INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

Inteligencia artificial avanzada para la ciencia de datos II

TC3007C - Gpo 501

Deep Learning

Integrantes:

Jorge Ignacio Reyes Pérez - A00573981

Fecha de entrega:
1 de Diciembre del 2025

Introducción

El reconocimiento automático de acciones humanas en video es una de las tareas más relevantes en visión por computadora moderna. Sus aplicaciones abarcan vigilancia inteligente, análisis deportivo, interacción persona-máquina, salud, y sistemas autónomos. En este proyecto se implementó un modelo de deep learning capaz de clasificar acciones humanas utilizando el conjunto de datos UCF101, un benchmark ampliamente utilizado que contiene más de 13,000 videos distribuidos en 101 clases diferentes.

A diferencia de los modelos basados en video crudo (RGB), este proyecto utiliza esqueletos 2D: coordenadas normalizadas de 17 articulaciones por frame extraídas previamente por un detector de poses. Esta representación reduce drásticamente el tamaño de los datos, hace posible entrenar el modelo sin GPU y permite enfocarse exclusivamente en los patrones de movimiento. En este proyecto se utilizaron todas las 101 clases disponibles.

El objetivo principal es demostrar el flujo completo de un sistema de reconocimiento de acciones: preprocesamiento, construcción del modelo, entrenamiento, evaluación, comparación contra baselines e inferencia sobre ejemplos reales.

Estrategia de trabajo

1. Revisión del dataset UCF101 Skeleton y del formato de anotaciones.
2. Organización del proyecto en Github, con scripts separados para entrenamiento e inferencia.
3. Carga y exploración del dataset
4. Implementación de dos modelos:
    a. ActionLSTM
    b. ActionMLP
5. Entrenamiento del modelo durante 20 épocas usando el split oficial train1/test1 por 20 épocas.
6. Evaluación del desempeño y comparación contra un baseline aleatorio.
7. Inferencia cualitativa utilizando nombres de clase.
8. Documentación completa de resultados, limitaciones y trabajo a futuro.

Pipeline implementado

- **Preprocesamiento**

    - Normalización de coordenadas.
    - Concatenación de x y por cada uno de los 17 joints → 51 features por frame.
    - Padding 120 frames por video.
    - Máscara temporal para distinguir frames reales de padding.

- **Modelos**
    - Se implementaron dos arquitecturas:

Modelo 1: ActionLSTM

El modelo base implementado es un LSTM bidireccional para capturar dependencias temporales en ambas direcciones de la secuencia. Esto es adecuado porque los keypoints forman una serie temporal que describe la evolución del movimiento humano.

Ventajas del LSTM:

- Los keypoints forman secuencias temporales naturales.
- Es ligero y eficiente sin GPU.
- Ha sido ampliamente utilizado como baseline para reconocimiento de acciones basado en esqueletos.

La salida final del LSTM se obtiene concatenando los estados finales forward y backward, lo que mejora el contexto temporal. La red concluye con una capa totalmente conectada que produce logits de tamaño 101.

Modelo 2: ActionMLP

Se implementó un segundo modelo para comparar arquitectura y fortalecer el análisis experimental.

El ActionMLP opera como baseline simple:

- Realiza un pooling temporal sobre todos los frames válidos → vector fijo por video.
- Pasa este vector por un MLP de dos capas.
- No utiliza información temporal explícita.

Es un modelo más limitado, pero útil para contrastar contra el LSTM

Resultados de entrenamiento

Se entrenaron ambos modelos durante 20 épocas sobre 9537 videos (train1) y se evaluaron en 3783 videos (test1).

Modelo	Arquitectura	Accuracy Test	Accuracy Train
Aleatorio	Predicción uniforme	0.99%	
ActionMLP	MLP con pooling temporal	28%	31%
ActionLSTM	LSTM bidireccional	31%	50%

Observaciones:

- Ambos modelos superan ampliamente al baseline aleatorio
- El LSTM obtiene mejor desempeño al capturar dependencias temporales
- El MLP funciona sorprendentemente bien considerando su simplicidad
- El LSTM muestra algo de overfitting, lo cual es esperado dada la complejidad de las 101 clases

Inferencia y ejemplos reales

Se probó el modelo ActionLSTM con el script “inference.py” obteniendo los siguientes resultados

```
None
Video idx: 0
Ground truth: 0 (ApplyEyeMakeup)
Top-5 predicciones:
  1) clase 77 (ShavingBeard)  prob=0.309
  2) clase 1 (ApplyLipstick)  prob=0.275
  3) clase 0 (ApplyEyeMakeup)  prob=0.092
  4) clase 17 (BoxingSpeedBag)  prob=0.073
  5) clase 19 (BrushingTeeth)  prob=0.068

Video idx: 10
```

```
Ground truth: 0 (ApplyEyeMakeup)
Top-5 predicciones:
  1) clase 1 (ApplyLipstick)  prob=0.277
  2) clase 77 (ShavingBeard)  prob=0.233
  3) clase 0 (ApplyEyeMakeup)  prob=0.197
  4) clase 19 (BrushingTeeth)  prob=0.083
  5) clase 61 (PlayingFlute)  prob=0.046

Video idx: 25
Ground truth: 0 (ApplyEyeMakeup)
Top-5 predicciones:
  1) clase 1 (ApplyLipstick)  prob=0.265
  2) clase 77 (ShavingBeard)  prob=0.241
  3) clase 0 (ApplyEyeMakeup)  prob=0.215
  4) clase 19 (BrushingTeeth)  prob=0.107
  5) clase 61 (PlayingFlute)  prob=0.045
```

Aunque la clase correcta aparece solo en top3 esto es razonable dado que es un modelo baseline sencillo sin técnicas avanzadas, además no usa video, sino únicamente keypoints 2D y se debe de tomar en cuenta que se usan las 101 clases totales.

Al utilizar el modelo ActionMLP se obtuvieron los siguientes resultados:

```
None
Video idx: 0
Ground truth: 0 (ApplyEyeMakeup)
Top-5 predicciones:
  1) clase 0 (ApplyEyeMakeup)  prob=0.397
  2) clase 77 (ShavingBeard)  prob=0.283
  3) clase 1 (ApplyLipstick)  prob=0.078
  4) clase 33 (Haircut)  prob=0.077
  5) clase 12 (BlowDryHair)  prob=0.047

Video idx: 10
Ground truth: 0 (ApplyEyeMakeup)
```

```
Top-5 predicciones:
  1) clase 1 (ApplyLipstick)  prob=0.532
  2) clase 0 (ApplyEyeMakeup)  prob=0.142
  3) clase 19 (BrushingTeeth)  prob=0.133
  4) clase 77 (ShavingBeard)  prob=0.105
  5) clase 12 (BlowDryHair)  prob=0.026

Video idx: 25
Ground truth: 0 (ApplyEyeMakeup)
Top-5 predicciones:
  1) clase 0 (ApplyEyeMakeup)  prob=0.545
  2) clase 1 (ApplyLipstick)  prob=0.190
  3) clase 77 (ShavingBeard)  prob=0.121
  4) clase 12 (BlowDryHair)  prob=0.054
  5) clase 19 (BrushingTeeth)  prob=0.029
```

En este caso podemos observar que el modelo tiene a confundir acciones faciales entre sí (ApplyLipstick, ShavingBeard, BlowDryHair), lo cual es razonable debido a la similitud en los patrones de movimiento del rostro, además, el modelo acierta en el top1 y top2 en varios casos, lo cual indica que sí aprende patrones relevantes.

Regularizacion

Dropout 0.3 en las capas LSTM
Bidireccional LSTM
Padding y masking
Normalizacion simple de los keypoints

Conclusiones

En este proyecto se implementó un sistema funcional para reconocimiento de acciones humanas utilizando únicamente keypoints 2D, sin necesidad de video completo. Los dos modelos implementados permiten comparar enfoques y demostrar:

- La ventaja de modelos temporales (LSTM) frente a modelos estáticos (MLP).
- La factibilidad de trabajar con representaciones comprimidas.

- Que un sistema relativamente ligero puede resolver una tarea compleja de 101 clases.

El modelo LSTM logró 31% de accuracy en test, superando ampliamente al baseline aleatorio y también al baseline MLP (28%).
Esto valida la capacidad de los LSTM para capturar patrones temporales en secuencias de movimiento humano

Repositorio

https://github.com/Gees14/LSTM