performance. It directly affects:

- Event rate (higher speed → more events)
- Detection window duration
- Buffer load in FPGA

If the conveyor moves too fast, the system may experience:

- Insufficient accumulation time for detection
- Increased event burst leading to buffer overflow

If the speed is too slow:

- Detection latency increases
- System throughput decreases

Therefore, the system is designed to operate within a controlled speed range where reliable detection can be achieved without exceeding processing capacity.

2.4.3 Inspection-to-Sorting Distance Design

The spatial distance between the sensing region and the sorting actuator defines the **available response time window**:

$$T_{budget} = \frac{d}{v}$$

where d is the distance between camera and actuator, and v is the conveyor speed.

This design parameter is critical because:

- It determines the maximum allowable processing latency
- It ensures that the actuator can respond before the object reaches the sorting point

The system is designed such that this timing budget is always greater than the total processing latency of the FPGA pipeline and actuator response.

2.5 Output Module

2.5.1 Output Signal Generation

The decision result from the FPGA is converted into control signals for both visual and physical outputs. This includes LED indicators and actuator control signals.

2.5.2 Accept / Reject Physical Mechanism

The system uses a servo-based mechanism to physically sort packages based on detection results.

Hardware: MG996R Servo Motor

Specification:

Weight: 55g
Dimension: 40.7×19.7×42.9mm
Stall torque: 9.4kg/cm (4.8v); 11kg/cm (6.0v)
Operating speed: 0.19sec/60degree (4.8v); 0.15sec/60degree (6.0v)
Operating voltage: 4.8~ 6.6v
Gear Type: Metal gear
Temperature range: 0- 55deg
Servo Plug: JR (Fits JR and Futaba)
Dead band width: 1us
Servo wire length: 32cm
Current draw at idle 10mA
No load operating current draw 170mA
Stall current draw 1400mA

Figure 7. The picture and specifications for Servo Motor.

The servo rotates a diverter to direct packages into accept or reject paths.

2.5.3 Actuation Timing Constraint

The actuator must respond within the available timing budget defined by conveyor speed and spacing. Delayed actuation may result in incorrect sorting even if detection is correct.

2.5.4 Sorting Accuracy vs. Conveyor Speed and Spacing

Sorting accuracy is influenced by the interaction between:

- Detection latency
- Conveyor speed
- Camera-to-actuator distance

The system is designed to ensure that correct decisions are translated into correct physical actions within the allowable time window.

2.6 Power and Standalone Integration Module

2.6.1 Regulated Power Supply

The system is powered by an external DC source and uses regulated power distribution to supply different modules.

Different voltage levels are provided to:

- FPGA board
- DVS sensor
- Servo actuator
- Conveyor motor

2.6.2 One-Button Startup and Standalone Operation

The system is designed as a standalone embedded device. After power is applied, the system automatically initializes and begins operation without external configuration or computer support.

This ensures that the system behaves as a complete product rather than a laboratory prototype.

3 Design Verification

3.1 Design verification for subsystems

3.1.1 Camera Sensing Module

Table 2. Requirements and verifications of Sensing Module.

Requirements	Verification
1. The sensing module shall use the DAVIS346 event camera to generate asynchronous event streams under controlled laboratory lighting.	1. Operate the sensing setup under fixed lighting and run 20 trials of package motion through the camera field of view. The module passes if valid event streams are obtained in all trials.
2. For a 10 ms accumulation window, background activity shall remain below 10 events/window.	2. Record event accumulation under no-package/background conditions over repeated 10 ms windows. The module passes if the measured activity remains below 10 events/window.
3. For a 10 ms accumulation window, package motion activity shall exceed 100 events/window, allowing a threshold near $\theta = 50$ to remain valid.	3. Record event accumulation during package motion over repeated 10 ms windows. The module passes if motion activity exceeds 100 events/window, so that $\theta \approx 50$ remains between noise and motion conditions.

3.1.2 Processing and Decision Module

Table 3. Requirements and verifications of Motion Platform Module.

Requirements	Verification
1. The conveyor shall provide controlled package motion through the sensing region.	1. Run repeated package transport tests and confirm that packages pass fully through the camera field of view in all trials.
2. The conveyor speed shall remain within the designed operating range so that reliable detection can be achieved without exceeding processing capacity.	2. Measure the conveyor speed during repeated operation and confirm that the chosen operating speed is the same as that used in detection and timing validation.
3. The camera-to-actuator spacing and conveyor speed shall satisfy the timing condition $L_{total} < d / v$.	3. Measure the distance d and speed v, then compare the available response time d / v against the measured system latency. The module passes if $L_{total} < d / v$.

3.1.3 Motion Platform Module

Table 4. Requirements and verifications of Processing and Decision Module.

Requirements	Verification
1. The FIFO buffer shall satisfy $B \geq (\lambda peak - \mu)\Delta t$. Using the design assumptions, the minimum required capacity shall be 50 events.	1. Verify the design using $\lambda peak = 2\times10^{4}$ events/s, $\mu = 1.5\times10^{4}$ events/s, and $\Delta t = 10$ ms. The module passes if the implemented FIFO capacity is no less than 50 events.
2. The implemented buffer shall include a 2× safety margin, i.e. $B \geq 100$ events.	2. Check the implemented FIFO depth in the design. The module passes if the actual capacity is at least 100 events.
3. The total latency shall satisfy $L_{total} < 50$ ms, with stage budgets of interface + parsing < 5 ms, buffering < 10 ms, detection < 30 ms, and output < 5 ms.	3. Measure or estimate each stage delay and sum them to obtain L_{total}. The module passes if each stage remains within its budget and the total latency is below 50 ms.

3.1.4 Output Module

Table 5. Requirements and verifications of Output Module.

Requirements	Verification
1. The output module shall convert FPGA decisions into LED indication and servo-based physical sorting.	1. Apply known decision outputs and confirm that the LED indicator and MG996R servo both respond to the decision signal.
2. The physical sorting action shall complete within the available timing budget defined by conveyor speed and spacing.	2. Measure the interval from decision generation to completed actuator response. The module passes if the response remains within the available time budget d / v.
3. The sorting mechanism shall preserve correct physical action when the detection result is correct.	3. Run repeated accept/reject tests with known decision outputs and confirm that the actuator directs the object to the intended path in each case.

3.1.5 Power and Standalone Integration Module

Table 6. Requirements and verifications of Power and Standalone Integration Module.

Requirements	Verification
1. The system shall operate from a single external DC source with regulated distribution to the FPGA board, DVS sensor, servo actuator, and conveyor motor.	1. Inspect the integrated power setup and confirm that all modules are supplied from the same external DC source through regulated distribution.
2. Supply variation shall remain within ±5% during operation.	2. Measure the regulated output voltages during normal operation. The module passes if all measured voltages remain within ±5% of nominal values.
3. The system shall boot automatically and enter detection mode without external computer support.	3. Perform repeated power-up tests. The module passes if FPGA configuration and peripheral initialization complete automatically and the system enters detection mode without laptop-side intervention.

3.2 Tolerance Analysis

3.2.1 Critical Design Component

The most critical design component of this project is the end-to-end event-to-actuation pipeline, specifically the combination of

(1) the DVS-to-FPGA event interface and buffering, and (2) the real-time detection and decision logic that drives the physical output actuator.

This is the highest-risk part of the design because it is the point where the project stops being a software-only classification task and becomes a standalone embedded hardware system. The success of the project is not determined only by whether events can be classified correctly in simulation, but by whether the system can reliably transform real sensor events into a timely physical response on a low-cost platform without the help of an external computer.

In the final demonstration, the audience should see a complete hardware loop: an object moves through the sensing region, the DVS camera generates events, the FPGA processes those events in real time, and the system produces an observable output such as LED indication and actuator or diverter motion. Therefore, the true critical function is not only detection accuracy, but deterministic real-time event handling under physical operating conditions.

3.2.2 Source of Risk

The main design risk is that the event stream generated by the DVS sensor is highly variable. Its rate depends on object speed, object contrast, ambient lighting variation, and scene noise. This creates three coupled risks:

1) Event Burst Overflow: If the instantaneous event rate is too high, the FPGA input buffer may overflow before downstream logic can process all events. In that case, useful events are lost, which may cause missed detections or unstable output behavior.

2) Threshold Instability: If the detection threshold is selected too low, noise or background flicker may trigger false detections. If it is selected too high, real objects may not produce enough accumulated activity to activate the sorter. This is a design tolerance issue because the correct threshold must remain valid across a range of real operating conditions rather than for only one ideal test case.

3) End-to-End Timing Failure: Even if the logic is correct, the project fails its intended function if the decision arrives too late for the actuator to respond while the object is still in the sorting region. Thus, latency tolerance is as important as detection correctness.

For this reason, the most critical risk is not that the algorithm is difficult, but whether the hardware pipeline can tolerate realistic variation in event rate and still produce a correct physical sorting decision within the allowed time budget.

3.2.3 Mathematical Tolerance Analysis

To demonstrate that the proposed system can operate reliably under realistic conditions, the key design risks identified above are further analyzed from a quantitative perspective. The following subsections correspond to the three primary sources of tolerance concern: detection robustness, buffer stability, and real-time response capability.

1) Quantitative Analysis of Detection Threshold

The reliability of the detection module depends on the separation between noise-induced activity and motion-induced activity within a fixed accumulation window. The accumulated event activity is determined by the event rate and the time window, and the detection threshold must be carefully selected to ensure robustness.

Let the event rate be λ (events/s) and the accumulation time window be T (s). The total accumulated activity is given by $A = \lambda T$

For reliable detection, the threshold θ must satisfy the condition:

$$\lambda_{nT} < \theta < \lambda_{mT}$$

where λ_n is the noise event rate and λ_m is the motion-induced event rate. This ensures that noise does not trigger detection while real motion does. For example, if $\lambda_n \approx 1\times10^3$ events/s and $\lambda_m \approx 1\times10^4$ events/s with T = 10 ms, then:

- Noise accumulation: $A_n = 10$
- Motion accumulation: $A_m = 100$

A practical threshold such as θ ≈ 50 provides a clear separation margin between these two cases. This margin is important because in real-world operation, environmental factors such as lighting variation and object speed may slightly shift event rates. A sufficiently large separation ensures that the system remains stable under such variations.

2) Buffer Stability and Overflow Tolerance

The second critical tolerance issue lies in the system's ability to handle bursty event streams without data loss. Since the DVS sensor produces asynchronous events, the incoming event rate can temporarily exceed the processing capability of the FPGA.
To prevent overflow, the FIFO buffer must handle peak event rates. If the maximum expected event rate is λ_{max} and the processing rate is μ, the buffer size B must satisfy:

$$B \geq (\lambda_{max} - \mu) \cdot \Delta t$$

where Δt is the maximum burst duration. For example, if λ_{max} = 2×10⁴ events/s, μ = 1.5×10⁴ events/s, and Δt = 10 ms, then B ≥ 50 events. Designing the buffer with at least 2× margin (e.g., B ≥ 100) ensures safe operation.

3) Real-Time Response and Latency Budget

The total system latency is the sum of delays across all blocks:

$$L_{total} = L_{interface} + L_{buffer} + L_{Processing} + L_{output}$$

To meet the requirement L_total < 50 ms, each stage must be bounded. For example:

- Interface + parsing: < 5 ms
- Buffering: < 10 ms
- Detection: < 30 ms
- Output: < 5 ms

This ensures the system meets real-time constraints.

4) Standalone Power and System Integration Tolerance

Another important tolerance issue is system startup and standalone operation. Since the device is intended to function as a one-button embedded system, it cannot depend on manual software configuration, laptop-side processing, or repeated parameter loading after power-up.

The integrated hardware is considered tolerant at the system level if all modules power on from a single regulated supply, voltage variation remains within ±5%, FPGA configuration and peripheral initialization complete automatically, and the system enters detection mode without external computer intervention.

This requirement reduces the risk that the project appears as a partially assembled prototype rather than a complete embedded product. It also addresses the instructor's concern that the project should be demonstrable as a real device rather than as a software pipeline assisted by external tools.

3.2.4 Tolerance Conclusion

The design can be considered feasible if the following conditions are simultaneously satisfied:

$$\lambda_n T_w < \theta < \lambda_m T_w$$

$$B \geq (\lambda_{peak} - \mu)\,\Delta t$$

$$L_{total} < \min(50\text{ ms}, d / v)$$

and the system can boot and run from a single standalone power source without external computing support.

If these conditions are met, then the proposed architecture has sufficient tolerance to variation in sensor input, event burst rate, and physical timing, and the design should remain functional in a real demonstration environment.

4 Costs

The following table summarizes the estimated component costs for this project based on current Chinese market prices (RMB). The FPGA development board and the event-based camera are both provided by the ECE 445 laboratory and are listed at no direct cost to the team. All remaining components are commercially available at low cost. The total estimated out-of-pocket cost is approximately ￥1,500 RMB.

Table 7. Estimated project component costs in RMB based on current Chinese market prices.

Component	Description	Qty	Total (RMB)
Event-based Camera	DVS sensor for asynchronous event generation	1	￥1100
FPGA Development Board	Real-time event processing platform	1	￥0
Servo Motor	Servo-based sorting diverter actuator	1	￥25
DC Motor Conveyor Belt	Conveyor platform for package transport	1	￥900
LED Indicators (Red/Green)	Visual accept/reject output feedback	4	￥12
Regulated Power Supply	External DC source with distribution module	1	￥120
Wiring, Connectors & Mounting Hardware	Cables, PCB headers, mechanical fixtures	—	￥80
Total			**≈ ￥1,137**

5 Conclusion

5.1 Ethics

The design and development of this smart package inspection and sorting system involve several ethical considerations aligned with the IEEE Code of Ethics and the ACM Code of Ethics.

Data Privacy and Sensing Scope: The event-based camera used in this system captures only brightness-change events rather than full video frames. This design choice inherently limits the amount of visual information collected, reducing the risk of unintended capture of personally identifiable information. The system is intended solely for package surface inspection in a controlled laboratory environment and is not designed to monitor individuals.

Fairness and Non-Discrimination: The sorting decisions made by this system are based entirely on physical event patterns generated by package surface characteristics. The system does not incorporate any demographic, social, or identity-related information. Sorting outcomes are determined objectively by the detection logic and are not influenced by any biased data source.

Reliability and Professional Responsibility: As engineers, we are responsible for ensuring that the system operates within its defined specification and does not produce harmful or misleading outputs. The system is designed for controlled demonstration with defined accuracy targets (≥90%) and latency bounds (<50 ms). Any extension to industrial deployment would require additional validation, redundancy, and fail-safe mechanisms beyond the scope of this prototype.

Environmental Responsibility: The system uses low-power event-driven sensing to minimize energy consumption compared to conventional frame-based inspection systems. By processing only motion-relevant data, the design reduces unnecessary computation and is consistent with energy-efficient engineering principles.

Transparency: The system's detection logic and classification criteria are fully documented and reproducible. Decisions made by the system can be traced back to measurable event patterns, ensuring transparency and interpretability of the sorting outcome.

5.2 Safety

The physical operation of the smart sorting system involves moving mechanical parts, electrical components operating at multiple voltage levels, and an event-based camera mounted above the conveyor. The following safety considerations are incorporated into the system design.

Electrical Safety: The system is powered by a single regulated DC source, and all subsystems receive power through regulated voltage rails. Supply voltage variation is maintained within ±5% of nominal values. No high-voltage AC connections are used in the system. All wiring is insulated and secured to prevent accidental contact during operation.

Mechanical Safety: The conveyor belt and servo-based diverter involve moving parts. During operation, users should maintain a safe distance from the sorting region. The conveyor speed is

bounded within a controlled range to prevent mechanical instability or uncontrolled ejection of packages. The servo motor's range of motion is hardware-limited by physical stops to prevent over-rotation.

Camera Mounting Safety: The event-based camera is mounted in a fixed overhead position above the conveyor. The mounting structure is secured to prevent accidental displacement during operation. Users should not reach into the sensing area while the system is active.

Fail-Safe Behavior: In the event of a detection timeout or signal loss, the system defaults to a safe output state (e.g., no actuation, or directing all objects to the accept path) to avoid unpredictable mechanical behavior. LEDs provide continuous visual indication of system state.

Standalone Operation Risks: Since the system is designed to operate without a connected computer, the operator should ensure the system is powered off before replacing or repositioning objects near the sensing and sorting region. A single power button controls the entire system, and all safety checks are performed during the boot sequence.

The physical operation of the smart sorting system involves moving mechanical parts, electrical components operating at multiple voltage levels, and an event-based camera mounted above the conveyor. The following safety considerations are incorporated into the system design.

Electrical Safety: The system is powered by a single regulated DC source, and all subsystems receive power through regulated voltage rails. Supply voltage variation is maintained within ±5% of nominal values. No high-voltage AC connections are used in the system. All wiring is insulated and secured to prevent accidental contact during operation.

Mechanical Safety: The conveyor belt and servo-based diverter involve moving parts. During operation, users should maintain a safe distance from the sorting region. The conveyor speed is bounded within a controlled range to prevent mechanical instability. The servo motor's range of motion is hardware-limited by physical stops to prevent over-rotation.

Camera Mounting Safety: The event-based camera is mounted in a fixed overhead position above the conveyor. The mounting structure is secured to prevent accidental displacement during operation. Users should not reach into the sensing area while the system is active.

Fail-Safe Behavior: In the event of a detection timeout or signal loss, the system defaults to a safe output state (e.g., no actuation, or directing all objects to the accept path) to avoid unpredictable mechanical behavior. LEDs provide continuous visual indication of system state.

Standalone Operation Risks: Since the system is designed to operate without a connected computer, the operator should ensure the system is powered off before replacing or repositioning objects near the sensing and sorting region. A single power button controls the entire system, and all safety checks are performed during the boot sequence.

References

[1] Terasic Technologies, *DE2-115 User Manual*. Terasic Technologies, Inc., 2023. [Online]. Available: https://www.terasic.com.tw/cgi-bin/page/archive.pl?CategoryNo=183&Language=English&No=502&PartNo=2

[2] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based Vision: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 154–180, Jan. 2022. [Online]. Available: https://arxiv.org/pdf/1904.08405

[3] T. Kryjak, "Event-based vision on FPGAs – a survey," *arXiv preprint arXiv:2407.08356*, Jul. 2024. [Online]. Available: https://arxiv.org/html/2407.08356v1

[4] F. Perez-Peña, A. Morgado-Estevez, A. Linares-Barranco, A. Jimenez-Fernandez, F. Gomez-Rodriguez, G. Jimenez-Moreno, and J. Lopez-Coronado, "Neuro-Inspired Spike-Based Motion: From Dynamic Vision Sensor to Robot Motor Open-Loop Control through Spike-VITE," *Sensors*, vol. 13, no. 11, pp. 15805–15832, Nov. 2013. https://doi.org/10.3390/s131115805

[5] Y. Gao, S. Wang, and H. K.-H. So, "A Reconfigurable Architecture for Real-time Event-based Multi-Object Tracking," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 16, no. 4, Sep. 2023. https://doi.org/10.1145/3593587