Proyecto de IA

Nombre del Proyecto: Detección de Violencia Física Escolar en Tiempo Real
Objetivo Principal: Desarrollar un software basado en inteligencia artificial para detectar comportamientos violentos o potencialmente violentos entre estudiantes en entornos escolares, utilizando video en tiempo real. El software identifica tensiones previas a peleas y agresiones físicas y distingue estas situaciones de convivencia normal (estudiantes caminando, hablando, en pasillos, en fila o jugando pacíficamente).

Contexto: El proyecto se centra en la vigilancia escolar para prevenir incidentes de violencia, con un enfoque en la detección temprana de comportamientos que puedan escalar a agresiones físicas. Está diseñado para ser implementado en un entorno escolar, utilizando cámaras de vigilancia (como cámaras IP o otras tecnologias) para capturar video en tiempo real.

Contexto detallado sobre el proyecto:

Proyecto consiste en desarrollar un software con modelos de IA para prevención de violencia escolar violencias como peleas, agresiones físicas.

El software debe funcionar de las siguiente manara general: Debe haber una cámara de seguridad que esté instalada, la cámara estará funcionando en vivo en tiempo real en tanto en los patios y pasillos, entonces a través de eso se detectará violencia si en caso ocurre, si en caso de que haya violencia el modelo debe predecir su hay violencia de estudiante peleando entonces debe alertar al personal de seguridad para prevenir la violencia, la forma de alerta será a través de un dispositivo sonara que está instalado en las oficinas de los directores u oficina de dirección que sonará cuando ocurra una violencia o agresión física y también se debe llegar o notificar sobre el hecho a la aplicación web que tendrá el sistema, el sistema contará con una aplicación web para poder monitorear o gestionar todos los casos de violencia que ocurre en el instituto, donde podrán monitorear la cámara que están detectando en tiempo real y visualizar las cámaras, además en la aplicación podrán gestionar las historiales de los hechos, sacar informes, etc. En la aplicación web también se deben poder registrar de manera automática el hecho de la violencia con los datos esto para poder tener pruebas y evidencias de los casos hechos datos como ubicación del hecho, fecha, hora, pasillo, y un video del momento hecho esto para ver las pruebas en videos registrados.

En resumen, que a través de las cámaras de seguridad el modelo de IA que vamos a entrenar debe detectar en tiempo real si se presenta violencia o no y si se presenta violencias como peleas se debe alertar de manera instantánea mediante una notificación a la aplicación web que esta en la administración y a la vez debe sonar la alarma de sonara y poder tener un altavoz que diga como ejemplo: “Alerta

de violencia en el pasillo Nro. 2", Y la vez se deben poder registrar los datos de manera automática en la base de datos esto si ocurre una violencia para poder tener registros y pruebas de los casos hechos en la institución. El objetivo es prevenir violencia escolar mediante una Inteligencia Artificial donde debemos entrenar modelos de IA a base de YOLOv11 para detección de las personas y TimesFormer basado en Transformers para la detección de violencia en tiempo real. Solo se entrenarán dos modelos.

Tecnologías Utilizadas:

- **Modelos de IA**:
    - **YOLOv11**: Para detección de personas en los frames de video. Identifica y localiza a los estudiantes en tiempo real.
    - **DeepSort**: Para seguimiento de personas a lo largo de los frames, asignando IDs únicos a cada estudiante detectado y rastreando sus movimientos.
    - **TimeSformer (transformers import AutoImageProcessor, TimesformerForVideoClassification)**: Para clasificar comportamientos violentos o no violentos analizando clips de video de 3 a 6 segundos. Clasifica secuencias de video en categorías de violencia (como amenazas o tensiones) o no violencia (convivencia normal).
- **Infraestructura**:
    - **Para el backend :** Backend para manejar la comunicación en tiempo real entre el servidor y el frontend, sirviendo frames procesados y eventos de detección.
    - Base de datos para registrar eventos de violencia detectados, incluyendo timestamps, frames y videos involucrados, probabilidad de violencia e IDs de los estudiantes implicados y otros datos importantes.
- **Librerías**:
    - OpenCV para procesamiento de video.
    - PyTorch y ONNX Runtime o otros para ejecutar los modelos en GPU.
    - React para el frontend, mostrando el video procesado con bounding boxes y alertas de violencia, transmitimos la cámara en tiempo real en la aplicación web.

Flujo de Trabajo:

1. **Captura de Video**: El software captura video en tiempo real desde una cámara.
2. **Detección de Personas**: YOLOv11 detecta personas en cada frame, identificando sus posiciones mediante bounding boxes.
3. **Seguimiento de Personas**: DeepSort rastrea a las personas detectadas, asignando un ID único a cada estudiante y siguiendo sus trayectorias a lo largo del video.
4. **Clasificación de Comportamientos**: TimeSformer (**transformers import AutoImageProcessor, TimesformerForVideoClassification**) analiza clips de 3 a 6 segundos con secuencias de frames y FPS para clasificar si hay violencia o no, basándose en un umbral de probabilidad (0.6).
5. **Visualización y Alertas**:
    - El video procesado se muestra en el frontend con bounding boxes alrededor de los estudiantes, indicando sus IDs.
    - Si se detecta violencia, se muestra un mensaje en el frontend ("Violencia: Sí") con la probabilidad y los IDs de los estudiantes involucrados.
    - Los eventos de violencia se registran en la BD para análisis posterior.
6. **Interacción en Tiempo Real**: El usuario puede iniciar o detener la detección desde el frontend, y los frames procesados se envían continuamente al frontend.

Dataset: Estás armando un dataset de videos con dos categorías:

- **Amenazas y Tensiones**: Videos de tensiones previas a peleas agresiones físicas.
- **Convivencia Normal**: Videos de estudiantes caminando, hablando, en pasillos, en fila o jugando pacíficamente.

Pipeline sobre le Proyecto o flujo de funcionamiento del software

Fase 1: Captura de video:

- Se utilizará una cámara de seguridad, la cámara capturara el video en tiempo real.
- Las herramientas que se utilizarían son una cámara como: (App webcam IP, otras tecnologías que existan para esto.), OpenCV
- La cámara capturará el video en tiempo real para procesarlo.
- La cámara capturará el video con una base de 1280x720 por defecto para que sean procesados.

Fase 2: Procedimiento del video:

- Herramientas como OpenCV y otros para procesar los videos
- El proceso que se debe hacer es que se debe extraer frames de la cámara y extraerlo a frames de 640x640 para el Modelo de Detección de personas con YOLOv11 que trabaja con esa dimensión para el procesamiento y DeepSort
- Para el modelo de TimesFormer para la detección de violencia se debe procesar el video capturado a clips de 224x224 de resolución que es que el modelo trabaja.
- Y la salida después de ser procesados son frames individuales de 640x640 y clips procesados 22x224 con FPS de 15.

Fase 3: Procedimiento con los modelos de IA

- Herramientas a utilizar son Ultralytics Yolo, DeepSort, PyTorchVideo TimesFormer (**transformers import AutoImageProcessor, TimesformerForVideoClassification**)
- Primero esta el modelo de detección de personas en tiempo real esto con el modelo de base YOLOv11 y el otro modelo es el DeepSort para rastrear a las personas asignado los IDs a cada persona esto para identificar que personas están involucradas en el evento.
- Detección y rastreo de personas en los frames en tiempo real, el primer proceso es detectar personas generando sus bounding boxes con YOLO, el modelos debe dar una salida en la de sus cordenas, estamos aplicando detección de objetos.

  "Detectar y rastrear personas en frames"

  - Proceso: Detectar personas (conf > 0.6), generar

  bounding boxes

- Salida: [x, y, w, h, confl por frame

- Depues entra el DeepSort para rastrear alas personas asignando IDs a base de las entradas de los Bounding Boxes de YOLO debe asignar los IDS únicos rastrear trayectorias su salida debe se rel ID, sus coordenadas de los frames.
  - DeepSORT:
  - Entrada: Bounding boxes de YOLO
  - Proceso: Asignar IDs únicos, rastrear trayectorias
  - Salida: [ID, x, y, w, h] por frame
- A la ves en paralelo esta el otro modelo que detecta la violencia en tiempo real a base del modelo de TimesFormer quien procesará el videos atraves de la cámara en tiempo real y predicirá la violencia. Como entrada tiene el video o clips de 224x224 de 15FPS procesará el videos para poderlos clasificar, su salida es su etiqueta, probabilidad de acción.
  "Clasificar violencia en clips"
  - Entrada: Clips (224x224, 15 FPS, 5s)
  - Proceso: Muestrear frames, clasificar
  "violence" o "no_violence" (conf > 0.7)
  - Salida: [etiqueta, probabilidad] por clip
- Estos modelos funcionan en paralelo de la siguiente manera el proceso es: Cuando el modelo detecta una situación de violencia en tiempo real entonces en ese evento se debe registrar las personas mediante IDs para saber que personas están involucradas en el evento y los datos o respuestas se deben de mostrar en la pantalla de cámara.
- **Integración y lógica de eventos]**
- **'Unir resultados de IA para generar eventos"**
- **Proceso:**
- **Sincronizar. Mapear IDs de DeepSORT al clip de TimeSformer por timestamps**
- **Decidir: Si "violence" y hay IDs Generar evento**
- **Enriquecer. Añadir metadata (timestamp, ubicación)**
- **Salida: Evento (ej. "Violencia en pasillo, IDs 1 y 2, 10:15:23")**

Fase 4: Generación de respuestas y alertas

"Activar sonido y voz para alertar"

Con una alarma sonora

- Herramientas: Pyttsx3, altavoces

- Proceso: Reproducir tono +

"Alerta de violencia en el pasillo"

- Salida: Audio en tiempo real

Mediante una Notificación:

[5.3 Notificaciones]

"Informar a aplicación web de los directivos en la oficina"

- Herramientas: SMTP, Twilio, etc. otros

- Proceso: Enviar notificación (ej.

"Violencia en pasillo, IDs 1 y 2")

- Salida: Mensaje enviado

Fase 5: de Monitoreo y FeedBack

- Base de datos – historial de incidentes
  "Guardar evidencia en base de datos"
  Herramientas cualquier db:
  Proceso: Almacenar clip, frames con IDs, metadata
  Salida: Registro en DB

- Aplicación Web (Monito eo y Gestión de Seguridad):

  "Proporcionar interfaz para monitoreo y revisión"

  - Herramientas: React (frontend) y para el backend lo mejor que exista.

  - Backend: Hacer eventos, tecnologías para video en vivo

  - Frontend: Mostrar streams, tabla de eventos, clips y frames con IDs

Fase 6 de Despliegue:

"Implementar y mantener el sistema en producción"

- Herramientas: Docker, AWS/GCP, logging

- Proceso:

- Contenerizar. IA, web, DB en contenedores Docker

- Desplegar. Servidor local o nube

- Monitorear: Registrar FPS, latencia, errores

- Salida: Sistema operativo 24/7

Estructuras de los dataset para detección de personas y violencia:

Dataset para detección de personas YOLOv11: El dataset cuenta con imágenes de 640x640 ya procesados listo para entrenar, en total son 13900 imágenes con sus labels que son las anotaciones bouding boxes remarcados en txt. Ya esta divido 80% para train y 20% para val y 10% para test, el dataset ya está todo procesado listo para entrenar, su estructura esta así:

```
dataset_people/
    images/
        train/
            people_001.jpg   // 10000 images
            people_002.jpg
            ......
        val/
        test/
    lables/
        train/
            people_001.txt
            people_002.txt
            ......
        val/
        test/

    data.yaml
```

Los bouding boxes en txt están sus etiquetas como ejemplo:

0 0.532898 0.592734 0.110391 0.297594

0 0.379297 0.619938 0.110406 0.326406

0 0.463297 0.603133 0.096000 0.286391

En el archivo data.yaml

train: ./images/train

val: ./images/val

test: ./images/test

nc: 1

names:

- persona

En resumen, el dataset ya está procesado listo para entrenarlo.

Dataset para detección de violencia en tiempo real TimesFormers (transformers import AutoImageProcessor, TimesformerForVideoClassification): El dataset ya está procesado listo para entrenar, contiene videos de 3 a 6 segundos, en total son 10300 videos, los videos ya están procesado con una resolución de 224x224 que el modelo trabaja así, además todos los videos tienen 15 de FPS, el dataset tiene dos clases como no_violencia, violencia. El dataset ya esta dividido tanto para train, val y test, El dataset tiene la siguiente estructura:

```
dataset_violencia/
    train/
        no_violence/
            no_violencia_001.mp4  // 3 a 6 segundos
            no_violencia _002.mp4
            .....
        violence/
            violencia_001.mp4  // 3 a 6 segundos
            violencia_002.mp4
            ......
    val/
        no_ violence /
        violence /
    test/
        no_ violence /
        violence /
```

En resumen, el dataset ya está procesado listo para entrenar

En RESUMEN

🧠 **Título del Proyecto**

“Seguridad y Prevención de Violencia Física en Instituciones Educativas mediante Inteligencia Artificial”

🎯 **Objetivo General**

Desarrollar un **software con modelos de inteligencia artificial capaces de detectar violencia física en tiempo real** en entornos escolares, alcanzando una precisión mínima del 90% y reduciendo el tiempo de respuesta ante incidentes en un 70%.

📌 **Objetivos Específicos**

1. Analizar el estado del arte sobre violencia escolar y su detección mediante IA.
2. Investigar técnicas de visión por computadora y aprendizaje profundo (CNNs, Transformers).
3. Construir e integrar modelos de IA para detectar y clasificar agresiones físicas.
4. Validar y optimizar el sistema en entornos simulados y reales.
5. Implementar una aplicación web para monitoreo y gestión de alertas.

🔍 **Problemática Detectada**

- Las peleas o agresiones físicas escolares no son atendidas a tiempo.
- La vigilancia tradicional es reactiva (se revisan las cámaras después del hecho).
- El personal escolar no tiene una forma efectiva de intervenir de inmediato.

💡 **Solución Propuesta**

Desarrollar un sistema basado en visión por computadora e IA que:

- Detecte personas (con **YOLOv11**).
- Las rastree (con **DeepSORT**).
- Clasifique violencia (con **TimeSformer**).
- Genere **alertas inmediatas** (sonoras y en una aplicación web).
- Registre automáticamente incidentes en una **base de datos** con evidencias.

🛠️ **Componentes del Sistema**

1. Modelos de IA

- **YOLOv11/YOLOv8n**: Detección de personas en video (640x640).
- **DeepSORT**: Seguimiento y asignación de IDs a personas.

- **TimeSformer**: Clasificación de violencia en clips (224x224, 15 FPS).

2. Infraestructura

- Backend: Manejo de video, lógica de eventos, base de datos.
- Frontend: Interfaz React para monitoreo y gestión.
- Sonido: Alarma sonora y mensaje por altavoz (usando pyttsx3, etc.).
- Notificaciones: A través de medios como Twilio o notificaciones locales.
- Despliegue: Docker, servidores locales o en la nube (AWS/GCP).

📜 **Funcionalidades del Software**

- Visualización en tiempo real del video con **bounding boxes** e **IDs**.
- Alerta sonora y mensaje por voz (ej: “Alerta de violencia en el pasillo 2”).
- Registro automático del evento: clip de video, IDs, ubicación, hora, etc.
- Acceso histórico de incidentes con opción de informes.
- Capacidad de monitorear desde oficina o sala de administración escolar.

🍼 **Datasets**

Ambos datasets están **ya procesados y listos para el entrenamiento**:

➤ **Detección de personas (YOLOv11)**

- 13,900 imágenes (640x640) anotadas.
- Estructura en train, val y test.
- Etiquetas en formato YOLO (.txt con bounding boxes).

➤ **Detección de violencia (TimeSformer)**

- 10,300 videos (3–6 s, 224x224, 15 FPS).
- Dos clases: violencia y no_violencia.
- Estructura organizada en train, val y test.

🔁 **Flujo General del Sistema**

1. **Captura del video en tiempo real**.
2. **Detección de personas (YOLO)**.
3. **Seguimiento (DeepSORT)**.
4. **Clasificación de clips de video (TimeSformer)**.
5. **Sincronización y generación de evento de violencia**.
6. **Alertas (sonido + voz)**.
7. **Notificación a aplicación web y registro en base de datos**.
8. **Interfaz web para monitoreo en tiempo real e histórico**.