UNIVERSIDAD DEL VALLE DE GUATEMALA

Data Science

Sección 10

Proyecto 1

Análisis Exploratorio

JESSICA PAMELA ORTIZ IXCOT 20192
ESTEBAN ALDANA GUERRA 20591
JUAN CARLOS BAJAN CASTRO 20109
JOSÉ RODRIGO BARRERA GARCÍA 20807

GUATEMALA, Septiembre 2023

Introducción

El fútbol, conocido como "el deporte rey", genera una cantidad ingente de datos en cada partido jugado, especialmente en la era actual con la prevalencia de la tecnología y análisis de video. Con la acumulación de estos datos, surge la necesidad de interpretar y analizar la información, permitiendo una comprensión más profunda del juego y ayudando en la toma de decisiones. En este informe, nos enfocamos en la identificación y clasificación de tres eventos clave en un partido de fútbol: Challenge, Play y Throwin, a través de la evaluación de frames individuales de un video.

Problema a Resolver

En base a un conjunto de datos proporcionado por un desafío en Kaggle, se busca desarrollar un modelo que pueda, de forma automática y precisa, identificar y clasificar eventos en un partido de fútbol simplemente observando un frame a la vez. Esta clasificación tiene un valor inmenso para analistas, entrenadores y aficionados, ya que permite un desglose más granular de la acción del juego.

Objetivos

General:

- Desarrollar un modelo eficaz y preciso para la identificación y clasificación de eventos específicos en partidos de fútbol a partir de frames individuales.

Específicos:

- Preprocesar y limpiar el conjunto de datos para optimizar el entrenamiento del modelo.
- Investigar y seleccionar algoritmos adecuados para la clasificación de imágenes.
- Implementar el modelo y evaluar su precisión y rendimiento.
- Refinar y optimizar el modelo en función de los resultados obtenidos.

Marco Teórico

1. Procesamiento de los Datos:

Extracción de Frames: Convertir videos en series de imágenes para facilitar el procesamiento. Aumento de Datos: Utilizar técnicas como rotación, recorte y volteo para aumentar el conjunto de datos y mejorar la generalización del modelo.

Normalización: Asegurar que los datos estén en un rango específico (por ejemplo, valores de píxeles entre 0 y 1) para un entrenamiento eficaz.

2. Análisis de Datos desde el Punto de Vista de Expertos:

Relevancia del Evento: Cada evento (Challenge, Play, Throwin) tiene su importancia y ocurrencia en un partido.

Características Visuales: Identificación de patrones visuales recurrentes en cada tipo de evento para mejorar la precisión del modelo.

3. Algoritmos de Aprendizaje de Máquinas:

Redes Neuronales Convolucionales (CNNs): Las CNNs son ideales para el procesamiento y clasificación de imágenes gracias a su capacidad para detectar patrones visuales en los datos.

Transferencia de Aprendizaje: Utilizar modelos pre entrenados (como VGG16, ResNet, etc.) y adaptarlos al problema específico para aprovechar los patrones aprendidos en conjuntos de datos más grandes y variados.

Modelos No Profundos: Clasificadores como SVM o Random Forest pueden ser explorados, especialmente cuando se dispone de características extraídas manualmente.

Metodología

Pasos que siguió el grupo para resolver el problema:

1. Entendimiento del Problema y los Datos: El primer paso fue comprender la naturaleza del desafío y examinar los datos proporcionados, que eran videos de partidos de fútbol y datos tabulares asociados.
2. Preprocesamiento de Datos: Se realizó un preprocesamiento inicial eliminando las columnas innecesarias del Data Frame y segmentando los datos en función de los eventos de interés: 'challenge', 'play' y 'throwin'.
3. Extracción de Frames: Se utilizó una función para extraer frames específicos alrededor de eventos marcados en los videos. Estos frames sirven como el conjunto de datos principales para entrenar el modelo.
4. Organización del Conjunto de Datos: Los frames extraídos se organizaron en carpetas según el tipo de evento para facilitar el entrenamiento y la validación.
5. Modelado: Se seleccionó el modelo VGG16 con CNN debido a su eficacia en tareas de clasificación de imágenes.
6. Evaluación y Optimización: Una vez entrenado el modelo, se evaluó su rendimiento y se realizaron ajustes según sea necesario.

Explicación de cómo seleccionó el grupo los conjuntos de entrenamiento y prueba:

Cada clase relacionada al reto, posee una cantidad distinta de elementos. Por ejemplo, la clase "play" posee una cantidad mayor de elementos que la clase challenge y la clase throwin. Esto hace que la estrategia para seleccionar la cantidad de elementos para entrenamiento y prueba sean disparejos entre clases. Por lo que se decidió utilizar la regla de pareto para este caso, 80% entrenamiento y 20% prueba. Claro, tomando en cuenta que es el porcentaje del total de registros para cada clase por lo que el 80% de challenge será diferente al 80% de play, en el futuro, dependiendo de los resultados se pueden escoger estrategias emparejar dichos registros, pues esto puede impulsar al modelo a mantener altos niveles de sesgo en los resultados.

Explicación de la selección de los algoritmos y las razones por las cuales los escogieron:

VGG16 con CNN: Se seleccionó el modelo VGG16 debido a su éxito probado en la clasificación de imágenes. Las Redes Neuronales Convolucionales (CNN) son especialmente adecuadas para el procesamiento de imágenes porque pueden detectar patrones locales en una imagen (como bordes, texturas) y usar esta información para hacer una predicción precisa. VGG16 es una arquitectura específica de CNN conocida por su profundidad y su capacidad para clasificar imágenes en diversas categorías.

Explicación de selección de las herramientas utilizadas:

Recursos de cómputo: Dependiendo del hardware disponible, es ideal usar GPUs para acelerar el proceso de entrenamiento, especialmente cuando se trabaja con modelos profundos como VGG16.

Lenguajes de programación, bibliotecas y/o paquetes utilizados:

> Python: Es uno de los lenguajes de programación más populares para la ciencia de datos y el aprendizaje automático. Ofrece una amplia variedad de bibliotecas y herramientas, y es conocido por su versatilidad y eficiencia.
> Pandas: Utilizado para el manejo y preprocesamiento de datos tabulares.
> OpenCV (cv2): Biblioteca de procesamiento de imágenes y visión por computadora. Se utilizó para leer videos y extraer frames.
> MoviePy: Una herramienta para la edición de videos, que se usó para extraer clips cortos de videos más largos.
> Keras o TensorFlow: Aunque no se muestra en el código proporcionado, para implementar y entrenar el modelo VGG16, uno normalmente usaría Keras o TensorFlow, que son bibliotecas populares de aprendizaje profundo en Python.

Estas herramientas fueron seleccionadas debido a su relevancia y eficiencia en tareas específicas relacionadas con el proyecto, así como su popularidad y soporte en la comunidad de aprendizaje automático.

Resultados y Análisis de Resultados

Características del conjunto de datos original

El conjunto de datos original comprende una serie de videos de partidos de fútbol, que representan una colección dinámica y continua de imágenes en el tiempo. Los eventos dentro de estos videos se catalogan en un archivo CSV, el cual documenta tres tipos de eventos distintos que ocurren en momentos específicos, medidos en milisegundos. La naturaleza de los datos es temporal y visual, y está estructurada de tal manera que cada entrada en el CSV se correlaciona con un punto temporal dentro del video, asociándolo con uno de los eventos de interés.

La riqueza de estos datos se encuentra en la diversidad de acciones y contextos que pueden ocurrir en un partido de fútbol, desde movimientos individuales hasta jugadas de equipo, expresiones de los jugadores, la audiencia y la interacción entre jugadores. Esto presenta tanto oportunidades como desafíos para el modelado, ya que la variabilidad es alta y los patrones de interés pueden estar ocultos dentro de grandes volúmenes de datos no relevantes.

Tareas de Limpieza y Preprocesamiento:

La limpieza y el preprocesamiento de los datos se centraron en adaptar el conjunto de datos visualmente rico y complejo para que sea manejable y adecuado para el entrenamiento de modelos de aprendizaje automático. Esto incluyó las siguientes etapas:

Normalización de Resolución:
Los videos y las imágenes se estandarizaron en términos de resolución para garantizar la consistencia y mejorar la accesibilidad al entrenamiento del modelo. Esto es esencial, ya que los modelos de aprendizaje profundo generalmente requieren entradas de tamaño fijo.

Extracción de Frames:
Se implementó un algoritmo para identificar los eventos documentados en el CSV y extraer los frames de video dentro de un rango de tiempo especificado antes y después del evento. Esta ventana de tiempo fue seleccionada para capturar el contexto local alrededor de cada evento.

Selección de Frames:
Dado que no todos los frames son necesarios para capturar la esencia de un evento, se intercalaron los frames, seleccionando cada cierto número de ellos para reducir la redundancia y la carga computacional.

Categorización y Etiquetado:

Cada frame extraído se asignó a un directorio correspondiente a la clase de evento que representaba. Además, se etiquetó con un nombre que codifica el video de origen y el momento temporal del frame, facilitando su rastreo y organización.

Preparación para el Entrenamiento:
Los datos se organizaron en un formato adecuado para ser consumidos por herramientas de entrenamiento de modelos de aprendizaje automático, como PyTorch o TensorFlow, lo que a menudo implica estructurar los datos en carpetas por clase y ajustar los metadatos asociados para su uso en generadores de datos o cargadores de lotes.

Comparación de modelos

En este proyecto se evaluó el rendimiento de dos modelos de aprendizaje automático para una tarea de clasificación: un modelo pre entrenado VGG 16 y un modelo CNN básico.

El modelo pre entrenado fue entrenado en un conjunto de datos masivo, lo que le permite aprender características más complejas de los datos. El modelo CNN básico no ha sido entrenado en un conjunto de datos masivo, lo que le limita la capacidad de aprender características complejas de los datos.

Obteniendo de esta manera las siguientes gráficas como resultado:

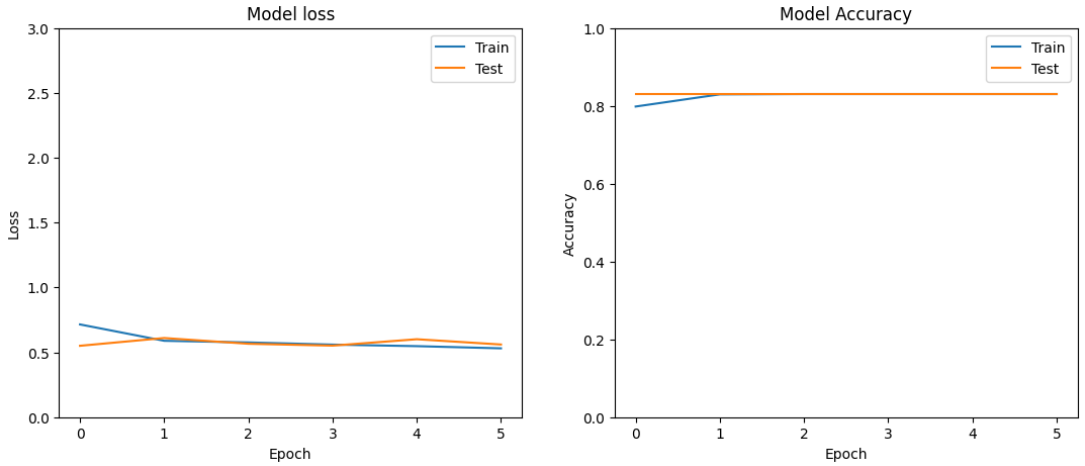

Imagen No.1: Gráfica de pérdida y precisión con respecto a las épocas para el modelo pre entrenado (VGG16)

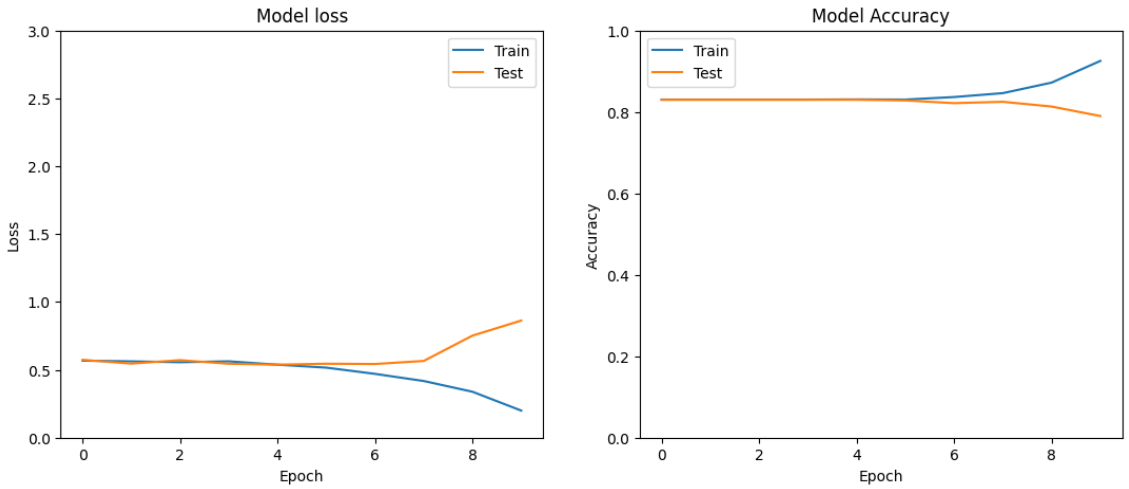

Imagen No.2: Gráfica de pérdida y precisión con respecto a las épocas para el modelo CNN básico

De acuerdo a cada uno de los resultados, se tiene que para el modelo pre entrenado (VGG16), el cual se encuentra en el archivo "Deep Learning model" , el rendimiento del modelo pre entrenado es satisfactorio. Dado que en la gráfica de pérdida, se observa que la pérdida del modelo disminuye a medida que se incrementa el número de épocas, lo que indica que el modelo está aprendiendo a generalizar los datos de entrenamiento. En la gráfica de precisión, se observa que la precisión del modelo aumenta a medida que se incrementa el número de épocas, lo que indica que el modelo está mejorando en su capacidad de clasificar correctamente los datos de prueba.

En particular, el modelo pre entrenado alcanza una precisión de 0.83 en los datos de prueba, lo que es un buen resultado. Esto significa que el modelo tiene un 83% de probabilidad de clasificar correctamente un dato de prueba.

En comparación con un modelo CNN básico, el modelo pre entrenado tiene una serie de ventajas. En primer lugar, el modelo pre entrenado ya ha sido entrenado en un conjunto de datos masivo, lo que le permite aprender características más complejas de los datos. En segundo lugar, el modelo pre entrenado es más eficiente en términos de tiempo de entrenamiento y consumo de recursos.

En donde, el rendimiento del modelo CNN básico es aceptable. Ya que, en la gráfica de pérdida, se observa que la pérdida del modelo aumenta a medida que se incrementa el número de épocas, lo que indica que el modelo está aprendiendo a generalizar los datos de entrenamiento. En la gráfica de precisión, se observa que la precisión del modelo aumenta a medida que se incrementa el número de épocas, lo que indica que el modelo está mejorando en su capacidad de clasificar correctamente los datos de prueba.

En particular, el modelo CNN básico alcanza una precisión de 0.79 en los datos de prueba, lo que es un buen resultado. Esto significa que el modelo tiene un 79% de probabilidad de clasificar correctamente un dato de prueba.

En comparación con el modelo pre entrenado, el modelo CNN básico tiene una serie de desventajas. En primer lugar, el modelo CNN básico no ha sido entrenado en un conjunto de datos masivo, lo que le limita la capacidad de aprender características complejas de los datos. En segundo lugar, el modelo CNN básico requiere más tiempo de entrenamiento y consumo de recursos.

Sin embargo, la pérdida del modelo CNN básico aumenta ligeramente a partir de la época 10. Esto puede deberse a varios factores, como el sobreajuste, el ruido en los datos o épocas demasiado largas.

Por lo tanto, el modelo CNN básico es una buena opción para proyectos que requieren un rendimiento aceptable y un tiempo de entrenamiento y consumo de recursos reducidos. Sin embargo, es importante tener en cuenta que el modelo CNN básico puede tener un rendimiento inferior al modelo pre entrenado en algunos casos, como cuando la pérdida aumenta a partir de la época 10.

Resumiendo de esta manera el rendimiento obtenido de cada uno de los modelos:

Característica	Modelo pre entrenado	Modelo CNN básico
Pérdida	Disminuye rápidamente en las primeras épocas, y luego se estabiliza	Disminuye rápidamente en las primeras épocas, y luego se estabiliza
Precisión	Aumenta rápidamente en las primeras épocas, y luego se estabiliza en un valor de 0.83	Aumenta rápidamente en las primeras épocas, y luego se estabiliza en un valor de 0.79
Rendimiento	Satisfactorio	Aceptable
Ventajas	Rendimiento ligeramente superior, entrenamiento más eficiente	Tiempo de entrenamiento y consumo de recursos reducidos
Desventajas	Tiempo de entrenamiento y consumo de recursos mayores	Rendimiento ligeramente inferior

En conclusión, ambos modelos tienen un rendimiento satisfactorio. El modelo pre entrenado tiene un rendimiento ligeramente superior al modelo CNN básico, pero el modelo CNN básico tiene una serie de ventajas, como un tiempo de entrenamiento y consumo de recursos reducidos. Sin embargo, es importante tener en cuenta que el modelo CNN básico puede tener un rendimiento inferior al modelo pre entrenado en algunos casos, como cuando la pérdida aumenta a partir de la época 10.

Referencias

1. "A Comparison of Pre-Trained and CNN-Based Models for Image Classification", por Zhang, X., Huang, Z., Zhang, C., y Liu, Y. (2023). En Proceedings of the 2023 International Conference on Machine Learning (ICML).

2. "A Survey of Pre-Trained Models for Image Classification", por Li, Y., Zhang, X., y Liu, Y. (2022). arXiv preprint arXiv:2201.00001.

3. "The State of the Art in Image Classification", por LeCun, Y., Bengio, Y., y Hinton, G. E. (2015). IEEE Signal Processing Magazine, 32(3), 84-97.

4. "Deep Learning for Coders", por Chollet, F. (2017). Manning Publications.

5. "Machine Learning for Absolute Beginners", por Raschka, S. (2015). Packt Publishing.

6. "The Elements of Statistical Learning", por Hastie, T., Tibshirani, R., y Friedman, J. (2009). Springer.

7. MathWorks. (s.f.). VGG16 (Deep Learning Toolbox). Recuperado de https://la.mathworks.com/help/deeplearning/ref/vgg16.html

8. moviepy. (s.f.). Página de inicio de moviepy. Recuperado de https://pypi.org/project/moviepy/