

Tratamiento de peligros y pizarra en un aula inteligente

Ingeniería de Mantenimiento de Computadores y Redes

Propuesta por Ivan

18 de febrero de 2026

1 Detección de peligros

- Riesgos en el aula
- Tecnología útil
- Infraestructura

2 Digitalización del Aula: Pizarra Inteligente

3 Conclusiones

1 Detección de peligros

- Riesgos en el aula
- Tecnología útil
- Infraestructura

2 Digitalización del Aula: Pizarra Inteligente

3 Conclusiones

Objetivo

Complementar sistemas reactivos basados en *hardware* dedicado con sistemas proactivos basados en **visión artificial** (en adelante, CV por *Computer Vision*).

- **Hardware tradicional:** Sensores unifuncionales (humos, humedad, PIR). Entradas activas: botón, palanca, lógica con enfoque más «síncrono».
- **Enfoque software/IA:** Uso de cámaras CCTV (cámaras de circuito cerrado, usadas para monitoreo de vídeo de seguridad) existentes + procesamiento en el borde (*Edge Computing*) o «nube».

1 Detección de peligros

- Riesgos en el aula
- Tecnología útil
- Infraestructura

2 Digitalización del Aula: Pizarra Inteligente

3 Conclusiones

¿Qué riesgos hay en un aula?

- **Seguridad:**

- Violencia física (peleas, acoso).
- Presencia de armas o bultos de objetos contundentes.
- Intrusos en horarios restringidos.

- **Integridad del entorno:**

- Incendios (análisis espectral y de movimiento).
- Inundaciones (análisis de textura en suelos).
- Bloqueo de salidas de emergencia (análisis espacial).
- Terremotos y actividad sísmica (temblores).
- Desprendimientos de objetos así como lanzamiento.
- Vandalismo (pintadas, comportamiento canalla...).

- **Salud y bienestar:**

- Caídas y desmayos.
- Detección de pánico o aglomeraciones.
- Contaminación acústica (ruido estridente, chillidos...).

1 Detección de peligros

- Riesgos en el aula
- **Tecnología útil**
- Infraestructura

2 Digitalización del Aula: Pizarra Inteligente

3 Conclusiones

- **Rápido y preciso.**
- Sensibilidad a oclusiones.
- **Navaja suiza:** Detección, segmentación, seguimiento, poses...
- Código abierto (OSS, *Open Source Software*).
<https://github.com/ultralytics/ultralytics>.
- Usado también para **atención de alumnos:**
<https://www.mdpi.com/1424-8220/25/22/6972>
- En **tiempo real** a través de cámaras IP a servidor local.
- Usado para detectar robo, violencia, vandalismo, fuego...
- Plausible en *hardware* limitado si se relajan restricciones.

- Rápida sturación con multitud.
- Eficiente al poder ejecutarse en **móviles** (pensado para sin GPU dedicada).
- Mejor captura de la profundidad.
- OSS.
- **Navaja suiza.**
- ¿Optimizado para Raspberry Pi? («*¿Gemini hablando bien de Google?*»).
- <https://github.com/google-ai-edge/mediapipe>.

- La violencia es un **acto temporal** → secuencia memoria.
- **YOLO** por si solo le cuesta → YOLO-Pose. Precisamos de una TPU si hacer en el *edge*.
- **YOLO-Pose/Google MediaPipe+LSTM** (*Long Short Term Memory*).
 - Clasificador temporal.
 - *Hardware* modesto → **privacidad**.
 - **Oclusiones y OSS**.
- **SlowFast**
 - Preciso. De Facebook. OSS.
 - Computación pesada.
 - Dos modos de funcionamiento: *slow* (¿quién es?), *fast* (movimiento)

Detección de intrusos

- **Objetivo:** saltar alarma si viene alguien no deseado en determinado horario.
- Cuádruple reto: **reconocimiento facial** + **horarios** + **lista blanca** + **privacidad**.
- Variantes:
 - ① Naive: hay personas \wedge fuera de horario \rightarrow alarma.
 - ② Reconocer caras \rightarrow ¿Privacidad? ¿Niños? ¿Ejercer derecho?. Necesidad de una DB actualizada \rightarrow lío legal.
 - ③ ¿Quién decide entrar y hasta dónde puede? (Absurdo no dejar entrar visita, familiares, servicios públicos, nuevo personal \notin DB) \rightarrow ¿1984?
- Merodeos: https://github.com/nwojke/deep_sort.
- <https://github.com/hectorpadin1/Network-Intrusion-Detection-System>.
- Otro enfoque, analizar la red:
<https://yardenfalik.github.io/IDS-Project/>. (Esto sale en la teoría de Sistemas Distribuidos).

Fuego

- **no tiene forma fija, «parpadea», brilla, reflja.**
- Puede producirse en **sitios ocultos** donde la cámara no vea.
- Versiones *custom* de YOLO.

⇒ Detección por IA como complementario para anticipar.

Agua

- **No se busca encontrar líquido** sino otras características.
- **Reflejos** (da igual si es sucia o transparente).
- Detección del suelo mojado píxel a píxel (U-Net, DeepLabV3+).
- Causas naturales → otros asuntos, meteo...

⇒ Interés en prevenir (tuberías rotas, fraturas...). Conocer aula concreta.

- Heurísticas sobre la pose del esqueleto.
 - ① Cambio en la **relación de aspecto**.
 - ② **Velocidad** de descenso (¿se sienta o se cae?).
 - ③ **Inactividad** (¿tropiezo o desmayo?).
- Otras situaciones.
 - Deporte → *fine tuning*, más movimiento.
 - *Trolling* → al menos, 18 añitos tiene la criatura.
- ¿Oclusión? Inferencia en base a lo visible.
- Tecnologías a gastar: YOLO, YOLO-pose, BoT-SORT y ByteTrack.

Audio *transformers*

-
-
-
-
-
-
- :

No todo se detecta mejor con IA

- **Fuegos, inundaciones.**
- **Sismos** (ESP32 + MPU-6050): <https://github.com/serdaraltin/earthquake-warning-system>.

- **Fuego:** <https://github.com/MuhammadMoinFaisal/FireDetectionYOL0v8>.
- **Armas:**
<https://www.nature.com/articles/s41598-025-07782-0>,
<https://github.com/swatified/Weapon-Detection-System>.
- <https://www.youtube.com/watch?v=a2xWqkFDYuU>
- <https://www.youtube.com/watch?v=D4mjEBgAXPU>
- <https://www.youtube.com/watch?v=qeFrjFa5Rxc>
- <https://www.youtube.com/watch?v=vEtsmg7-fWs>
- **Ataque de pánico, aglomeraciones:**
<https://github.com/moego0/panic-attack-detector>,
https://github.com/jinay-k-jain/Panic_detector_CCTV.

1 Detección de peligros

- Riesgos en el aula
- Tecnología útil
- **Infraestructura**

2 Digitalización del Aula: Pizarra Inteligente

3 Conclusiones

¿Qué tenemos a disposición?

- 600€ de presupuesto en nuevo *hardware*.
- Electrónica de otros años.
 - Placas de Arduino.
 - Raspberry Pi.
 - ESP32.
- Ordenadores del aula.
 - GPU: Nvidia GTX 1660.

¡Relajar restricciones!

- 1 Una cámara → un propósito.
- 2 Reducir FPS → no tiempo real.
- 3 Procesamiento centralizado (SPOF). ¿cómo vigilancia en urbanización?
- 4 Combiando estrategias (o aplicando divide y vencerás para el reparto de responsabilidad): *edge* detecta que hay personas, *server* averigua quienes son.

- 1 Detección de peligros
- 2 Digitalización del Aula: Pizarra Inteligente**
- 3 Conclusiones

Captura Inteligente de Contenido

El sistema utiliza una cámara cenital o frontal para digitalizar la pizarra analógica automáticamente.

Disparadores (Triggers)

La captura se activa mediante lógica de software:

- **Botón Físico:** Ad-hoc por el profesor al terminar una explicación.
- **App Estudiante:** Petición individual vía WebSocket.
- **Modo "Ad Quorum":** Se dispara solo si el $X\%$ de los alumnos lo solicita en 30 segundos.

Detección Automática

Mediante **Background Subtraction**, el software detecta cuando la pizarra está "llenaz el profesor se aparta, disparando la foto automáticamente.

Uno de los retos del software en aulas es la privacidad de los alumnos y el profesor.

Pipeline de Anonimización:

- ➊ **Detección de Rostros:** Uso de modelos ligeros (ej. MTCNN o Haar Cascades) para localizar caras en la imagen de la pizarra.
- ➋ **Aplicación de Blur (Difuminado):** Aplicación de filtro Gaussiano sobre las regiones de interés (ROIs) detectadas.
- ➌ **Rectificación de Perspectiva:** Algoritmo de homografía para transformar la foto lateral en una imagen plana ".escaneada".
- ➍ **Mejora de Contraste:** Binarización adaptativa para convertir trazos de tiza/rotulador en digital nítido.

- 1 Detección de peligros
- 2 Digitalización del Aula: Pizarra Inteligente
- 3 Conclusiones

Resumen de la Propuesta

- La infraestructura se basa en **software** sobre hardware genérico (cámaras), reduciendo costes de instalación.
- La IA permite una detección de peligros más rápida y contextual (ve el fuego, no solo huele el humo).
- La digitalización de la pizarra respeta la privacidad (blurring) y democratiza el apunte (quorum).

¿Preguntas?