

```
<!--Universidad del Valle-->
```

Detección de Eventos en partidos de Fútbol{

```
<Por="Carlos Raxtúm  
Jose Gutierrez  
Walter Saldaña  
Juan Marroquin  
Javier Cotto"/>
```

```
}
```



Contenidos

01

Introducción

02

Problemática

03

Objetivos

04

Metodología

05

Librerías

06

Resultados

07

Conclusión

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.

}

Equipo {



JUAN
MARROQUIN

PROGRAMADOR



JAVIER
COTTO

CUM MASTER



WALTER
SALDAÑA

PROJECT
MANAGER



CARLOS
RÁXTUM

GYM RAT



JOSE
GUTIERREZ

BANQUERO

}

Objetivos {

Objetivo General

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

Objetivos Especificos

- 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.



}

Metodología {

Hallazgos

Separación
Train - Test

ANALISIS

Análisis
Exploratorio

Cruce de
Variables

Selección
Algoritmo

Eventos
Posibles

}



Planificación {

FASE 1

- Análisis del problema planteado y los datos, junto a detectar las tareas de limpieza y preprocesamiento que se deben llevar a cabo.

FASE 2

- Realizar un análisis exploratorio de los datos. 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
- 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.

ACTUALIDAD

FASE 3

Mostras avances preliminares. donde se realiza la aplicacion de la carga de archivo, aplicacion donde se puede ver el resultado de cada algoritmo y a la vez se muestra la informacion con graficas iterativas.

FASE 4

Se culmina con el proyecto, en esta fase se muestran los resultados finales del proyecto y se comentan los hayazgos encontrados y las conclusiones a las que se llevo.

}

Librerías {

01

Pandas

02

Numpy

03

Matplotlib

04

Sklearn

05

Scipy

06

Seaborn

07

Pandas
Profiling

08

MoviePy

}

Algoritmo Escogido {

Open Source Computer Vision

El primer algoritmo que escogimos se apoyo de la herramienta OpenCV, que nos brinda tecnología de IA para poder trackear objetos en videos.

- Primeramente para perseguir a los jugadores de cada equipo.
- Como segundo paso, detectaríamos la pelota.
- Como último punto, se recabara información de los partidos para luego obtener los eventos donde ocurre un pase.



}

Modelo {

El modelo utilizado para recabación de datos es una aplicación con OpenCV. Este permitiría analizar los videos de cada partido, para luego entrenar el modelo y que este detecte la cantidad de pases de cada equipo y sus respectivos tiempos.

Ademas de que esta tecnología podria servir para detectar otros eventos, tales como tiros libres, tiros a puerta, entre otros eventos.

```
▶ <dom-if style="display: none;">...</dom-if>
  <!-- TODO(crbug.com/1168361): Instead of hidden$="[[!logoEnabled_]]"
  it would
    be nicer to use a dom-if. However, that breaks

  StartupBrowserCreatorPickerNoParamsTest.ShowPickerWhenAlreadyLaunched
  on
    the msan builder. See crbug.com/1169070. -->
  ▼ <ntp-logo id="logo" single-colored> flex
    ▶ #shadow-root (open)
    </ntp-logo>
  ▶ <ntp-realbox id="realbox" shown is-dark>...</ntp-realbox>
  ▶ <cr-most-visited id="mostVisited" is-dark_ visible_ custom-links-
    enabled_>...</cr-most-visited>
  ▼ <dom-if style="display: none;">
    ▶ <template>...</template>
    </dom-if>
  ▼ <ntp-middle-slot-promo>
    ▶ #shadow-root (open)
    </ntp-middle-slot-promo>
  ▶ <dom-if style="display: none;">...</dom-if>
  ▶ <dom-if style="display: none;">...</dom-if>
  ▶ <a id="backgroundImageAttribution" hidden>...</a>
    <!-- cr-button has a transparent background. This leads to incorrect
    results when a custom background is set. Therefore, wrap
    customize
      button in container to enforce solid background color. -->
  ▶ <div id="customizeButtonContainer">...</div>
```

}

Resultados del Modelo {

- El modelo detecta varios jugadores en el clip.
- Deficiencias en diferenciar jugadores de diferentes equipos
- Deficiencias en encontrar la pelota

2.3s

Por clips de 6s

}

Conclusiones {

- El modelo tiene la capacidad de encontrar jugadores pero hace falta afinarlo más para extraer la información necesaria para cortar el clip en las secciones deseadas.
- OpenCV facilitó el modelo pero aún se requirió de agregar varios filtros al clip.



}

<!--Universidad del Valle de Guatemala-->

Gracias {

- "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/.

}