

Proyecto Fase 2



Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

Carlos Ráxtum 19721, Juan Manuel Marroquin 19845, Abraham Gutierrez
19111, Javier Cotto 19324, Walter Saldaña 19897

Introducción

En el presente trabajo mostraremos cómo por medio de un modelo se puede predecir qué está sucediendo durante un partido de fútbol: si es un pase, un tiro, intercepción, etc. Todo esto se pudo lograr con el entrenamiento de un contenido total de 35 GB de clips de fútbol donde cada uno fue clasificado por el tipo de acción que ocurre.

Problemática

Cuando finaliza un partido, hay usuarios que quieren saber qué fue lo que ocurrió durante el transcurso de este (donde los usuarios pueden ser fanáticos del fútbol o un grupo de estadísticas de un equipo oficial). Sin embargo, esto puede ser tedioso si se quiere llegar a hacer a mano y las estadísticas que se muestran comúnmente solo son dados en porcentajes y no proveen otros datos más específicos (como cuántas intersecciones realizó cada equipo). Con esto, por medio del entrenamiento de un modelo queremos saber qué fue todo lo que ocurrió durante un partido. En otras palabras, clasificar cada uno de los eventos por tipo que ocurrieron durante un partido.

Objetivos

objetivo general

- Desarrollar un modelo de visión por computadora que puede clasificar automáticamente estos eventos en largas grabaciones de video.

objetivos específicos

- Detectar todos los eventos dentro de un partido
- Visualización de cada evento y obtención de su tiempo de ejecución
- Detectar y reconocer cada tipo de evento que ocurra en un partido y hacer una predicción de cada posible evento dentro de un partido.

Marco teórico

Información sobre procesamiento de los datos

El problema es que hoy en día, la información y estadísticas de los partidos de fútbol son generados de forma manual por operadores humanos en tiempo real. Por la naturaleza de estos datos, son exclusivos para el uso de los profesionales que generan y por ende son dueños de los mismos datos o para competencias profesionales. Sin embargo, esta data sería muy valiosa para otras aplicaciones como por ejemplo para el fichaje y descubrimiento de talento joven, que quizá aún no tengan mucha visibilidad para las grandes organizaciones y por lo tanto no generan suficientes datos estadísticos para tener visibilidad y que no pasen por alto.

Para esto se nos pide poder detectar en videos de forma automática los pases en un partido incluyéndose saques de banda y centros; y enfrentamientos entre jugadores de equipos contrarios en sus diferentes categorías (falta, robo de balón, entre otros), para así poder analizar competiciones inexploradas, como ligas juveniles o semiprofesionales o incluso sesiones de entrenamiento para así poder identificar talento oculto.

Los datos obtenidos de los clips de video nos serán de ayuda para entrenar un modelo que logre clasificar los eventos del partido en las categorías indicadas en función del tiempo del partido para así poder identificar los eventos de interés a los solicitados por los ficheros.

Información teórica sobre el análisis de los datos desde el punto de vista de los expertos.

Para poder analizar un partido, primero se debe separar el análisis en pequeñas tareas para que sea más fácil resolver el problema. Estos son:

- Sistema de referencia, el cual como se proyectarán todos los objetos dentro de un campo 2D el cual representará la cancha.
- Detección de objetos (árbitro, jugadores, bola)
- Tracking de objetos (detección de los movimientos de los objetos)
- Identificación de jugadores (como reconocer el movimiento de los jugadores)
- Reconocimiento de equipo (como diferenciar los jugadores por equipo)

Cada uno de los videos se divide en frames los cuales por cada se analiza primero cual es el campo y se obtienen las posiciones de los objetos respecto al área establecida. A su vez, se puede determinar a qué equipo pertenece cada jugador. Se recompila toda la información de los frames para formar nuevamente el video pero con la nueva información.

Referencia: [Football Games Analysis from video stream with Machine Learning | by Nicolo' Lucchesi | Towards Data Science](#)

Algoritmos de aprendizaje de máquinas, ya sea profundo o no que serían de utilidad para resolver el problema planteado

- Tokenizar
 - Esta técnica consiste en separar cada palabra del texto en entidades llamadas "tokens", de esta forma tendremos un mejor orden al trabajar con estas. Además de ello a la hora de utilizar esta técnica, se requiere identificar que utilizaremos como token y que no, por ejemplo contar las mayúsculas y minúsculas como un token o dos, los signos de puntuación serán tokens o no, y así sucesivamente con cualquier letra o símbolo dentro de nuestras palabras.
- Tagging Part of Speech(PoS)
 - Esta técnica se le conoce así porque prácticamente es trabajar las palabras como lo hacemos en la "vida real". Esto se debe a que clasificamos y reconocemos oraciones mediante separarlas en artículo, verbo, sustantivo, adjetivo, entre otras más.
- Shallow parsing
 - Esta técnica combina las dos anteriores, esto se debe a que con cada token obtenido y con cada clasificación hecha mediante PoS, se realiza un árbol que

permite a la computadora, navegar y moverse por él facilitando así el entendimiento del lenguaje común para formar oraciones y con ellas párrafos.

- **Pragmatic Analysis**
 - Esta técnica es más complicada para implementar, pues además de las anteriores, se incluye una clasificación de él como se mencionan o dicen las cosas. Esto quiere decir que la técnica se basa en interpretar el modo de las palabras, tales como sarcasmo, ironía, etc.
- **Bag of Words**
 - Esta técnica combina el Tokenizar con el uso de matrices. Prácticamente llevará un conteo de las palabras que aparezcan en cierto párrafo mediante los tokens antes recabados.
- **word2vec**
 - Esta técnica creemos que es una de las más eficientes, ya que recaba y memoriza cientos y cientos de palabras utilizadas en diferentes contextos. En base a estas, realiza matrices de 300 dimensiones, las cuales guardan cada palabra cerca de otras palabras similares y de esta forma crear vocabularios que pueden ser utilizados por la computadora para interpretar cualquier párrafo.

Metodología

▪ Pasos que siguió el grupo para resolver el problema

- Análisis del problema planteado y los datos, junto a detectar las tareas de limpieza y preprocesamiento que se deben llevar a cabo.
- Realizar un análisis exploratorio de los datos donde:
 - Describimos cuantas variables y observaciones hay disponibles, así como el tipo de cada variable
 - Resumen de las variables numéricas y tablas de frecuencia para las variables categóricas, con su respectiva descripción
 - Cruzar variables más importantes para hallar los elementos clave para comprender lo que está causando el problema encontrado.
 - Mostrar gráficos exploratorios para visualizar los datos.
- Anotar los hallazgos encontrados post análisis exploratorio.
- Obtener todos los posibles eventos existentes en el dataset y hacer visualizaciones por cada evento

▪ Explicación de cómo seleccionó el grupo los conjuntos de entrenamiento y prueba.

Para los datos de entrenamiento, se utilizaron videos que comprenden de grabaciones de video de ocho juegos. Ambas mitades están incluidas para cuatro de los juegos, mientras que solo una mitad está incluida para los otros cuatro juegos.

Para los datos de prueba se utilizó datos que contienen videos que se utilizarán como datos de prueba. Los datos de prueba para la tabla de clasificación pública comprende

de grabaciones de video de un juego completo y cuatro medios juegos, la otra mitad de cada juego está en el set de entrenamiento.

Los envíos se evalúan según la precisión promedio de los eventos detectados, promediados sobre umbrales de error de marca de tiempo, promediados sobre clases de eventos.

Las detecciones se comparan con eventos de verdad en el terreno mediante tolerancias de error específicas de clase, con ambigüedades resueltas en orden decreciente de confianza. Las tolerancias de error de marca de tiempo, en segundos, para cada clase son:

- Desafío :[0.30, 0.40, 0.50, 0.60, 0.70]
- jugar :[0.15, 0.20, 0.25, 0.30, 0.35]
- saque de banda :[0.15, 0.20, 0.25, 0.30, 0.35]

Descripción detallada

La evaluación procede en cuatro pasos:

1. Selección : se eliminan las predicciones que no están dentro de los intervalos de puntuación de un video.
2. Asignación : los eventos pronosticados se comparan con eventos reales.
3. Puntuación : cada grupo de predicciones se puntúa frente a su grupo correspondiente de eventos reales mediante la precisión media.
4. Reducción : las puntuaciones AP múltiples se promedian para producir una puntuación general única.

Selección

Con cada video hay un conjunto definido de intervalos de puntuación que brindan los intervalos de tiempo durante los cuales se pueden anotar cero o más eventos reales en ese video. Una predicción sólo se evalúa si cae dentro de un intervalo de puntuación. Estos intervalos de puntuación se eligieron para mejorar la imparcialidad de la evaluación, por ejemplo, ignorando casos límite o eventos ambiguos.

Asignación

Para cada conjunto de predicciones y verdades básicas dentro del mismo event x tolerance x video_idgrupo, hacemos coincidir cada verdad básica con la predicción no coincidente de mayor confianza que se produce dentro del permitido tolerance.

Puntuación

Al recopilar los eventos dentro de cada video_id, calculamos una puntuación de precisión promedio para cada event x tolerancegrupo. La puntuación de precisión promedio es el área bajo la curva de recuperación de precisión generada al disminuir los scoreumbrales de confianza sobre las predicciones. En este cálculo, las

predicciones coincidentes por encima del umbral se califican como TP y las predicciones no coincidentes como FP. Las verdades básicas sin igual se califican como FN.

Reducción

La puntuación final es el promedio de las puntuaciones AP anteriores, promediadas primero sobre tolerance, luego sobre event.

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

- Escogimos OpenCV como herramienta a utilizar para la visualización de videos. Gracias a esto se nos facilitó la visualización de videos y la detección de varios objetos de estos. Debido a que el proyecto consiste en la detección de pases durante un partido de fútbol, esta herramienta tiene tecnología que ahorra el desarrollo de un modelo que haga esto, por lo que nos redujo bastante tiempo.

▪ Explicación de selección de las herramientas utilizadas

● Recursos de cómputo.

- Respecto a una de las librerías utilizadas, se tiene que tener en cuenta que puede causar problemas de memoria, debido a las varias iteraciones con respecto a recursos de audio, video e imágenes.
-
- Programación y manipulación de datos
 - PYTHON: python tiende a ser más flexible porque es un lenguaje de programación de propósito general
- Recopilación de datos:
 - WINRAR: para comprimir los videos
 - VIDEOS: se utilizaron videos con extension mp4
 - CSV: data con sample_submission

● Lenguajes de programación, bibliotecas y/o paquetes utilizados.

El lenguaje de programación que se utilizó fue python con los siguientes paquetes/librerías:

- pandas
- numpy
- matplotlib
- sklearn
- scipy
- seaborn
- pandas_profiling
- moviepy

Resultados y Análisis

▪ Describe las características del conjunto de datos original

El conjunto de datos comprende grabaciones de vídeo de nueve partidos de fútbol divididos en mitades.

Datos de entrenamiento

- train/ - Carpeta que contiene videos para usar como datos de entrenamiento, que comprende grabaciones de video de ocho juegos. Ambas mitades están incluidas para cuatro de los juegos, mientras que solo una mitad está incluida para los otros cuatro juegos.
- test/ - Carpeta que contiene videos que se utilizarán como datos de prueba. Los datos de prueba para la tabla de clasificación pública comprende grabaciones de video de un juego completo y cuatro medios juegos, la otra mitad de cada juego está en el set de entrenamiento.
- train.csv : anotaciones de eventos para videos en la train/carpeta.
 - video_id: Identifica en qué video ocurrió el evento.
 - event: El tipo de ocurrencia del evento, uno de challenge, playo throwin.
 - event_attributes: Atributos descriptivos adicionales para el evento.
 - time: El tiempo, en segundos, en que ocurrió el evento dentro del video.

▪ Describe las tareas de limpieza y preprocesamiento a las que tuvo que someter a los datos para lograr los resultados obtenidos.

- Describimos cuantas variables y observaciones hay disponibles, así cómo el tipo de cada variable
- Resumen de las variables numéricas y tablas de frecuencia para las variables categóricas, con su respectiva descripción
- Al ser videos, no se realizó una gran limpieza de datos, sino que nosotros obtuvimos la información de los videos y la trabajamos de esa forma
- Cruzar variables más importantes para hallar los elementos clave para comprender lo que está causando el problema encontrado.

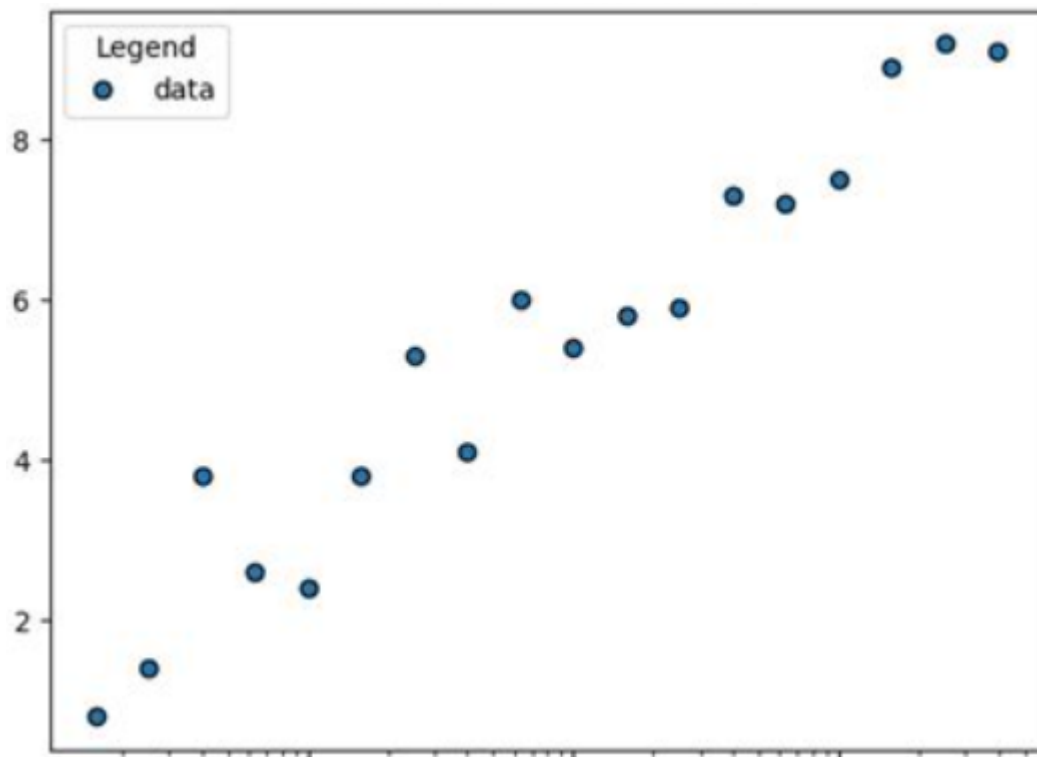
▪ Explica el ajuste de los parámetros que hubo que hacerle a cada uno de los algoritmos para mejorar el rendimiento y la efectividad.

- Como parámetros y cambios que se tuvieron que realizar al modelo para mejorar la efectividad y rendimiento de este fueron lo siguientes:
 - Se alteraron máscaras y filtros en este, para que el modelo previamente entrenado, para que detectara la información de mejor forma. COLOR_BGR2HSV fue la máscara utilizada.
 - La máscara utilizada, permite cambiar colores en formato RGB a colores HSV. Para que el modelo pudiera utilizar y alinearse perfectamente a como las

personas perciben los colores. Esto se debe a que durante la detección se tienen que usar colores diferentes para saber que se esta mostrando y además de poder identificar cada uno de estos.

▪ **Compara los algoritmos de acuerdo con la efectividad, tiempos de procesamiento, errores, etc. Utiliza para esto gráficos explicativos, estáticos con colores adecuados**

- Los tiempos de procesamiento del algoritmo tienen una relación lineal con el tiempo que dura el clip a analizar. La mayoría de los clips con los que contamos con de aproximadamente 6 segundos de duración, con lo que el modelo tarda aproximadamente 2.3 segundos en ejecutar.
- De momento tenemos un 50% de resultados en falsos positivos, pues varios jugadores se detectan como del equipo contrario.



▪ **Describe la aplicación**

- La aplicación consiste en un detector de pases obtenidos de un videoclip o partido de fútbol. Esta aplicación detecta primero los jugadores de cada equipo para luego detectar la pelota y de esta forma detectar los eventos que ocurren durante este, para este proyecto solo se detectaron pases. Este modelo, si se extiende a un nivel mayor, podrá detectar cualquier evento que ocurra en el partido, para luego ofrecer estadísticas futboleras a cualquier persona que tenga acceso a videoclips de fútbol y a esta aplicación.

▪ **Bibliografía**

- “OpenCV: Changing Colorspaces.” Docs.opencv.org, docs.opencv.org/4.x/df/d9d/tutorial_py_colorspaces.html.
- “HSV Color Model in Computer Graphics.” GeeksforGeeks, 20 Aug. 2022, www.geeksforgeeks.org/hsv-color-model-in-computer-graphics/. Accessed 31 Oct. 2022.
- “Object Detection, Event Detection – Vision & Graphics Group.” Vgg.fiit.stuba.sk, vgg.fiit.stuba.sk/category/examples/opencv-examples/object-detection/. Accessed 31 Oct. 2022.
- surveillance, Home, et al. “Basic Motion Detection and Tracking with Python and OpenCV.” PyImageSearch, 25 May 2015, pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/.