PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
Escuela de Posgrado

Identificación automática de las fases del gesto de recepción en el vóley mediante análisis de videos usando redes neuronales convolucionales

Tesis para optar por el Grado Académico de

MAGÍSTER EN INFORMÁTICA CON MENCIÓN EN CIENCIAS DE LA COMPUTACIÓN

Autor

Jose Gustavo Garcia Sulca

Asesor

Dr. César Armando Beltrán Castañón

Lima, Noviembre del 2019

RESUMEN

El presente trabajo presenta un modelo algorítmico que permite la identificación automática a partir de videos de las fases temporales que ocurren durante la ejecución de la técnica de recepción en el vóley.

En la etapa inicial se muestra la definición de dichas fases temporales a analizar, así como algunos trabajos relacionados al ámbito de reconocimiento de actividades en el área de ciencias de la computación. De igual manera, se presenta el marco teórico que contiene los conceptos necesarios para el desarrollo de este trabajo.

Luego se procedió a definir dos módulos en los que se divide el modelo algorítmico: módulo de detección de jugador y módulo de clasificación de fases. En cada uno de estos módulos se detalla las arquitecturas de los modelos a utilizar así como el pre-procesamiento de los datos y el respectivo método de entrenamiento.

Finalmente, se muestra lo obtenido tras la implementación de los módulos detallados anteriormente. Para ello se realizó adicionalmente la recolección de una base de datos de videos con su respectivo etiquetado, la cual fue desarrollada para la presente tesis como parte del proyecto *"Caracterización biomecánica del gesto técnico de recepción en el voleibol puesta al servicio del entrenamiento deportivo mediante el desarrollo de un aplicativo móvil integrado a un sistema de captura de movimiento low-cost"*, el cual viene siendo desarrollado por el Grupo de Investigación en Robótica Aplicada y Biomecánica. Con ello, se muestran los resultados obtenidos al realizar el entrenamiento de los módulos con esta base de datos. Estos muestran que el modelo implementado consigue identificar correctamente la fase temporal a nivel de *frames* con una precisión de 92.19%. Además a ello, en los casos donde ocurre un error en la identificación, la fase identificada por el modelo es una contigua a la real, mostrando que el modelo pudo captar la esencia temporal de las fases.

Agradecimientos

En primer lugar, agradezco a mi madre Flor Sulca Zaconetta por todo el apoyo y la guía brindada durante esta etapa de mi vida.

A mis compañeros del Laboratorio de Investigación en Biomecánica y Robótica Aplicada por el apoyo y la amistad brindada durante la realización del presente trabajo.

Agradezco además a mi asesor, el Dr. César Beltrán, por la orientación brindada a lo largo del desarrollo de la presente tesis.

Agradecimiento especial al Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica del Perú por apoyar la realización de este trabajo mediante el proyecto “Caracterización biomecánica del movimiento asociado a la técnica de recepción en el vóley categoría juvenil damas” con contrato N° 206-2015 FONDECYT y el proyecto “Caracterización biomecánica del gesto técnico de recepción en el voleibol puesta al servicio del entrenamiento deportivo mediante el desarrollo de un aplicativo móvil integrado a un sistema de captura de movimiento low-cost” con contrato N° 058-2018 FONDECYT.

INDICE DE TABLAS

Pág.

INDICE DE FIGURAS

Pág.

ÍNDICE DE CONTENIDO

CAPÍTULO 1

Generalidades

1.1 Problemática

El estudio sobre el rendimiento de los deportistas es un área de investigación importante que se desarrolla principalmente por dos motivos: incrementar el rendimiento del deportista y evitar lesiones en el entrenamiento. Este trabajo lo desarrollan principalmente los entrenadores deportivos, quienes a partir de la observación directa toman decisiones en base a información cualitativa. Con el tiempo, se han propuesto sistemas de cuantificación que permitan eliminar la subjetividad que se presenta en esta valoración cualitativa. Sin embargo, estos sistemas llegan a ser muy costosos, consumen mucho tiempo o son difíciles de transportar al campo [Suárez, 2009].

En el Laboratorio de Investigación en Biomecánica y Robótica Aplicada de la PUCP se desarrollan trabajos de investigación relacionados a caracterizar la biomecánica del movimiento de algunos deportes. Uno de los proyectos trabajados buscó desarrollar una metodología que permita realizar la valoración cuantitativa del gesto de la recepción que se realiza en el vóley [GIRAB, 2019]. Esta valoración se basa en dos sistemas de captura: uno de movimiento y otro de fuerza; los cuales, como se mencionó anteriormente, son costosos y difíciles de transportar. Adicionalmente, como parte de estos experimentos, se realiza la grabación de videos de los ensayos que involucran la realización de gestos deportivos con condiciones controladas. Sin embargo, estos videos son archivados para un uso posterior de visualización por parte de los entrenadores.

Como parte de este estudio, en conjunto con los entrenadores, se definió una división temporal del gesto en cuatro fases. Esto fue considerado muy importante por parte de los entrenadores para poder realizar un mejor análisis del gesto y de esta manera poder brindar una mejor realimentación a los

jugadores. Sin embargo, la identificación de estas fases resulta tediosa e implica analizar un video muy detenidamente, resultando inviable para efectos de entrenamiento de los deportistas.

Como alternativa, existen sistemas de visión computacional los cuales poseen la capacidad de realizar esta tarea de manera automática. Inclusive, son capaces de realizarlo con mejor precisión que un especialista, tal como ya se ha reportado en la tarea de clasificación sobre el conjunto de datos *ImageNet* [Russakovsky, 2015].

Es por ello que en el presente trabajo se busca responder si un algoritmo basado en visión computacional para el análisis de videos pueda estimar las fases temporales de un movimiento deportivo.

1.2 Objetivos

OBJETIVO GENERAL

Implementar un modelo algorítmico de detección de fases durante el gesto de recepción en el vóley mediante el análisis del movimiento en video con redes neuronales profundas.

OBJETIVOS ESPECÍFICOS

OE1. Determinar las fases de un gesto deportivo y sus periodos en una secuencia de video, de manera que permita constituir la base de datos.
OE2. Detectar a un jugador de vóley dentro de una secuencia de videos mediante la implementación de un algoritmo de detección basado en redes neuronales profundas.
OE3. Identificar las fases de la realización del gesto de recepción mediante la implementación de un algoritmo de clasificación basado en redes neuronales profundas.

1.3 Resultados Esperados

1. Para el OE1 se plantea obtener una base de datos de videos del gesto de recepción en el vóley anotados con estampas temporales de inicio y fin de las fases de dicho gesto.
2. Para el OE2 se plantea obtener un algoritmo funcional basado en redes neuronales profundas que permita detectar en una imagen la región sobre la cual se encuentra un jugador de vóley.
3. Para el OE3 se plantea obtener un algoritmo funcional basado en redes neuronales profundas que permita identificar correctamente las fases temporales del gesto realizado sobre un video.

1.4 Métodos y procedimientos

En la Figura 1.1 se muestra los distintos pasos seguidos en el presente trabajo. Esto comprendió cinco componentes importantes: la generación de conjunto de datos, el pre-procesamiento, la determinación de región de interés, el entrenamiento de módulo de clasificación y la predicción de las fases del gesto.

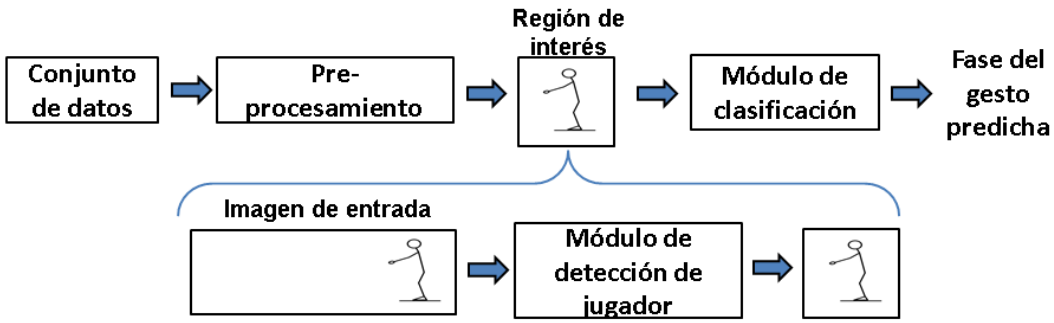


Figura 1.1: Metodología empleada en la identificación de las fases del gesto de recepción del vóley que comprende cinco componentes importantes.

En primera instancia se realizó la adquisición de una base de datos que permita desarrollar el sistema propuesto. Esta adquisición se desarrolló como parte del proyecto *"Caracterización biomecánica del gesto técnico de recepción en el voleibol puesta al servicio del entrenamiento deportivo mediante el desarrollo de un aplicativo móvil integrado a un sistema de captura de*

movimiento low-cost"[1]. Esto comprendió el desarrollo de ensayos con sus respectivos protocolos para la captura de los videos que conformaron el conjunto de datos final así como el etiquetado de las estampas temporales por cada video. Posteriormente, se realizó el pre-procesamiento de los videos, lo cual consistió en la separación de *frames* de los videos así como el procesamiento propio de las técnicas a utilizar. A continuación, se realizó el entrenamiento de un modelo algorítmico que permita realizar la detección de la región donde se encuentra el jugador sobre una imagen. Para ello se realizó el etiquetado de estas regiones sobre las imágenes y se entrenó un modelo de detección de objetos. Luego se utilizó las imágenes recortadas como entrada para el entrenamiento de un módulo con el propósito de realizar la clasificación de las fases deseadas. Finalmente, se usó este módulo para poder obtener las fases temporales en gestos.

1.5 Alcance y limitaciones

El alcance de la presente investigación abarcó la clasificación de la fase del gesto de la recepción en videos obtenidos en las instalaciones del polideportivo PUCP. Estos videos son analizados desde dos vistas diagonales de aproximadamente 45°. Al tenerse un enfoque sobre el entrenamiento de jugadores, estos videos estuvieron restringidos a contener solamente al jugador y se realizaron en condiciones controladas.

[1] Proyecto PUCP ID672 – FONDECYT 058-2018 PUCP

CAPÍTULO 2

Estado del Arte y Marco Conceptual

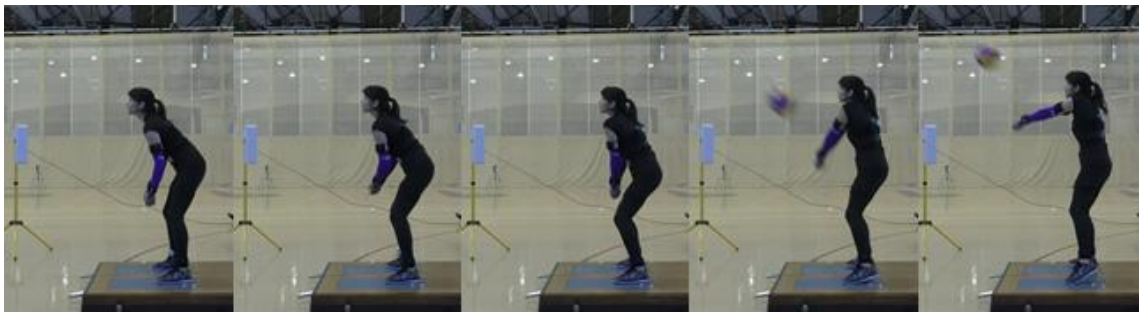
En el presente capítulo se presentará la revisión bibliográfica realizada para el presente trabajo así como la definición de conceptos importantes para la realización de la presente tesis.

2.1 Estado del Arte

2.1.1. Secuencia de fases en el gesto de recepción en el vóley

En el desarrollo del proyecto “Caracterización biomecánica del movimiento asociada a la técnica de recepción en el vóley categoría juvenil damas” [GIRAB, 2019] se ha definido, con el apoyo de entrenadores de vóley, la existencia de cuatro fases que se ejecutan en secuencia durante el gesto de recepción, una de las cuales se divide en dos subfases por motivos de análisis.

- Fase 1: Esta fase se caracteriza por mostrar la postura de espera de la jugadora. Culmina cuando se empieza el desplazamiento para ir al contacto con el balón (Figura 2.1(a)).
- Fase 2: Esta fase se caracteriza por mostrar el movimiento que realiza la jugadora para hacer contacto con el balón. Se divide en dos subfases:
    - Fase 2.1: Inicia con el despegue de los pies y culmina cuando la jugadora fija los pies en la posición final deseada (Figura 2.1(b)).
    - Fase 2.2: Esta fase culmina cuando la jugadora forma una plataforma rígida con sus brazos (Figura 2.1(c)).
- Fase 3: Esta fase se caracteriza por mostrar el movimiento de los brazos para hacer contacto con el balón, momento en el cual culmina la fase (Figura 2.1(d)).
- Fase 4: Esta fase se caracteriza por mostrar el movimiento post-contacto de la jugadora hasta dejar la postura rígida mostrada anteriormente, usualmente cuando relaja los brazos (Figura 2.1(e)).



(a) (b) (c) (d) (e)

Figura 2.1: Fases de la técnica de recepción en el vóley. (a) Fase 1, (b) Fase 2.1, (c) Fase 2.2, (d) Fase 3 y (e) Fase 4.

2.1.2. Trabajos relacionados

Esta tarea puede ser considerada en el ámbito de reconocimiento de actividades. Existen revisiones que muestran los temas que se han abarcado recientemente en este ámbito y en cuanto a deportes solo se presenta el reconocimiento de que deporte se realiza o el tipo de gesto realizado [Singh, 2019], [Ramasamy, 2018]. Al no encontrar trabajos enfocados en el reconocimiento de actividades al nivel de análisis planteado para el presente trabajo, se revisarán trabajos que realizan clasificación a nivel de gestos deportivos tanto en voleibol como en otros deportes.

2.1.2.1. Activity recognition in beach voleyball using Deep Convolutional Neural Network [Kautz, 2017]

Este trabajo se enfoca en la clasificación de actividades a partir de la información obtenida por sensores inerciales ubicados sobre los deportistas. Con ello utilizaron una red convolucional profunda 1D para el análisis de las series temporales. Las clases a predecir fueron 10: servicio (3 variaciones), recepción (2 variaciones), ataque, mate, bloqueo, clavada y una clase nula.

2.1.2.2. A Hierarchical Deep Temporal Model for Group Activity Recognition [Ibrahim, 2016]

En este trabajo se propone un modelo jerárquico constituido por modelos LSTM

que permiten realizar la clasificación de actividades sobre videos. Uno de los conjuntos de datos utilizados consistía en videos de partidos de voleibol en los cuales clasificaban el lado de la cancha donde se realizaba la acción así como el tipo de acción realizado: pase, mate y recepción.

2.1.2.3. Proyecto JUMP [JUMP, 2014]

Este proyecto fue realizado por un equipo multidisciplinario en el cual se desarrolló un sistema integrado de plataformas de fuerza y un sistema de tracking para deportistas y personas realizando acciones comunes.

Entre estos resultados desarrollaron algoritmos que permiten discernir la actividad que está desarrollando un jugador de básquetbol durante un partido a partir de los *frames* de un video, esto se realiza por medio de redes convolucionales 3D mezcladas con un modelo que permite estimar articulaciones, a la cual se le agrega una red tipo *encoder-decoder* para detectar el campo y poder discernir si el jugador se encuentra en juego o en la banca de suplentes [Francia, 2018].

2.2 Marco Conceptual

A continuación se mostrarán algunos conceptos teóricos relacionados al tema desarrollado en el presente trabajo.

2.2.1 Técnicas de detección de objetos en imágenes

La detección de objetos consiste en determinar la región dentro de la imagen que engloba a un objeto y clasificarlo con la categoría correspondiente. De manera general estas técnicas se pueden clasificar en dos tipos [Zhao, 2019].

A. Estructura basada en proposición de región
   Esta estructura consiste en dos pasos: realizar un escaneado de toda la imagen y luego enfocarse en las regiones de interés encontradas. Para el primer paso se hace uso del método de ventana deslizante. Entre las

técnicas más usadas se tienen R-CNN, SPP-Net, Faster R-CNN entre otras.

B. Estructura basada en regresión y clasificación

   Esta estructura, a diferencia de la ya mencionada, realiza la detección en un solo paso. Para ello realiza un mapeo de pixeles hacia coordenadas de los recuadros de las regiones (*bounding box*) y las probabilidades de las clases. Al no tener la necesidad de recorrer toda la imagen previo a encontrar las zonas de interés, este método llega a ser más rápido comparado a los ya mencionados en 'A'. Entre las técnicas más usadas se tienen YOLO y SSD.

En cuanto a la métrica usada en estos casos, actualmente la más usada es la métrica GIoU (intersección sobre unión generalizada) [Rezatofighi, 2019]. Esta métrica es una generalización del IoU (intersección sobre unión), la cual calcula la razón entre la intersección y la unión del área de predicción y el área real. Esta métrica tiene problemas en los casos donde no hay intersección, puesto que la métrica IoU devuelve el valor 0 y no brinda un acercamiento sobre si otra predicción llega a ser mejor o no. Es por ello que en la métrica GIoU se plantea usar la menor capsula convexa que contenga al área de la predicción y al área real. Con ello se sustrae a la métrica IoU una razón sobre el área de la cápsula convexa que disminuye conforme se posee una mejor predicción, generando un mejor indicador. En la Figura 2.2 se puede observar las áreas utilizadas para determinar la métrica GIoU.

(a) (b) (c)

Figura 2.2: Representación de las áreas utilizadas para el cálculo de la métrica GIoU[2]. (a) área real, (b) y (c) malas predicciones con sus respectivas cápsulas convexas.

2.2.2 Fusión de información temporal

Al trabajar con videos se posee una característica que no se tiene en las imágenes: el tiempo. Es por ello que esta característica puede ser de interés al momento de definir los modelos a trabajar. En el artículo "*Large-scale Video Classification with Convolutional Neural Networks"* proponen cuatro métodos para trabajar con videos [Karpathy, 2014].

a. Frames individuales

En este acercamiento se propone trabajar con cada *frame* del video de manera individual, lo cual permitiría analizar la capacidad de la red para clasificar el video en base a información estática.

b. Fusión tardía

En este acercamiento se propone trabajar con los *frames* inicial y final de un clip del video, para los cuales se utilizaría una CNN individual. Finalmente estas dos CNN se unen en una capa totalmente conectada.

[2] Generalized Intersection over Union. https://giou.stanford.edu/

c. Fusión temprana

En este acercamiento se propone trabajar con una secuencia de *frames* consecutivos, los cuales son fusionados para entrar a una CNN, esto permitiría conocer la dirección y velocidad del movimiento.

d. Fusión lenta

Este acercamiento es una mezcla de la fusión temprana y la fusión tardía. Acá se propone mezclar secuencias de *frames* consecutivos en una ventana de tiempo móvil.

En la Figura2.3 se puede observar gráficamente cómo se realizaría cada uno de estos métodos.

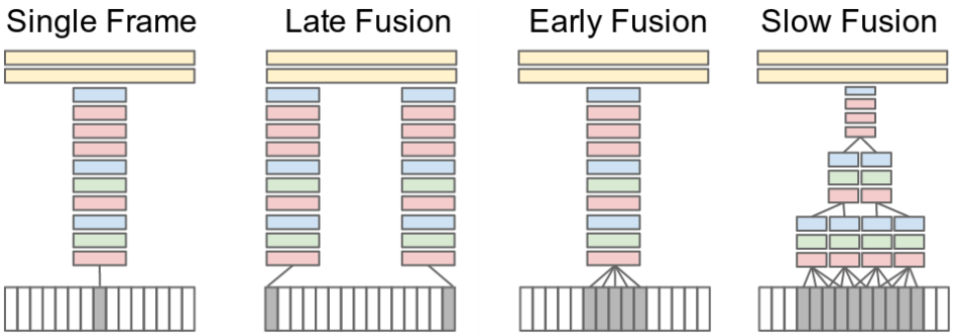


Figura 2.3: Representación gráfica de los métodos de fusión de información temporal. Extraído de [Karpathy, 2014]

CAPÍTULO 3

Módulo de detección de jugador en imagen

Como primera parte, el algoritmo a desarrollar consta de un módulo que permita detectar la región de la imagen donde se encuentra el jugador. El principal motivo de este módulo es poder trabajar posteriormente con dicha región delimitada eliminando todo el posible ruido que se presente como fondo en la imagen.

3.1. Modelo a trabajar

Se escogió trabajar con el algoritmo YOLO (*You Only Look Once*) [Redmon, 2016] debido a que este módulo no requiere mayor complejidad: solo se requiere detectar a un jugador. Este algoritmo en su tercera versión consiste en dividir la imagen en regiones y para cada región realizar una predicción de tres recuadros que encierren a los objetos a buscar, como se puede observar en la Figura 3.1. Por cada una de estas predicciones se obtienen cuatro coordenadas que definen las esquinas del recuadro, un valor de confianza de la predicción y un código *one-hot* que predice la clase del objeto detectado. En este caso en particular, solo se busca detectar un jugador, por lo que cada una de estas predicciones constará de (4+1+1) valores.

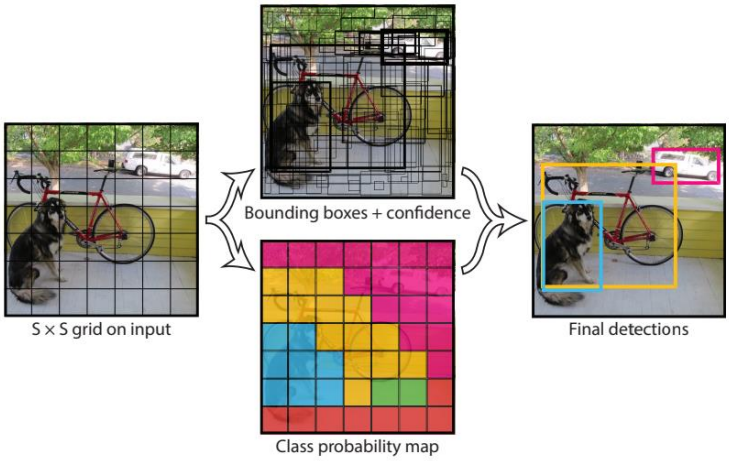


Figura 3.1: Ejemplificación del proceso de detección del algoritmo YOLO.
Extraído de [Redmon, 2016]

3.2. Etiquetado

El etiquetado que se requiere para el entrenamiento del modelo a trabajar consiste en brindar por cada imagen las dos coordenadas de la esquina superior izquierda, el ancho y el alto del recuadro, y la clase del objeto que se está encerrando. El modelo requiere estas coordenadas normalizadas en relación al tamaño de la imagen. Para ello se utilizará la herramienta *Yolo-Annotation-Tool-New*[3] la cual brinda una interfaz que muestra las imágenes a etiquetar y guarda un archivo *.txt* por cada imagen con las coordenadas y tamaño normalizado del recuadro dibujado manualmente. Esta interfaz puede ser observada en la Figura 3.2.

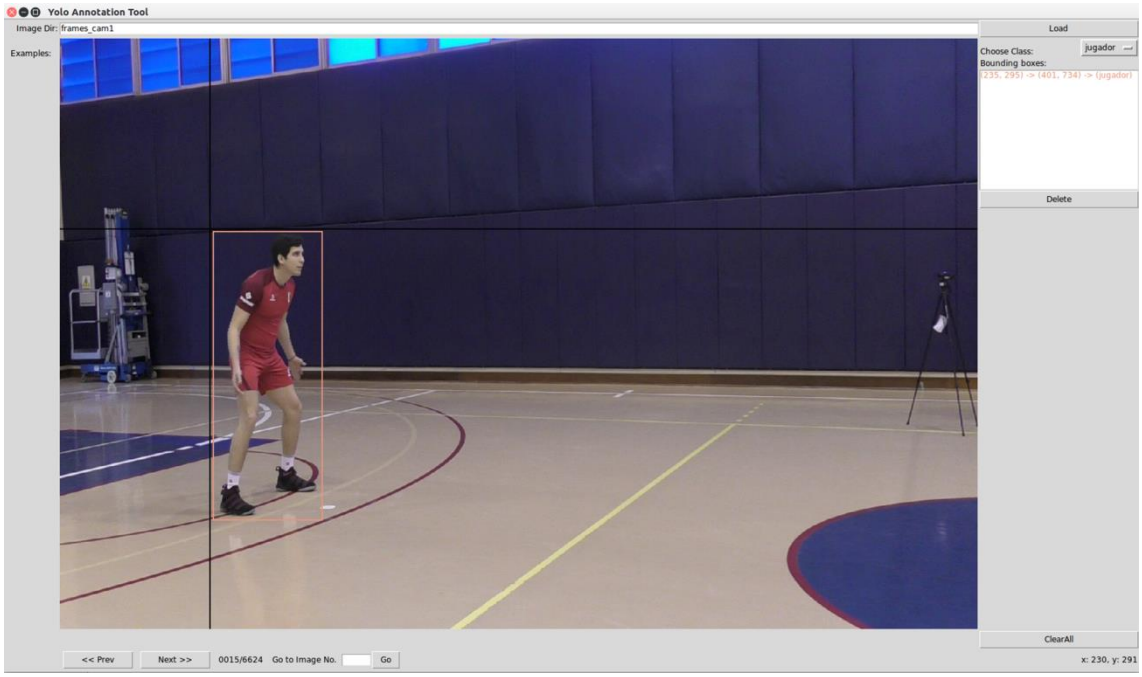


Figura 3.2: Interfaz de herramienta utilizada para el etiquetado de las imágenes.

3.3. Entrenamiento

El entrenamiento de este módulo está basado en la versión desarrollada en Pytorch por Ultralytics LLC [Glenn, 2019]. Esta versión incluye el entrenamiento personalizado del modelo en base a un conjunto de datos etiquetado con los

[3] Yolo-Annotation-Tool-New. **https://github.com/ManivannanMurugavel/Yolo-Annotation-Tool-New-**

objetos que se deseen detectar. Adicionalmente incluye en el proceso de entrenamiento la técnica de incremento de imágenes, para lo cual utiliza las transformaciones mostradas en la Tabla 3.1.

Tabla 3.1: Transformaciones realizadas en el incremento de imágenes

Transformación	**Descripción**
Traslación	$\pm 10\%$ (vertical y horizontal)
Rotación	$\pm 5^{\circ}$
Cizallamiento (shear)	$\pm 2^{\circ}$
Escalamiento	$\pm 10\%$
Reflejo Horizontal	50% de probabilidad
Saturación HSV	$\pm 50\%$
Intensidad HSV	$\pm 50\%$

CAPÍTULO 4

Módulo de clasificación de fases de la recepción en video

Este módulo tiene como objetivo identificar la fase de la recepción partiendo del resultado del módulo de detección de jugador.

4.1. Pre-procesamiento

Con el objetivo de introducir la información temporal como entrada, se utilizará las técnicas de análisis por *frames* individuales y análisis por *Early Fusion* explicadas en el inciso 2.2.2. Para ello, se plantea los siguientes métodos de pre-procesamiento de las imágenes, los cuales serán comparados posteriormente durante la experimentación.

4.1.1. Unión de tres *frames* de video consecutivos

Con este método lo que se busca es poder juntar *frames* consecutivos sin perder información de forma de cada uno de los *frames*. Para ello, cada imagen es primero transformada a una imagen en escala de grises y posteriormente se crea una imagen cuyas capas R, G y B son cada una de estas imágenes. En la Figura 4.1 se observa el resultado del proceso realizado para un instante del video.

4.1.2. Promedio de *frames* consecutivos

Esta propuesta de unión implica crear una imagen como el promedio de cierta cantidad de *frames* consecutivos. A diferencia del método mostrado en 4.1.1, este promedio junta la información temporal pero pierde la información de cada instante de tiempo así como también pierde la secuencia de los *frames*; sin embargo, permite utilizar una mayor cantidad de *frames* en el análisis. En la Figura 4.2 se muestra el resultado obtenido por medio de este método.

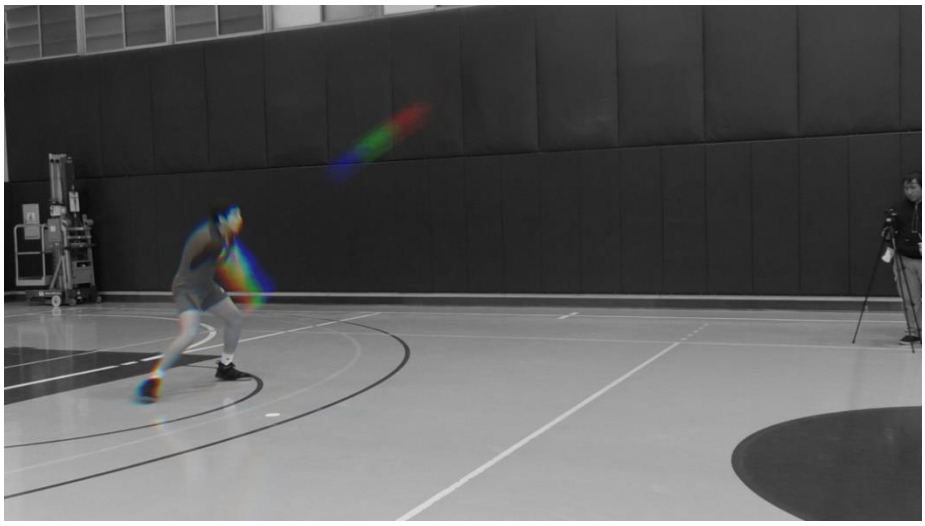
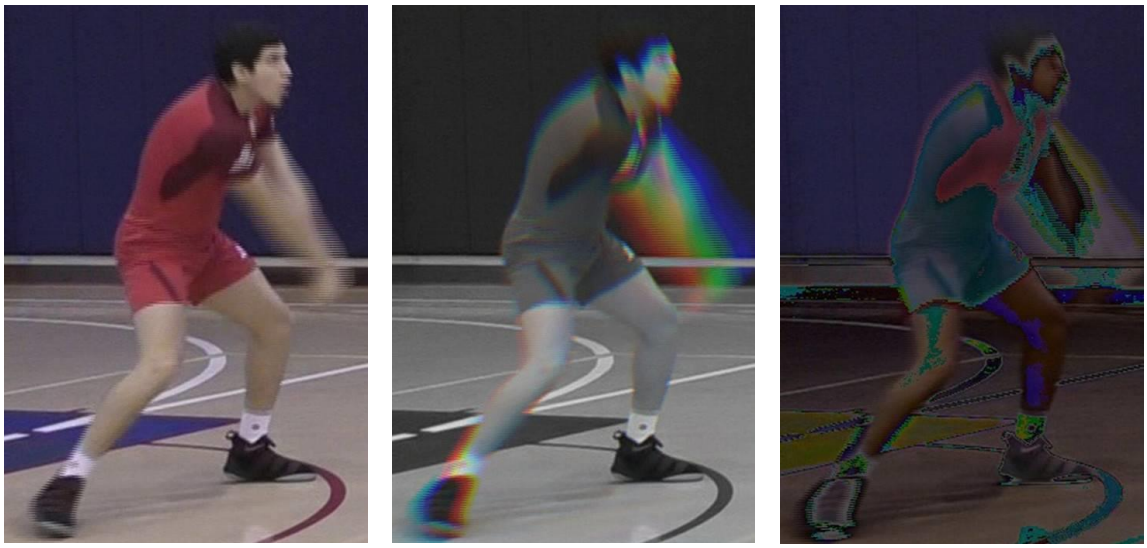


Figura 4.1: Resultado de unión de *frames* consecutivos en capas R, G y B

Figura 4.2: Resultado de promedio de tres *frames* consecutivos

4.1.3. Recorte de imágenes a la región de interés

Luego de haber realizado la unión de *frames* consecutivos, se procede a utilizar el modelo entrenado en el módulo de detección de jugador. Al ser este módulo entrenado sobre el conjunto de imágenes originales, se considerará la imagen central utilizada en estas uniones para determinar el área de interés. Es en base a dichas coordenadas que se realizará el recorte de estas imágenes. En la Figura4.3 se puede observar el resultado del recorte de la imagen original y las imágenes pre-procesadas.



(a) (b) (c)

Figura4.3: Resultado de recorte de imágenes. (a) Imagen original, (b) Imagen unida por capas RGB y (c) Imagen promedio

4.2. Definición de la arquitectura de la CNN

La red convolucional a utilizar estará basada en modelos ya existentes, tales como *ResNet*, esto con el objetivo de realizar un entrenamiento por medio de *Transfer Learning*. De acuerdo al modelo escogido, para el análisis de *frames* individuales se realizará una modificación sobre la última capa densa, la cual será una capa densa de 128 neuronas y se le adicionará una capa densa de 5-6 neuronas, de acuerdo al número de fases a trabajar, con lo cual será posible realizar la clasificación deseada por medio de una función de regresión logística softmax[4] (Figura 4.4).

Adicionalmente, para realizar el análisis de *Early Fusion*, se utilizará la concatenación de la salida de tres modelos ResNet a los cuales se les modificó la última capa densa por una capa densa de 120 neuronas. De esta manera, se concatena a un vector de 360 características y esto se continúa trabajando como una CNN común, culminando en una capa densa de 5-6 neuronas con las cuales se obtendrá la clasificación por medio de una función de regresión logística softmax, tal como se muestra en la Figura 4.5. El objetivo de esto es poder utilizar como entrada tres *frames* consecutivos tal como se describió el

[4] Función log_softmax. https://pytorch.org/docs/stable/nn.functional.html#torch.nn.functional.log_softmax

análisis de *Early Fusion* en el inciso 2.2.2.

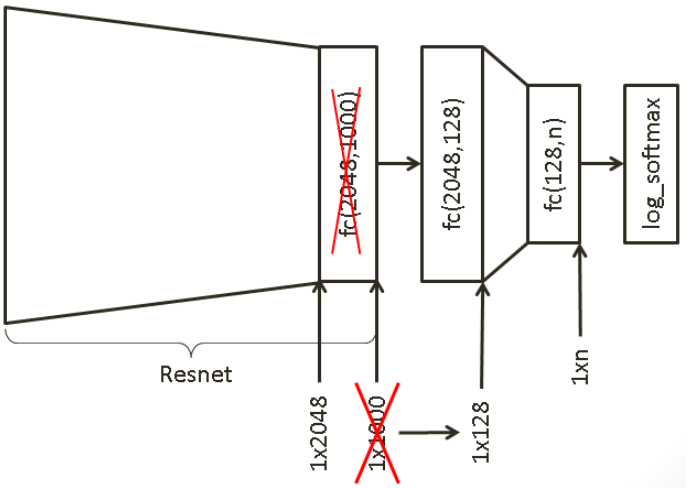


Figura 4.4: Esquema de modelo para análisis de frames individuales usando como base un modelo *ResNet* donde la capa tachada representa la última capa que fue reemplazada. El valor "n" indica la cantidad de fases a analizar.

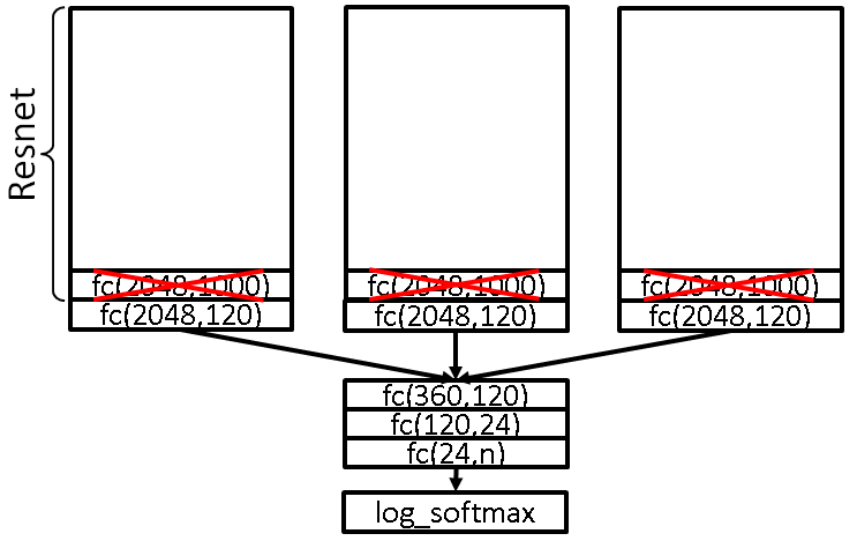


Figura 4.5: Esquema de modelo para análisis de *Early Fusion* usando tres columnas basadas en un modelo *ResNet* donde la capa tachada representa la capa que fue reemplazada. El valor "n" indica la cantidad de fases a analizar.

4.3. Entrenamiento

El entrenamiento de este módulo estará dividido en dos partes. En primera instancia se entrenará los modelos para análisis de frames individuales. Para ello se usará la arquitectura mostrada en la Figura 4.3. Se usará el principio de *Trasfer Learning*, por lo que se cargará un modelo *ResNet* preentrenado. El

entrenamiento se realizará usando un optimizador SGD[5], para el cual se fijará dos parámetros: tasa de aprendizaje y *momentum* [Ruder, 2016]. Adicional a ello se usará un *learning rate scheduler*[6] que se encargará de reducir la tasa de aprendizaje cada cierto número de épocas de entrenamiento.

Luego, se entrenará los modelos para el análisis por *Early Fusion*. Para ello se cargarán las imágenes en paquetes de tres *frames* consecutivos los cuales serán ingresados a la arquitectura mostrada en la Figura 4.4. De igual manera a los modelos descritos para el análisis por *frames* individuales, se utilizará modelos *ResNet* preentrenados y se usarán los mismos parámetros de optimización mencionados anteriormente.

[5] Optimizador SGD. https://pytorch.org/docs/stable/_modules/torch/optim/sgd.html
[6] Learning Rate Scheduler. https://pytorch.org/docs/stable/_modules/torch/optim/lr_scheduler.html

CAPÍTULO 5

Entrenamiento y Evaluación de Resultados

En este capítulo se mostrará cómo se llevó a cabo el proceso de etiquetado, entrenamiento y los resultados obtenidos.

5.1. Hardware y software utilizado

El entrenamiento de los modelos presentados se llevó a cabo usando una máquina que contaba con dos módulos GPU NVIDIA RTX2080®[7] de 8GB cada uno, un procesador Intel Core i7-9700K que posee una frecuencia de 3.6GHz, un disco duro de 2TB, un disco de estado sólido de 512GB y 64GB de RAM.

En cuanto al software utilizado, la máquina contó con el sistema operativo Ubuntu 16.04. Todo el trabajo fue realizado usando el lenguaje de programación *Python*[8], en su versión 3.6. El entrenamiento de los modelos a presentar se realizó en *Jupyter Notebook*[9] y la principal librería utilizada para manejar los modelos fue *Pytorch*[10], en su versión 1.3.

5.2. Conjuntos de datos

Los conjuntos de datos utilizados parten de dos sesiones de grabación realizadas en el polideportivo de la Pontificia Universidad Católica del Perú. En total se contó con tres jugadores (dos varones y una dama) y se utilizaron dos vistas a aproximadamente 45° del jugador, como se muestra en la Figura5.1.

[7] Tarjeta Gráfica GEFORCE RTX 2080. https://www.nvidia.com/en-us/geforce/graphics-cards/rtx-2080/
[8] Python. https://www.python.org/
[9] Jupyter Notebook. https://jupyter.org/
[10] Pytorch. https://pytorch.org/

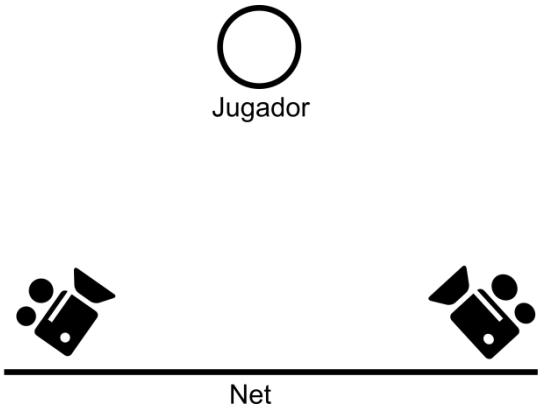


Figura 5.1.: Ubicación de cámaras usadas en la grabación ubicadas aproximadamente a 45° del jugador.

El conjunto de datos utilizado para el módulo de detección de jugador consistió de 19869 imágenes provenientes de los videos obtenidos de las dos cámaras utilizadas. Para cada una de estas imágenes se obtuvo un archivo .txt con las coordenadas descritas en 3.2.

En cuanto al conjunto de datos utilizado para el módulo de clasificación de fases, el etiquetado consistió en determinar la fase correspondiente para cada imagen. Usando que estas fases son temporalmente continuas, de cada video se extrajeron todos los *frames* y se determinó los *frames* correspondientes a los cambios de fases. Esta tarea fue realizada por tres personas capacitadas para identificación visual de las fases a analizar con el objetivo de eliminar la posible subjetividad al analizar estos cambios de fase. Se encontró que no hubo una diferenciación mayor de 5 *frames* entre los cambios de fase determinados por cada una de estas tres personas.

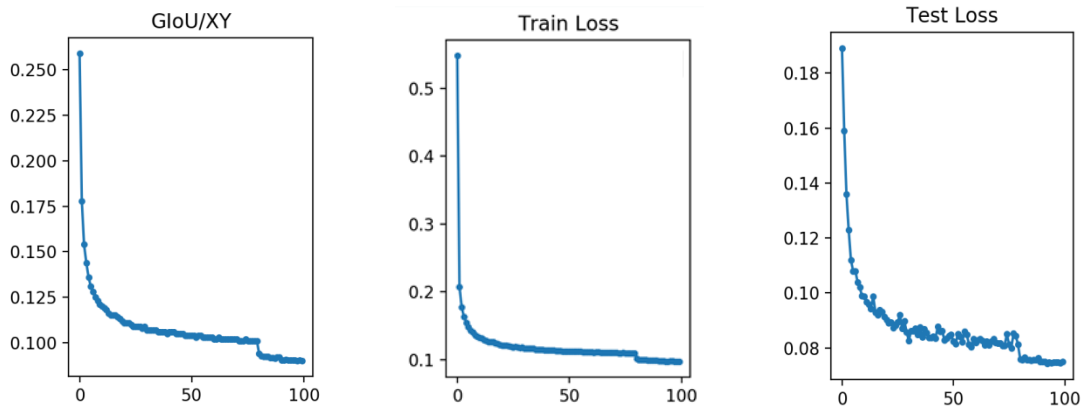
En total, se obtuvieron 14769 imágenes a partir de 69 videos grabados a 60fps. Luego de la clasificación, la distribución de imágenes por fases resultó ser la mostrada en la Tabla5.1. Se adicionó además imágenes correspondientes a una fase de reposo que indicaban estar fuera del gesto a analizar.

Tabla 5.1: Distribución de imágenes por fases

	Fase1	Fase2.1	Fase2.2	Fase3	Fase4	Reposo
Entrenamiento	2321	3107	931	852	1411	3165
Prueba	577	712	244	229	401	819

5.3. Entrenamiento de módulos

El entrenamiento del primer módulo se llevó a cabo durante 100 épocas, lo cual tomó un tiempo de 8 horas en total. El resultado del entrenamiento puede ser observado en la Figura 5.1, donde se muestra la métrica GIOU y la pérdida obtenida durante el entrenamiento.



(a) (b) (c)

Figura5.2: (a) Métrica GIoU, (b) Pérdida en entrenamiento y (c) Pérdida en prueba durante las 100 épocas de entrenamiento realizadas.

En cuanto al entrenamiento de los modelos de clasificación se llevó a cabo el planteamiento realizado en la sección 4.3. Para ambas arquitecturas a analizar se realizó el entrenamiento por un total de 25 épocas. En un principio se realizó variaciones sobre los parámetros del optimizador y del *learning rate scheduler* y los mejores resultados se obtuvieron con una tasa de aprendizaje de 0.001 y un *momentum* de 0.9 para el optimizador; y una reducción de la tasa de aprendizaje por un factor de 0.1 por cada 7 épocas para el *learning rate scheduler*. Es por ello que estos parámetros fueron mantenidos para todas las experimentaciones que se mencionan a continuación.

En el entrenamiento de los modelos para el análisis de *frames* individuales considero principalmente dos variaciones. En primer lugar se varió el conjunto de datos. Este se cambiaba entre las imágenes sin preprocesamiento (original), unidas por las capas RGB y las imágenes promediadas (Prom3) y también se analizó el efecto de considerar la fase de reposo en el entrenamiento, por lo que este se realizó considerando dicha fase y sin considerarla. Además a ello,

se varió el modelo *ResNet* base utilizado para la arquitectura. Este se cambió entre sus 5 variaciones: ResNet18, ResNet34, ResNet50, ResNet101, ResNet152. Los resultados de estos entrenamientos se pueden observar en las Tablas 5.2 y 5.3. En promedio, el tiempo de entrenamiento por cada uno de estos modelos fue de 1 hora.

Tabla5.2: Tasas de precisión considerando la fase de reposo en el conjunto de evaluación para los modelos de análisis por *frames* individuales. Cada modelo *ResNet* fue entrenado dos veces por separado.

Conj. Datos	ResNet18		ResNet34		ResNet50		ResNet101		ResNet152	
Original	89.13	90.17	89.54	66.67	85.18	90.17	80.21	90.85	90.88	**91.31**
RGB	91.01	71.09	90.27	90.98	**91.72**	91.48	77.90	90.91	91.35	91.68
Prom3	88.53	89.30	78.60	83.70	89.34	77.13	78.00	**90.31**	89.67	89.74

Tabla5.3: Tasas de precisión sin considerar la fase de reposo en el conjunto de evaluación para los modelos de análisis por *frames* individuales. Cada modelo *ResNet* fue entrenado dos veces por separado.

Conj. Datos	ResNet18		ResNet34		ResNet50		ResNet101		ResNet152	
Original	91.17	91.08	**91.72**	90.85	73.46	82.94	90.75	90.61	90.98	91.12
RGB	91.17	90.89	**92.19**	91.08	73.88	56.17	90.80	91.26	91.54	91.49
Prom3	89.32	90.43	82.89	82.85	**90.94**	72.86	90.43	82.34	82.34	74.11

De estos resultados, se observó que el mejor modelo sin usar la fase de reposo se consiguió con el conjunto de datos original y usando el modelo ResNet34, brindando un porcentaje de precisión de 92.19%. En cambio, el mejor modelo usando la fase de reposo se consiguió con el conjunto de datos RGB y usando el modelo Resnet50, brindando un porcentaje de precisión de 91.72%. Los errores generados en cada uno de estos modelos se muestran en las matrices de confusión mostradas en las Tablas 5.4 y 5.5.

En el entrenamiento de los modelos para el análisis de *Early Fusion* se varió el modelo ResNet base utilizado para la arquitectura. De igual manera, este se cambió entre sus 5 variaciones: ResNet18, ResNet34, ResNet50, ResNet101, ResNet152 y se consideró el análisis usando y sin usar la fase de reposo. Los resultados de estos entrenamientos se pueden observar en la Tabla 5.6. En promedio, el tiempo de entrenamiento por cada uno de estos modelos fue desde 1 hora para el modelo más simple (ResNet18) hasta 5 horas para el modelo más pesado (ResNet152).

Tabla5.4: Matriz de confusión para mejor modelo de análisis de frames individuales sin usar la fase de reposo. Conjunto de datos: RGB. Modelo ResNet34

	Fase1	Fase2.1	Fase2.2	Fase3	Fase4
Fase1	550	27	0	0	0
Fase2.1	41	655	16	0	0
Fase2.2	0	26	201	17	0
Fase3	0	1	31	197	0
Fase4	0	0	0	10	391

Tabla5.5: Matriz de confusión para mejor modelo de análisis de frames individuales usando la fase de reposo. Conjunto de datos: RGB. Modelo ResNet50

	Fase1	Fase2.1	Fase2.2	Fase3	Fase4	Reposo
Fase1	536	41	0	0	0	0
Fase2.1	24	668	20	0	0	0
Fase2.2	0	20	207	17	0	0
Fase3	0	0	33	190	6	0
Fase4	0	0	1	10	354	36
Reposo	0	0	0	1	38	780

Tabla5.6: Tasas de precisión en el conjunto de evaluación para los modelos de análisis de Early Fusion

Conj. Datos	ResNet18	ResNet34	ResNet50	ResNet101	ResNet152
Reposo	88.63	69.35	58.52	**90.71**	84.81
No Reposo	89.60	63.99	44.66	72.72	**91.08**

De estos resultados, se observó que el mejor modelo usando la fase de reposo se consiguió con el modelo ResNet101, brindando un porcentaje de precisión de 90.71%. En cambio, el mejor modelo sin usar la fase de reposo se obtuvo con el modelo ResNet152 obteniendo un porcentaje de precisión de 91.08%. Los errores generados en cada uno de estos modelos se muestran en las matrices de confusión mostradas en la Tablas 5.7 y 5.8.

Tabla 5.7: Matriz de confusión para mejor modelo de análisis de Early Fusion sin usar la fase de reposo. Modelo ResNet152

	Fase1	Fase2.1	Fase2.2	Fase3	Fase4
Fase1	545	32	0	0	0
Fase2.1	26	649	37	0	0
Fase2.2	0	36	196	12	0
Fase3	0	0	34	191	4
Fase4	0	0	2	10	389

Tabla 5.8: Matriz de confusión para mejor modelo de análisis de Early Fusion usando la fase de reposo. Modelo ResNet101

	Fase1	Fase2.1	Fase2.2	Fase3	Fase4	Reposo
Fase1	533	39	0	0	0	5
Fase2.1	20	651	40	0	0	1
Fase2.2	1	28	198	17	0	0
Fase3	0	0	35	194	0	0
Fase4	0	0	0	10	359	32
Reposo	1	11	3	0	34	770

Conclusiones, Observaciones y Recomendaciones

En el presente trabajo se consiguió construir un conjunto de videos de tres jugadores peruanos que realizan la técnica de recepción en el vóley. Estos videos se encuentran etiquetados con estampas temporales para la división de fases temporales de dicho gesto.

Es en base a este conjunto de datos que se propone un modelo algorítmico que permite identificar a partir de un video las fases temporales de un gesto de recepción en el vóley. Esto fue constituido en dos partes: detección del jugador y clasificación de las fases. La primera parte consiguió buenos resultados, siendo usado actualmente para el recorte de *frames* de videos no usados en este trabajo.

En cuanto al módulo de clasificación de fases, se obtuvo que el mejor de los modelos entrenados predecía correctamente la fase con una certeza de 92.19%. Además a esto, se observó que en los errores que se llegan a cometer, la fase predicha es una de las contiguas a la fase real, salvo pocas excepciones. Esto indica que este modelo tiene buena capacidad de predecir la temporalidad con la que ocurren las fases y de esta manera no presentar un error muy grande en la clasificación.

Como trabajo futuro se plantea incrementar esta base de datos con mayor diversidad de jugadores y posiblemente otros entornos que permitan obtener un conjunto de datos en diferentes condiciones de ambiente. Parte de esto ayudaría a poder analizar la influencia de utilizar diversidad de jugadores, tanto en género como en morfología de los jugadores. Adicionalmente se plantea explorar otros modelos que puedan incorporar mejor la información temporal, tales como redes neuronales recurrentes u otros métodos de fusión temporal.

Bibliografía

[Francia, 2018]
Francia, S., Calderara, S., & Lanzi, D. F. (2018). Classificazione di Azioni Cestistiche mediante Tecniche di Deep Learning.

[GIRAB, 2019]
GIRAB (2019). Caracterización biomecánica del movimiento asociada a la técnica de recepción en vóley categoría juvenil damas. Retrieved from http://investigacion.pucp.edu.pe/grupos/girab/proyecto/biomecanica-del-movimiento-la-recepcion-del-voley/

[Glenn, 2019]
Glenn Jocher et al. (2019). ultralytics/yolov3: Rectangular Inference, Conv2d + Batchnorm2d Layer Fusion (Version v6). Zenodo. http://doi.org/10.5281/zenodo.2672652

[Ibrahim, 2016]
Ibrahim, M. S., Muralidharan, S., Deng, Z., Vahdat, A., & Mori, G. (2016). A hierarchical deep temporal model for group activity recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1971-1980).

[JUMP, 2014]
JUMP - Una piattaforma sensoristica avanzata per rinnovare la pratica e la fruizione dello sport, del benessere, della riabilitazione e del gioco educativo. (n.d.). Retrieved January 25, 2019, from http://www.aimagelab.unimore.it/imagelab/project.asp?idProgetto=56

[Karpathy, 2014]
Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1725-1732).

[Kautz, 2017]
Kautz, T., Groh, B. H., Hannink, J., Jensen, U., Strubberg, H., & Eskofier, B. M. (2017). Activity recognition in beach volleyball using a Deep Convolutional Neural Network. Data Mining and Knowledge Discovery, 31(6), 1678-1705.

[Ramasamy, 2018]
Ramasamy Ramamurthy, S., & Roy, N. (2018). Recent trends in machine learning for human activity recognition—A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4), e1254.

[Redmon, 2016]
Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[Rezatofighi, 2019]
Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 658-666).

[Ruder, 2016]
Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

[Russakovsky, 2015]
Russakovsky, O. et al. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.

[Singh, 2019]
Singh, T., & Vishwakarma, D. K. (2019). Human Activity Recognition in Video Benchmarks: A Survey. In Advances in Signal Processing and Communication (pp. 247-259). Springer, Singapore.

[Suárez, 2009]
Suárez, G. R. (2009). Biomecánica deportiva. Biomecánica deportiva y control del entrenamiento,4, 15.

[Zhao, 2019]
Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems.