TỔNG QUAN BÁO CÁO SEMI

**🎯 GIAI ĐOẠN 1: HIỂU ĐỀ – KHOAN SÂU VẤN ĐỀ**

| **Việc cần làm** | **Trạng thái** | **Ghi chú** |
| --- | --- | --- |
| ✅ 1. **Hiểu rõ đề bài thi** | ✔ Đã hoàn thành | Đề yêu cầu “smart sampling”, “AI models”, “real-time QC” |
| ✅ 2. **Xác định chuỗi quy trình bán dẫn** | ✔ Đã có | 9 bước, đã tô đậm 3 điểm kiểm lỗi chính |
| ✅ 3. **Phóng to vào quy trình Final Inspection** | ✔ Đã xác định | Dựa trên AQL → đang là điểm nghẽn |
| ✅ 4. **Phân tích điểm yếu AQL sampling** | ✔ Rất rõ | Bỏ sót lỗi, không sinh dữ liệu, tốn nhân công |
| 🔜 5. **Xác định nỗi đau doanh nghiệp (Singapore)** | ⚠️ Cần trình bày rõ hơn | Nhân lực kiểm lỗi thiếu, lỗi lọt, khó mở rộng AI |
| 🔜 6. **Tìm hiểu các công ty đang làm gì hiện nay** | ⚠️ Cần thêm thông tin | (Ví dụ: TSMC, Intel, GlobalFoundries...) |

**🛠️ GIAI ĐOẠN 2: THIẾT KẾ GIẢI PHÁP**

| **Việc cần làm** | **Trạng thái** | **Ghi chú** |
| --- | --- | --- |
| 🔜 7. **Thiết kế quy trình mới: Vision AI + Smart Sampling** | ✅ Có ý tưởng, cần vẽ pipeline | Vision kiểm toàn bộ, AI tự học, phân tích lỗi, adaptive sampling |
| 🔜 8. **Xác định phần cứng sử dụng (IoT)** | ⚠️ Cần chọn mô-đun: Jetson, Pi + camera + cảm biến? |  |
| 🔜 9. **Xác định phần mềm (AI models)** | ✅ Gợi ý: YOLOv8n, MobileNet + LightGBM cho dữ liệu máy |  |
| 🔜 10. **Tạo logic lấy mẫu linh hoạt (Smart Sampling Engine)** | ⚠️ Rất quan trọng – thay thế AQL bằng AI logic |  |

**🧠 GIAI ĐOẠN 3: TRÌNH BÀY & THUYẾT PHỤC**

| **Việc cần làm** | **Trạng thái** |  |
| --- | --- | --- |
| 🔜 11. **Chiến lược trình bày không dựa vào thuật toán** | ⚠️ Ưu tiên gây ấn tượng = cảm xúc + tính thực tế + lợi ích |  |
| 🔜 12. **Soạn 1-slide poster: Gọn – Đẹp – 3 vùng** | ⚠️ Nêu vấn đề – giải pháp – hiệu quả |  |
| 🔜 13. **Slide pitch (5–7 slide)** | ⚠️ Cấu trúc: Vấn đề → Hệ thống → Demo → Lợi ích → Mở rộng |  |
| 🔜 14. **Viết đoạn thuyết trình 3 phút** | ⚠️ Truyền cảm – súc tích – đúng trọng tâm |  |
| 🔜 15. **Tạo sơ đồ hệ thống trực quan (pipeline)** | ✅ Có thể hỗ trợ vẽ Canva, Lucidchart, v.v. |  |

**📊 GIAI ĐOẠN 4: DỮ LIỆU & MÔ PHỎNG**

| **Việc cần làm** | **Trạng thái** | **Ghi chú** |
| --- | --- | --- |
| 🔜 16. **Tải dataset từ bảng AQL nếu cần** | ✅ Có thể viết script Python để tạo bảng mẫu từ website |  |
| 🔜 17. **Tạo tập ảnh lỗi mô phỏng (dùng để demo Vision AI)** | ✅ Có thể dùng ảnh wafer/PCB bị lỗi |  |
| 🔜 18. **Tạo file mô phỏng output AI (CSV, JSON)** | ⚠️ Để trình bày: AI phát hiện gì? quyết định gì? |  |

**CHIẾN LƯỢC TỔNG THỂ ĐỂ CHIẾN THẮNG**

Chúng ta KHÔNG cần chứng minh mình là chuyên gia AI.  
**CHÚNG TA cần chứng minh:**

* Bạn hiểu rõ điểm đau ngành bán dẫn
* Giải pháp bạn đưa ra “giải quyết được vấn đề” và có thể triển khai thực tế
* Chi phí rẻ hơn, nhanh hơn, thông minh hơn
* Trình bày gãy gọn, chuyên nghiệp, tạo cảm hứng

**GIAI ĐOẠN 1 : PHÂN TÍCH CHALLENGE**

PHẦN 1 : PHÂN TÍCH ĐỀ

**Phân tích đề bài – từng yêu cầu rõ ràng**

| **Thành phần của đề** | **Ý nghĩa** | **Việc bạn cần làm** |
| --- | --- | --- |
| **“Smart automation and AI”** | Bạn phải dùng AI & tự động hóa, không còn thủ công | Hệ thống của bạn phải có AI thực sự – không chỉ làm thủ công rồi nói suông |
| **“Smart sampling”** | Cải tiến phương pháp lấy mẫu – không còn AQL cứng nhắc | Phải có logic lấy mẫu thông minh: dựa trên dữ liệu, cảnh báo lỗi, hoặc mức rủi ro |
| **“Monitor production line”** | Phải giúp giám sát dây chuyền, không chỉ kiểm tra từng sản phẩm đơn lẻ | Hệ thống có thể báo cáo theo ca, theo khu vực sản xuất, hoặc theo máy |
| **“Anticipate failures before it happens”** | Không chỉ phát hiện lỗi, mà còn dự đoán lỗi trước khi xảy ra | Cần AI học từ lịch sử và dự đoán xu hướng lỗi (nếu có dữ liệu thiết bị) |
| **“Machine Learning models”** | Nêu cụ thể thuật toán gì, dùng ở đâu | Trình bày mô hình nhẹ nhưng thực tế: YOLOv8n, MobileNet, hoặc LightGBM |
| **“Computer vision… in real time”** | Thị giác máy tính là bắt buộc, dùng để kiểm lỗi trực tiếp | Hệ thống AI của bạn cần có camera + model xử lý ảnh – không chỉ nói về lý thuyết |

**🛠️ PHẦN CỨNG: HỆ THỐNG IoT KIỂM LỖI THÔNG MINH**

**🎯 Mục tiêu:**

Tạo **một hệ thống phần cứng thực tế** gồm các thiết bị giá rẻ, dễ triển khai, dùng để:

* Gắn lên dây chuyền sản xuất bán dẫn
* Thu thập hình ảnh sản phẩm/die/wafer
* Gửi về AI model xử lý real-time
* Có thể mở rộng số lượng node theo cụm máy/dây chuyền

**Cần làm:**

| **Thành phần** | **Gợi ý lựa chọn** |
| --- | --- |
| **Camera module** | Raspberry Pi Cam / ESP32-CAM / USB webcam HD |
| **Thiết bị xử lý tại chỗ** | Raspberry Pi 4 / Jetson Nano / Khả năng chạy YOLO nhỏ gọn |
| **Bộ lưu trữ & truyền dữ liệu** | WiFi hoặc LAN nội bộ (Edge), có tùy chọn gửi lên Cloud Gateway |
| **Nguồn điện & vỏ chống bụi** | Tùy môi trường nhà máy (có thể dùng pin/adapter + housing 3D print) |
| **Khả năng mở rộng** | Hệ thống dạng mô-đun – plug-and-play nhiều node song song |

**Việc cần viết:**

* Sơ đồ bố trí hệ thống IoT (1 node + nhiều node)
* Ghi rõ vai trò từng phần cứng
* Mô tả chi phí (chi tiết chi phí thấp là điểm cộng lớn!)

**🤖 PHẦN MỀM: AI / MACHINE LEARNING – BỘ NÃO CỦA HỆ THỐNG**

**🎯 Mục tiêu:**

Tạo **một mô hình học máy hiệu quả, nhẹ**, hoạt động real-time để:

* Phát hiện lỗi trực tiếp từ hình ảnh sản phẩm
* Phân loại lỗi: Critical / Major / Minor
* Ghi nhớ và học từ kết quả trước đó (reinforcement)
* Gợi ý ngưỡng lấy mẫu động thay AQL cố định

**Gợi ý mô hình:**

| **Ứng dụng** | **Mô hình đề xuất** |
| --- | --- |
| **Phát hiện lỗi hình ảnh (real-time)** | YOLOv8-nano / MobileNetV3 / EfficientNet-lite |
| **Phân tích dữ liệu cảm biến/máy** | LightGBM / Random Forest (nếu có dữ liệu thiết bị) |
| **Smart Sampling logic** | Rule-based + Threshold AI / Reinforcement (nếu có thời gian) |

**Việc cần làm:**

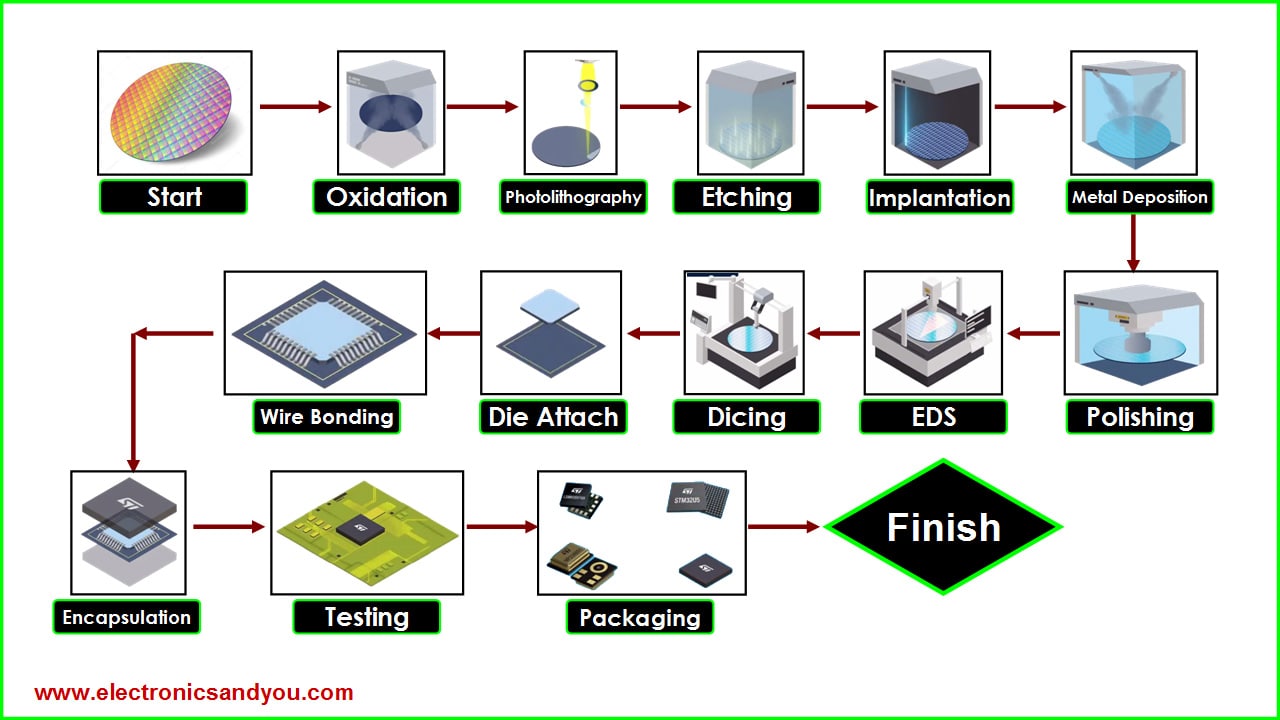
* Viết phần trình bày “tại sao chọn mô hình đó”
* Vẽ sơ đồ: Input → AI → Output → Decision
* Nếu cần: mô phỏng đầu vào bằng ảnh mẫu + đầu ra bằng kết quả phân loại

PHẦN 2 : QUY TRÌNH

**I. Chuỗi Quy Trình Sản Xuất Bán Dẫn (Tổng Quan)**

Chuỗi sản xuất bán dẫn gồm nhiều giai đoạn phức tạp, thường chia làm 4 khâu chính:

1. **Xử lý tổ ong silic (Front-End Wafer Fabrication)**
   * Khởi nguồn từ silicon wafer
   * Bao gồm: oxide growth, photolithography, etching, ion implantation
2. **Kiểm tra & dicing wafer**
   * Kiểm tra bằng hệ thống quang để phát hiện khuyết tật wafer
   * Cắt die (từng chip nhỏ) ra khỏi wafer
3. **Đóng gói chip (Back-End Packaging)**
   * Đóng chip vào khuôn, wire bonding, molding
4. **Kiểm tra cuối và giao hàng (Final Testing & QC)**
   * Electrical test, Visual Inspection
   * Lúc này sử dụng AQL Sampling là chính



**II. Xác Định Lỗi Thất Trong Chuỗi Quy Trình**

Trong chuỗi trên, các vấn đề chủ yếu phát sinh tại:

* **Wafer Inspection**: Khuyết tật nhỏ, lặp lỉ 2D hoặc 3D
* **Die Inspection**: Sai lệch bonding, lỗi chồi nối, bễ mặt
* **Final Visual Inspection**: Dựa vào AQL sampling, **để lại nhiều lỗi nghiêm trọng**

Vì vậy, **chúng ta tập trung vào quy trình cuối - Final QC**,

nơi hiện nay vẫn sử dụng phương pháp kiểm tra lấy mẫu theo tình ngẫu nhiên (AQL Sampling)

Quy trình này là **vấn đề** của nhiều nhà máy do:

* Bỏ sót lỗi nguy hiểm
* Tốn nhân lực
* Không sinh dữ liệu huấn luyện AI
* Không tích hợp với quy trình Edge AI hiện đại

PHẦN 3 **Phóng To: Quy Trình Final Visual Inspection**

Quy trình Final Visual Inspection là công đoạn kiểm tra sau cùng trước khi chip bán dẫn được đóng gói và giao cho khách hàng. Các yếu tố chính bao gồm:

* **Mục tiêu**: Phát hiện các lỗi nhẹ như nứt, trầy xước, lỗi in laser, sai logo, sai ID, vỏ góc
* **Công cụ**: Thường là camera, kính hiển vi quang học, và nhân sự
* **Phương pháp**: Đánh giá theo mẩu AQL (Acceptable Quality Limit) hoặc theo checklist
* **Nguy cơ**:
  + Bỏ sót lỗi nhỏ (do mắt người hoặc camera không đủ độ phân giải)
  + Tính chủ quan cao (nhân sự đánh giá khác nhau)
  + Chi phí vận hành cao (phải kiểm tra tay/tại chỗ theo ca)
  + Dữ liệu không được ghi nhận để AI học hoặc tracking

=> **Kết luận**: Đây chính là công đoạn chủ lực trong quy trình AQL hiện nay, và là nơi cần được tự động hoá hoàn toàn bằng AI + thị giác máy tính nhằm:

* Loại bỏ rủi ro
* Giảm chi phí nhân sự
* Sinh dữ liệu để huấn luyện AI feedback
* Tăng độ đáng tin cho khách hàng trong chuyển giao chip

**Dựa vào đâu mà chọn đúng “Final Inspection” để tự động hóa?**

**🧠 Dựa vào 3 yếu tố:**

**a. Đề thi yêu cầu rõ:**

“...Smart Sampling... to monitor production line and anticipate failures…”

⟶ Vậy ta phải chọn **một công đoạn nào đó mà:**

* Có sampling (lấy mẫu)
* Có kiểm lỗi
* Có khả năng gắn AI vào để theo dõi và “anticipate failure”

→ Trong chuỗi 9 bước sản xuất bán dẫn, chỉ có **Final Visual Inspection** hội đủ cả 3 yếu tố đó.

**b. Trong thực tế ngành bán dẫn:**

* Final inspection hiện vẫn **dùng nhân lực + lấy mẫu**
* **AQL sampling** thường đặt ở giai đoạn cuối
* Các doanh nghiệp đang **tốn chi phí + để lọt lỗi + không có dữ liệu AI** ở giai đoạn này

→ Vậy đây là nơi tốt nhất để:

* **Đưa AI vào kiểm lỗi trực tiếp bằng vision**
* **Thay AQL bằng Smart Sampling**
* **Sinh dữ liệu đầy đủ để huấn luyện mô hình AI**

**c. Tư duy chiến lược:**

Trong một cuộc thi đổi mới, ta phải:

* Không làm dàn trải
* **Tập trung đột phá đúng một điểm nghẽn lớn nhưng có thể giải quyết tốt**  
  → Và Final Visual Inspection + AQL là lựa chọn tối ưu.

PHẦN 4 : **Phân Tích Chi Tiết Về Phương Pháp AQL Sampling**

**A. AQL Sampling là gì?**

AQL (Acceptable Quality Limit) là một chuẩn quốc tế (ISO 2859-1) quy định số lượng mẫu sản phẩm cần kiểm tra trong một lô, và số lỗi tối đa cho phép để quyết định Accept/Reject.

**B. Cách vận hành:**

* Bóc ngẫu nhiên một số sản phẩm trong lô (theo bảng AQL)
* Kiểm tra theo tiêu chí Critical / Major / Minor
* So sánh với giới hạn lỗi: và Accept/Reject toàn bộ lô

**C. Nhược điểm nghiêm trọng**

1. **Tính ngẫu nhiên cao**: Không đảm bảo sản phẩm lỗi nặng được phát hiện.
2. **Lỗi chỉ là đại diện**: Dẫn đến hiện tượng "pass ảo", khi toàn bộ lô có thể bị sai.
3. **Không sinh dữ liệu đủ để AI học**: Chỉ kiểm 5–10% → không đủ dữ liệu để huấn luyện mô hình AI.
4. **Tốn nhân lực đối chiếu**: Cần người kiểm thử nghiệm để thực hiện.
5. **Không linh hoạt**: Các sản phẩm mới hoặc quy trình thay đổi cần thiết lập AQL mới thủ công.
6. **Không tích hợp Edge AI**: AQL mang tính tứ động, không có feedback loop.

**D. Liên quan đến đề bài**

Đề thi yêu cầu "Smart Sampling + AI + Vision" → **AQL sampling trở thành lỗi thời**, không còn phù hợp với nhu cầu "real-time, feedback, intelligent inspection"

**E. Kết luận: Vì sao cần thay thế AQL?**

* Đáp ứng tự động hoá và Edge AI
* Giúc AI sinh dữ liệu và học theo thời gian thật
* Tránh bỏ sót lỗi nguy hiểm
* Phân loại lỗi theo cách người thật không thể làm

PHẦN 5 : **Phân Tích Nỗi Đau Của Doanh Nghiệp Bán Dẫn (Singapore & Khu Vực ASEAN)**

Trong bối cảnh bán dẫn ASEAN đang chuyển dần sang chuẩn hoá tự động hóa, Singapore là đầu tàu, nhưng vẫn đang đối mặt với nhiều "nỗi đau" trong quy trình kiểm lỗi cuối:

**1. Thiếu nhân sự kiểm lỗi tay nghề cao**

* Chi phí nhân công cao (theo USD/h)
* Lực lượng lao động kỹ thuật giới hạn, luôn trong tình trạng thấp
* Việc kiểm tra phụ thuộc nhiều vào kinh nghiệm

**2. Tỷ lệ lỗi bỏ sót cao với AQL**

* Kiểm theo mẫu AQL → không đại diện cho toàn lô
* Dễ bỏ qua lỗi nhỏ nguy hiểm đối với chip logic cao cấp

**3. Không sinh dữ liệu cho AI / Analytics**

* Nhà máy không thu thập đầy đủ hình ảnh và label
* Không có dữ liệu huấn luyện cho AI/ML
* Mệt mỏi khi tích hợp quy trình cũ với chuyển đổi số

**4. Không scale được kiểm tra khi nhu cầu tăng**

* Tăng 20% sản lượng → tăng 30% chi phí kiểm tra tay
* Không linh hoạt: cần AI để scale nhanh theo lô & ca

**5. Chưa có quy trình feedback loop cho chất lượng**

* Kiểm tra AQL dừng lại sau "pass/fail"
* Không học từ lỗi để cải thiện quy trình upstream

=> **Tổng hợp:**

Final inspection vẫn còn mang tính "cổ điển, nhân công, rời rạc". Trong khi đó, xu hướng toàn cầu là Smart Factory, Zero Defect, và AI-driven Sampling

**Đây chính là "nỗi đau" thật sự để chúng ta đề xuất giải pháp đổi mới có giá trị với doanh nghiệp.**

PHẦN 6 : **ình hình hiện tại: Các phương pháp kiểm tra chất lượng bằng AI trong ngành bán dẫn**

**1. TSMC – Hệ thống phân loại lỗi tự động (ADC) với học máy có sự hỗ trợ của con người (HITL)**

* TSMC sử dụng học máy để triển khai phân loại lỗi tự động (ADC) và duy trì độ chính xác của việc xác định lỗi nâng cao. Phương pháp học máy mà họ sử dụng là "Human-in-the-loop" (HITL), kết hợp trí tuệ của con người và máy móc để tạo ra các mô hình học máy. [tsmc.com](https://www.tsmc.com/english/dedicatedFoundry/services/apm_intelligent_packaging_fab/intelligentFab_automation?utm_source=chatgpt.com)

**2. Intel – Kiểm tra wafer thông minh với AI và thị giác máy tính**

* Intel đã áp dụng tự động hóa dựa trên trí tuệ nhân tạo (AI) và thị giác máy tính (CV) như một thành phần quan trọng trong kiểm soát chất lượng tại các nhà máy chế tạo của mình trong hơn một thập kỷ. Họ thu thập hình ảnh từ nhiều kênh trong quá trình chế tạo và tính toán hàng nghìn đặc trưng để tự động phát hiện lỗi. [Intel](https://www.intel.com/content/dam/www/central-libraries/us/en/documents/2023-11/intel-it-smart-manu-using-ai-inline-paper.pdf?utm_source=chatgpt.com)

**3. GlobalFoundries – Hệ thống kiểm tra hình ảnh tự động với AutoML**

* GlobalFoundries đã triển khai hàng trăm mô hình trong các nhà máy của mình để phát hiện và kiểm soát lỗi quy trình. Các tính năng quản lý dữ liệu và mô hình của AutoML Vision giúp làm mới dữ liệu liên tục và hiệu quả, cung cấp cho công ty khả năng quan sát tất cả các mô hình đó. [Google Cloud](https://cloud.google.com/blog/products/ai-machine-learning/ai-and-machine-learning-improve-manufacturing-visual-inspection-process?utm_source=chatgpt.com)

**4. Samsung – Sử dụng AI cho kiểm tra PCB và hàn**

* Samsung tận dụng thị giác máy tính để kiểm tra PCB và hàn, tối ưu hóa chất lượng sản xuất. [Musashi AI](https://musashiai.com/2025/05/07/ai-inspections-in-action-how-manufacturing-industry-leaders-are-elevating-quality-control/?utm_source=chatgpt.com)

**5. Foxconn – Kiểm tra quang học tự động (AOI) với AI**

* Foxconn sử dụng kiểm tra quang học tự động (AOI) được hỗ trợ bởi AI để phát hiện lỗi tốc độ cao trong các cụm điện tử. [Musashi AI](https://musashiai.com/2025/05/07/ai-inspections-in-action-how-manufacturing-industry-leaders-are-elevating-quality-control/?utm_source=chatgpt.com)

**6. Texas Instruments – Bảo trì dự đoán và tối ưu hóa quy trình**

* Texas Instruments tích hợp AI vào sản xuất bán dẫn để bảo trì dự đoán và tối ưu hóa quy trình. [Musashi AI](https://musashiai.com/2025/05/07/ai-inspections-in-action-how-manufacturing-industry-leaders-are-elevating-quality-control/?utm_source=chatgpt.com)

**Phân tích chiến lược: Những điểm chung và khác biệt**

**✅ Điểm chung:**

* **Tự động hóa kiểm tra chất lượng**: Các công ty đều hướng đến việc tự động hóa quy trình kiểm tra chất lượng bằng AI để giảm thiểu lỗi và tăng hiệu suất.
* **Sử dụng học máy và thị giác máy tính**: Hầu hết các giải pháp đều tích hợp học máy và thị giác máy tính để phát hiện và phân loại lỗi.
* **Tích hợp dữ liệu và phản hồi**: Các hệ thống đều chú trọng vào việc thu thập dữ liệu và phản hồi để cải thiện mô hình AI liên tục.

**❌ Khác biệt:**

* **Mức độ tự động hóa**: Một số công ty như TSMC vẫn duy trì yếu tố con người trong quy trình (HITL), trong khi Intel và GlobalFoundries hướng đến tự động hóa hoàn toàn.
* **Phạm vi ứng dụng**: Các công ty khác nhau tập trung vào các giai đoạn khác nhau trong quy trình sản xuất, từ kiểm tra wafer đến kiểm tra PCB và hàn.[tsmc.com](https://www.tsmc.com/english/dedicatedFoundry/services/apm_intelligent_packaging_fab/intelligentFab_automation?utm_source=chatgpt.com)

**🚀 Cơ hội đột phá: Đề xuất giải pháp mới**

Dựa trên các phương pháp hiện tại, có thể thấy rằng việc tích hợp một hệ thống kiểm tra thông minh, tự động hóa hoàn toàn, sử dụng AI và thị giác máy tính, đặc biệt tập trung vào giai đoạn kiểm tra cuối cùng (Final Visual Inspection), sẽ là một bước đột phá. Hệ thống này không chỉ giúp phát hiện lỗi chính xác hơn mà còn tạo ra dữ liệu phong phú để huấn luyện mô hình AI, từ đó cải thiện chất lượng sản phẩm và hiệu suất sản xuất.

**GIAI ĐOẠN 2 : THIẾT KẾ GIẢI PHÁP**

PHẦN 7 : **Thiết Kế Quy Trình Mới: Vision AI + Smart Sampling**

Dựa trên phân tích điểm nghẽn tại công đoạn Final Visual Inspection,  tôi đề xuất một quy trình kiểm lỗi thông minh tích hợp Vision AI và Smart Sampling, nhằm thay thế hoàn toàn AQL sampling truyền thống.

**🎯 Mục tiêu hệ thống mới:**

* Tự động kiểm tra toàn bộ sản phẩm bằng thị giác máy tính
* Phát hiện lỗi theo mức độ nghiêm trọng: Critical / Major / Minor
* Ghi nhận toàn bộ dữ liệu hình ảnh và kết quả để đào tạo AI liên tục
* Hệ thống sampling thông minh: quyết định chọn mẫu sâu hơn hoặc bỏ qua dựa trên dữ liệu thực tế và tần suất lỗi

**Thành phần chính của quy trình mới:**

1. **Camera AI Vision Node (Edge Device)**
   * Camera công nghiệp gắn tại trạm kiểm tra cuối
   * Gửi hình ảnh đến model AI tại chỗ hoặc trên cloud
2. **AI Model – Vision Defect Detector**
   * Mô hình nhẹ như YOLOv8n / MobileNet
   * Phát hiện và phân loại lỗi real-time
3. **Smart Sampling Engine (Adaptive)**
   * Sử dụng feedback từ AI để:
     + Xác định tỉ lệ lỗi theo lô, theo thời điểm, theo máy
     + Ra quyết định kiểm tra kỹ hơn các cụm nguy cơ cao
     + Giảm kiểm tra với lô liên tục đạt chuẩn
4. **Data Feedback Loop & Training Set Builder**
   * Tự động lưu lại các mẫu ảnh, lỗi, phân loại
   * Cập nhật dữ liệu để huấn luyện lại mô hình định kỳ
5. **Dashboards & Alert System**
   * Hiển thị heatmap lỗi, phân bố theo thời gian, khu vực máy
   * Báo động khi lỗi vượt ngưỡng hoặc khi có xu hướng bất thường

**Luồng hoạt động tổng quát:**

1. Sản phẩm sau đóng gói được quét bởi camera
2. Hình ảnh gửi đến model Vision AI
3. AI nhận diện lỗi, đánh nhãn mức độ, và gửi kết quả về hệ thống trung tâm
4. Smart Sampling Engine phân tích kết quả để:
   * Đề xuất kiểm thêm sản phẩm cùng lô nếu lỗi cao
   * Tự động giảm sampling nếu chất lượng ổn định liên tục
5. Dữ liệu hình ảnh + kết quả được lưu để cải tiến mô hình AI
6. Dashboard báo cáo + Cảnh báo theo thời gian thực

**✅ Ưu điểm của quy trình mới:**

* **Toàn diện**: kiểm tra toàn bộ thay vì chỉ lấy mẫu
* **Thông minh**: sampling dựa trên dữ liệu thực, không theo bảng cố định
* **Tự học**: mô hình AI học từ dữ liệu thật phát sinh
* **Tích hợp**: phù hợp với xu hướng Smart Factory, dễ scale
* **Tiết kiệm chi phí nhân công và giảm lỗi lọt**

Quy trình này đặt nền tảng cho việc thay thế hoàn toàn mô hình kiểm lỗi xác suất AQL, giúp doanh nghiệp bán dẫn đạt được kiểm soát chất lượng theo thời gian thực, với khả năng phản ứng nhanh và học tập liên tục.

PHẦN 8 : **Phần Cứng Đề Xuất Cho Hệ Thống IoT Kiểm Lỗi Bằng AI**

**🎯 Mục tiêu phần cứng**

* **Thu thập hình ảnh** từ dây chuyền sản xuất
* **Xử lý tại chỗ (Edge)** hoặc **gửi lên server**
* **Tích hợp dễ dàng** với nhà máy hiện tại
* **Giá thành thấp, khả năng scale cao**

**Các Phương Án Tối Ưu Nhất (Xếp theo cấp độ)**

| **Cấp độ** | **Phần cứng đề xuất** | **Ưu điểm** | **Hạn chế** |
| --- | --- | --- | --- |
| **1. Cơ bản – tiết kiệm – linh hoạt** | 🎯 **Raspberry Pi 4 + Pi Camera HQ** | - Giá rẻ (~90–120 USD) - Dễ triển khai và mở rộng - Có thể chạy model YOLOv5n/MobileNet lite tại chỗ | - Giới hạn về FPS và độ phân giải nếu dùng nhiều camera - Không chạy tốt model lớn |
| **2. Trung cấp – chuyên xử lý hình ảnh** | 🎯 **NVIDIA Jetson Nano + Camera CSI** | - Hỗ trợ CUDA, TensorRT - Có thể chạy YOLOv8 real-time - Lý tưởng cho thị giác máy tính | - Giá cao hơn (~150–250 USD) - Hơi phức tạp khi triển khai số lượng lớn |
| **3. Cao cấp – công nghiệp hoá** | 🎯 **Jetson Xavier NX / AGX Orin + Camera công nghiệp** | - Hiệu suất AI cực mạnh - Dùng được mô hình lớn - Phù hợp môi trường công nghiệp nặng | - Giá cao (~500–900 USD/node) - Thừa tài nguyên nếu chỉ cần nhận diện cơ bản |
| **4. Siêu tiết kiệm – chỉ gửi ảnh (Edge Light)** | 🎯 **ESP32-CAM / Arduino + WiFi + Sensor** | - Siêu rẻ (~5–15 USD) - Chụp ảnh và gửi về trung tâm - Tốt để giám sát vị trí phụ | - Không xử lý AI tại chỗ - Độ phân giải thấp, dễ nhiễu |
| **5. Công nghiệp tích hợp** | 🎯 **Intel Movidius / Luxonis DepthAI** | - Tích hợp AI + camera + depth sensing - Có API Python dễ tích hợp - Đã dùng trong nhiều dây chuyền SMT/PCB | - Chi phí khoảng 250–400 USD - Phụ thuộc vào nguồn cấp ổn định |

[Camera Node] → [Edge Device: Pi/Jetson] → [Smart Sampling Engine]

↓

[WiFi/LAN/5G – optional]

↓

[Central Dashboard + Data Logger]