Valeria Aguilar Meza (A01741304)

Tecnológico de Monterrey

Inteligencia artificial avanzada para la ciencia de datos II (Gpo 501)
Clave: TC3007C.501

Diciembre 2025

1. Introducción

En este proyecto se desarrolló un modelo de reconocimiento de acciones basado en una arquitectura híbrida que combina redes convolucionales (CNN) y redes recurrentes (RNN). El objetivo fue clasificar videos del conjunto de datos UCF101 utilizando tanto características visuales extraídas de cada frame como anotaciones de esqueletos 2D cuando estaban disponibles.

La estrategia se centró en construir un pipeline funcional y eficiente bajo recursos limitados, priorizando experimentación con diferentes arquitecturas recurrentes.

2. Conjunto de Datos

El dataset utilizado fue **UCF101**, que contiene 101 clases de acciones humanas. Para este proyecto se seleccionó un subconjunto específico de 5 clases, definido en el archivo `config.py`:

- Basketball
- BasketballDunk
- ApplyEyeMakeup
- ApplyLipstick
- Archery

Adicionalmente, se utilizaron anotaciones de esqueletos en formato `.pkl` siguiendo el estándar de *MMAction2*. Cada frame contiene:

$$17 \text{ keypoints} \times 3 \text{ canales (x, y, score)} = 51 \text{ características}$$

3. Representación de los Datos

Cada video se representa mediante la combinación de dos tipos de características:

1. **Características visuales (CNN):** Extraídas utilizando **InceptionV3** preentrenada en ImageNet, generando un vector de 2048 características por frame.
2. **Esqueleto 2D:** Cuando disponible, se agregan 51 características por frame; en videos externos estas se rellenan con ceros.

Cada secuencia se normaliza a **20 frames por video**. La dimensión total por frame es:

$$2048 + 51 = 2099$$

4. Pipeline de Procesamiento

4.1. Preprocesamiento

- Lectura de videos desde UCF101.
- Selección de 20 frames equidistantes.
- Extracción de características con InceptionV3.
- Carga de anotaciones de esqueleto en formato `.pkl`.
- Concatenación de características CNN + skeleton.
- Filtrado automático de videos para conservar sólo las 5 clases seleccionadas.

4.2. Formato de Esqueletos

El formato de anotaciones sigue el estándar de MMAction2:

$$(\text{num\_persons},\ \text{num\_keypoints},\ 3)$$

Documentación oficial disponible en [1].

5. Modelos Implementados

Se implementaron tres arquitecturas RNN distintas, cada una definida en un archivo independiente dentro de `models/`.

5.1. 1. Modelo Base (Baseline)

Arquitectura real según `baseline_model.py`:

- LSTM (64 unidades, `return_sequences=True`)
- LSTM (32 unidades)
- Dropout 0.5
- Capa densa de 32 neuronas (ReLU)
- Softmax final
- Optimizador Adam (lr=0.001 por defecto)

Precisión en validación: 65.57 %

5.2. 2. Modelo Optimizado

Arquitectura real según `rnn_model.py`:

- LSTM Bidireccional (64 unidades, dropout 0.2)
- Batch Normalization
- LSTM (32 unidades, dropout 0.1)
- Batch Normalization
- Dense 128 (ReLU) + Dropout 0.3
- Dense 64 (ReLU) + Dropout 0.2
- Softmax final
- Adam (lr=0.0005)

Precisión en validación: 58.47 %

5.3. 3. Modelo Mejorado

Arquitectura real según `improved_model.py`, que incluye mecanismos avanzados:

- LSTM Bidireccional (128 unidades) con L1/L2 y *recurrent dropout*
- Mecanismo de atención temporal
- Conexión residual combinando atención y una LSTM adicional de 64 unidades
- Capas densas: 256 → 128 → 64 (ReLU/ELU)
- Regularización L1/L2 y Dropout 0.4–0.5
- Optimizador Adam (lr=0.001, `clipnorm=1.0`)

Precisión en validación: 43.72 %

6. Detalles de Entrenamiento

- **EarlyStopping**: paciencia 8
- **ReduceLROnPlateau**: paciencia 4, factor 0.5
- **ModelCheckpoint**: guarda sólo el mejor modelo en validación

7. Evaluación y Resultados

Los resultados por clase (tomados del `README.md`) muestran variabilidad según el modelo:

- Basketball: 75.00 % (Baseline), 34.09 % (Optimizado), 63.64 % (Mejorado)
- BasketballDunk: 68.75 % (Baseline), **100.00 %** (Optimizado), 31.25 % (Mejorado)
- ApplyEyeMakeup: 60.98 % (Baseline), 53.66 % (Optimizado), 26.83 % (Mejorado)
- ApplyLipstick: **94.29 %** (Baseline), 85.71 % (Optimizado), 0.00 % (Mejorado)
- Archery: 22.58 % (Baseline), 25.81 % (Optimizado), **100.00 %** (Mejorado)

8. Predicciones en Nuevos Videos

Para videos externos:

- Se extraen únicamente características CNN.
- El esqueleto se rellena con ceros al no existir archivos `.pkl`.
- Se genera una secuencia final de 20 frames.
- El modelo produce una distribución de probabilidad sobre las 5 clases.

9. Conclusiones

A partir de los experimentos realizados se concluye que:

- El modelo base obtuvo el mejor rendimiento general (65.57 %), y es el más estable.
- El modelo optimizado es útil para clases específicas como BasketballDunk.
- El modelo mejorado presenta especialización extrema, con aciertos perfectos en algunas clases y fallas totales en otras.
- Modelos más complejos no necesariamente generalizan mejor bajo datos y recursos limitados.

Referencias

1. MMAction2 Team. *Skeleton Dataset Format Documentation*. Disponible en: `https://mmaction2.readthedocs.io/en/latest/dataset_zoo/skeleton.html`