

Detección de escenas y cortes automáticos con IA

Tecnologías Emergentes e Impacto Social | Inteligencia Artificial

Descripción del Curso

Este curso ofrece un marco práctico para aplicar Inteligencia Artificial en contextos de procesamiento de video, con foco en el desarrollo de prototipos reproducibles y útiles para resolver problemas reales. Aunque está estructurado por unidades, la propuesta general busca combinar fundamentos teóricos con implementación concreta, permitiendo a los estudiantes traducir ideas de diseño en soluciones funcionales. En particular, la Unidad 3 se centra en la implementación de un prototipo básico con IA para la detección de escenas y la generación de cortes. La unidad propone trabajar con herramientas de IA como PyTorch o TensorFlow y con OpenCV para capturar y procesar video, detectar cambios de escena y derivar segmentos de interés. Se ejecutará el prototipo en videos de muestra, se documentarán las configuraciones utilizadas y se analizarán los resultados obtenidos para evaluar la viabilidad y la reproducibilidad de la solución. El objetivo es convertir un diseño conceptual en una solución operativa y reproducible, que pueda ser extendida o adaptada a otros conjuntos de videos y escenarios. El enunciado de la Unidad 3 especifica: “Implementar un prototipo básico que utilice herramientas de IA (p. ej., PyTorch/TensorFlow y OpenCV) para detectar escenas y generar cortes, ejecutarlo en videos de muestra y documentar las configuraciones utilizadas.” Sus objetivos específicos orientan a: (1) desarrollar una versión inicial que integre adquisición de video, extracción de características, inferencia y generación de cortes; (2) probar el prototipo en videos de muestra registrando resultados y métricas simples de detección (por ejemplo, recortes generados y intervalos entre cortes); (3) documentar configuraciones, dependencias y pasos para reproducir el prototipo, promoviendo la trazabilidad y la reutilización. En suma, el curso busca construir capacidades técnicas y de documentación que permitan al estudiante transformar conceptos de IA aplicada en prototipos funcionales y compartibles.

Competencias

- Comprender conceptos fundamentales de IA aplicados al procesamiento de video y visión por computadora.
- Diseñar y desarrollar prototipos que integren adquisición de video, extracción de características, inferencia y generación de cortes.
- Implementar pipelines que utilicen PyTorch o TensorFlow junto con OpenCV para tareas de detección de escenas.
- Evaluar resultados de manera básica mediante métricas simples y registrar observaciones para mejorar el prototipo.
- Documentar configuraciones, dependencias y pasos necesarios para reproducir el prototipo, promoviendo trazabilidad y reutilización.
- Trabajar de forma colaborativa y comunicar de manera clara los resultados técnicos y las decisiones de diseño.

Requerimientos

- Conocimientos previos de programación en Python.

- Conocimientos básicos de IA y visión por computadora.
- Entorno de desarrollo Python 3.x (virtualenv o conda) y gestión de dependencias.
- Instalación de PyTorch o TensorFlow y OpenCV, así como otras dependencias necesarias para el prototipo.
- Acceso a una computadora con recursos razonables de procesamiento; se recomienda GPU para acelerar inferencia, aunque es posible ejecutar en CPU.
- Acceso a videos de muestra para realizar pruebas y ejercicios prácticos.
- Habilidad para documentar configuraciones, pasos de reproducción y resultados experimentales.

Unidades del Curso

Unidad 1: Unidad 1: Fundamentos de detección de escenas y cortes automáticos con IA

Objetivos de Aprendizaje

1. Definir qué es una escena, qué es una transición y qué es un corte en material audiovisual y distinguir sus características principales.
2. Explicar, con ejemplos, cómo la detección de cambios de escenario facilita la edición automática y la generación de resúmenes o clips temáticos.
3. Identificar componentes básicos y criterios de evaluación para un pipeline conceptual de detección de escenas.

Contenidos Temáticos

1. Conceptos clave: escena, transición y corte.
Definición de cada concepto y ejemplos visuales para distinguirlos en videos reales.
2. Visualización de cambios de escena y técnicas básicas de detección.
Idea general de cómo se detectan cambios entre tomas y cuáles señales pueden indicar un corte o una transición.
3. Relevancia en la edición automatizada.
Justificación de por qué la detección de escenas ayuda a crear clips automáticos, resúmenes y formatos adaptados.

Actividades

1. **Actividad: Exploración conceptual** – Analizar ejemplos de videos cortos para identificar escenas, transiciones y cortes. Tema de aprendizaje: comprensión de conceptos y diferencias. Puntos clave: definiciones, ejemplos, diferencias prácticas. Aprendizajes: capacidad de clasificar segmentos y justificar su selección en edición automática.
2. **Actividad: Análisis de clips de muestra** – Etiquetar manualmente escenas y cortes en 3 clips proporcionados y justificar las decisiones. Puntos clave: criterios de detección, consistencia entre clips. Aprendizajes: criterios de validación y revisión crítica de resultados.

Evaluación

Evaluación formativa centrada en el objetivo general de la unidad: - Cuestionario corto sobre conceptos clave (definiciones, diferencias entre escena, transición y corte). - Ejercicio de análisis de clips con justificación de las transiciones detectadas. - Participación y aportes en discusiones de clase.

Unidad 2: Unidad 2: Diseño de pipeline conceptual de detección de escenas y cortes

Objetivos de Aprendizaje

1. Identificar y describir cada bloque funcional del pipeline: adquisición, preprocesamiento, extracción de características, inferencia y generación de cortes.
2. Explicar la interconexión entre bloques y justificar las decisiones de diseño mediante criterios de rendimiento y robustez.
3. Elaborar un diagrama de flujo de alto nivel que represente el pipeline y preparar una breve justificación de cada componente.

Contenidos Temáticos

1. Arquitectura de un pipeline de detección de escenas.
Visión general de los módulos y sus responsabilidades dentro del flujo de trabajo.
2. Adquisición de video y preprocesamiento.
Fuentes de video, formatos, normalización, y preparación de frames para el análisis.
3. Extracción de características, inferencia y generación de cortes.
Conceptos de características (frame-level, temporal) y modelos (basados en IA) para detectar cambios y generar cortes.
4. Diagrama de flujo y justificación de componentes.
Representación gráfica del pipeline y argumentos para cada decisión de diseño.

Actividades

1. **Actividad: diseño de un diagrama de flujo** – Crear un diagrama de flujo de alto nivel para un pipeline de detección de escenas y explicar la función de cada bloque. Puntos clave: claridad, consistencia entre bloques, casos de uso, y criterios de rendimiento. Aprendizajes: habilidad para estructurar soluciones complejas y justificar decisiones de diseño.
2. **Actividad: especificación de preprocesamiento** – Describir y justificar las técnicas de preprocesamiento (rescalado, normalización, manejo de frame drops) adecuadas para la detección de escenas. Aprendizajes: selección de técnicas adecuadas a datos y efectos en la precisión.
3. **Actividad: análisis de flujo de datos** – Evaluar cómo la información se transmite entre bloques del pipeline y proponer mejoras para reducir latencia y aumentar robustez.

Evaluación

Evaluación basada en el objetivo general 2: - Entrega de un diagrama de flujo de alto nivel acompañado de una breve justificación escrita de cada componente. - Informe de análisis de preprocesamiento y elección de características con criterios de rendimiento. - Participación en revisión de pares y defensa de decisiones de diseño.

Unidad 3: Unidad 3: Implementación de prototipo básico con IA para detección de escenas y generación de cortes

Objetivos de Aprendizaje

1. Implementar una versión inicial del prototipo que integre adquisición de video, extracción de características, inferencia y generación de cortes.
2. Ejecutar el prototipo en videos de muestra y registrar resultados, incluyendo métricas simples de detección (p. ej., recortes generados, intervalos entre cortes).
3. Documentar configuraciones, dependencias y pasos para reproducir el prototipo, promoviendo la trazabilidad y reutilización.

Contenidos Temáticos

1. Configuración del entorno y herramientas.
Instalación de Python, PyTorch/TensorFlow, OpenCV y dependencias, manejo de entornos y versiones.
2. Detección de escenas: enfoque práctico con IA y visión computacional.
Ejemplos simples como comparación de frames, extracción de características y umbrales, con posibilidad de extender a modelos ligeros.
3. Generación de cortes y exportación de resultados.
Conversión de detecciones en una lista de cortes o un archivo de decisiones de edición (EDL), exportación de resultados y documentación de salida.
4. Documentación y reproducibilidad.
Guía de uso, requerimientos, ranuras de configuración y ejemplos de ejecución para facilitar la repetición.

Actividades

1. **Actividad: configuración del entorno** – Configurar un entorno con Python, PyTorch/TensorFlow y OpenCV; verificar versiones y crear un script de prueba simple para leer un video. Aprendizajes: manejo de dependencias, reproducibilidad básica.
2. **Actividad: prototipo de detección de cambios** – Implementar un detector de cambios entre frames utilizando diferencias de histogramas o embeddings simples y generar una lista de cortes. Puntos clave: umbral, suavizado de detecciones, robustez ante iluminación.
3. **Actividad: ejecución en videos de muestra** – Correr el prototipo en 1-2 videos de muestra, registrar tiempos de procesamiento, cantidad de cortes y crear un pequeño informe de resultados.

4. **Actividad: documentación de configuraciones** – Preparar un README con dependencias, pasos para reproducir, cualquier hiperparámetro utilizado y consideraciones de rendimiento. Aprendizajes: buenas prácticas de desarrollo y documentación.

Evaluación

Evaluación centrada en el objetivo general 3: - Demostración funcional del prototipo en videos de muestra. - Calidad y claridad de la documentación de configuraciones y reproducibilidad. - Informe de resultados con métricas básicas y reflexión sobre limitaciones y posibles mejoras.