MÁSTER EN INTELIGENCIA ARTIFICIAL

TRABAJO FIN DE MÁSTER

Detector de eventos clave en eventos deportivos basados en análisis de vídeos

Presentado por:

CESÁREO TELLO SÁNCHEZ

Dirigido por:

MIGUEL ÁNGEL TORRES FONT

CURSO ACADÉMICO 2024-2025

Resumen

Este trabajo de fin de máster aborda el desarrollo de un sistema inteligente capaz de identificar y contextualizar los momentos más trascendentales en partidos de fútbol. La detección manual de estos sucesos es un desafío significativo por su naturaleza laboriosa, inherente subjetividad humana y altos costos. Este proyecto propone una herramienta innovadora que busca democratizar el acceso a tecnologías avanzadas de análisis deportivo.

El sistema integra avanzadas técnicas de visión por computador e inteligencia artificial para procesar vídeos de fútbol. El corazón del proyecto reside en la detección de eventos claves (goles, tarjetas amarillas, sustituciones y saques de esquina), una tarea que demanda comprender secuencias de acciones en el tiempo. Se emplean modelos de aprendizaje profundo para el análisis espacio-temporal de vídeos, entrenados con el dataset SoccerNet. La vista isométrica de este dataset es fundamental para que el modelo aprenda patrones visuales estables y entienda el contexto de cada evento.

La inferencia del modelo se realiza mediante un proceso de ventana deslizante sobre vídeos de partidos, generando probabilidades de ocurrencia para cada evento. Estas predicciones son sometidas a una rigurosa fase de post-procesamiento, aplicando umbrales de confianza y técnicas de filtrado para destilar los eventos más significativos y obtener sus marcas temporales precisas. Finalmente, el sistema culmina con la generación automática de un vídeo resumen dinámico, combinando los clips detectados con efectos visuales y textuales para una experiencia de usuario atractiva.

Los resultados demuestran la efectividad del enfoque para detectar automáticamente eventos clave de alto impacto narrativo y clara firma visual. Este sistema contribuye a la creación eficiente de contenido destacado y ofrece una potente herramienta para el análisis táctico asistido por ordenador, superando las limitaciones de los métodos tradicionales y estableciendo una base sólida para futuras investigaciones en el análisis deportivo inteligente.

Palabras claves: redes neuronales convolucionales 3D, visión por computador, detección de eventos, vídeo resumen, análisis deportivo, aprendizaje por transferencia.

Abstract

This master's thesis (MFP) addresses the development of an intelligent system capable of identifying and contextualizing the most momentous moments in soccer matches. Manual detection of these events is a significant challenge due to its laborious nature, inherent human subjectivity, and high costs. This project proposes an innovative tool that seeks to democratize access to advanced sports analysis technologies.

The system integrates advanced computer vision and artificial intelligence techniques to process soccer videos. The core of the project lies in the detection of key events (goals, yellow cards, substitutions, and corners), a task that requires understanding sequences of actions over time. Deep learning models are used for the spatiotemporal analysis of videos, trained with the SoccerNet dataset. The isometric view of this dataset is essential for the model to learn stable visual patterns and understand the context of each event.

The model's inference is performed using a sliding window process on match videos, generating probabilities of occurrence for each event. These predictions undergo a rigorous post-processing phase, applying confidence thresholds and filtering techniques to distill the most significant events and obtain their precise timestamps.

Finally, the system culminates with the automatic generation of a dynamic summary video, combining the detected clips with visual and textual effects for an engaging user experience.

The results demonstrate the effectiveness of the approach in automatically detecting key events with high narrative impact and clear visual signatures. This system contributes to the efficient creation of outstanding content and offers a powerful tool for computer-assisted tactical analysis, overcoming the limitations of traditional methods and laying a solid foundation for future research in intelligent sports analysis.

Keywords: 3D convolutional neural networks, computer vision, event detection, video summary, sports analysis, transfer learning.

Índice de contenidos

Índice de figuras

Índice de tablas

1. Introducción

El fútbol, como fenómeno global, genera una demanda constante de análisis profundo y contextualizado en todos sus aspectos. Los sistemas tradicionales para obtener esta información dependen en gran medida de la observación y el registro manual por parte de analistas humanos. Esta dependencia, si bien ha sido el pilar del análisis deportivo durante décadas, se encuentra estrechamente relacionada con una serie de limitaciones inherentes, tales como la naturaleza lenta y laboriosa de la tarea, la inherente subjetividad humana y los altos costos asociados. El presente trabajo de fin de máster (TFM) aborda esta problemática, posicionándose como una herramienta innovadora que busca democratizar el acceso a tecnologías avanzadas de análisis deportivo, ofreciendo una alternativa asequible frente a las soluciones comerciales existentes.

Este proyecto se centra en el desarrollo de un sistema autónomo e inteligente capaz de identificar y contextualizar los momentos más trascendentales en partidos de fútbol a partir del análisis de su contenido en vídeo. La finalidad última de este sistema es proporcionar una herramienta eficaz que contribuya al análisis táctico asistido por ordenador, siendo capaz de generar reportes precisos de los eventos detectados y, fundamentalmente, la generación automática de un vídeo resumen dinámico, combinando clips destacados con efectos visuales y textuales para una experiencia de usuario atractiva.

El objetivo general que guía este trabajo es diseñar e implementar un sistema completo y funcional capaz de detectar automáticamente eventos clave en partidos de fútbol mediante la aplicación de avanzadas técnicas de visión por computador e inteligencia artificial. Para lograrlo, el trabajo contempla el desarrollo de un módulo de preprocesamiento de vídeo que adapta la entrada, utilizando la vista isométrica del dataset SoccerNet para que el modelo aprenda patrones visuales más estables y entienda el contexto de cada evento. Seguidamente, se realiza el entrenamiento y validación de modelos de clasificación de vídeos, empleando redes neuronales convolucionales 3D y técnicas de aprendizaje profundo para detectar eventos complejos (como goles, tarjetas amarillas, sustituciones y saques de esquina) basándose en patrones visuales y temporales. Posteriormente, una rigurosa fase de post-procesamiento

transforma las probabilidades de detección en marcas temporales precisas, aplicando umbrales de confianza y técnicas de filtrado. Como culminación de este proceso, se automatiza la generación del vídeo resumen dinámico. Finalmente, el sistema será evaluado tanto cuantitativa como cualitativamente para validar su utilidad y aplicabilidad en casos de uso reales de análisis deportivo, buscando superar las limitaciones de los métodos tradicionales.
Más allá de los retos técnicos intrínsecos, este trabajo plantea ser una herramienta base sólida orientada a democratizar el acceso a tecnologías de análisis deportivo. Busca ofrecer una alternativa de bajo costo e inteligente, a diferencia de las soluciones comerciales existentes cuyo alto precio las hace inaccesibles para la mayoría de clubes amateur.

1.1. Justificación

La identificación manual de eventos en partidos de fútbol toma mucho tiempo, sin considerar el pre y post procesamiento que debería tener un vídeo de un partido completo
Por muy imparciales que queramos ser como personas, la percepción humana, las emociones y el peso del partido impactan notoriamente en las detecciones resultantes. Los entrenadores de fútbol son menos del 45% precisos en su evaluación posterior al partido sobre los eventos críticos que habían ocurrido [1].

Un sistema automatizado reduciría estas brechas, ofreciendo análisis accesible a equipos amateur y permitiendo generar resúmenes inteligentes. Dejando a un lado las limitaciones de los sistemas tradicionales.
Aunque existen soluciones comerciales, su alto costo las hace inaccesibles para la mayoría de clubes amateur. Por tanto este trabajo brinda una solución asequible mediante técnicas de visión por computador, código abierto y hardware intermedio.

1.2. Problema y finalidad

Problema: El análisis del fútbol, un fenómeno global de alta complejidad y dinamismo, demanda una comprensión profunda y ágil que los métodos tradicionales, basados en la observación y el registro manual, no pueden satisfacer eficientemente. Estos procesos se caracterizan por su lentitud, laboriosidad, inherente subjetividad humana y elevados costos operativos. La identificación manual de eventos clave en un partido consume una cantidad considerable de tiempo, incluso sin considerar las fases de pre y post-procesamiento necesarias para un vídeo completo. Además, la percepción individual y las emociones asociadas al juego pueden introducir un sesgo que compromete la objetividad y consistencia del análisis. Paralelamente, las soluciones comerciales avanzadas, si bien eficaces, suelen ser económicamente inaccesibles para la mayoría de las organizaciones deportivas, limitando su adopción generalizada y creando una barrera significativa para el acceso a herramientas analíticas sofisticadas. Esta situación deja una laguna en el campo, requiriendo alternativas más accesibles y eficientes.

Finalidad: Frente a la problemática expuesta, la finalidad de este TFM es proporcionar una herramienta eficaz que contribuya al análisis táctico asistido por ordenador. Para ello, el objetivo fundamental es el desarrollo de un sistema autónomo e inteligente capaz de identificar y contextualizar los momentos más trascendentales en partidos de fútbol a partir del análisis de su contenido en vídeo. Este sistema busca generar reportes precisos de los eventos detectados y, fundamentalmente, la creación automática de resúmenes dinámicos. Dichos resúmenes combinarán los clips destacados con efectos visuales y textuales para ofrecer una experiencia de usuario clara y atractiva. Al automatizar este proceso y ofrecer una alternativa accesible a las costosas soluciones comerciales, este proyecto busca democratizar el acceso a capacidades analíticas sofisticadas, beneficiando a una audiencia más amplia en el ámbito deportivo y de retransmisiones.

2. Marco teórico

2.1. Herramientas y técnicas

El análisis de vídeo en el ámbito deportivo, particularmente en el fútbol, se ha consolidado como un pilar fundamental para la mejora del rendimiento táctico, la estrategia de juego, y la optimización de las retransmisiones para los aficionados. Este capítulo establece la base teórica sobre la que se construye el presente TFM, abordando desde la relevancia del análisis videográfico en el deporte hasta las técnicas de inteligencia artificial y visión por computador que lo hacen posible, pasando por la importancia de los datos y la justificación de la selección de eventos.

2.1.1. Análisis de vídeo en el deporte

El fútbol, un deporte de alta complejidad y dinamismo global, ha evolucionado más allá de la mera observación a simple vista. Los equipos profesionales y, cada vez más, las categorías semiprofesionales y formativas, buscan una ventaja competitiva mediante la comprensión profunda de los patrones de juego, el rendimiento individual de los jugadores y las estrategias del equipo rival. El análisis de vídeo emerge como la herramienta más potente para desentrañar esta complejidad.

Las aplicaciones del análisis de vídeo en el deporte son diversas y de impacto inmediato. En primer lugar, en el análisis táctico, permite a los entrenadores y analistas revisar partidos para identificar aciertos y errores, tanto propios como del adversario. Esto incluye el estudio de movimientos sin balón, transiciones, presiones defensivas, y la eficacia de jugadas a balón parado. Un sistema que automatice la detección de eventos libera tiempo valioso para que los analistas se enfoquen en la interpretación y la toma de decisiones, en lugar de en la tediosa tarea de revisión manual. En segundo lugar, facilita la generación de resúmenes dinámicos para retransmisiones deportivas y plataformas de contenido. Esto no solo mejora la experiencia del espectador al permitirle consumir rápidamente los momentos más emocionantes de un partido, sino que también ofrece nuevas vías para la creación de contenido multimedia automatizado y personalizado. La automatización en este campo se convierte,

por tanto, en un área activa de investigación con aplicaciones prácticas inmediatas.

2.1.2. Detección de eventos clave

Una tarea también conocida como reconocimiento de acciones en el tiempo. Esta consiste en la identificación precisa del cuándo y qué ocurre en un vídeo de un partido de fútbol, marcando momentos trascendentales.

La dificultad de esta tarea radica en varios factores inherentes a la naturaleza del fútbol y de las retransmisiones de vídeo:

- **Variabilidad de la cámara:** Las cámaras de televisión cambian constantemente de ángulo, zoom y posición, lo que presenta una perspectiva visual inconsistente para el modelo.
- **Oclusiones:** Los jugadores pueden interponerse entre la cámara y el balón o entre otros jugadores, ocultando la acción relevante.
- **Velocidad de la acción:** Muchos eventos son fugaces, ocurriendo en fracciones de segundo, lo que dificulta su detección precisa.
- **Ambigüedad visual:** Algunas acciones pueden parecer similares, y el contexto es crucial para una correcta clasificación. Por ejemplo, un pase largo puede preceder a un fuera de juego, y la señal del árbitro asistente es clave para la detección.

El objetivo no es solo identificar el tipo de evento, sino también su ubicación temporal exacta dentro del vídeo.

2.1.3. Enfoques basados en visión por computador e inteligencia artificial

Para abordar los desafíos de la detección de eventos en vídeo, se emplean técnicas avanzadas de Visión por Computador e Inteligencia Artificial. Estas buscan replicar y superar la capacidad de análisis humano mediante el procesamiento automatizado de información visual.

Un sistema integral para la detección de eventos generalmente combina varias etapas de procesamiento:

- **Detección de objetos:** Inicialmente, es fundamental identificar los elementos básicos presentes en el campo, como jugadores, el balón y el

árbitro. Para esta tarea, modelos de detección de objetos como YOLO (You Only Look Once) son altamente eficientes. Estos modelos están diseñados para localizar y clasificar múltiples objetos en un solo paso, generando "cajas delimitadoras" (bounding boxes) alrededor de cada objeto detectado. Si bien la detección inicial de estos elementos es un paso necesario, cabe destacar que para la tarea de detectar acciones directamente en el dataset SoccerNet, no se requiere el entrenamiento de YOLO, ya que los eventos se clasifican a partir de clips completos y la vista isométrica del dataset permite una comprensión contextual directa.

- **Seguimiento (tracking):** Una vez que los objetos son detectados, es crucial seguir su movimiento a lo largo del tiempo para obtener sus trayectorias. Algoritmos como SORT, DeepSORT, FairMOT o ByteTrack son esenciales para esta función. El seguimiento de la pelota y los jugadores es vital para comprender la dinámica de la jugada y contextualizar los eventos.
- **Proyección al campo (homografía):** Para eventos que dependen de la posición relativa de los elementos en el campo (como el fuera de juego), se utiliza la homografía. Esta técnica permite mapear las posiciones de los jugadores y el balón desde la perspectiva de la cámara a una vista aérea normalizada del campo. Esto proporciona un contexto espacial crucial para el análisis táctico.
- **Reconocimiento de acciones / clasificación temporal:** La inteligencia artificial analiza secuencias de vídeo para identificar la ocurrencia de eventos específicos. Las redes neuronales son la herramienta principal para esta tarea.
    - **Redes neuronales convolucionales 3D:** Modelos como C3D, I3D, o R(2+1)D son especialmente diseñados para el análisis de vídeo. A diferencia de las CNNs 2D que procesan imágenes estáticas, las CNNs 3D procesan el vídeo como un volumen espacio-temporal (ancho, alto y tiempo). Esto les permite aprender características que combinan información espacial (formas, texturas) y temporal (movimiento) de manera conjunta, siendo muy potentes para reconocer acciones como un tiro a puerta o un gesto de tarjeta [15].

- **Modelos basados en CNNs 2D + modelos temporales:** Otra aproximación es extraer características espaciales de cada fotograma individualmente usando una CNN 2D, y luego alimentar la secuencia de estas características a una red recurrente (como LSTM o GRU) o a un Transformer para modelar la dependencia temporal.
- **Modelos "transformer" para vídeo:** Arquitecturas más recientes que adaptan el concepto de los Transformers (originalmente exitosos en procesamiento de lenguaje natural) para procesar secuencias de vídeo, como Video Swin Transformer o Timesformer. Son capaces de capturar dependencias espacio-temporales muy complejas.

2.1.4. La Importancia de los datasets anotados

El desarrollo y entrenamiento de este tipo de modelos (como la detección de acciones en vídeo), dependen críticamente de la disponibilidad de grandes cantidades de datos de entrenamiento de alta calidad y meticulosamente anotados. Sin estos datos, los modelos no tendrían la información necesaria para aprender a reconocer los patrones visuales y temporales asociados a cada evento.

Los datasets deben contener vídeos de partidos de fútbol acompañados de etiquetas temporales precisas que indiquen el instante o la ventana de tiempo en la que ocurre cada evento de interés. La calidad de estas anotaciones es tan importante como la cantidad de vídeo, ya que los errores en el etiquetado pueden confundir al modelo durante el entrenamiento. La gestión de estos datasets, incluyendo el etiquetado manual si es necesario, puede ser una tarea laboriosa, para lo cual herramientas como Roboflow son útiles, aunque en este proyecto se aprovecha un dataset pre-anotado.

2.1.5. El dataset SoccerNet

La elección del dataset SoccerNet es un pilar metodológico fundamental y estratégico. Este conjunto de datos ha sido meticulosamente diseñado y curado específicamente para la tarea de reconocimiento de acciones en el tiempo en el contexto del fútbol. Ofrece un recurso robusto y relevante al proporcionar 500 partidos completos de fútbol profesional, acompañados de anotaciones detalladas de eventos clave.

Ofrece vídeos y archivos .json (etiquetado de acciones) por cada partido, pudiendo elegir entre tamaños de 224p o 720p.
El formato de las anotaciones incluye la etiqueta del evento y el timestamp preciso de su ocurrencia, lo que permite entrenar un modelo para identificar cuándo sucede cada acción.

Una característica distintiva y crucial de SoccerNet es la vista isométrica de los vídeos. Esta perspectiva ofrece ventajas y desventajas significativas que serán detalladas y estudiadas más adelante.

La selección de SoccerNet se justifica plenamente debido a su diseño específico para la tarea de detección de acciones y la riqueza de sus datos curados, que proporcionan una base sólida para el entrenamiento y validación del modelo [9].

2.2. Estado del arte

Tradicionalmente, la investigación en la detección de eventos en vídeos deportivos se ha concentrado de manera exclusiva en la modalidad visual. Sin embargo, para explorar el potencial de sistemas de alto rendimiento, estudios como el pionero realizado por Nergård Rongved (2021), en adelante "el estudio de Nergård", se adentran en la evaluación de enfoques multimodales, combinando información visual y de audio para la detección y clasificación de eventos [13]. Aunque el estudio de Nergård aborda la sinergia de ambas modalidades, el presente TFM, dadas las delimitaciones de su alcance y objetivos específicos, se centra exclusivamente en el análisis de la modalidad visual para la detección de eventos. No obstante, se reconoce la importancia de este tipo de investigaciones para el avance del campo.

El estudio de Nergård empleó el conjunto de datos SoccerNet, una de las bases de datos de referencia en el análisis de fútbol, que incluye 500 partidos completos de ligas europeas con miles de anotaciones de eventos clave.
Para lograr la detección y clasificación de eventos, Nergård investigó y combinó diversos modelos de aprendizaje profundo:

- **Modelos visuales:** Evaluaron modelos de última generación basados en redes neuronales convolucionales. Entre ellos, destacan el modelo CALF (Context-Aware Loss Function), reconocido por su enfoque en el contexto temporal alrededor de las acciones, y un modelo CNN 3D basado en ResNet, pre-entrenado en el dataset Kinetics-400 para capturar características espacio-temporales del vídeo. Además, se incluyó un modelo CNN 2D como referencia. Estos modelos son capaces de analizar el vídeo como un volumen espacio-temporal, permitiendo identificar movimientos y formas de manera conjunta.

- **Modelo de audio:** Se desarrolló un modelo que transforma las señales de audio en espectrogramas Log-Mel, una representación visual del sonido, que luego son analizados mediante una red ResNet 2D. Este enfoque permite al sistema capturar patrones de sonido asociados a los eventos.

La fusión de las modalidades (visual y audio) fue un aspecto central del estudio, explorando dos estrategias principales:

- **Fusión temprana:** Consistió en la concatenación de las características extraídas de las modalidades visual y de audio antes de que el modelo realizara el proceso de clasificación. Esto permite que el modelo aprenda directamente de una representación combinada de los datos.

- **Fusión tardía:** Se realizó promediando o tomando el valor máximo de las predicciones de probabilidad (softmax) de modelos entrenados de forma independiente para cada modalidad. Este método combina las decisiones de los modelos unimodales para llegar a una predicción final.

Los resultados del estudio de Nergård demuestran de manera contundente el potencial de un enfoque multimodal. Se observó que, para eventos con una firma de audio predecible, como los goles, la combinación de información visual y de audio mejoró consistentemente el rendimiento de la detección. La euforia de la multitud o los comentarios excitados de los narradores son patrones sonoros que preceden o acompañan un gol, lo que ofrece al sistema una valiosa pista adicional. Sin embargo, para otros eventos como las tarjetas o las sustituciones, donde el audio es menos distintivo o predecible, la mejora fue menos clara y dependía de la configuración específica, e incluso en algunos casos, la adición de audio podría resultar perjudicial. A pesar de esto, la investigación subraya que los modelos visuales más avanzados siguen siendo potentes por sí solos y que el beneficio real de la multimodality reside en su aplicación contextualizada.

En definitiva, el estudio de Nergård no sólo validó la utilidad de un sistema inteligente de detección de eventos en fútbol a través del aprendizaje automático, sino que también iluminó la importancia de considerar la sinergia entre diferentes tipos de datos. Este enfoque multimodal, especialmente al aprovechar las características acústicas distintivas de ciertos eventos, sienta las bases para sistemas de análisis deportivo más robustos, precisos y eficientes,

capaces de capturar la riqueza informativa de un partido de fútbol de una manera que las soluciones tradicionales no pueden. Dicha investigación proporciona un marco conceptual y empírico sólido para el presente trabajo, al respaldar la elección de modelos de aprendizaje profundo, el uso del dataset SoccerNet y la hipótesis de que la integración de diferentes fuentes de datos puede conducir a una mejora sustancial en la detección de eventos clave en los partidos de fútbol.

3. Metodología

La presente investigación se fundamenta en una metodología rigurosa y sistemática, diseñada para abordar el desafío de la detección y clasificación automatizada de eventos clave en partidos de fútbol. Adoptando un paradigma de investigación cuantitativo, el proyecto busca generar conocimiento medible, comparable y repetible sobre la ocurrencia de eventos. Además, su naturaleza se define como un diseño no experimental, ya que se centra en el análisis de datos preexistentes (vídeos de partidos de fútbol) para construir y evaluar un sistema inteligente, en lugar de manipular variables en un experimento controlado.
Esta metodología reside en la sinergia entre las técnicas de visión por computador y las avanzadas redes neuronales profundas, lo que configura un enfoque híbrido para el análisis videográfico. La estructura del trabajo se desglosa en fases interconectadas que abarcan desde la preparación de los datos de entrada hasta la generación de un producto final tangible y útil.

3.1. Objetivo general

El objetivo general que guía este trabajo es diseñar e implementar un sistema completo y funcional capaz de detectar automáticamente eventos clave en partidos de fútbol mediante la aplicación de avanzadas técnicas de visión por computador e inteligencia artificial. Este sistema aspira a ir más allá de la mera detección, proporcionando una herramienta eficaz para el análisis táctico asistido por ordenador, capaz de generar reportes precisos de los eventos identificados y, crucialmente, producir vídeos resumen dinámicos de forma automática. Al hacerlo, se busca democratizar el acceso a tecnologías de análisis deportivo sofisticadas, ofreciendo una alternativa inteligente y de bajo costo frente a las soluciones comerciales existentes. Este objetivo general se concibe como una guía constante que asegura la coherencia y el propósito de cada fase del proyecto.

3.2. Objetivos específicos

Para alcanzar el ambicioso objetivo general, el trabajo se articula en una serie de objetivos específicos que representan los pilares fundamentales del sistema. Estos objetivos se han formulado de manera precisa y coherente, alineándose directamente con la problemática identificada y el desarrollo previsto del proyecto. Cada uno de ellos, describe una acción concreta necesaria para la consecución del propósito final:

- **Desarrollar un módulo de preprocesamiento de vídeos:** Este componente inicial es muy importante para adaptar y preparar la información visual y auditiva de los vídeos de fútbol. Se centrará en utilizar la vista isométrica del conjunto de datos SoccerNet, la cual, al ofrecer un ángulo y zoom consistentes, permite al modelo de inteligencia artificial aprender patrones visuales más estables y contextualizar mejor las jugadas. Además, este módulo se encargará de corregir perspectivas, redimensionar los vídeos y aplicar técnicas de aumento de datos para enriquecer el conjunto de entrenamiento y mejorar la robustez del modelo. Una adecuada normalización será el paso final para asegurar la coherencia de la entrada al sistema.

- **Entrenar y validar modelos de clasificación de vídeos:** La esencia de la inteligencia del sistema reside en esta fase, donde se capacitará a los modelos para identificar las acciones principales dentro del juego. Se emplearán redes neuronales convolucionales 3D (CNN 3D) y otras técnicas de aprendizaje profundo, capaces de analizar el vídeo como un volumen espacio-temporal para reconocer eventos complejos como goles, tarjetas amarillas, sustituciones y saques de esquina, basándose en patrones visuales y temporales específicos. El foco principal será la clasificación de los eventos clave a partir de sus firmas visuales características.

- **Implementar una rigurosa fase de post-procesamiento:** Las predicciones iniciales del modelo, aunque indicativas, requieren un refinamiento para convertirse en marcas temporales precisas y fiables.

Este objetivo implica transformar las probabilidades de ocurrencia generadas por el modelo en instantes exactos. Se aplicarán umbrales de confianza por cada clase para filtrar las detecciones, y técnicas de filtrado como la supresión no máxima (NMS), para asegurar que sólo se retengan los picos de probabilidad más significativos dentro de una ventana de tiempo definida. Este proceso es fundamental para destilar la información más relevante y eliminar el "ruido" de las predicciones.

- **Generar el vídeo resumen:** Es el producto final del TFM, en base a las predicciones destiladas que marcan un tiempo (inicio y fin) de un evento, se agregan efectos de texto, imágenes y animaciones superpuestas sobre el vídeo inferido. Cada clase (evento) tiene sus propios efectos y recursos a utilizarse.

3.3. Hipótesis

El presente TFM se fundamenta en la problemática inherente a la identificación manual de eventos clave en partidos de fútbol. La naturaleza laboriosa, la subjetividad humana y los elevados costos asociados a los métodos tradicionales demandan una solución automatizada que democratice el acceso a análisis deportivos avanzados. En este contexto, la investigación persigue validar una proposición fundamental que guiará el desarrollo del sistema.

- **Hipótesis principal:** Se hipotetiza que un sistema basado en técnicas avanzadas de visión por computador e inteligencia artificial, que aproveche modelos de aprendizaje profundo convolucionales 3D entrenados con el dataset SoccerNet, sea capaz de detectar automáticamente eventos clave de alto impacto narrativo y clara firma visual en partidos de fútbol, superando las limitaciones de los métodos tradicionales en términos de eficiencia, objetividad y accesibilidad.

3.4. Recursos requeridos

El desarrollo e implementación del sistema ha requerido la confluencia estratégica de diversos recursos técnicos, humanos y materiales. La selección de estas herramientas se ha guiado por principios de eficiencia, robustez, accesibilidad y la naturaleza del proyecto.

Recursos técnicos:

- **Núcleo de inteligencia artificial y visión por computador:**
  - **PyTorch:** Como framework principal de aprendizaje profundo, ha sido indispensable para la construcción, entrenamiento y evaluación de las redes neuronales. Ha permitido la implementación de arquitecturas complejas y la gestión del proceso de aprendizaje de los modelos.
  - **Modelos Convolucionales 3D:** Se exploró y entrenó modelos de reconocimiento de vídeo desde cero (SimpleCNN3D) y también se adoptó modelos pre-entrenados como R(2+1)D_18 de la familia ResNet para el aprendizaje por transferencia (transfer learning). Estos modelos son fundamentales para analizar el vídeo como un volumen espacio-temporal, permitiendo al sistema comprender el movimiento y las formas que definen los eventos clave en el fútbol. La capacidad de estos modelos para aprender directamente de los clips de vídeo es la clave para la detección de acciones [11].

- **Procesamiento y composición de vídeo:**
  - **MoviePy:** Esta versátil librería de Python, que opera como una interfaz amigable para FFmpeg, ha sido la columna vertebral de la fase 1 (generación y recortes de eventos sobre los partidos crudos de SoccerNet) y fase 4 (generación del vídeo resumen). Permite la extracción precisa de clips correspondientes a los eventos detectados, la aplicación de efectos visuales (como cámara lenta, transición dinámica, parpadeo, congelamiento de área, reproducción inversa, superposición de imágenes y animaciones,

entre otros), y la composición final de un vídeo resumen dinámico y atractivo.

  - **FFmpeg:** Aunque opera en segundo plano, esta herramienta de código abierto es el motor de procesamiento de vídeo subyacente a MoviePy, facilitando la decodificación, el corte y la codificación de los archivos de vídeo.

- **Análisis y gestión de datos:**
  - **OpenCV:** Principalmente utilizada para pruebas de concepto, tareas de procesamiento de imágenes y manejo de vídeos a un nivel más fundamental, asegurando la compatibilidad y manipulación de los fotogramas del vídeo.
  - **Scikit-learn:** Su módulo "model_selection" ha sido trascendental para la división estratificada de los datos en conjuntos de entrenamiento, validación y prueba, garantizando una evaluación imparcial del modelo. Además, su módulo "metrics" ha permitido la obtención de métricas de evaluación avanzadas (precisión, recall, F1-score y matriz de confusión), ofreciendo una visión profunda del rendimiento del modelo más allá de la simple exactitud.

- **Python:** Como lenguaje de programación principal, Python ha proporcionado el entorno para integrar todas estas librerías, desarrollar la lógica del sistema y construir una arquitectura de código modular y reutilizable (a través de scripts como: dataset_soccernet.py, transforms.py, engine_training.py, video_resumen_engine.py, entre otros).

- **Librerías auxiliares:** "tqdm" para barras de progreso, y los módulos logging, sys, os, numpy y pandas para el manejo de logs, operaciones del sistema y manipulación de datos.

- **Dataset SoccerNet:** Este es el dataset principal a utilizar. Contiene 500 partidos completos de fútbol ya etiquetados.

- **Tensores:** Como buena práctica y dado los recursos limitados para el entrenamiento (falta de GPU potente), se pretendió pasar los clips recortados a tensores de PyTorch (los tensores vienen a ser el input refinado que alimenta el proceso de entrenamiento). Google Colab, por ejemplo, tarda más en cargar los vídeos desde Google Drive y también se puede utilizar TensorBoard para el monitoreo del entrenamiento. Para la versión final se retiró el script que realiza esta tarea, dado que los tensores ocupaban demasiado espacio y no se contaba con grandes almacenes disponibles; este script se puede revisar en el histórico de commits del repositorio.

- **Data augmentation:** Se pretende aplicar al momento de entrenar el modelo para tener más aleatoriedad. Es muy importante mencionar que el sistema maneja transformaciones tanto para 1 canal (escala de grises) como para 3 canales (RGB). Transformaciones como:
  - Volteo horizontal.
  - Brillo, contraste y saturación.
  - Rotaciones pequeñas.
  - Recortes aleatorios y redimensionamiento.
  - Normalización.

Recursos Humanos:

- El Autor del presente TFM (Cesáreo Tello Sánchez).
- El Director del TFM (Miguel Ángel Torres Font).

Recursos Materiales:

- Laptop ThinkPad P16s Gen 3: Una estación de trabajo con CPU Intel, utilizada para el desarrollo inicial del código, depuración, y pruebas con cargas de datos reducidas. Este entorno permitió validar la lógica del pipeline del sistema antes de escalar a recursos de mayor rendimiento.

- PC de la UEV: Una estación de trabajo con GPU Nvidia de 16 GB de VRAM, esencial para el entrenamiento de modelos de Deep Learning con grandes

volúmenes de datos. La disponibilidad de VRAM en esta PC fue de gran importancia para manejar el tamaño de los lotes y las activaciones intermedias de modelos como el R(2+1)D_18.

Posibles licencias de software:

Se ha priorizado el uso de herramientas de código abierto, alineándose con el objetivo de ofrecer una solución accesible y de bajo costo.

Hasta la fecha, el proyecto se ha desarrollado enteramente con software de código abierto, incluyendo Python y todas las librerías mencionadas, sin necesidad de adquirir licencias de pago.

3.5. Presupuesto

El presente apartado tiene como finalidad cuantificar los recursos económicos estimados para el desarrollo y la culminación del sistema. Aunque este TFM se enmarca en un contexto académico, la estimación de un presupuesto detallado refleja el rigor metodológico y la planificación inherente a un proyecto de esta envergadura. Se ha priorizado la utilización de herramientas y plataformas de código abierto o con modelos de suscripción accesibles, lo que ha permitido optimizar los costes operativos del proyecto.

3.5.1. Fases

Las fases están interconectadas, cada una demandando una dedicación específica que, en su conjunto, suman las 400 horas de trabajo comprometidas por el autor:

- **Fase 1, investigación preliminar y conceptualización (64 horas):** En esta etapa inicial, se realizó una profunda investigación bibliográfica sobre redes neuronales 3D y detección de acciones. Se definieron con precisión el problema, los objetivos y la hipótesis del proyecto. Además, se exploró a fondo el dataset SoccerNet para comprender su estructura y el formato

de las anotaciones, sentando las bases teóricas y metodológicas del trabajo.

- **Fase 2, diseño de arquitectura de código y planificación (56 horas):** Durante esta fase, se diseñó una arquitectura de software modular y escalable para garantizar la mantenibilidad del código. Se definió el pipeline técnico completo, desde la detección de elementos hasta la clasificación de eventos. Finalmente, se elaboró un cronograma detallado con hitos y fechas de entrega para gestionar eficazmente el progreso del proyecto.

- **Fase 3, preparación y preprocesamiento de datos (88 horas):** Esta etapa se centró en el análisis exploratorio del dataset SoccerNet para justificar la selección de las 4 clases de eventos clave. Se implementó un módulo para la carga eficiente de vídeos, aplicando técnicas de data augmentation para mejorar el entrenamiento. También se optimizó la gestión de los datos para asegurar la representatividad de las clases y el uso eficiente de la memoria.

- **Fase 4, implementación y entrenamiento de modelos (96 horas):** Aquí se desarrolló y optimizó un modelo propio de CNN 3D. Se implementó un segundo modelo mediante Transfer Learning, ajustándolo para la detección de eventos de fútbol. Además, se creó un motor de entrenamiento genérico para gestionar el proceso, el guardado de progreso y el registro de métricas de rendimiento.

- **Fase 5, inferencia y post-procesamiento (48 horas):** En esta fase, se implementó un script para aplicar los modelos entrenados a partidos completos usando una técnica de ventana deslizante. Se desarrolló un algoritmo de post-procesamiento para refinar las predicciones, agrupar detecciones consecutivas y filtrar resultados ruidosos. Se ajustaron los parámetros de manera iterativa para equilibrar la precisión y la exhaustividad de los resultados.

- **Fase 6, generación del vídeo resumen y efectos visuales (48 horas):** La última etapa consistió en desarrollar un motor de composición de vídeo para crear un resumen dinámico a partir de los eventos detectados. Se diseñaron y aplicaron efectos visuales y textuales personalizados para cada tipo de evento, como cambios de velocidad y animaciones. Por último, se optimizó el proceso de renderizado para asegurar la calidad y estabilidad del vídeo final.

	Abril 2025										Mayo 2025								
	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9
Fase 1																			
Fase 2																			

	Mayo 2025																			
	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Fase 3																						
Fase 4																						

	Junio 2025																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Fase 4																											
Fase 5																											
Fase 6																											

Tabla 1: Diagrama de Gantt.

Fuente: Elaboración propia.

3.5.2. Costes humanos

Representan la inversión principal en cualquier proyecto de investigación y desarrollo, reflejando el valor del conocimiento, la experiencia y el tiempo dedicado. Se estima la participación de los siguientes roles:

- **Autor del TFM:** Se ha dedicado un esfuerzo intensivo a lo largo de la fase de investigación, desarrollo, implementación y redacción. Considerando una dedicación equivalente a 400 horas de trabajo directo en las actividades clave del proyecto, y asignando una tarifa estimada de 20.00 €/hora para un perfil de investigador en inteligencia artificial, el coste directo asciende a 8,000.00 €.

- **Director del TFM:** El rol de dirección y supervisión académica ha sido fundamental, proporcionando orientación estratégica, feedback y apoyo metodológico. Estimando una dedicación de 40 horas a lo largo del proyecto, con una tarifa de 50.00 €/hora para un perfil de investigador senior, el coste directo asciende a 2,000.00 €.

Subtotal coste humano directo: 10,000.00 €

Para reflejar el coste total asociado al personal en un contexto laboral real en España, se aplica un incremento del 21% en concepto de seguridad social:

- Incremento de 21%: 10,000.00 € * 0.21 = 2,100.00 €

- Total coste humano estimado: 10,000.00 € + 2,100.00 € = 12,100.00 €

3.5.3. Costes materiales y tecnológicos

Esta categoría engloba los recursos tangibles y las plataformas necesarias para la ejecución del proyecto. Se ha buscado minimizar la inversión mediante la selección de alternativas eficientes.

- **Hardware personal (Laptop ThinkPad P16s Gen 3):** Al ser una herramienta que se adquirió para estudiar el máster, se incluye en el costo; aunque no fue una inversión ideal porque no se supo de la importancia de tener GPU. Costo del hardware: 1,700.00 €.

- **Google Colab Pro:** Este servicio fue inicialmente elegido para tareas de entrenamiento de modelos, facilitando el acceso a recursos computacionales de alto rendimiento; sin embargo los resultados esperados no fueron satisfactorios. Una única suscripción de 12.00 €.

- **Hardware de la UEV (PC con GPU):** Al ser un equipo propio de la UEV, no se imputa un coste de adquisición directo al proyecto.

- **Software y librerías:** El proyecto se ha construido enteramente sobre la base de software de código abierto, incluyendo lenguajes de programación y librerías. Esta elección elimina cualquier coste directo de licencia, fomentando la accesibilidad y la replicabilidad del proyecto.

Total coste material y tecnológico estimado: 1,712.00 €

3.5.4. Presupuesto total

La suma de los costes humanos y materiales tecnológicos permite obtener una estimación global de la inversión necesaria para el desarrollo de este TFM.
Presupuesto total estimado: 12,100.00 € (humanos) + 1,712.00 € (materiales tecnológicos) = 13,812.00 €

Esta cifra representa una estimación razonable de los recursos necesarios para un proyecto de estas características, destacando la eficiencia lograda mediante la adopción de soluciones de software libre y plataformas en la nube de coste controlado, lo cual se alinea con el objetivo de democratizar el acceso a tecnologías avanzadas de análisis deportivo.

Concepto	**Costo unitario**	**Unidades empleadas**	**Sub total**
Autor del TFM	20.00 €	400.00	8,000.00 €
Director del TFM	50.00 €	40.00	2,000.00 €
Seguridad social (21%)			2,100.00 €
Laptop personal	1,700.00 €	1.00	1,700.00 €
Google Colab Pro	12.00 €	1.00	12.00 €
PC con GPU (UEV)	0.00 €	1.00	0.00 €
Dataset SoccerNet	0.00 €	1.00	0.00 €
Software utilizado	0.00 €	1.00	0.00 €
Total			**13,812.00 €**

Tabla 2: Presupuesto del proyecto.

Fuente: Elaboración propia.

3.6. Diseño y procedimiento

3.6.1. Diseño de investigación

El presente TFM se enmarca como un proyecto de ingeniería que aplica los principios de la investigación cuantitativa para construir y validar una solución tecnológica. Su naturaleza es no experimental, dado que no se manipulan variables independientes ni se realizan intervenciones; en su lugar, se analizan datos preexistentes para identificar patrones y construir un sistema de detección.

La metodología se clasifica como cuantitativa, priorizando la objetividad en la adquisición de información y buscando aumentar el conocimiento comparable, medible y repetible. El fenómeno a estudiar, la ocurrencia de eventos en partidos de fútbol, es externo al investigador y se aborda mediante instrumentos válidos y fiables, en este caso, técnicas de visión por computador y modelos de inteligencia artificial.

3.6.2. Proceso de desarrollo

El proceso central del sistema se basa en la aplicación de modelos de inteligencia artificial especializados en reconocimiento de vídeo. La inferencia del modelo se realiza mediante un proceso de ventana deslizante sobre los vídeos de los partidos. Por cada segmento de vídeo (3 segundos), el modelo predice la probabilidad de ocurrencia de cada tipo de evento de interés.

Una vez obtenidas estas probabilidades, se aplica un post-procesamiento riguroso para transformar las predicciones iniciales en marcas temporales precisas de los eventos. Esto implica la identificación de picos de probabilidad que superan un umbral de confianza predefinido y la aplicación de técnicas de filtrado, como la supresión no máxima (NMS), para discernir el instante exacto en que se considera que ha ocurrido un evento. El resultado de este análisis es una lista de eventos detectados, cada uno con su correspondiente marca temporal precisa [3].

3.6.3. Análisis de datos (pre-procesamiento)

SoccerNet proporciona 500 partidos completos, cada partido se estructura de la siguiente manera:

- **Vídeo:** 2 archivos de 45 min aproximadamente (es decir primera y segunda mitad). Cada partido se ofrece en 2 tamaños, calidad LQ (398 x 224) y calidad HQ (1280 x 720).
- **Json:** 1 archivo de etiquetado de acciones (versión 2) con la siguiente estructura por cada acción:

```
{
      "gameTime": "1 - 13:10",
      "label": "Goal",
      "position": "790722",
      "team": "home",
      "visibility": "visible"
}
```

Cabe mencionar que la librería python "SoccerNet" maneja divisiones de datos: train/valid/test, sin embargo estos no serán tomados en cuenta para el entrenamiento, validación o evaluación del modelo, dado que no analizaremos el partido completo, sino los eventos que serán videoclips de 3 segundos.

A continuación se detallan los pros y contras de la vista isométrica, todos el dataset está compuesto con esta vista:

- **Ventajas de la vista isométrica:**

    - **Consistencia:** A diferencia de los ángulos y zooms variables de las cámaras de retransmisión televisiva, la vista isométrica mantiene un ángulo y zoom más consistentes a lo largo del dataset. Esta uniformidad facilita que el modelo de Inteligencia Artificial aprenda patrones visuales más estables y consistentes para las diferentes acciones.

- **Contexto de la jugada:** Proporciona una visión equilibrada del campo donde ocurre la acción y de las interacciones entre múltiples jugadores. Esto es importante para comprender la dinámica de muchos eventos, permitiendo al modelo captar movimientos en 3D como las trayectorias del balón o los saltos de los jugadores de manera más efectiva que una vista puramente cenital.

- **Disponibilidad y curación:** La existencia de este dataset anotado para 500 partidos, fruto de un minucioso proceso de curación, lo convierte en un recurso listo para usar y entrenar, lo cual es invaluable para este TFM.

- **Desventajas de la vista isométrica:**

  - **Distorsión de perspectiva:** Es la principal limitación frente a una vista cenital pura. Los objetos y las distancias no se escalan uniformemente; un jugador lejano parece más pequeño que uno cercano, dificultando análisis geométricos precisos (por ejemplo, medir distancias exactas o verificar el fuera de juego basándose en la posición del jugador respecto a la línea y el balón).

  - **Visión parcial del campo:** Aunque ofrece contexto, no muestra el campo completo simultáneamente con el mismo nivel de detalle, lo que puede limitar el análisis de eventos que dependen de la posición de jugadores en partes muy distantes del campo.

  - **Dificultad para homografía precisa:** Calcular una homografía perfecta para mapear a una vista cenital es más complejo desde una vista isométrica fija sin ver las líneas completas del campo, a diferencia de una vista donde las líneas son claramente visibles y pueden usarse como puntos de referencia.

3.6.4. Selección de eventos

Lista de acciones etiquetadas (17) que ofrece SoccerNet:

Etiqueta en SoccerNet	Traducido a castellano
Penalty	Penalti
Kick-off	Saque inicial
Goal	Gol
Substitution	Sustitución
Offside	Fuera de juego
Shots on target	Disparos a portería
Shots off target	Disparos desviados
Clearance	Despeje
Ball out of play	Balón fuera de juego
Throw-in	Saque de banda
Foul	Falta
Indirect free-kick	Tiro libre indirecto
Direct free-kick	Tiro libre directo
Corner	Saque de esquina
Yellow card	Tarjeta amarilla
Red card	Tarjeta roja
Yellow->red card	Tarjeta amarilla->roja

Tabla 3: Traducción de etiquetas del dataset.

Fuente: Elaboración propia.

Orden de prioridad de las acciones etiquetadas que el modelo-IA aprenderá: una vez comprendido las ventajas y desventajas de la vista isométrica, debemos centrarnos en las acciones con firmas visuales claras y localizadas, y aquellas que son cruciales en el juego, aunque algunas representan más desafíos.

Se plantean 4 aspectos para agrupar los 17 eventos [14]:

- Eventos de alto impacto o firma visual clara:
    - Gol
    - Tarjeta amarilla
    - Tarjeta roja
    - Tarjeta amarilla->roja
    - Sustitución
    - Saque inicial
- Eventos tipo "jugada preparada":
    - Penalti
    - Saque de esquina
    - Saque de banda
    - Tiro libre directo
    - Tiro libre indirecto
- Eventos de interacción o movimiento del balón:
    - Falta
    - Despeje
    - Disparos a portería
    - Disparos desviados
    - Balón fuera de juego
- Eventos complejos con la vista isométrica:
    - Fuera de juego

Este dataset ofrece una riqueza de información clave, con aproximadamente 109.000 anotaciones distribuidas entre las 17 acciones.
La selección de los eventos específicos a detectar y clasificar (Gol, Tarjeta amarilla, Sustitución, Saque de esquina) no es arbitraria, sino que se deriva de un proceso de filtrado metodológico en tres etapas:

- **Filtro por frecuencia y representatividad de la clase:** Se evaluó la cantidad de instancias de cada evento en el dataset para asegurar que hubiera suficientes datos para un entrenamiento robusto y para priorizar eventos de alto impacto narrativo en el resumen final.

- **Filtro por aislamiento temporal:** Se analizó la "pureza" de la señal visual de cada evento, midiendo su distancia temporal respecto a otras acciones cercanas. Esto permite seleccionar eventos con clips "puros" y menos susceptibles a la ambigüedad, lo que es crucial para la fiabilidad de la clasificación.

- **Filtro de impacto narrativo y claridad de firma visual:** Finalmente, se aplicó un criterio cualitativo para seleccionar eventos que, además de cumplir los filtros anteriores, tuvieran un alto impacto en la dinámica del juego y una firma visual distintiva (ejemplo: el balón en la red para un gol, el gesto arbitral para una tarjeta, etc.). Esta selección equilibra la relevancia (resultado, disciplina, estrategia, peligro) con la viabilidad técnica para la detección automatizada.

3.6.5. Implementación de modelos

Las herramientas utilizadas para la implementación incluyen Python como lenguaje principal, PyTorch como framework de aprendizaje profundo para el entrenamiento y ejecución de las redes neuronales, y OpenCV y MoviePy para el preprocesamiento, manipulación y generación de vídeos. La evaluación de los modelos se realizará utilizando librerías como scikit-learn para el cálculo de métricas detalladas.

Este proyecto se basa en la capacidad de los modelos de Deep Learning para identificar eventos clave en partidos de fútbol. Para ello, se han entrenado y validado dos arquitecturas distintas: una Red Neuronal Convolucional 3D (CNN3D) personalizada entrenada desde cero, y un modelo R(2+1)D_18 pre-entrenado de la familia ResNet adaptado mediante Transfer Learning y fine-tuning granular. Ambas estrategias se han evaluado rigurosamente utilizando métricas de rendimiento avanzadas, vitales para comprender su efectividad en la tarea de clasificación.

Asimismo, librerías como OpenCV y PyAV facilitan el preprocesamiento y la lectura eficiente de vídeos, mientras que MoviePy permite la manipulación y

composición de los clips para generar el vídeo resumen final. Scikit-learn, mediante su módulo "sklearn.model_selection" es utilizada para dividir los datos de manera estratificada y, mediante su módulo "sklearn.metrics" es utilizada para calcular métricas de evaluación detalladas como la matriz de confusión, precisión, recall y F1-score.

Es importante entender las métricas empleadas para una evaluación precisa del rendimiento del modelo:

- **Matriz de Confusión:** Es la base para un análisis detallado. Para cada clase, clasifica las predicciones en cuatro categorías:
  - Verdaderos positivos (VP): El modelo predijo correctamente un evento y realmente ocurrió.
  - Falsos positivos (FP): El modelo predijo un evento, pero no ocurrió (falsa alarma).
  - Falsos negativos (FN): El modelo no predijo un evento, pero sí ocurrió (evento no detectado).
  - Verdaderos negativos (VN): El modelo predijo correctamente que un evento no ocurrió.
- **Precisión (Precision):** Responde a la pregunta: "De todas las veces que el modelo predijo esta clase, ¿cuántas fueron correctas?" Una alta precisión indica pocos falsos positivos. Se calcula como VP / (VP + FP).
- **Exhaustividad (Recall):** Responde a la pregunta: "De todos los eventos reales de esta clase, ¿qué porcentaje el modelo fue capaz de detectar?" Una alta exhaustividad indica pocos falsos negativos. Se calcula como VP / (VP + FN).
- **Puntuación F1 (F1-Score):** Es la media armónica de la Precisión y la Exhaustividad. Es especialmente útil en datasets con desequilibrio de clases (como el de SoccerNet), ya que proporciona una métrica equilibrada que penaliza los modelos que ignoran las clases minoritarias. Se calcula como 2 * (Precisión * Exhaustividad) / (Precisión + Exhaustividad).
- **Soporte (Support):** El número de ocurrencias reales de cada clase en el conjunto de datos evaluado.

- **Promedio macro (Macro Average):** Calcula la métrica (Precisión, Recall, F1-Score) para cada clase y luego promedia esos resultados, dando el mismo peso a todas las clases. Es útil para saber si el modelo es consistentemente bueno en todas las clases, incluidas las menos frecuentes.
- **Promedio Ponderado (Weighted Average):** Similar al promedio macro, pero el promedio se pondera por el número de muestras de cada clase. Las clases con más ejemplos tienen más peso en el promedio final.

3.6.6. Arquitectura del proyecto Python y buenas prácticas

Código fuente (rama: version_1): https://github.com/cesareots/uev-tfm.git

Más allá de la mera implementación de algoritmos, la fortaleza y escalabilidad de un sistema inteligente residen en la rigurosidad de su arquitectura de código fuente. Este TFM no solo ha logrado avances en la detección de eventos y la generación de vídeo, sino que también es un testimonio de un diseño ingenieril meticuloso, concebido para la robustez, la modularidad y la mantenibilidad.
El código, disponible por invitación en GitHub, no es simplemente una colección de scripts, sino una estructura orquestada que refleja un compromiso profundo con las mejores prácticas de desarrollo.
Esta arquitectura se basa en el principio de separación de responsabilidades, donde cada componente del sistema tiene un propósito único y bien definido. Esta filosofía se materializa en una serie de módulos interconectados, cada uno actuando como un engranaje vital en el flujo de trabajo global:

Preparación y preprocesamiento de datos: Esta fase asegura que los datos de vídeo se adapten y optimicen para el entrenamiento del modelo, garantizando una entrada estandarizada y de alta calidad.

- **Organización del dataset:**
  - La clase DatasetSoccernet (ubicada en src/preprocess/dataset_soccernet.py) es la encargada de leer los vídeos directamente desde el disco en formato .mkv.
  - La función crear_dividir_dataset_t_v_t (también en dataset_soccernet.py) gestiona la creación y división estratificada de los datos en conjuntos de entrenamiento, validación y prueba.

- **Aplicación de transformaciones:**
  - Las transformaciones de preprocesamiento y aumento de datos se definen de forma modular en src/models/transforms.py. Funciones como get_transforms_cnn3d_grayscale y get_transforms_resnet encapsulan pipelines de transformaciones.
  - Normalización y formato del tensor: Dentro de DatasetSoccernet.__getitem__, la línea video_tensor_processed.permute(1, 0, 2, 3) es fundamental para asegurar que el tensor de vídeo tenga el formato (C, T, H, W) esperado por las capas nn.Conv3d.
  - Detectar tamaño de salida de transformaciones: La función get_output_size_from_transforms (en transforms.py) permite extraer programáticamente el tamaño de salida final de una pipeline de transformaciones (Resize, RandomResizedCrop, etc.), haciendo el código más robusto y flexible.

Implementación y entrenamiento de modelos: Esta fase abarca la definición de las arquitecturas de red y el ciclo de entrenamiento supervisado.

- **Definición del modelo personalizado (SimpleCNN3D):**
  - La clase SimpleCNN3D (definida en src/models/main_cnn3d.py) implementa una Red Neuronal Convolucional 3D personalizada.
  - Arquitectura mejorada: Incluye capas nn.Conv3d, nn.BatchNorm3d (para estabilizar el entrenamiento), nn.ReLU (para no-linealidad), y nn.Dropout3d (para combatir el sobreajuste).

- Conexión con el clasificador: Utiliza nn.AdaptiveAvgPool3d((1, 1, 1)) antes de las capas densas (nn.Linear) para condensar las características y simplificar la conexión entre el extractor de características y el clasificador.
- Cálculo dinámico de dimensiones: Un extracto clave que demuestra robustez es el cálculo dinámico del tamaño de entrada a la capa lineal (fc) utilizando un tensor ficticio, como se logra con num_features_for_fc = output_shape_features después de AdaptiveAvgPool3d.

- **Implementación del modelo de transfer learning (RESNET):**
  - La función model_r2plus1d_fine_tuning_granular (en src/models/main_resnet.py) gestiona la carga del modelo R(2+1)D_18 pre-entrenado en Kinetics-400.
  - Congelación de capas: Se utiliza para congelar el "backbone" (cuerpo) del modelo pre-entrenado, preservando el conocimiento aprendido.
  - Reemplazo de la cabeza clasificadora: Reemplaza la capa de salida original por una nueva adaptada a las clases, que es entrenable por defecto.
  - Fine-tuning granular y tasa de aprendizaje diferencial: El código permite descongelar selectivamente capas (layer4.1.) y configura el optimizador con params_to_optimize como una lista de diccionarios, aplicando diferentes tasas de aprendizaje (muy bajas para el backbone, más altas para la cabeza).

- **Motor de entrenamiento centralizado:**
  - Las funciones train_model y evaluate_model (en src/models/engine_training.py) son el corazón del ciclo de entrenamiento, diseñadas para ser agnósticas al modelo.
  - Checkpointing robusto: train_model implementa un sistema de guardado de progreso completo, incluyendo model.state_dict(), optimizer.state_dict(), scheduler.state_dict(), la época actual, y la mejor métrica de validación alcanzada (best_val_metric_value).

- Reanudación del entrenamiento: La función extras (también en engine_training.py) gestiona la lógica de reanudación desde un checkpoint, incluyendo la detección automática del dispositivo (map_location=device en torch.load) y la organización de directorios de ejecución.
- Ajuste de la tasa de aprendizaje: train_model integra torch.optim.lr_scheduler.ReduceLROnPlateau que ajusta la tasa de aprendizaje si la métrica de validación se estanca.
- Métricas detalladas: evaluate_model no solo reporta la pérdida y precisión, sino que también utiliza sklearn.metrics.classification_report y sklearn.metrics.confusion_matrix. La matriz de confusión se visualiza con seaborn.heatmap para un análisis rápido y claro del rendimiento del modelo para cada clase.

Inferencia y post-procesamiento: Esta fase aplica el modelo entrenado a vídeos completos para extraer y refinar los eventos clave.

- **Script principal de inferencia:** main_infe.py (ubicado en src/inference/main_infe.py) orquesta todo el proceso de inferencia.

- **Carga de modelo y transformaciones para inferencia:** La función load_model_and_transforms se encarga de cargar el modelo entrenado (.pth) y las transformaciones adecuadas para la inferencia, asegurando que la arquitectura del modelo se reconstruya idénticamente a como fue entrenada (especialmente crucial para RESNET).

- **Técnica de ventana deslizante:** La función run_sliding_window procesa el vídeo largo dividiéndolo en clips superpuestos. Utiliza torchvision.io.read_video para leer eficientemente sólo los segmentos necesarios, evitando cargar todo el vídeo en memoria.

- **Refinamiento de predicciones:** La función post_process_predictions es clave para transformar las predicciones crudas del modelo en una lista de eventos discretos y significativos.
    - Utiliza umbrales de confianza configurables por clase (confidence_thresholds diccionario) para filtrar detecciones dudosas.
    - Realiza agrupación temporal de predicciones consecutivas.
    - Aplica filtrado por duración mínima para eliminar falsos positivos muy cortos.

Generación del vídeo resumen final: Esta es la etapa culminante donde los eventos detectados se materializan en un vídeo dinámico; cumpliendo así uno de los objetivos del TFM.

- **Motor de vídeo resumen:** video_resumen_engine.py (ubicado en src/inference/video_resumen_engine.py) contiene la lógica para componer el vídeo.

- **Composición de clips:** La función create_summary_video utiliza moviepy para extraer y concatenar los segmentos de vídeo de los eventos detectados.

- **Efectos visuales personalizados:**
    - Se aplican efectos específicos a cada tipo de evento (match evento_str:), encapsulados en funciones modulares como fn_multiply_speed_blink (cámara lenta y parpadeo), fn_time_mirror (reproducir al revés), fn_freeze_region (congelar una parte de la pantalla).
    - Superposiciones: Utiliza CompositeVideoClip para superponer elementos visuales:
        - Texto (TextClip) con el nombre del evento y timestamp.
        - Imágenes estáticas con capa alfa (transparencia).

- Animaciones: Se integran pequeños vídeos animados o GIFs, demostrando un control avanzado sobre la composición.

- **Manejo de recursos:** El script asegura la liberación de recursos de los clips de vídeo (.close()) y de FFmpeg en bloques try...finally para evitar fugas de memoria.

3.6.7. Post-procesamiento

Lógica de eventos: Se refinan las predicciones del modelo. Esto incluye filtrar detecciones por umbrales de confianza (por cada clase) [12], agrupar predicciones consecutivas para formar un evento único, y aplicar filtros de "duración mínima" para eliminar ruido o detecciones demasiado cortas.

- **Paso 1:** Texto con timestamp:
  - Se guarda la lista de eventos detectados en un archivo .json, con el siguiente formato:

```
{
      "evento": "Goal",
      "inicio": 450.0,
      "fin": 456.0,
      "confianza_promedio": 0.9735955893993378
}
```

- **Paso 2:** Vídeo resumen:
  - Para cada evento, se extrae un pequeño segmento del vídeo original (el inferido) centrado en el timestamp del evento (ejemplo: extraer los 5 segundos antes y 5 segundos después del timestamp del "Goal").
  - Se concatenan estos segmentos de vídeo en orden cronológico para crear un nuevo archivo de vídeo más corto.
  - Se agregan efectos superpuestos tales como texto, imágenes y animaciones por cada evento.

3.6.8. Evaluación del sistema

La evaluación del rendimiento del sistema se realizará tanto cuantitativamente como cualitativamente.

Las métricas cuantitativas clave serán la precisión y el recall, que permitirán medir la exactitud de las detecciones del sistema y la proporción de eventos reales que son correctamente identificados.

La evaluación cualitativa se centrará en la utilidad práctica y la aplicabilidad del sistema en casos de uso reales de análisis deportivo, asegurando que el producto final cumpla con su propósito de democratizar el acceso a un análisis deportivo avanzado y objetivo.

3.6.9. Visor web interactivo del producto final

Como GitHub no permite almacenar archivos grandes, estos resultados se encuentran en Google Drive. Descargar cualquier directorio y ejecutar su correspondiente HTML:
https://drive.google.com/drive/folders/1eRm5FVHZF_CiTcA_wNRwNqJ5zKTvqJfy?usp=sharing

Para culminar el pipeline de análisis y presentar los resultados de una manera accesible, intuitiva e impactante, se ha desarrollado un visor web interactivo y auto-contenido. Este componente no es una aplicación web que requiera un servidor, sino un generador programático que, utilizando Python, ensambla dinámicamente una interfaz HTML5 a partir de los resultados de la inferencia. El proceso toma como entrada la lista de eventos depurada (en formato JSON) y las rutas a los vídeos correspondientes para producir un único archivo HTML portable.

La interfaz ha sido diseñada con un enfoque en la claridad y la usabilidad, presentando dos módulos de visualización principales que permiten una exploración completa de los resultados del sistema:

- **Módulo de vídeo resumen:** En la parte superior de la página, se presenta un reproductor dedicado exclusivamente al vídeo resumen final generado por el sistema. Esta sección permite al usuario consumir de forma directa y concisa los momentos clave del partido, concatenados y enriquecidos con los efectos visuales implementados, validando así el producto final del proyecto.

- **Módulo de reproductor interactivo:** Presenta el vídeo completo del partido en paralelo a una lista cronológica y navegable de los eventos detectados. Cada elemento en esta lista no es meramente informativo; es un marcador de tiempo funcional. Al hacer clic sobre un evento (por ejemplo, "⚽ Goal" en el minuto 44:15), la reproducción del vídeo principal salta instantáneamente a ese preciso momento, permitiendo al usuario contextualizar la jugada de inmediato, revisar la acción y validar cualitativamente la precisión de la detección del modelo. La presentación se ha enriquecido con iconos distintivos para cada tipo de evento (Goal, Yellow card, Substitution, Corner) y un diseño moderno, asegurando una experiencia de usuario atractiva y profesional.

Este visor cumple un doble propósito fundamental. No solo sirve como una potente herramienta de validación cualitativa que permite auditar visualmente el rendimiento del modelo, sino que materializa el objetivo final del TFM: transformar datos analíticos complejos en un producto final útil, accesible e impactante, cumpliendo con la misión de democratizar el acceso a un análisis deportivo avanzado.

4. Resultados

El desarrollo de este TFM culmina en la implementación exitosa de un sistema inteligente capaz de detectar eventos clave en partidos de fútbol y generar automáticamente vídeos resumen dinámicos. Los resultados obtenidos no solo validan la hipótesis principal de este proyecto (demostrando la efectividad de las técnicas avanzadas de visión por computador e inteligencia artificial para superar las limitaciones de los métodos manuales), sino que también establecen una base sólida para el análisis deportivo asistido por ordenador.

Los logros principales del sistema se estructuran en las siguientes fases:

4.1. Preparación y análisis estratégico de datos

Se realizó un análisis exploratorio de datos (EDA) que está en el directorio "notebooks" del código fuente. Tuvo como objetivo comprender la estructura y características del dataset de SoccerNet, identificar patrones de juego realistas y, fundamentalmente, establecer un pipeline de filtrado de eventos "consistentes" que sirvan de base para el entrenamiento de un modelo CNN 3D orientado a la detección de eventos clave en partidos de fútbol. El dataset original comprende 500 partidos de las 6 principales ligas europeas. Tras descontar 6 partidos para tareas de inferencia y 1 partido con errores de decodificación, se trabajó con un total de 493 partidos.

4.1.1. Filtrado y selección de eventos clave

La fase más crítica del EDA fue el filtrado y selección de eventos clave, considerando la viabilidad para un modelo de detección visual. Se propusieron tres criterios principales:

- **Frecuencia y relevancia:** El análisis de frecuencia de los 17 eventos únicos reveló un marcado desequilibrio de clases. Eventos como 'Ball out of play' (31,379), 'Throw-in' (18,671), y 'Foul' (11,534) dominan el dataset, mientras que eventos como 'Penalty' (170), 'Red card' (55), y 'Yellow->red card' (46) son muy raros. Este desequilibrio justificó la exclusión de eventos de muy alta y muy baja frecuencia.
-

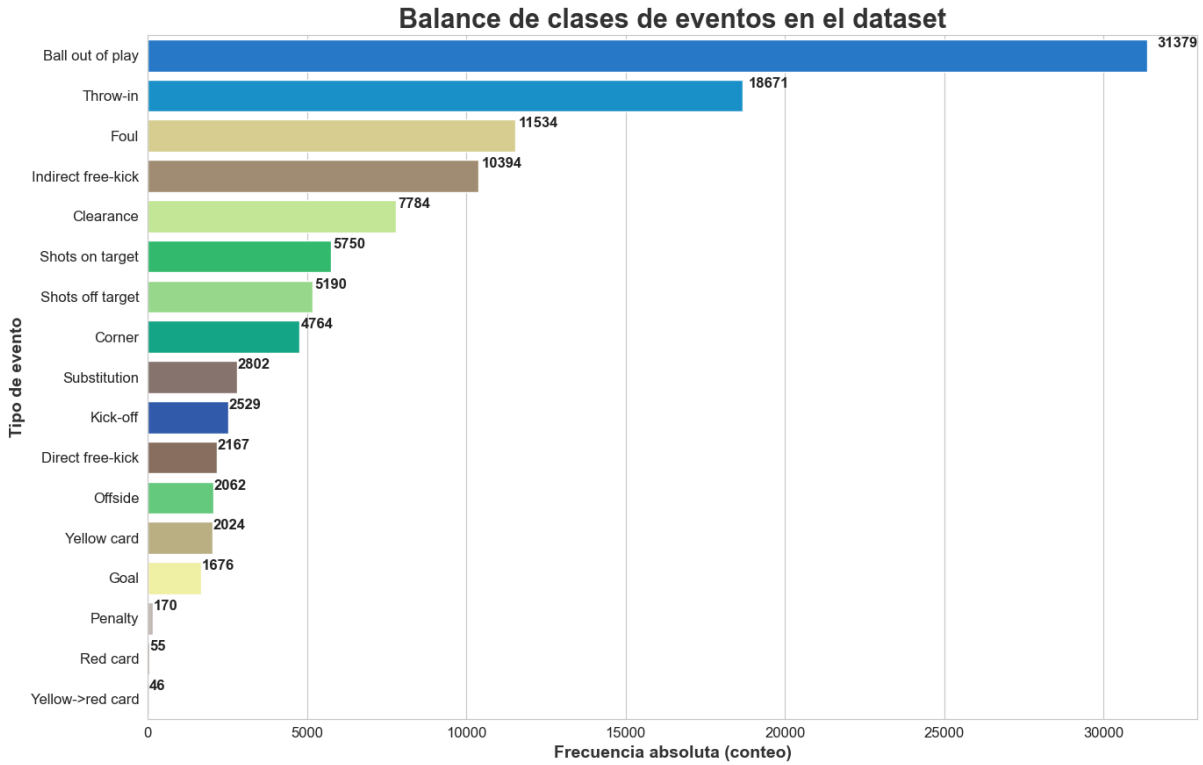

Figura 1: Frecuencia de acciones de SoccerNet.

Fuente: Elaboración propia (jupyter notebook).

- **Visibilidad:** Se analizó el porcentaje de ocurrencia de cada evento que fue etiquetado como "visible". Este análisis es más que obvio para un modelo visual. Se observó que algunos eventos, a pesar de su frecuencia, tenían un porcentaje significativo de ocurrencias no visibles. Eventos como 'Kick-off', 'Clearance', y 'Indirect free-kick' mostraron porcentajes de no visibilidad que justificaron su exclusión, ya que un modelo visual tendría dificultades para aprender a detectarlos consistentemente.

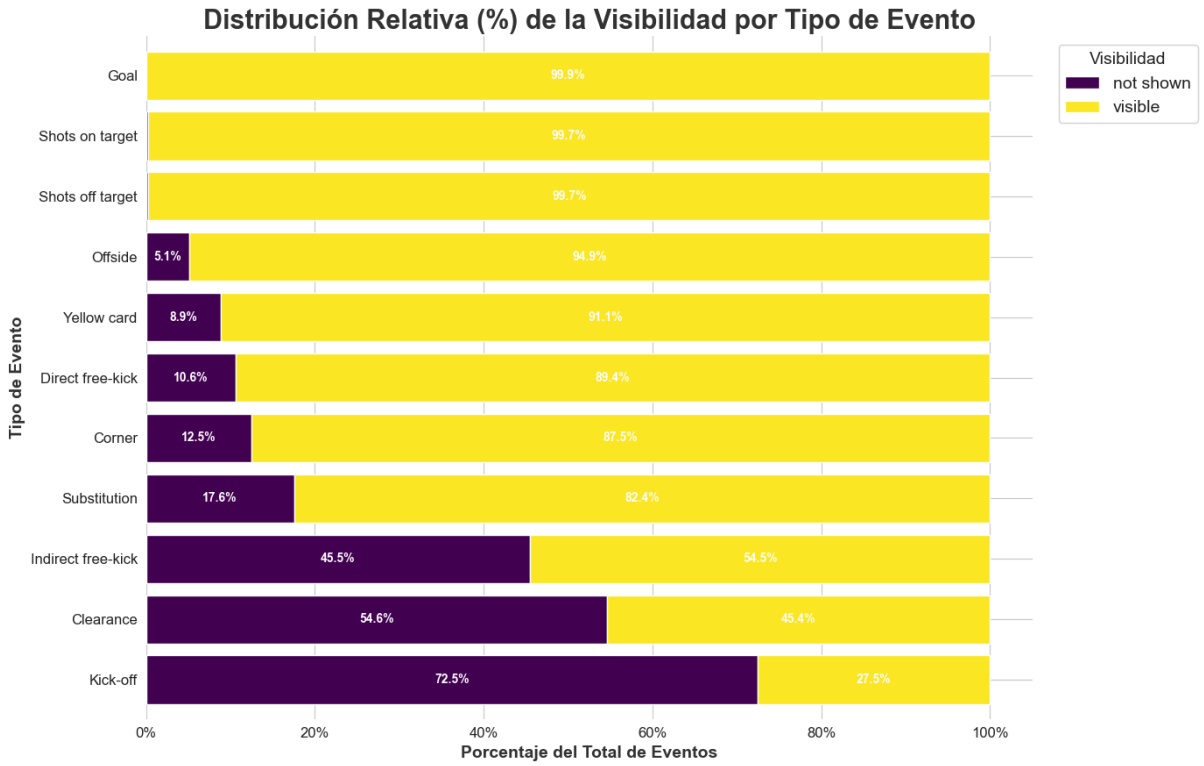

Figura 2: Visibilidad de acciones de SoccerNet.
Fuente: Elaboración propia (jupyter notebook).

- **Aislamiento temporal:** Se calculó el tiempo transcurrido entre un evento y su evento más cercano en el tiempo. Un mayor aislamiento temporal sugiere que el evento ocurre en un momento "limpio" del partido, facilitando su detección visual. Un boxplot mostró la distribución del aislamiento temporal por evento. Eventos con menor aislamiento, como 'Shots on target' y 'Direct free-kick', tendían a ocurrir en secuencias rápidas con otros eventos, lo que dificultaría su discriminación visual. Estos eventos fueron excluidos.

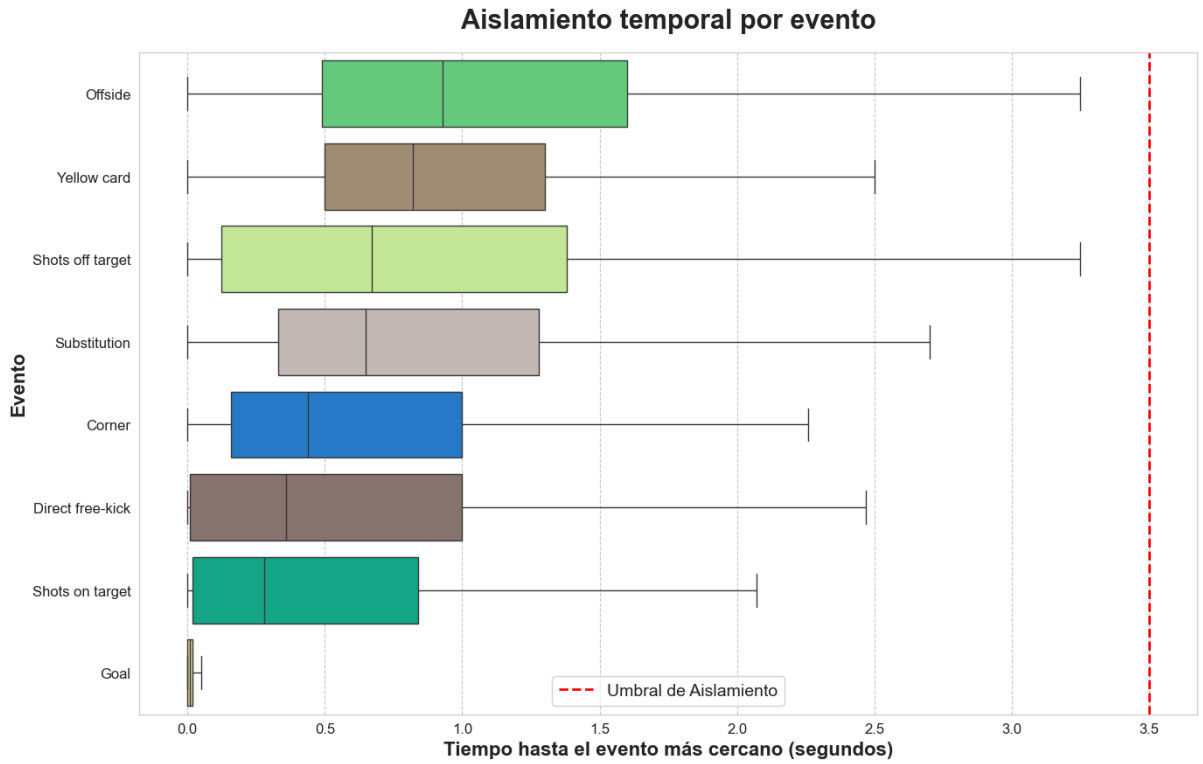

Figura 3: Aislamiento temporal de acciones de SoccerNet.
Fuente: Elaboración propia (jupyter notebook).

- **Justificando los 4 eventos finales:** Tras aplicar los criterios de filtrado y en los criterios de la agrupación de aspectos (sección: diseño y procedimiento), el conjunto final de eventos seleccionados se redujo a 4 clases: 'Goal', 'Yellow card', 'Substitution' y 'Corner'. Estos eventos fueron elegidos por su combinación de frecuencia suficiente para el entrenamiento, alto porcentaje de visibilidad, mayor aislamiento temporal y relevancia para un vídeo resumen de los momentos clave del partido. El evento 'Goal' fue considerado un evento no negociable debido a su impacto y relevancia, a pesar de tener una frecuencia moderada.

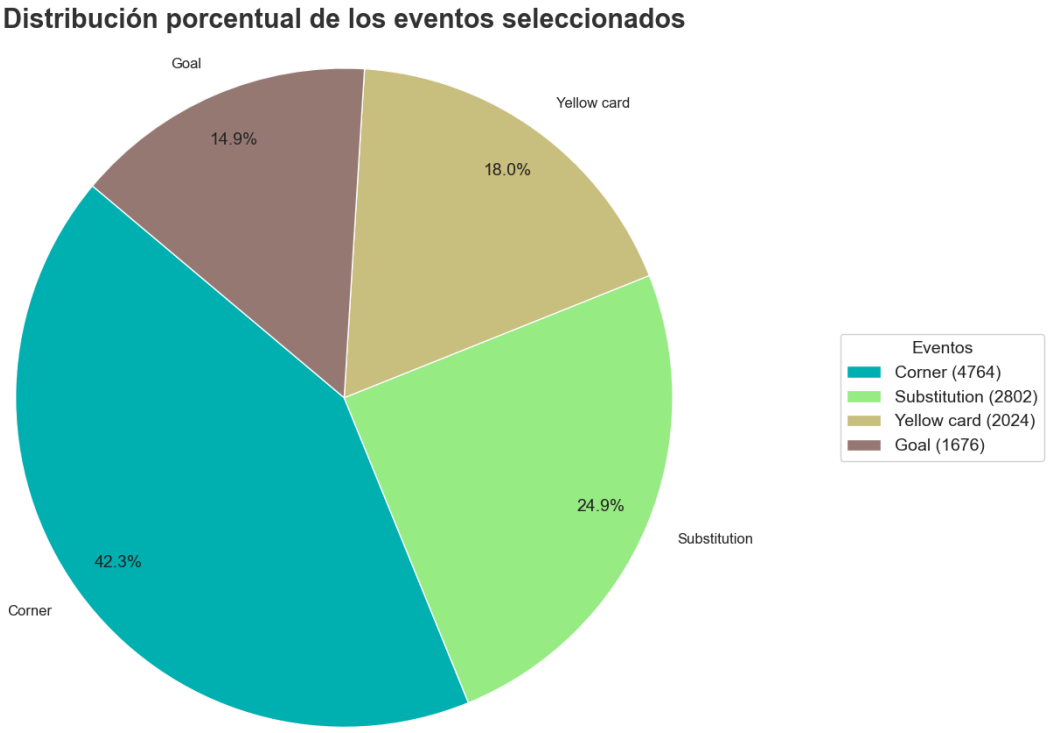

Figura 4: Eventos finalistas para el proyecto.
Fuente: Elaboración propia (jupyter notebook).

4.2. Eventos con transformaciones aplicadas

Mediante el script "src/models/main_transforms_visualize.py" se generaron las siguientes muestras.

- **Gol:**

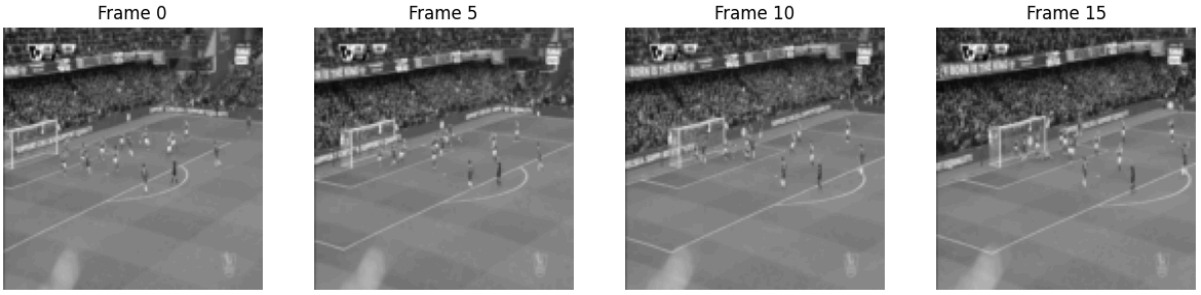

Figura 5: Transformaciones "Gol" modelo CNN3D (escala grises).
Fuente: Elaboración propia.

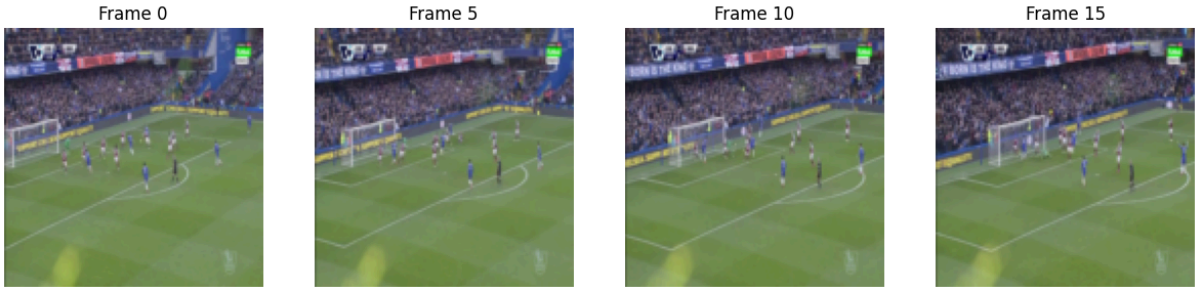

Figura 6: Transformaciones "Gol" modelo CNN3D (rgb).
Fuente: Elaboración propia.

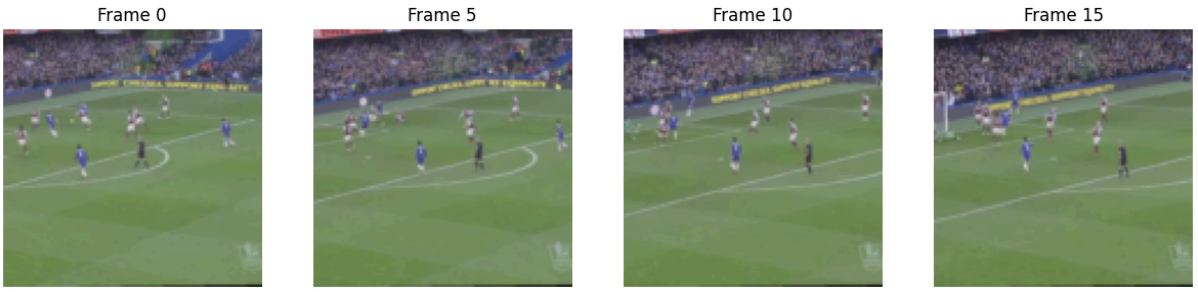

Figura 7: Transformaciones "Gol" modelo RESNET (rgb).
Fuente: Elaboración propia.

- **Tarjeta amarilla:**

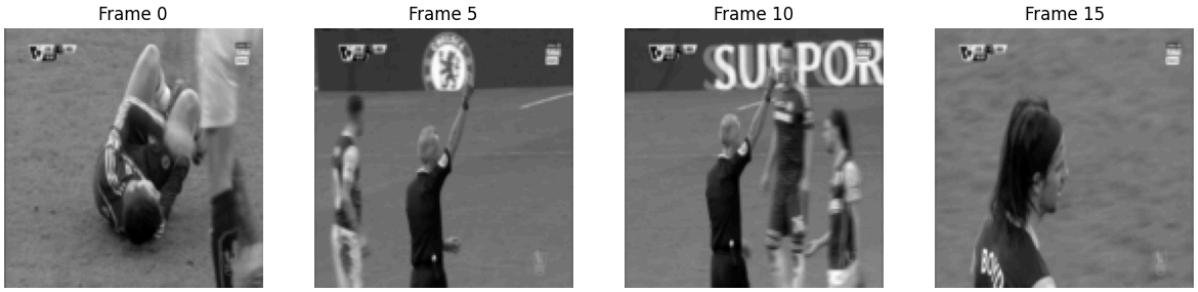

Figura 8: Transformaciones "Tarjeta amarilla" modelo CNN3D (escala grises).

Fuente: Elaboración propia.

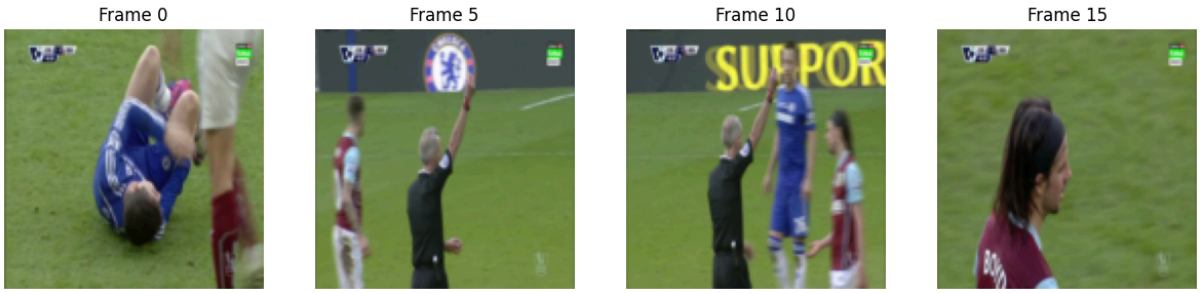

Figura 9: Transformaciones "Tarjeta amarilla" modelo CNN3D (rgb).

Fuente: Elaboración propia.

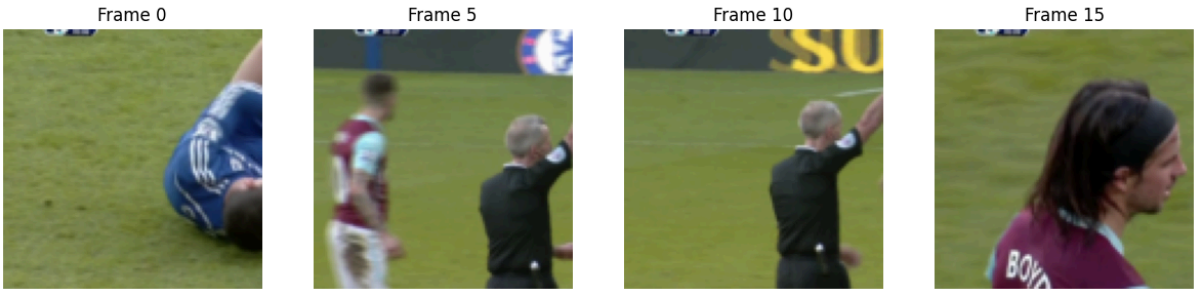

Figura 10: Transformaciones "Tarjeta amarilla" modelo RESNET (rgb).

Fuente: Elaboración propia.

- **Sustitución:**

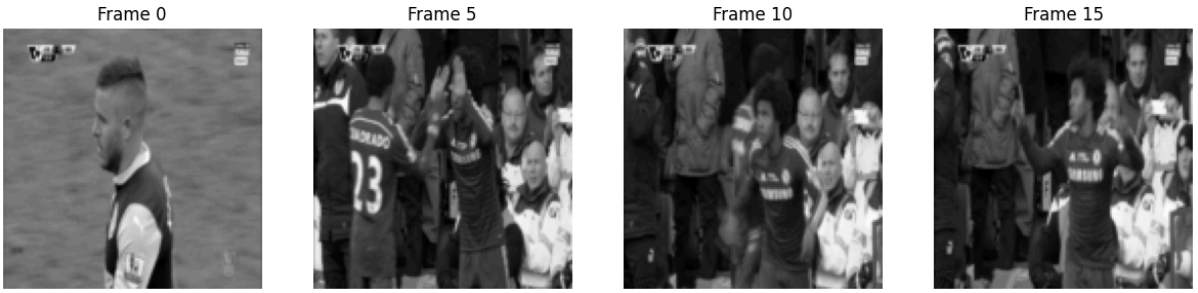

Figura 11: Transformaciones "Sustitución" modelo CNN3D (escala grises).

Fuente: Elaboración propia.

Figura 12: Transformaciones "Sustitución" modelo CNN3D (rgb).

Fuente: Elaboración propia.

Figura 13: Transformaciones "Sustitución" modelo RESNET (rgb).

Fuente: Elaboración propia.

- **Saque de esquina:**

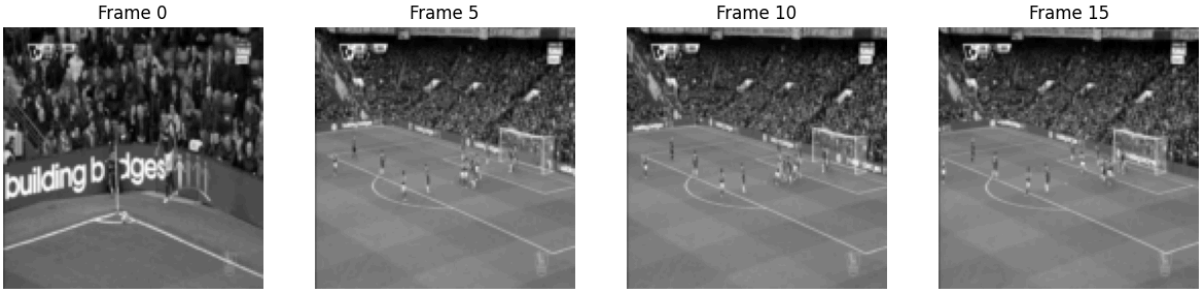

Figura 14: Transformaciones "Saque de esquina" modelo CNN3D (escala grises).

Fuente: Elaboración propia.

Figura 15: Transformaciones "Saque de esquina" modelo CNN3D (rgb).

Fuente: Elaboración propia.

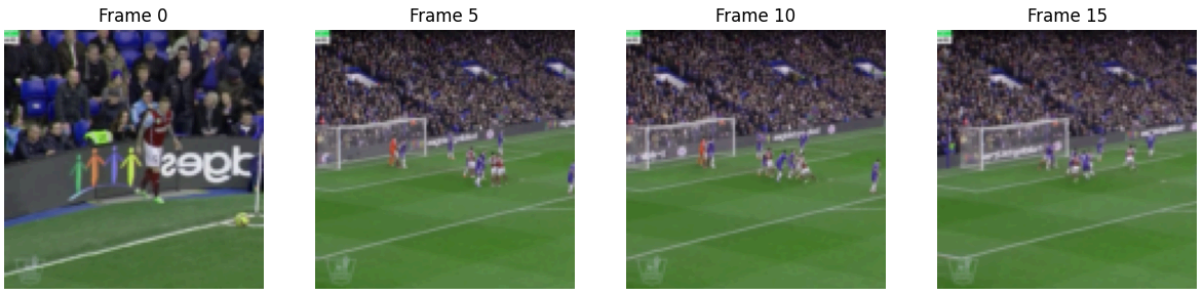

Figura 16: Transformaciones "Saque de esquina" modelo RESNET (rgb).

Fuente: Elaboración propia.

4.3. Resultados del entrenamiento y validación

4.3.1. Modelo CNN3D

El modelo CNN3D personalizado fue entrenado con una configuración que incluía 1 canal (escala de grises) como entrada, un BATCH_SIZE de 64, y un total de 80 épocas, con los checkpoints guardándose cada 5 épocas. El entrenamiento fue monitoreado con un ReduceLROnPlateau con paciencia de 5 y un factor de 0.5, monitoreando la pérdida de validación.

La duración total de las tareas, de entrenamiento y validación para las 80 épocas, fue de aproximadamente 4130.67 minutos (casi 69 horas). La duración promedio por época de entrenamiento fue de alrededor de 51 minutos.

El scheduler ReduceLROnPlateau redujo la tasa de aprendizaje en las épocas 20, 34, 56, 69 y 76, lo que demuestra su capacidad para optimizar la convergencia del modelo.

Los resultados finales de la evaluación en el conjunto de TEST (datos no vistos) para el mejor modelo guardado (model_CNN3D_best.pth) son los siguientes:

Informe de clasificación (sklearn.metrics.classification_report):

	precision	**recall**	**f1-socre**	**support**
Goal	0.9421	0.9306	0.9363	245
Yellow card	0.7325	0.8007	0.7651	301
Substitution	0.8044	0.8005	0.8024	411
Corner	0.9435	0.9130	0.9280	713
accuracy			0.8677	1670
macro avg	0.8556	0.8612	0.8580	1670
weighted avg	0.8710	0.8677	0.8690	1670

Tabla 4: Informe clasificación del modelo CNN3D.

Fuente: Elaboración propia.

Resumen evaluación final: Loss: 0.3712, Accuracy: 0.8677.

Interpretación:

- Rendimiento general: El modelo CNN3D personalizado alcanzó un Accuracy general del 86.77% en el conjunto de prueba. El F1-Score ponderado (weighted avg) fue del 86.90%, lo que indica un buen equilibrio general considerando el número de muestras por clase.
- Rendimiento por clase:
    - Las clases "Goal" y "Corner" muestran un rendimiento excepcional, con F1-Scores superiores al 92%, Precision y Recall muy elevados. Esto sugiere que el modelo es muy capaz de identificar estas acciones con alta fiabilidad y exhaustividad. Esto se alinea con la hipótesis de que eventos de alto impacto narrativo suelen tener una clara firma visual.
    - "Substitution" también presenta un buen desempeño con un F1-Score de 80.24%.
    - La clase "Yellow card" es la que muestra el rendimiento más bajo, con un F1-Score de 0.7651 y una precisión (0.7325) notablemente inferior a su exhaustividad (0.8007). Esto implica que el modelo es relativamente bueno detectando la mayoría de las tarjetas amarillas (alto recall), pero tiende a generar más falsos positivos al predecirlas (menor precision). Este hallazgo justifica la necesidad de utilizar umbrales de confianza por clase en la fase de inferencia para ajustar la compensación entre precisión y exhaustividad.

Matriz de confusión:

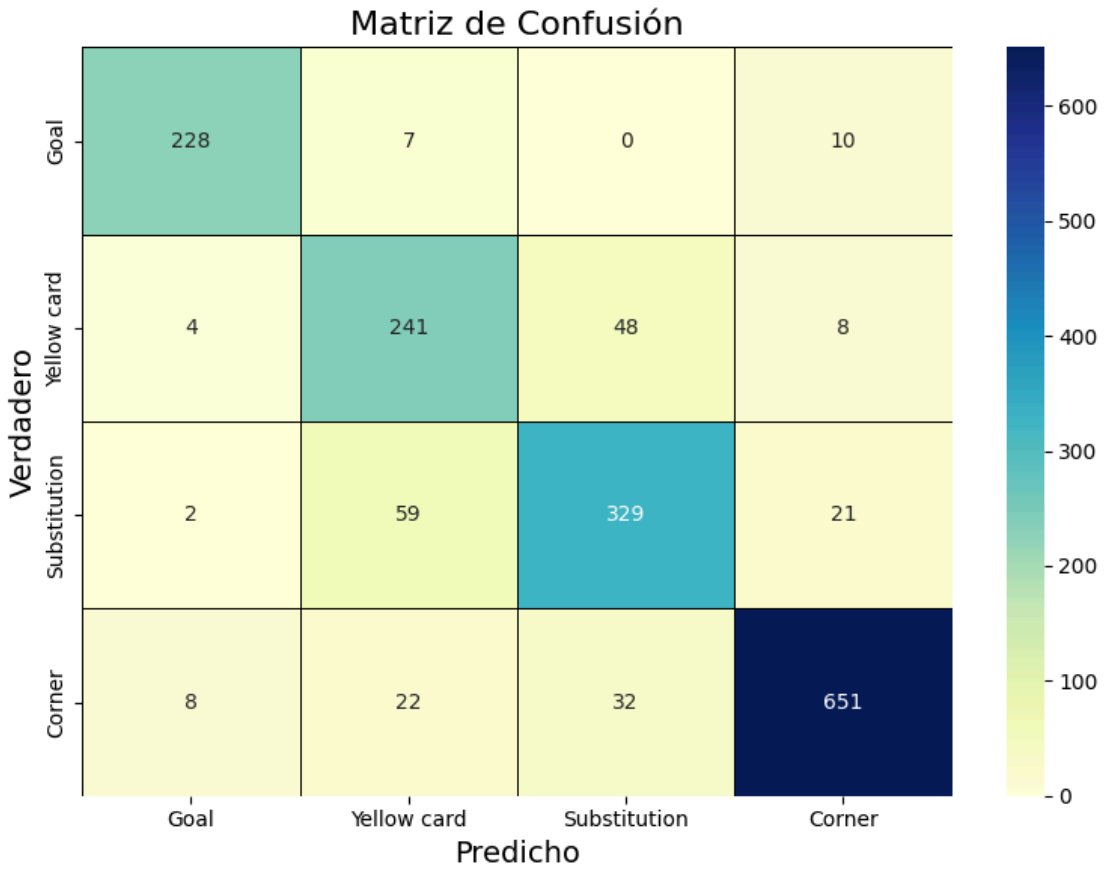

Figura 17: Matriz de confusión del modelo CNN3D.

Fuente: Elaboración propia.

Falsos positivos: ocurre cuando el modelo predice que algo es de una clase específica (positiva), pero en realidad no lo es. Es como una "falsa alarma".

- Goal:
  - El modelo predijo "Goal", pero fue una "Yellow card": 7
  - El modelo predijo "Goal", pero fue una "Substitution": 0
  - El modelo predijo "Goal", pero fue un "Corner": 10
  - **Total Falsos Positivos para "Goal": 7 + 0 + 10 = 17**
- Yellow card:
  - El modelo predijo "Yellow card", pero fue un "Goal": 4
  - El modelo predijo "Yellow card", pero fue una "Substitution": 48
  - El modelo predijo "Yellow card", pero fue un "Corner": 8
  - **Total Falsos Positivos para "Yellow card": 4 + 48 + 8 = 60**
- Substitution:

    - El modelo predijo "Substitution", pero fue un "Goal": 2
    - El modelo predijo "Substitution", pero fue una "Yellow card": 59
    - El modelo predijo "Substitution", pero fue un "Corner": 21
    - **Total Falsos Positivos para "Substitution": 2 + 59 + 21 = 82**
- Corner:
    - El modelo predijo "Corner", pero fue un "Goal": 8
    - El modelo predijo "Corner", pero fue una "Yellow card": 22
    - El modelo predijo "Corner", pero fue una "Substitution": 32
    - **Total Falsos Positivos para "Corner": 8 + 22 + 32 = 62**

Falsos negativos: ocurre cuando el modelo no predice algo que sí es de una clase específica (positiva), pero en realidad sí lo es. Es como una "omisión" o "no detectar" algo que debería haber detectado.

- Goal:
    - Realmente fue "Goal", pero el modelo predijo "Yellow card": 4
    - Realmente fue "Goal", pero el modelo predijo "Substitution": 2
    - Realmente fue "Goal", pero el modelo predijo "Corner": 8
    - **Total Falsos Negativos para "Goal": 4 + 2 + 8 = 14**
- Yellow card:
    - Realmente fue "Yellow card", pero el modelo predijo "Goal": 7
    - Realmente fue "Yellow card", pero el modelo predijo "Substitution": 59
    - Realmente fue "Yellow card", pero el modelo predijo "Corner": 22
    - **Total Falsos Negativos para "Yellow card": 7 + 59 + 22 = 88**
- Substitution:
    - Realmente fue "Substitution", pero el modelo predijo "Goal": 0
    - Realmente fue "Substitution", pero el modelo predijo "Yellow card": 48
    - Realmente fue "Substitution", pero el modelo predijo "Corner": 32
    - **Total Falsos Negativos para "Substitution": 0 + 48 + 32 = 80**
- Corner:
    - Realmente fue "Corner", pero el modelo predijo "Goal": 10
    - Realmente fue "Corner", pero el modelo predijo "Yellow card": 8
    - Realmente fue "Corner", pero el modelo predijo "Substitution": 21

- **Total Falsos Negativos para "Corner": 10 + 8 + 21 = 39**

El modelo está haciendo un buen trabajo al identificar correctamente la mayoría de los eventos, como lo demuestran los altos números en la diagonal principal. Sin embargo, este modelo muestra una confusión notable entre "Tarjetas Amarillas" y "Sustituciones". Este es el principal punto débil:

- El modelo a menudo confunde una "Sustitución" con una "Tarjeta Amarilla" (59 veces) y viceversa, prediciendo una "Tarjeta Amarilla" cuando en realidad es una "Sustitución" (48 veces). Esto sugiere que hay características en los datos que usa el modelo que son similares para estos dos eventos, o que necesita más entrenamiento para distinguirlos mejor.

- También hay una tendencia a confundir "Corners" con "Substitutions", aunque en menor medida.

4.3.2. Modelo RESNET (Transfer Learning)

El modelo RESNET (R(2+1)D_18), utilizando una estrategia de fine-tuning granular (descongelando el último bloque de model.layer4 y la capa model.fc) y pesos pre-entrenados de Kinetics-400, fue entrenado con BATCH_SIZE de 64. Se configuró para 60 épocas, con el optimizador Adam utilizando una tasa de aprendizaje diferencial: 0.00001 para el backbone afinado y 0.001 para la cabeza clasificadora.

La duración total de las tareas, de entrenamiento y validación para las 60 épocas, fue de aproximadamente 7664.33 minutos (alrededor de 127.7 horas). El tiempo de entrenamiento por época fue considerablemente más alto que para el modelo CNN3D, oscilando entre 126 y 130 minutos; esto se debe a la mayor complejidad del modelo. El scheduler ReduceLROnPlateau redujo la tasa de aprendizaje en las épocas 31, 37, 43, 49 y 55, lo que demuestra su capacidad para optimizar la convergencia del modelo.

Los resultados finales de la evaluación en el conjunto de TEST (datos no vistos) para el mejor modelo guardado (model_RESNET_best.pth) son los siguientes:

Informe de clasificación (sklearn.metrics.classification_report):

	precision	**recall**	**f1-socre**	**support**
Goal	0.9837	0.9837	0.9837	245
Yellow card	0.9692	0.9402	0.9545	301
Substitution	0.9706	0.9635	0.9670	411
Corner	0.9766	0.9930	0.9847	713
accuracy			0.9749	1670
macro avg	0.9750	0.9701	0.9725	1670
weighted avg	0.9748	0.9749	0.9748	1670

Tabla 5: Informe clasificación del modelo RESNET.

Fuente: Elaboración propia.

Resumen evaluación final: Loss: 0.0847, Accuracy: 0.9749.

Interpretación:

- Rendimiento general: El modelo RESNET alcanzó una exactitud general del 97.49% y un F1-Score ponderado del 97.48% en el conjunto de prueba. Este es un resultado excepcional y demuestra la efectividad del transfer learning y el fine-tuning para esta tarea. La reducción significativa en la pérdida (0.0847) en comparación con el modelo CNN3D también subraya su superioridad.

- Rendimiento por clase: Muestra un rendimiento sobresaliente en todas las clases, con F1-Scores superiores a 0.95.
    - Las clases "Goal" y "Corner" mantienen sus altas puntuaciones (cercanas al 0.98).
    - "Yellow card" y "Substitution" muestran una mejora drástica en comparación con el modelo CNN3D, alcanzando F1-Scores de 0.9545 y 0.9670, respectivamente. Esto indica que el modelo pre-entrenado y el fine-tuning lograron capturar características más robustas para estas acciones.

Matriz de confusión:

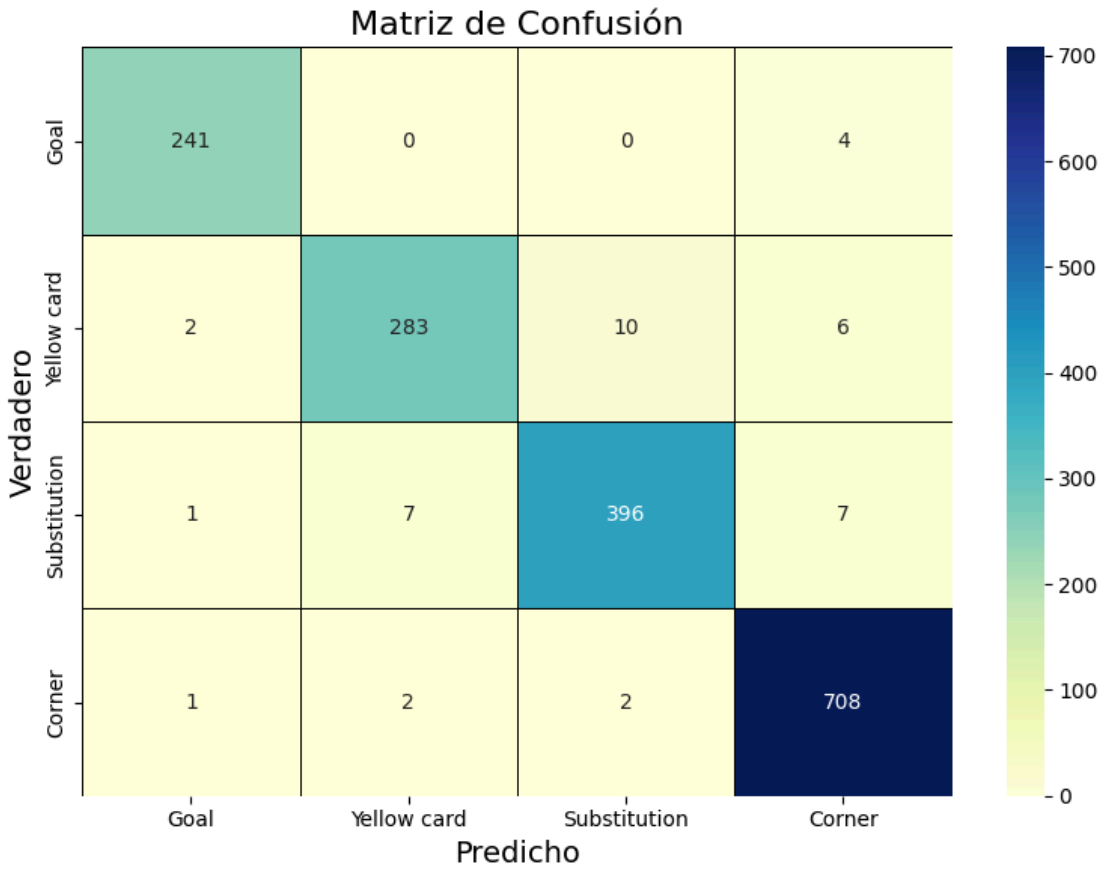

Figura 18: Matriz de confusión del modelo RESNET.

Fuente: Elaboración propia.

Falsos positivos:

- Goal:
  - El modelo predijo "Goal", pero fue una "Yellow card": 0
  - El modelo predijo "Goal", pero fue una "Substitution": 0
  - El modelo predijo "Goal", pero fue un "Corner": 4
  - **Total Falsos Positivos para "Goal": 0 + 0 + 4 = 4**
- Yellow card:
  - El modelo predijo "Yellow card", pero fue un "Goal": 2
  - El modelo predijo "Yellow card", pero fue una "Substitution": 10
  - El modelo predijo "Yellow card", pero fue un "Corner": 6
  - **Total Falsos Positivos para "Yellow card": 2 + 10 + 6 = 18**
- Substitution:
  - El modelo predijo "Substitution", pero fue un "Goal": 1

    - El modelo predijo "Substitution", pero fue una "Yellow card": 7
    - El modelo predijo "Substitution", pero fue un "Corner": 7
    - **Total Falsos Positivos para "Substitution": 1 + 7 + 7 = 15**
- Corner:
    - El modelo predijo "Corner", pero fue un "Goal": 1
    - El modelo predijo "Corner", pero fue una "Yellow card": 2
    - El modelo predijo "Corner", pero fue una "Substitution": 2
    - **Total Falsos Positivos para "Corner": 1 + 2 + 2 = 5**

Falsos negativos:

- Goal:
    - Realmente fue "Goal", pero el modelo predijo "Yellow card": 2
    - Realmente fue "Goal", pero el modelo predijo "Substitution": 1
    - Realmente fue "Goal", pero el modelo predijo "Corner": 1
    - **Total Falsos Negativos para "Goal": 2 + 1 + 1 = 4**
- Yellow card:
    - Realmente fue "Yellow card", pero el modelo predijo "Goal": 0
    - Realmente fue "Yellow card", pero el modelo predijo "Substitution": 7
    - Realmente fue "Yellow card", pero el modelo predijo "Corner": 2
    - **Total Falsos Negativos para "Yellow card": 0 + 7 + 2 = 9**
- Substitution:
    - Realmente fue "Substitution", pero el modelo predijo "Goal": 0
    - Realmente fue "Substitution", pero el modelo predijo "Yellow card": 10
    - Realmente fue "Substitution", pero el modelo predijo "Corner": 2
    - **Total Falsos Negativos para "Substitution": 0 + 10 + 2 = 12**
- Corner:
    - Realmente fue "Corner", pero el modelo predijo "Goal": 4
    - Realmente fue "Corner", pero el modelo predijo "Yellow card": 6
    - Realmente fue "Corner", pero el modelo predijo "Substitution": 7
    - **Total Falsos Negativos para "Corner": 4 + 6 + 7 = 17**

Este segundo modelo es mucho más preciso y fiable para clasificar los eventos de fútbol. Ha superado las principales debilidades del modelo anterior, logrando una alta tasa de acierto y minimizando las confusiones entre las diferentes categorías de eventos.

- Precisión excepcional en "Goal" y "Corner": El modelo es casi perfecto prediciendo "Goals" (241 aciertos y solo 4 falsos positivos a "Corner") y "Corners" (708 aciertos y muy pocos errores).
- Gran mejora en "Yellow Card" y "Substitution": A diferencia del modelo anterior, este modelo ya no confunde las "Tarjetas Amarillas" y las "Sustituciones" entre sí de manera significativa. Ahora acierta la inmensa mayoría, 283 y 396 respectivamente.
- Errores mínimos y distribuidos: Los pocos errores que comete están muy distribuidos entre las clases, sin que una confusión específica destaque negativamente como ocurría antes. Esto indica que el modelo RESNET es mucho más robusto y fiable.

4.3.3. Comparación de los modelos

	CNN3D	**RESNET**
Tiempo entrenamiento	69 horas (80 épocas)	127.7 horas (60 épocas)
Tiempo por época	51 minutos	128 minutos
batch zise	64	64

Tabla 6: Resumen comparativo de modelos.

Fuente: Elaboración propia.

Aunque el modelo RESNET requirió más tiempo total de entrenamiento (debido a que cada época es más larga por su mayor complejidad y al fine-tuning granular), la mejora en el rendimiento justifica ampliamente esta inversión. La capacidad del Transfer Learning para aprovechar el conocimiento previo de un modelo entrenado en un vasto dataset (Kinetics-400) se traduce en una convergencia hacia un rendimiento muy superior en la tarea específica de detección de eventos en fútbol. Esto valida la hipótesis de que las Redes

Neuronales Convolucionales 3D pueden aprender patrones visuales y temporales específicos, y que el post-procesamiento con umbrales de confianza por clase es efectivo para transformar predicciones crudas en marcas temporales precisas.

La alta Precisión y Exhaustividad en la mayoría de las clases para el modelo RESNET confirman su efectividad para detectar eventos clave de alto impacto narrativo y clara firma visual, como se planteó en los objetivos del TFM. Estos resultados establecen una base sólida para la fase de generación del vídeo resumen, ya que se contará con una lista de eventos altamente fiable para la creación de contenido de valor añadido.

4.4. Inferencia y post-procesamiento de eventos

Una vez entrenado y validado el modelo, la fase de inferencia permite su aplicación práctica a partidos completos. El proceso se basó en una técnica de ventana deslizante:

- **Análisis continuo del vídeo:** El modelo se aplica repetidamente a clips de vídeo cortos y solapados (clips de 3 segundos con un stride de 2 segundos) que se extraen secuencialmente del vídeo completo del partido. Cada clip genera un conjunto de probabilidades de ocurrencia para cada evento:


```
{
      "timestamp": 50.0,
      "prediction_idx": 0,
      "class_name": "Goal",
      "confidence": 0.976847231388092
},
{
      "timestamp": 52.0,
      "prediction_idx": 0,
      "class_name": "Goal",
      "confidence": 0.9703439474105835
},
{
      "timestamp": 54.0,
```

```
        "prediction_idx": 0,
        "class_name": "Goal",
        "confidence": 0.8714113235473633
    }
```


- **Refinamiento de predicciones:** Las predicciones crudas del modelo son sometidas a una rigurosa fase de post-procesamiento para transformar las probabilidades en marcas temporales precisas y fiables de los eventos:
    - **Umbralización de confianza:** Se aplicaron umbrales de confianza personalizados para cada clase de evento. Esto permitió filtrar predicciones poco fiables y mitigar el "ruido", priorizando solo aquellos momentos donde el modelo mostró alta certeza.
    - **Agrupación y filtrado temporal:** Se implementó una lógica para agrupar predicciones consecutivas de un mismo evento, consolidándolas en un único suceso. Además, se aplicó un filtro de duración mínima de evento (3.5 segundos), eliminando así detecciones muy cortas y probablemente erróneas.

El resultado de esta fase es una lista estructurada y limpia de eventos detectados, cada uno con su etiqueta de clase, tiempo de inicio y tiempo de fin en el partido:


```
{
    "evento": "Goal",
    "inicio": 50.0,
    "fin": 54.0,
    "confianza_promedio": 0.9735955893993378
}
```


4.5. Generación del vídeo resumen final

La fase final del proyecto materializa todo el trabajo previo en un producto tangible y atractivo: el vídeo resumen dinámico. Este proceso se orquestó utilizando la librería MoviePy, que interactúa con la potente herramienta FFmpeg en segundo plano:

- **Composición de escenas:** Para cada evento detectado en la fase anterior, el sistema recorta el segmento de vídeo correspondiente del partido original. Estos clips se concatenan secuencialmente para formar el hilo narrativo del resumen.

- **Efectos visuales personalizados:** Se implementó una estrategia modular para la aplicación de efectos visuales, donde cada tipo de evento tiene asociados uno o varios efectos específicos. Esto permitió añadir un valor narrativo y estético al resumen:
  - **Cámara lenta:** Utilizada en momentos clave de los goles para realzar el dramatismo de la jugada.
  - **Superposiciones gráficas:** Se incorporaron elementos gráficos deslizando, como marcadores o nombres de jugadores, para contextualizar la información de forma visualmente atractiva.
  - **Imágenes y animaciones con transparencia:** Se logró superponer imágenes estáticas y pequeños vídeos animados (GIFs) con capas alfa (transparencia) directamente sobre los clips de vídeo, añadiendo un toque profesional y dinámico al resumen.

- **Modularidad y reutilización:** La arquitectura del código para la generación de vídeo se diseñó con funciones específicas para cada efecto. Esto facilita la experimentación y la adición de nuevos efectos en el futuro.

- **Salida de alta calidad:** El vídeo final se codifica utilizando códecs estándar (libx264 para vídeo y aac para audio), asegurando una amplia compatibilidad y una experiencia de usuario fluida.

En síntesis, los resultados demuestran la viabilidad y efectividad del sistema para automatizar un proceso que, hasta ahora, requería una intervención humana intensiva y costosa. La combinación de la detección precisa de eventos y la generación de un vídeo resumen dinámico representa un avance significativo en la democratización del análisis deportivo avanzado, ofreciendo una herramienta potente y accesible para clubes, analistas y aficionados.

5. Conclusiones

El presente TFM ha abordado con éxito el desafío de automatizar la detección de eventos clave en partidos de fútbol y la generación de vídeo resúmenes dinámicos, superando las limitaciones inherentes a los métodos manuales, que son laboriosos, subjetivos y de alto coste. Este proyecto no sólo valida la hipótesis central de que un sistema basado en técnicas avanzadas de Visión por Computador e Inteligencia Artificial puede lograrlo, sino que establece una sólida base para la democratización del análisis deportivo asistido por ordenador.

Los logros más significativos del sistema se materializan a través de las siguientes conclusiones, alineadas con los objetivos específicos planteados:

- **Preparación y preprocesamiento de datos vanguardista:** Se desarrolló un módulo de preprocesamiento robusto, capaz de manejar grandes volúmenes de vídeo del dataset SoccerNet. La estrategia de segmentación en clips de duración fija y la implementación de transformaciones personalizadas, incluyendo la flexibilidad entre canales RGB y escala de grises, han garantizado una entrada de datos optimizada para el aprendizaje profundo. El riguroso EDA fue fundamental para justificar la selección de los cuatro eventos clave, basándose en su frecuencia, visibilidad y, crucialmente, su alto aislamiento temporal, lo que asegura una "firma visual" limpia y de alta calidad para el modelo.

- **Entrenamiento y validación de modelos de detección con rendimiento sobresaliente:**
    - **Modelo CNN3D Personalizado:** Se diseñó y entrenó una red neuronal convolucional 3D desde cero, demostrando su capacidad para aprender patrones espacio-temporales de forma autónoma. Este modelo estableció una base funcional, alcanzando una precisión general del 86.77% en el conjunto de test. La matriz de confusión reveló valiosos conocimientos sobre las confusiones específicas entre clases, como "Tarjeta Amarilla" y "Sustitución".

- **Modelo RESNET (Transfer Learning):** La implementación de un modelo R(2+1)D_18, pre-entrenado en Kinetics-400 y adaptado mediante una estrategia de fine-tuning granular (descongelando el último bloque de "layer4" y la capa "fc") con tasas de aprendizaje diferenciales, representó un salto cualitativo y cuantitativo fundamental. Este enfoque de Transfer Learning catapultó el rendimiento del sistema, logrando una exactitud general del 97.49% y un F1-Score ponderado del 97.48% en el conjunto de prueba. La drástica mejora en la discriminación de "Tarjeta Amarilla" y "Sustitución" (F1-Scores de 0.9545 y 0.9670 respectivamente) valida la potencia de las arquitecturas pre-entrenadas y la técnica de ajuste fino para capturar características complejas con una fiabilidad excepcional.

- **Inferencia y post-procesamiento riguroso:** La fase de inferencia se implementó mediante un eficiente proceso de ventana deslizante sobre vídeos completos. Las predicciones crudas del modelo fueron sometidas a un riguroso post-procesamiento, aplicando umbrales de confianza configurables por clase, agrupación temporal y filtrado por duración mínima. Esta etapa ha sido crucial para transformar las probabilidades en una lista estructurada y precisa de eventos discretos y significativos, lista para la fase final del proyecto.

- **Generación de vídeo resúmenes dinámicos y atractivos:** Como culminación del proyecto, se ha desarrollado un motor de composición de vídeo modular y sofisticado utilizando MoviePy. Este sistema permite recortar y concatenar los segmentos de vídeo detectados, aplicando una variedad de efectos visuales personalizados para realzar el impacto narrativo de cada evento. La capacidad de integrar cámara lenta, superposiciones de texto e imágenes con transparencia, y animaciones, ha transformado un proceso técnico en una experiencia visual atractiva y profesional, cumpliendo plenamente con el objetivo de generar contenido de valor añadido.

En síntesis, se ha demostrado la viabilidad y la eficacia de la Inteligencia Artificial y la Visión por Computador para revolucionar el análisis deportivo, proporcionando una herramienta autónoma, objetiva y de alto rendimiento. El sistema desarrollado no solo cumple con sus objetivos técnicos, sino que sienta un precedente para la creación de soluciones accesibles que democratizan el acceso a capacidades analíticas avanzadas, beneficiando a una audiencia más amplia en el ámbito deportivo y de las retransmisiones.

Conclusiones personales

La realización de este proyecto ha sido una experiencia profundamente enriquecedora y transformadora en mi formación como especialista en Inteligencia Artificial y Visión por Computador. He consolidado una comprensión práctica y profunda de todo el ciclo de vida de un proyecto de Deep Learning, desde la preparación minuciosa de datos hasta la materialización de un producto final tangible.

- **Dominio técnico y resolución de desafíos:** Enfrentar y superar desafíos como la optimización de la carga de datos para GPUs, la implementación de complejas lógicas de refactorización (como la separación del engine de entrenamiento o la gestión de DataLoaders), ha fortalecido mis habilidades de resolución de problemas y mi capacidad para construir sistemas robustos y eficientes. La iteración constante en el diseño de la arquitectura del modelo CNN3D y, especialmente, la implementación exitosa del Transfer Learning con arquitecturas de vanguardia como R(2+1)D, me han proporcionado una visión invaluable sobre cómo las decisiones arquitectónicas y de entrenamiento impactan directamente en el rendimiento final.

- **Importancia de las buenas prácticas de ingeniería de software:** Este TFM ha sido una demostración práctica de la importancia de la refactorización y la modularidad. Encapsular la lógica en funciones reutilizables, separar responsabilidades entre módulos (engine_training, dataset_soccernet, transforms, etc.), y mantener los scripts principales

como "orquestadores", ha sido una elección fundamental que seguiré aplicando en futuros proyectos. La implementación de sistemas de checkpointing y logging robustos, así como el manejo consciente de la memoria de la GPU, son prácticas que considero ya indispensables.

- **Visión integral de los procesos:** Más allá de los modelos, el proyecto ha enfatizado la relevancia de cada fase: desde un EDA riguroso para justificar decisiones de diseño, hasta un post-procesamiento inteligente para refinar las predicciones, y una fase final de generación de contenido que transforma datos en una experiencia de usuario. Esta perspectiva completa me capacita para abordar problemas complejos no solo desde la óptica del modelo, sino desde un enfoque holístico de sistema.

En definitiva, esta aventura no sólo marca un hito en mi formación académica, sino que me dota de las competencias y la confianza necesarias para contribuir significativamente al campo de la Inteligencia Artificial, especialmente en el desarrollo de soluciones prácticas y de alto impacto.

6. Limitaciones y futuras líneas de investigación

Se ha logrado con creces los objetivos, sin embargo, como todo proyecto de investigación, este sienta las bases para futuras exploraciones, y es crucial reconocer las fronteras actuales que definen sus limitaciones inherentes y las promisorias avenidas para su evolución.

6.1. Limitaciones del trabajo actual

La eficacia demostrada por el sistema se contextualiza dentro de un marco específico, cuyas particularidades, aunque necesarias para la viabilidad del proyecto, delinean su alcance actual:

- **Dependencia de la vista isométrica y generalización limitada:** El entrenamiento del modelo se realizó exclusivamente con el dataset SoccerNet, caracterizado por una vista isométrica consistente. Si bien esto simplifica la tarea de reconocimiento visual al proporcionar patrones estables, limita la capacidad de generalización del sistema a escenarios reales con ángulos de cámara variables, diferentes calidades de transmisión, o condiciones ambientales diversas (iluminación, clima). Un despliegue universal requeriría una mayor robustez ante la heterogeneidad visual de los partidos.

- **Alcance semántico de eventos restringido:** Aunque el proyecto se centró en la detección precisa de cuatro eventos clave de alto impacto narrativo, el universo completo de acciones en un partido de fútbol es mucho más amplio. Esta focalización, justificada por el EDA en términos de visibilidad y aislamiento temporal, implica que el sistema, en su estado actual, no proporciona una taxonomía exhaustiva de todos los sucesos relevantes del juego.

- **Monosensorialidad:** Al concentrarse exclusivamente en la modalidad visual, diversas investigaciones como el estudio de Nergård, han demostrado que la fusión de modalidades visuales y auditivas puede potenciar significativamente la precisión en la identificación de eventos.

Por lo tanto, una prometedora línea de investigación futura es la integración de la modalidad de audio al sistema. Esto permitiría no solo mejorar la robustez general de las detecciones, sino también identificar eventos que poseen características acústicas distintivas (por ejemplo, el sonido de un silbato, el impacto del balón en el poste o las celebraciones del público), que complementarían la información visual y enriquecerían la comprensión del contexto del juego.

- **Intensidad computacional para el entrenamiento y dependencia de GPU:** Si bien se lograron optimizaciones en el procesamiento de vídeo, el entrenamiento del modelo, particularmente el R(2+1)D con grandes volúmenes de clips, es intensivo en recursos. Las épocas requerían considerable tiempo y el aprovechamiento total de la arquitectura de transfer learning demandó un entorno con GPU, lo que podría ser una barrera para usuarios con recursos de hardware limitados.

- **Ajuste manual de hiperparámetros de post-procesamiento:** La fase de post-procesamiento, aún requiere un ajuste manual de parámetros como los umbrales de confianza por clase y la duración mínima de los eventos. Este ajuste se realiza empíricamente para optimizar el equilibrio entre precisión y exhaustividad, pero no está completamente automatizado.

6.2. Futuras líneas de investigación

Las limitaciones inherentes al estado actual de este trabajo abren un vasto horizonte de futuras investigaciones que podrían llevar el análisis deportivo automatizado a un nivel sin precedentes. Estas líneas no solo buscan pulir la funcionalidad existente, sino explorar nuevas fronteras tecnológicas y metodológicas:

- **Expansión y refinamiento de la taxonomía de eventos:** Más allá de los cuatro eventos actuales, se podría ampliar el catálogo de detecciones

para incluir otras acciones. Esto requeriría investigar cómo mejorar la detección de eventos con firmas visuales menos "limpias" o con menor aislamiento temporal, potencialmente a través de un análisis contextual más profundo o la incorporación de información táctica inferida.

- **Mejora de la generalizabilidad y robustez en entornos heterogéneos:** Para que el sistema sea verdaderamente aplicable en la industria, sería fundamental investigar técnicas de adaptación de dominio o el entrenamiento con datasets más diversos que incluyan múltiples ángulos de cámara, calidades de vídeo y condiciones de iluminación. Esto podría implicar el uso de modelos de homografía avanzados para mapear las acciones a una vista cenital del campo, permitiendo un análisis posicional más preciso.

- **Optimización para inferencia en tiempo real y despliegue en borde (edge computing):** Aunque la inferencia ya es eficiente, el siguiente paso sería adaptar los modelos y el pipeline para un procesamiento en tiempo real, permitiendo su integración en transmisiones en vivo o en dispositivos a pie de campo. Esto implicaría la exploración de modelos más ligeros, técnicas de cuantificación o poda de redes para reducir la latencia.

- **Desarrollo de una interfaz de usuario intuitiva:** La creación de una aplicación o plataforma web con una interfaz gráfica amigable permitiría a usuarios no técnicos cargar vídeos, ajustar parámetros de cada efecto del vídeo resumen y generarlos con mayor facilidad, democratizando aún más el acceso a esta tecnología.

- **Personalización y narrativa dinámica del vídeo resumen:** El motor de vídeo es potente, pero se podría investigar la creación de resúmenes personalizados o sistemas que generen narrativas más sofisticadas, quizás ajustando la duración de los clips destacados o la intensidad de los efectos basándose en el impacto real del evento dentro del partido. Esto podría incluir la adición de música o audio dinámico.

- **Aprendizaje semi-supervisado:** Para reducir la dependencia de datasets masivos y meticulosamente anotados, se podrían explorar técnicas que permitan al modelo aprender de vídeos con anotaciones más escasas o de etiquetas a nivel de vídeo (ejemplo: Este partido tuvo 3 goles) en lugar de marcas temporales precisas.

En definitiva, este TFM no sólo cierra un ciclo de investigación y desarrollo, sino que ilumina un sinfín de oportunidades para que la Inteligencia Artificial siga revolucionando el análisis deportivo, transformando la forma en que interactuamos con el deporte más popular del planeta. Las puertas de la innovación permanecen abiertas para seguir construyendo soluciones que superen las expectativas actuales y establezcan nuevos paradigmas en la comprensión y disfrute del fútbol.

7. Referencias bibliográficas

[1] Aguilera-Eguía, R. (2014). ¿Revisión sistemática, revisión narrativa o metaanálisis? Revista de la Sociedad Española del Dolor, 21(6), 359-360. https://dx.doi.org/10.4321/S1134-80462014000600010

[2] Ato, M., López, J. J., & Benavente, A. (2013). Un sistema de clasificación de los diseños de investigación en psicología. Anales de Psicología, 29(3), 1038-1059.
https://doi.org/10.6018/analesps.29.3.178511

[3] Bhalerao, C. (2023). A deep dive into non-maximum suppression (NMS). Built In.
https://builtin.com/machine-learning/non-maximum-suppression

[4] Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., & Zisserman, A. (2018). A short note on the Kinetics-600 human action dataset.
https://arxiv.org/abs/1808.01340

[5] Creswell, J. W. (2009). Research design: Qualitative, quantitative, and mixed methods approaches (3rd ed.).
https://www.ucg.ac.me/skladiste/blog_609332/objava_105202/fajlovi/Creswell.pdf

[6] Deliège, A., Cioppa, A., Giancola, S., Seik, C., Van Droogenbroeck, M., & Ghanem, B. (2021). SoccerNet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. En Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.
https://arxiv.org/abs/2011.13367

[7] Franks, I. M., & Goodman, D. (1986). A systematic approach to analysing sports performance. Journal of Sports Sciences, 4(1), 49-59.
https://doi.org/10.1080/02640418608732098

[8] Franks, I. M., & Miller, G. (1986). Eyewitness testimony in sport. Journal of Sport Behavior.

[9] Giancola, S., Amine, M., Dghaily, T., & Ghanem, B. (2018). SoccerNet: A scalable dataset for action spotting in soccer videos. En Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 1735-1744).
https://doi.org/10.1109/CVPRW.2018.00223

[10] Gong, Y., Sin, L. T., Chuan, C. H., Zhang, H., & Sakauchi, M. (2003). Automatic parsing of TV soccer programs. En Proceedings of the Second IEEE International Conference on Multimedia and Expo (Vol. 2, pp. II-373).
https://www.ri.cmu.edu/pub_files/pub2/gong_yihong_1995_1/gong_yihong_1995_1.pdf

[11] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., & Zisserman, A. (2017). The Kinetics human action video dataset.
https://arxiv.org/abs/1705.06950

[12] LeveReg. (s.f.). Computer vision basics: How confidence, accuracy, and thresholds impact performance.
https://www.leverege.com/blogpost/computer-vision-basics-how-confidence-accuracy-and-thresholds-impact-performance

[13] Rongved, O. A. N., Stige, M., Hicks, S. A., Thambawita, V. L., Midoglu, C., Zouganeli, E., Johansen, D., Riegler, M. A., & Halvorsen, P. (2021). Automated event detection and classification in soccer: The potential of using multiple modalities. Machine Learning and Knowledge Extraction, 3(4), 1030-1054.
https://doi.org/10.3390/make3040051

[14] Tiwari, S., & Vyas, A. (2014). Video Summarization: Survey on Event Detection and Summarization Techniques. International Journal of Computer Applications, 106(11), 1-5. https://thesai.org/Downloads/Volume6No11/Paper_33-Video_Summarization_Survey_on_Event_Detection_and_Summarization.pdf

[15] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. En Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6450-6459).