Escenas de playground usando MP-GCN para reconocimiento de actividades basadas en esqueletos

1st Carlos Alberto Mentado Reyes
A01276065

2nd Fernanda Díaz Gutiérrez
A01639572

3rd José Eduardo Puentes Martínez
A01733177

4th Raymundo Iván Díaz Alejandre
A01735644

***Abstract*—En zonas recreativas infantiles, la supervisión manual de las actividades de los niños es costosa, subjetiva y poco escalable, este trabajo explora el uso de grafos espaciotemporales basados en esqueletos 2D y el modelo MP-GCN para clasificar, a nivel de frame y por persona, actividades de riesgo y no riesgo en playgrounds a partir de cámaras fijas, la representación panorámica persona–objeto permite capturar relaciones intra-persona, inter-persona y persona–objeto, favoreciendo modelos más ligeros, más privados y robustos frente a cambios de iluminación, apariencia y perspectiva.**

***Index Terms*—Graph Convolutional Networks, MP-GCN, reconocimiento de actividades, esqueletos 2D, visión computacional, playground safety**

I. Introducción al problema

A. Claridad y descripción del problema

En zonas recreativas para niños (playgrounds), los cuidadores y operadores tienen la necesidad de detectar de manera eficiente las actividades que los niños realizan, con el fin de prevenir accidentes, mejorar su seguridad y priorizar su bienestar sin vulnerar sus datos, realizar esa supervisión hoy en día se lleva a cabo de manera manual, lo cual puede llegar a ser muy costoso, subjetivo (y por lo tanto ineficiente), y no escalable.

Problema a resolver. A partir de videos de cámaras fijas en dichos playgrounds, debemos clasificar cada uno de los frames en una de tres categorías dadas (que son mutuamente excluyentes y resumen los eventos importantes para el proyecto): Transit (tránsito), Social_People (interacción social) y Play_Object_Normal (juego normal con el mobiliario del parque). Para clasificar la actividad, la unidad de decisión es solo un frame del video; por lo tanto, para cada frame se emiten máximo 4 etiquetas (para 4 personas, pues es el máximo) que describen la actividad realizada.

Entrada. La unidad de entrada son frames de videos crudos de las cámaras de seguridad fijas en los playgrounds. Para la fase de decisión, el sistema deriva de dichos videos:

- *Esqueletos 2D:* De hasta K personas por frame (donde $K = 4$), basados en las articulaciones del cuerpo y muestreados a 12 fps.
- *Tracking:* Seguimiento de cada persona a lo largo del tiempo (asignación de ID por persona) para mantener coherencia entre frames.
- *Centroides de objetos:* Para los objetos fijos del playground como columpios o bancas, centroides por cámara en coordenadas normalizadas.

Enfoque de representación. Para cada frame t de cada video, el sistema produce una representación tipo grafo panorámico MP-GCN (persona–objeto):

- *Nodos:* (i) articulaciones humanas (17 por persona) y (ii) centroides de objetos fijos del parque (dependientes de la cámara).
- *Aristas*:
  - **Intra-persona:** Topología del cuerpo/conectividad del esqueleto.
  - **Persona ↔ Objeto:** (manos↔columpio/lomas), proximidad/contacto de las manos con el mobiliario.
  - **Inter-persona:** (pelvis–pelvis), relación entre personas.
- *Streams* considerados de forma interna: J (joints), B (bones), JM (ΔJ) y BM (ΔB).

Salida esperada. Para cada frame t y para cada persona $m \in \{1, \ldots, K_{\text{vis}} \leq K_{\max}\}$, el sistema produce una **etiqueta de la actividad realizada** $y_{m,t}$ que pertenece al conjunto de clases descrito.

Supuestos y limitaciones.

- El número de personas visibles por frame es de máximo 4 ($K_{\max}$ personas por ventana: 4).
- FPS de proceso: ~ 12; ventana $T \approx 60$ muestreada a $T = 48$ para el modelo.
- Las interacciones relevantes suceden dentro del campo de visión de las cámaras.
- "Riesgo" se entiende como un patrón anómalo del uso del mobiliario (alturas/trayectorias no esperadas).
- La decisión de etiquetado es por persona y no emite identificadores diferentes (personales) de la información mínima (pose/trayectoria) para clasificar.

B. Justificación y relevancia del problema

Durante los últimos años se ha visto un aumento en el uso de redes neuronales basadas en grafos; particularmente ha incrementado la implementación de redes neuronales convolucionales de grafos con información obtenida a partir de esqueletos humanos.

Se han adaptado este tipo de modelos para el reconocimiento de actividades humanas en video para aplicaciones en vigilancia, interacciones humano–computador, salud, robótica y entretenimiento.

De igual manera, este aumento en el uso de GCNs ha sido impulsado por las ventajas existentes cuando se compara con el análisis de video en crudo:

- Los datos obtenidos de los esqueletos son más ligeros que los datos obtenidos de un video en crudo.
- El uso de esqueletos en vez de procesamiento de frames fomenta la conservación de la privacidad de las personas.
- Un modelo basado en grafos de esqueletos humanos es mucho más robusto a cambios en apariencias, luces y perspectivas de cámaras.

Además, el rendimiento de los modelos de GCN muestra mejores resultados a comparación de otros modelos, sobrepasando resultados de modelos como redes neuronales de convolución en cuanto a reconocimiento de actividades basadas en esqueletos.

Gracias al auge que están teniendo los modelos GCN, también es relevante mencionar que se están buscando mejoras en los resultados y la eficiencia de estos modelos, pero también se incluyen retos como:

- Reconocimiento de interacciones complejas.
- Reconocimiento de acciones sutiles.
- Generalización en ambientes diversos.

Un proyecto con este alcance puede ser muy relevante para el campo de visión computacional y la ingeniería, tomando en cuenta que las actividades que se van a etiquetar incluyen diferentes espacios geográficos dentro de la misma toma.

II. Estado del arte

A. Trabajos previos relacionados

La fuente principal y guía del proyecto es el artículo *"Skeleton-based Group Activity Recognition via Spatial-Temporal Panoramic Graph"* (Springer, 2023). Este trabajo introduce el modelo MP-GCN, que propone un grafo panorámico persona–objeto–persona para representar las interacciones colectivas a partir de esqueletos 2D, integrando tres tipos de relaciones: dentro del cuerpo, entre distintos sujetos e interacciones con elementos del entorno. Este enfoque demuestra que los esqueletos pueden sustituir a los videos RGB, teniendo como resultado un modelo más liviano y éticamente responsable.

A partir de este núcleo se seleccionaron diez fuentes complementarias que fortalecen distintos componentes del proyecto:

- Trabajos de Pamplona Berón y Calvo Salcedo, que sentaron bases del reconocimiento multimodal mediante SVM y HMM.
- Komang et al. y Aguado, que aportan estrategias efectivas de clasificación a partir de esqueletos 2D, normalización anatómica y extracción de ángulos y velocidades.
- Li et al. (SGM-Net) amplía la visión hacia la fusión multimodal guiada por esqueleto, relevante para considerar información contextual o de objetos del entorno.
- Olmo explora el uso de información de color y profundidad con selección de características y modelado del movimiento, aportando ideas sobre filtrado temporal y reducción de redundancia.
- Estudios más recientes, como Bengtson et al. y Forini et al., demuestran la aplicabilidad de la estimación de esqueleto en entornos reales y simulados.

En conjunto, estas fuentes proporcionan una cobertura integral y actualizada del estado del arte, combinando modelos multimodales, espaciotemporales y de atención, los cuales responden a las necesidades del reto.

- MP-GCN (Springer) establece la arquitectura base y justifica la elección del enfoque esquelético.
- ST-GCN y SGM-Net ofrecen referencias técnicas para estructurar el grafo y sus flujos.
- Modelos previos basados en SVM/HMM inspiran la jerarquía de reconocimiento entre subestados y escenas completas.

Estas fuentes permiten un sustento teórico sólido y actualizado para el desarrollo del pipeline propuesto para el reto.

B. Análisis y síntesis de la información

El análisis del artículo *"Skeleton-based Group Activity Recognition via Spatial-Temporal Panoramic Graph"* (Springer, 2023) demuestra que el uso de esqueletos 2D como *input* ofrece una gran cantidad de ventajas frente a posibles cambios de iluminación, perspectiva, ángulos o apariencia.

El modelo logra capturar patrones espaciales de forma muy eficaz, ya que hace uso de la topología del cuerpo humano como una estructura de grafos con distintos cambios. Esto permite modelar un mayor número de interacciones entre múltiples personas y objetos sin depender de videos RGB, lo que ayuda a reducir tareas de procesamiento y a brindar una mayor privacidad a los usuarios. No obstante, también existen limitaciones, como la necesidad de calibrar cámaras, la dificultad de detectar acciones genéricas que puedan confundir al sistema y la pérdida de información en movimientos muy rápidos.

Comparado con arquitecturas previas como ST-GCN, el modelo MP-GCN agrega una capa adicional de relación persona–objeto, lo que brinda mayor contexto del entorno. Mientras que ST-GCN daba un enfoque principal a los movimientos del cuerpo en el tiempo, MP-GCN integra de manera explícita las relaciones con el mobiliario y otros sujetos, lo cual representa una ventaja en escenarios donde las acciones humanas dependen en gran medida de los objetos del entorno o las interacciones entre muchas personas.

Las conclusiones del estudio subrayan la importancia del uso de grafos, ya que permiten representar un modelo eficiente y escalable. El modelo MP-GCN proporciona una base sólida para comprender el reconocimiento de distintas actividades existentes en entornos reales. Su diseño, eficiencia

computacional y el fuerte enfoque en las relaciones persona–objeto (por ejemplo, niños, adultos y objetos del playground) ofrecen una representación estructurada que puede ejecutarse en tiempo real y habilitar una solución eficaz para monitoreo y prevención de riesgos.

III. Metodología propuesta

A. Etiquetado de videos

Para la solución del reto se contaba con videos archivo de 4 cámaras ubicadas en diferentes posiciones geográficas dentro de la zona TEC en Monterrey, primero se seleccionaron 400 videos en base a cantidad de personas presentes, de estos 400 videos 330 estaban archivados de forma errónea por lo que no pudieron ser descargados y se corrompieron. Después se realizó un filtrado para decidir cuáles si podrían resultar útiles para el modelo, resultando en 213 videos.

Una vez listos los videos, se creó una interfaz gráfica con el propósito de facilitar el seguimiento del etiquetado de los videos, esta interfaz gráfica permitía ver el video, seleccionar el tiempo de inicio y final de la acción a clasificar y decidir que clase es la que se presenta en el video. Gracias a esta interfaz se obtuvo un archivo *CSV* con la información necesaria para etiquetar y preparar los videos, este archivo contiene:

TABLE I
Estructura del CSV para anotaciones.

Field	Description
scene_id	Identificador único para describir una escena, puede haber más de una escena dentro del mismo video.
video_id	Identificador del video del que proviene una escena.
camera	Identificador de la cámara de la cuál se obtuvo el video.
timestamp_start	Marca de tiempo donde empieza una escena dentro de un video.
timestamp_end	Marca de tiempo donde termina una escena dentro de un video.
blob_path	Directorio dentro de la base de datos donde se encuentra un video.
fps	Frames por segundo del video.
label	Clase que se le asignó a un video.

B. Extracción de esqueletos

Con las escenas listas, el siguiente paso fue procesar los videos y extraer los esqueletos de las personas presentes en los video, para procesar los videos se creó un script de python que usa la librería *moviepy*, este archivo consta de una función que requiere de parámetros un string que representa el directorio de donde están guardados los videos en crudo, y los campos del archivo csv generado en el etiquetado de los videos, después corta los videos en las marcas de tiempo especificadas y guarda el resultado en un nuevo directorio.

En paralelo, cada que un video es cortado y guardado, una segunda función extrae los esqueletos del video usando un modelo *yolov8m*, normalizando los valores de la coordenadas basándose en el promedio de la distancia entre pelvis y cuello. Con la limitación de 4 personas por video. Los resultados de cada video fueron guardados en un archivo csv con el nombre del id de la escena más la clase asignada a la escena.

C. Detección de objetos

Para añadir la información de los centroides de los objetos presentes en cada cámara se creó un archivo yaml con las coordenadas de estos objetos

D. Creación de grafos para red

Para esta etapa, se busca transformar la información es cruda de detecciones (esqueletos y objetos) en una representación estructurada en forma de grafo espaciotemporal, apta para el modelo MP-GCN. Para cada escena etiquetada se construye un tensor panorámico persona-objeto y un conjunto de matrices de adyacencia que modelan las relaciones dentro del cuerpo, entre personas, y objeto-persona.

1) Ventanas temporales y normalización: Cada una de las escenas se recorta en una ventana fija de $T = 48$ frames, muestreada a $\sim$ 12 fps (aprox. 4 segundos efectivos por muestra). Para cada frame $t \in \{1, \ldots, T\}$ se tienen máximo se $M = 4$ personas visibles. Como las coordenadas originales de la pose se expresan en píxeles, se realizó una normalización persona a persona para que la representación sea invariante a height, resolución o distancia a la cámara. La normalización se basa en la distancia torso / neck y pelvis, de modo que:

$$\text{escala} = \|J_{\text{neck}} - J_{\text{pelvis}}\|_2 .$$

Para cada articulación j, la coordenada ya normalizada se obtiene como:

$$J_{\text{norm}}[j] = \frac{J[j] - J_{\text{pelvis}}}{\text{scale}}.$$

Este proceso realiza una traslación (re-centra las poses respecto a la pelvis) y las re-escala por la longitud del torso. Las coordenadas se llevan finalmente al rango $[0, 1]$.

2) Definición de nodos: El grafo combina joints y objetos (nodos de humanos y nodos de objetos). El total de nodos por frame se fija como:

$$V = 17 \text{ (joints humanos)} + 15 \text{ (objetos)} = 32.$$

Donde 15 es el número máximo de objetos por cámara.

- **Nodos humanos.** Para cada persona y frame se usan 17 joints del modelo de pose. Cada joint es un par de coordenadas normalizadas (x, y).
- **Nodos de objetos.** Para cada cámara se define un YAML con centroides de los objetos del parque (columpios, bancas, domos, etc.) en coordenadas (x_c, y_c) (normalizadas). Los nodos son estáticos en el tiempo (entre frames).
- **Padding de nodos.** El MP-GCN pide un V fijo. En nuestro caso el vocabulario de los objetos por cámara considera 15 elementos como máximo. Si en alguna cámara hay menos objetos, los nodos restantes se completan con padding (coordenadas en cero).

3) Streams de características: Acorde a la arquitectura original de MP-GCN, se construyen cuatro streams para cada nodo:

1) **J (joints).** Coordenadas espaciales de cada nodo:

$$J_{t,v,m} = (x_{t,v,m},\, y_{t,v,m}),$$

donde t es el frame, v el nodo y m el índice de persona u objeto (fijo en el tiempo).

2) **B (bounding boxes).** Para las personas se usa el bounding box de la detección de pose, para los objetos se construye un bounding box alrededor del centroide (sintético):

$$B_{t,v,m} = (x_{\min}, y_{\min}, x_{\max}, y_{\max}).$$

3) **JM (Joint Motion).** Captura el movimiento frame a frame de cada nodo:

$$JM_{t,v,m} = J_{t,v,m} - J_{t-1,v,m}, \quad t > 1,$$

y $JM_{1,v,m} = \mathbf{0}$.

4) **BM (Box Motion).** Diferencia temporal para los bounding boxes:

$$BM_{t,v,m} = B_{t,v,m} - B_{t-1,v,m}, \quad t > 1,$$

y $BM_{1,v,m} = \mathbf{0}$.

Cada stream se guarda como un tensor de forma $[C, T, V, M]$, donde $C = 2$ para coordenadas tipo (x, y) y $C = 4$ para bounding boxes.

En la implementación se trabaja con tensores de forma $[2, 48, 32, 4]$ por stream.

4) Matrices de adyacencia: Las relaciones entre nodos se codifican con 3 matrices de adyacencias (siguiendo el esquema del MP-GCN original).

- **Matriz A_0 (intra-cuerpo).** Es la estructura anatómica de cada persona, conecta joints vecinos (cadera–rodilla, rodilla–tobillo, hombro–codo, etc.). Es fija y se extiende a los $V = 32$ nodos mediante padding para incluir los nodos de objetos.
- **Matriz A_{intra} (persona–objeto).** Codifica relaciones entre joints humanos y de objetos. Se conectan manos/pies con objetos cuando se encuentran cerca. Se usa un umbral de distancia euclidiana τ en coordenadas normalizadas:

$$\|\mathbf{p}_{\text{joint}} - \mathbf{p}_{\text{obj}}\|_2 < \tau \Rightarrow A_{\text{intra}}(i, j) = 1.$$

- **Matriz A_{inter} (objeto–objeto / persona–persona).** Representa la proximidad entre nodos dentro de la escena como bloques contiguos (objetos agrupados) o personas cercanas entre sí. Se define también mediante un umbral sobre distancia:

$$\|\mathbf{p}_i - \mathbf{p}_j\|_2 < \tau' \Rightarrow A_{\text{inter}}(i, j) = 1.$$

Las tres matrices se apilan para obtener el grafo:

$$A = \begin{bmatrix} A_0 \\ A_{\text{intra}} \\ A_{\text{inter}} \end{bmatrix} \in \mathbb{R}^{3\times 32\times 32}.$$

En el código, dichas matrices se generan usando la función `build_adjacency_matrices(V_human=17, n_obj=15)` y se reutilizan para todas las muestras (ejemplos) del dataset.

5) Empaquetado final y almacenamiento: Para cada escena (etiquetada) se genera un archivo `.npy` que contiene:

- Los cuatro streams previamente descritos:

$$J,\, B,\, JM,\, BM \in \mathbb{R}^{2\times 48\times 32\times 4},$$

- Las 3 matrices de adyacencia apiladas, representadas como: $A \in \mathbb{R}^{3\times 32\times 32}$,
- La etiqueta de la escena, ya sea: (`Transit`, `Social_People` o `Play_Object_Normal`).

Y después, el dataloader construye el tensor de input para el modelo MP-GCN, apilando los cuatro streams a lo largo de una nueva dimensión de entrada:

$$X \in \mathbb{R}^{4\times 2\times 48\times 32\times 4},$$

donde los cuatro canales iniciales corresponden a $\{J, B, JM, BM\}$. Esta representación panorámica persona-objeto mantiene la información temporal-espacial necesaria para que la red aprenda patrones de actividades en el playground de manera exitosa y considerando la privacidad.

E. Representación tensorial para el modelo

Una vez que se construyó el grafo persona-objeto para cada frame, la información se organiza en tensores alineados al formato esperado por el modelo MP-GCN, con el fin de que se estandarizen el número de personas, articulaciones y objetos. Esto permite la aplicación de convoluciones espaciales-temporales sobre los grafos.

1) Dimensiones base: Para cada muestra se produce un conjunto de tensores organizados en las dimensiones fijas:

$$X \in \mathbb{R}^{I\times C\times T\times V\times M},$$

donde:

- $I = 4$: número de *streams* (J, B, JM, BM),
- $C = 2$: coordenadas espaciales (x, y),
- $T = 48$: frames por ventana temporal,
- $V = 32$: nodos totales (17 articulaciones humanas + 15 objetos),
- $M = 4$: número máximo de personas (simultáneas).

Esta estructura ayuda a convertir cada una de las escenas en una representación estandarizada (tensorial) que se vuelve invariante al número real de personas/objetos en cámara.

2) Tensor del stream J (joints): Este tensor contiene las coordenadas normalizadas de cada joint y de los centroides de objetos:

$$J[i, c, t, v, m] = \text{coordenada } (x \text{ o } y) \text{ del nodo } v.$$

En el caso de tener nodos inexistentes, como padding de objetos o personas faltantes, se asigna el valor cero.

3) Tensor del stream B (bounding boxes): Para cada persona se usa su bounding box estimado a partir del esqueleto, y para cada objeto se crea un bounding box centrado en su centroide (sintético). El tensor guarda las coordenadas normalizadas:

$$B = (x_{\min},\, y_{\min},\, x_{\max},\, y_{\max}).$$

4) Tensor JM (movimiento de articulaciones): Se calcula como:

$$JM_t = J_t - J_{t-1}, \qquad JM_1 = \mathbf{0}.$$

Este stream nos ayuda a capturar los movimientos suaves, y resulta importante para distinguir las actividades estáticas como la socialización (Social_People) de las actividades con movimiento (ya sea translacional o de juego) como (Transit, Play_Object_Normal).

5) Tensor BM (movimiento de bounding boxes): De forma análoga:

$$BM_t = B_t - B_{t-1}.$$

Este nos ayuda a capturar la amplitud de los movimientos y las aceleraciones a nivel global del esqueleto (cuerpo).

6) Apilamiento final: Estos cuatro tensores se apilan a lo largo de la dimensión del stream:

$$X = \begin{bmatrix} J \\ B \\ JM \\ BM \end{bmatrix} \in \mathbb{R}^{4\times 2\times 48\times 32\times 4}.$$

Y finalmente, este tensor, junto con la matriz de adyacencia

$$A \in \mathbb{R}^{3\times 32\times 32},$$

es la entrada final al modelo MP-GCN.

7) Ventajas de la representación tensorial:

- Elimina la dependencia del video RGB.
- Cuida la privacidad al utilizar solo poses y objetos.
- Reduce el tamaño de los datos.
- Mantiene dimensiones constantes, permitiendo un entrenamiento más estable.
- Facilita agregar relaciones (persona–objeto, persona–persona) a través de A.

Podemos decir que la representación tensorial es lo que permite que el modelo MP-GCN aprenda patrones (tanto espaciales como cinemáticos) sin necesitar información visual explícita, logrando un pipeline ligero y basado por completo en estructuras de grafos.

Este pipeline produce una representación estandarizada y completamente basada en información estructural (poses y objetos), permitiendo entrenar MP-GCN sin depender de imágenes RGB y maximizando privacidad y eficiencia.

IV. Arquitectura del modelo MP-GCN

Para el modelo, la arquitectura usada comprende inputs de secuencias de 48 frames, con los cuales construye un grafo por frame unificando personas y objetos.

La entrada del modelo comprende una secuencia de 48 frames, el cual se convierte en un tensor multistream de tamaño: [N, 4, 2, 48, 32, 4], donde son 4 streams (J, B, JM, y BM), 2 canales (x, y), 48 frames, 32 nodos (17 joints y 15 objetos) y 4 conjuntos de personas-objetos por escena.

Para la estructura interna del modelo, cuenta con el módulo de los Input Branches, en donde cada stream se procesa de manera independiente usando bloques ST-GCN que combinan las convoluciones espaciales usando la matriz de adyacencia A y las convoluciones temporales (kernel temporal)

Luego pasa por la fusión de los 4 streams, se concatenan para producir una representación conjunta humano - objeto, posteriormente se usan bloques GCN profundos en donde se incluyen capas con Normalización, Activación, Dropout y Conexiones residuales.

Al final se hace un global average pooling sobre nodos y frames, el vector resultante pasa por un MLP que produce 3 logits, las 3 clases a predecir.

V. Conjunto de datos y protocolo experimental

A. Descripción del dataset

Los datos provienen de cuatro cámaras fijas instaladas en el playground. Inicialmente se seleccionaron 400 videos, pero tras un proceso de filtrado (archivos dañados, escenas irrelevantes o vacías) se obtuvieron 213 videos utilizables. Cada video puede contener múltiples escenas; por ello el pipeline produce un archivo CSV con *scene_id*, rangos de tiempo, cámara, FPS y la etiqueta asignada por el equipo (`Transit`, `Social_People`, `Play_Object_Normal`).

El dataset original contenía clases muy desbalanceadas, por lo que se redujo a tres categorías con cantidad suficiente de muestras por clase.

B. Extracción y preparación de datos

Para cada escena se generaron:

- Esqueletos 2D mediante un modelo de pose basado en YOLOv8.
- Tracking de personas (máximo 4 por escena).
- Centroides de objetos por cámara mediante un archivo YAML.

Cada escena se convirtió en una ventana temporal fija de $T = 48$ frames (aprox. 4 s a 12 fps). Se construyó un grafo panorámico persona–objeto con $V = 32$ nodos y se generaron los cuatro streams requeridos por MP-GCN: J, B, JM, BM. Finalmente, para cada muestra se guarda un archivo $.npy$ con el tensor y las matrices de adyacencia.

C. Particiones de entrenamiento, validación y prueba

Las particiones se generaron mediante un script reproducible (`build_train_val_csv.py`). La distribución fue:

Clase	Total	Train	Val	Test
Transit	71	50	11	10
Social_People	71	50	11	10
Play_Object_Normal	70	49	10	11

El conjunto de prueba contiene 31 muestras y no se utilizó durante el entrenamiento.

D. Configuración experimental

El modelo se entrenó con:

- Optimizador Adam: $lr = 10^{-3}$, $weight\_decay = 10^{-4}$
- Épocas: 40
- Batch size: 4
- Semilla fija: 7
- Módulo de atención desactivado (`use_att = False`)

El mejor modelo se seleccionó por exactitud en validación (71.9%). Luego se evalúo sobre el conjunto de prueba, obteniendo una exactitud global de 48.39%.

E. Métricas utilizadas

Se usaron las métricas estándar para clasificación multiclase:

- Exactitud global.
- Precision, Recall y F1 por clase (Classification Report).
- Matriz de confusión.

Las métricas permiten analizar el desempeño entre clases, en específico la confusión entre `Transit` y `Social_People`.

Es importante mencionar que durante las primeras corridas se ajustaron hiperparámetros básicos (lr, batch size y número de épocas) hasta obtener un entrenamiento estable y sin overfitting extremo en entrenamiento. A partir de ahí se fijó la configuración descrita y se seleccionó el mejor modelo con base en la exactitud de validación.

VI. Resultados

El modelo obtuvo una exactitud global de 48.39 por ciento en el conjunto de prueba. Aunque este valor puede parecer algo moderado, es bastante consistente con la complejidad del problema y las limitaciones de utilizar únicamente esqueletos 2D como entrada. Además, se trata de un conjunto de datos reducido, con 31 muestras, lo cual introduce variabilidad significativa en las métricas por clase.

Las métricas muestran comportamientos diferenciados dependiendo de la clase, para Play_Object_Normal es la clase con mejor desempeño (Recall = 0.72, F1 = 0.66) Transit obtiene un desempeño de (F1=0.42) Social_People es la más difícil de discriminar con un (F1 = 0.31)

Estos resultados se resumen en la matriz de confusión (Fig. 1) y en los puntajes por clase de F1, precisión y recall (Fig. 2 y Fig. 3).

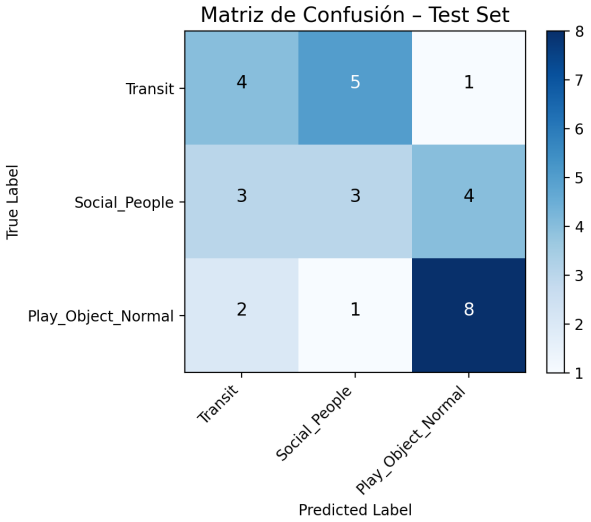

Fig. 1. Matriz de confusión en el conjunto de prueba.

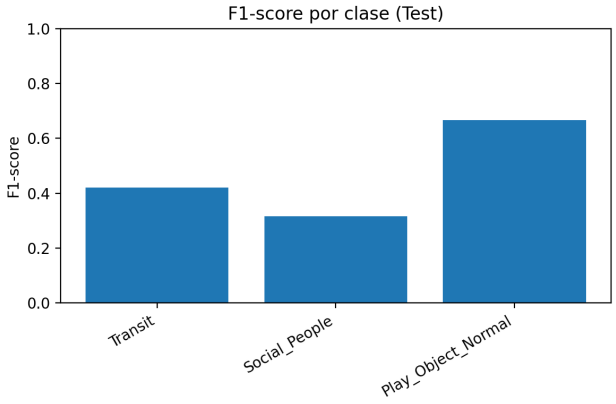

Fig. 2. Puntajes F1 por clase en el conjunto de prueba.

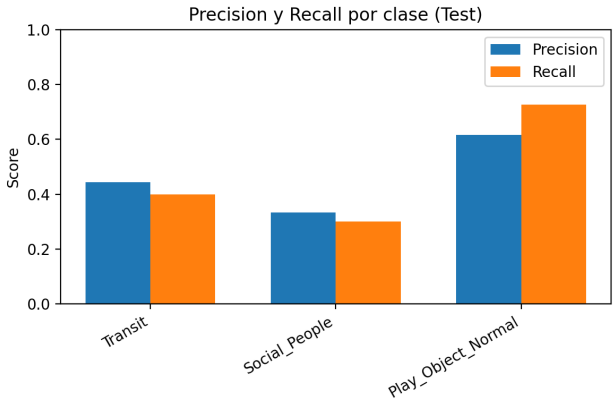

Fig. 3. Precisión y Recall por clase en el conjunto de prueba.

VII. Discusión

La mayor fuente de error ocurre entre Transit y Social_People, en total, 8 muestras se confunden entre estas dos

clases, representando más del 25 por ciento del conjunto de prueba.

Las poses de los esqueletos en Transit y Social_People suelen ser muy parecidas, caminar, estar de pie, moverse en grupo, etc.

Solo tomando en cuenta las articulaciones 2D es difícil distinguir si dos personas están simplemente cruzándose o en verdad están interactuando, además, por la naturaleza de ser un parque, dentro de las mismas interacciones sociales, llega a haber desplazamiento por parte de las personas.

VIII. Conclusiones y trabajo futuro

El desempeño desigual entre clases refleja la naturaleza del problema, no un fallo del modelo, las clases que más se confundían, Transit y Social_People, son semánticamente cercanas, dependen de información social y contextos aparte de los esqueletos 2D, y requieren temporalidad más larga, en cambio, Play_Object_Normal posee relaciones estructurales claras que el modelo puede capturar.

Como trabajo futuro, se propone refinar el modelo ajustando sistemáticamente la profundidad de las capas GCN, el uso del módulo de atención y esquemas de regularización más fuertes (por ejemplo mayor dropout o weight decay). También sería interesante comparar MP-GCN con otras arquitecturas de referencia para esqueletos, como ST-GCN o Shift-GCN, usando el mismo pipeline y protocolo experimental, para estudiar qué diseño ofrece mejor compromiso entre desempeño, costo computacional y robustez.

References

[1] T. Liu *et al.*, "Skeleton-based Group Activity Recognition via Spatial-Temporal Panoramic Graph," in *Proc. of a Computer Vision Conference*, 2023.

[2] S. Yan, Y. Xiong, and D. Lin, "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition," in *AAAI Conf. on Artificial Intelligence*, 2018.

[3] L. Shi, Y. Zhang, C. Jian, and H. Lu, "Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] K. Cheng, Y. Zhang, X. He, W. Chen, C. Jian, and H. Lu, "Skeleton-Based Action Recognition With Shift Graph Convolutional Network," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[5] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang, "Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[6] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[7] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] T. S. Kim and A. Reiter, "Interpretable 3D Human Action Analysis with Temporal Convolutional Networks," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[9] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaïd, "A New Representation of Skeleton Sequences for 3D Action Recognition," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] T. Liu, R. Zhao, K.-M. Lam, and J. Kong, "Visual-semantic Graph Neural Network With Pose-Position Attentive Learning for Group Activity Recognition," *Neurocomputing*, 2022.

[11] A. Bansal, "Detecting and Recognizing Humans, Objects, and Their Interactions," Ph.D. dissertation, 2019.