

Universidad Rafael Landívar
Inteligencia Artificial
Primer Semestre 2025

PROYECTO FINAL: Reconocimiento de Movimientos

Descripción General

El objetivo de este proyecto es aplicar los conceptos de **Visión por Computadora y Aprendizaje Automático** para desarrollar un sistema capaz de reconocer gestos manuales y utilizarlos para controlar dispositivos o traducir lenguaje de señas a texto. Los estudiantes podrán elegir entre las siguientes opciones:

1. **Control de Dispositivos mediante Gestos:** Crear un sistema donde los gestos de la mano activen acciones específicas, como cambiar diapositivas, controlar el volumen, controlar un vehículo, motor, circuito o pausar/reproducir música.
 - o **Dataset recomendado:** Se pueden capturar imágenes propias o utilizar **MediaPipe Hands**.
2. **Traducción de Lenguaje de Señas a Texto:** Diseñar un modelo que detecte señas del lenguaje de señas y las convierta en texto en tiempo real.
 - o **Dataset recomendado:** ASL Dataset o capturas personalizadas.
3. **Reconocimiento de Expresiones Faciales para Feedback en Clases:** Crear un sistema que, a través de una cámara, detecte las expresiones faciales de los estudiantes durante la clase (por ejemplo, si están atentos, distraídos, confundidos, etc.) y proporcione recomendaciones en tiempo real. Por ejemplo, si se detecta que los estudiantes están cansados, el sistema podría sugerir una pausa; si están distraídos, podría recomendar realizar una actividad interactiva; y si muestra expresiones de confusión sugiera contenido adicional. El sistema debe enviar recomendaciones al catedrático a través de una plataforma de mensajería.
 - o **Dataset recomendado:** EmoReact Dataset, FER2013, o capturas personalizadas.

El proyecto debe desarrollarse usando técnicas de **Machine Learning y/o Deep Learning**, aplicando procesamiento de imágenes y modelos de clasificación.

Objetivos Específicos

- Capturar y procesar imágenes de gestos con una cámara en tiempo real.
- Aplicar técnicas de preprocesamiento de imágenes (segmentación, detección de contornos, extracción de características).
- Entrenar un modelo de clasificación (SVM, Random Forest, y/o RNA).
- Evaluar el modelo con métricas como precisión, recall y F1-score.
- Implementar una interfaz que permita al usuario interactuar con el sistema.
- Integrar el modelo de IA con una aplicación práctica.

Producto a Entregar

Fecha de entrega y presentación: 19/05/2025 en clase.

Valor: 15 puntos netos.



Cada grupo deberá entregar los siguientes componentes:

1. Repositorio del Proyecto (GitHub o similar)
Debe contener:
 - Código fuente completo.
 - Dataset utilizado (o enlace al dataset si es muy pesado).
 - Archivo **README.md** con descripción del proyecto, instalación y ejecución.
 - Modelo entrenado guardado.
 - Archivos de la aplicación web o interfaz gráfica.
2. Documento (PDF)
Debe incluir:
 - Introducción y motivación del problema.
 - Definición del problema y objetivos (generales y específicos).
 - Descripción del dataset utilizado.
 - Explicación del preprocesamiento aplicado.
 - Implementación del modelo (justificación del algoritmo elegido).
 - Evaluación del modelo con métricas y análisis de resultados.
 - Diagramas:
 - Arquitectura de la solución (motor de inferencia + interfaz).
 - Casos de uso.
 - Flujo general del sistema.
 - Componentes y secuencia de interacción.
 - Evidencias de funcionamiento (capturas de pantalla, resultados del modelo en acción).
 - Conclusiones y aprendizajes.
3. Video de Presentación (5 minutos máximo)
 - Cada grupo debe grabar un video explicando el funcionamiento del sistema.
 - Debe mostrar el código, la página web, y varios ejemplos prácticos.
 - Explicar brevemente el proceso de preprocesamiento, entrenamiento y evaluación.

Requisitos Técnicos

1. Motor de Inferencia (Backend)
 - Lenguaje: Python.
 - Frameworks recomendados: TensorFlow/Keras, PyTorch, Scikit-Learn, OpenCV, MediaPipe.
 - Se debe entrenar un modelo basado en imágenes de gestos.
 - Dataset: Pueden seleccionar uno de los siguientes datasets recomendados o capturar el propio: ASL, MediaPipe Hands.
2. Interfaz Web (Frontend)
 - Opciones: Web App con Flask/Django, aplicación móvil (Android o IOS) o Aplicación en OpenCV con Tkinter.
 - Debe permitir:
 - Capturar gestos en tiempo real.

- Enviar datos al backend para inferencia.
- Mostrar/enviar/notificar el resultado de la predicción.

Observaciones Adicionales

- Plagio Cero: Cada entrega será revisada contra otros grupos y entregas previas. Proyectos con coincidencias elevadas serán anulados.
- El trabajo debe reflejar un equilibrio de responsabilidades entre los miembros del grupo.
- El enfoque es práctico y aplicado, por lo que la calidad del código y la claridad de la explicación son clave para una buena calificación.
- En grupos de 3 estudiantes. Se valorará la creatividad en la implementación.

Rúbrica de Evaluación

Criterio	Descripción	Ponderación
1. Documento del Proyecto	Explicación clara, importancia del problema, diagramas, análisis técnico.	10 pts
2. Preprocesamiento de datos	Segmentación de imágenes, extracción de características, uso de OpenCV.	15 pts
3. Implementación del Modelo	Entrenamiento correcto usando ML o DL.	15 pts
4. Evaluación del Modelo	Uso de métricas como precisión, recall, matriz de confusión.	10 pts
5. Guardado y Carga del Modelo	Se debe entregar el modelo entrenado en un archivo. Pickle, HDF5 u otro.	5 pts
6. Desarrollo de la Interfaz	Aplicación funcional para interactuar con el modelo.	10 pts
7. Integración Modelo + Interfaz	Comunicación fluida entre backend y frontend. Sugerencias y retroalimentación que entregue el modelo al usuario.	10 pts
8. Documentación Técnica	Incluir README con instalación, uso y arquitectura.	10 pts
9. Creatividad y Presentación	Diseño atractivo, interfaz intuitiva, implementación novedosa.	10 pts
10. Presentación Final	Exposición clara, justificación técnica y participación de todos los integrantes.	5 pts