

**Instituto Tecnológico y de Estudios Superiores de Monterrey  
GDL Campus**



**Tecnológico  
de Monterrey**

**Inteligencia artificial avanzada para la ciencia de datos II (Gpo 501)**

Clave: TC3007C

Arah Rojas Blanco (A00834299)

**Profesor:**

Prof. Obed Noé Sámano Abonce

**Campus:** Guadalajara

1 dic 2025

## LSTM+RNN con memoria de largo plazo

Para este proyecto decidí trabajar con esqueletos 2D (coordenadas de articulaciones por frame) en lugar de usar los videos directamente. Mi razón principal fue que así el modelo se enfoca en lo más importante para reconocer acciones: el movimiento, y al mismo tiempo se reduce muchísimo la carga computacional. También elegí un subconjunto de 5 clases porque mis recursos son limitados y porque esas clases siguen representando un reto real (varias son parecidas entre sí). Las clases que usé fueron:

- ApplyEyeMakeup
- ApplyLipstick
- BlowDryHair
- BrushingTeeth
- HeadMassage

Baseline:

Mi primera aproximación fue un modelo LSTM sencillo. Elegí LSTM porque está pensada para datos que vienen en secuencia, y aquí cada video es justamente eso: frames en orden donde el modelo necesita “recordar” lo que pasó antes para entender la acción.

Este baseline es simple: una LSTM y una capa final para clasificar en 5 acciones. La idea era tener un punto de partida real y medible para después mejorarlo.

Resultado del baseline: alrededor de ~51–52% de precisión (dependiendo del split), lo cual me sirvió como referencia para justificar cambios.

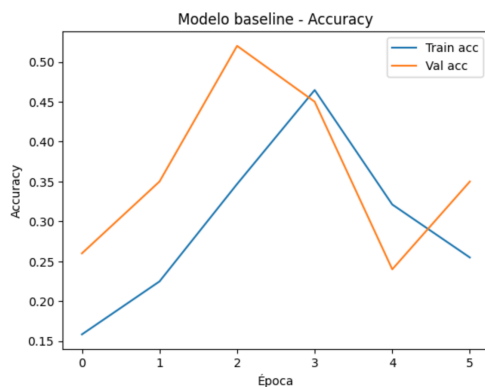


Figura 3. Curvas de entrenamiento del baseline

Model: "sequential"

Layer (type)	Output Shape	Param #
masking (Masking)	(None, 300, 34)	0
lstm (LSTM)	(None, 32)	8,576
dense (Dense)	(None, 5)	165

Total params: 8,741 (34.14 KB)  
Trainable params: 8,741 (34.14 KB)  
Non-trainable params: 0 (0.00 B)

Figura 4. Captura del model.summary() del baseline.

Modelo mejorado:

Cambios que se hicieron:

- Pasó de 1 LSTM a 2 capas LSTM
- Se agregó Dropout entre capas (usé 0.5) para evitar que el modelo se “memorice” el entrenamiento
- Se aplicó Early Stopping para detener el entrenamiento cuando el modelo ya no mejora en validación

Resultado del modelo mejorado: alrededor de 63–65% de precisión en test, y lo más importante: resultados similares entre validación y prueba, lo cual sugiere que sí generaliza.

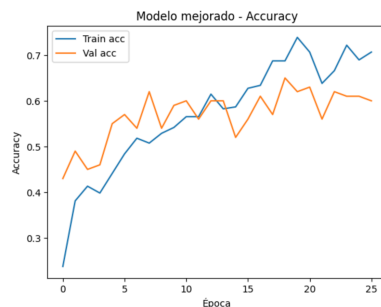


Figura 5. Curvas de entrenamiento del modelo mejorado.

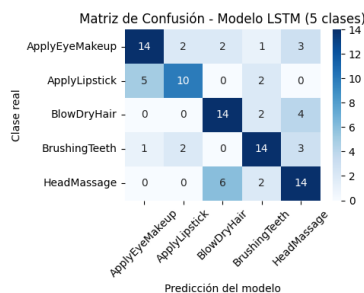


Figura 6. Matriz de confusión

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
masking_1 (Masking)	(None, 300, 34)	0
lstm_1 (LSTM)	(None, 300, 64)	25,344
dropout (Dropout)	(None, 300, 64)	0
lstm_2 (LSTM)	(None, 64)	33,024
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 5)	325

Total params: 58,693 (229.27 KB)  
Trainable params: 58,693 (229.27 KB)  
Non-trainable params: 0 (0.00 B)

Figura 7. Captura del model.summary() del modelo mejorado.

Inferencia (predicciones):

1. Busca un video\_id en data['annotations'] (por ejemplo v\_ApplyEyeMakeup\_g08\_c01)
2. Aplica el mismo preprocesamiento (normalización + padding a L)
3. Corre model.predict() y devuelve clase + confianza

Interfaz

Comparación Baseline vs Modelo Mejorado				
Accuracy	Mostrar Loss	Mostrar Ambas	Mostrar Métricas	Demo: Ejemplo #0
Hiperparámetro	Baseline	Mejorado		
0	LSTM units	32	64	
1	Capas LSTM	1	2	
2	Dropout	0.0	0.5	
3	Batch size	32	32	
4	Épocas	20	50	
5	Learning rate	0.001	0.001	

Conclusiones

En este proyecto logré desarrollar un modelo capaz de clasificar acciones humanas usando representaciones de esqueletos 2D del dataset UCF101. Comencé con un baseline sencillo basado en una LSTM, que sirvió como referencia inicial y me permitió entender el comportamiento del modelo con una arquitectura mínima. Después de analizar sus resultados, implementé un modelo mejorado más profundo y con regularización (dropout), lo que incrementó la precisión y redujo el sobreajuste.

Los resultados muestran una mejora clara entre el modelo base y el modelo final, lo que confirma que los ajustes fueron útiles. Además, implementé un método de predicción para probar videos individuales, asegurando que el modelo realmente puede generar inferencias y cumplir con los requisitos del módulo.