



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
SEDE MORONA SANTIAGO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA TECNOLOGÍAS DE LA INFORMACIÓN

**DESARROLLO DE UN SOFTWARE DE RECONOCIMIENTO DE
ERRORES EN POWERLIFTING USANDO VISIÓN ARTIFICIAL**

Trabajo de Titulación

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

AUTOR:

JAIME ALEXANDER JIMÉNEZ PERALTA

Macas - Ecuador

2024



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
SEDE MORONA SANTIAGO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA TECNOLOGÍAS DE LA INFORMACIÓN

**DESARROLLO DE UN SOFTWARE DE RECONOCIMIENTO DE
ERRORES EN POWERLIFTING USANDO VISIÓN ARTIFICIAL**

Trabajo de Titulación

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

AUTOR: JAIME ALEXANDER JIMÉNEZ PERALTA

DIRECTOR: DR. XAVIER SORIA POMA

Macas - Ecuador

2024

© 2024, Jaime Alexander Jiménez Peralta

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Jaime Alexander Jiménez Peralta, declaro que el presente Trabajo de Titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este Trabajo de Titulación; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

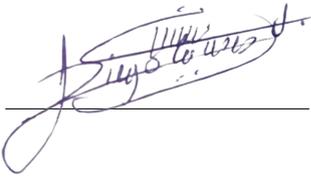
Macas, 11 de Diciembre de 2024



Jaime Alexander Jiménez Peralta
1400814420

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
SEDE MORONA SANTIAGO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA TECNOLOGÍAS DE LA INFORMACIÓN

El Tribunal del Trabajo de Titulación certifica que: El Trabajo de Titulación; Tipo: Proyecto Técnico: **DESARROLLO DE UN SOFTWARE DE RECONOCIMIENTO DE ERRORES EN POWERLIFTING USANDO VISIÓN ARTIFICIAL**, realizado por el señor: **JAIME ALEXANDER JIMÉNEZ PERALTA**, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Mgtr. Diego Alejandro Cáceres Veintimilla. PRESIDENTE DEL TRIBUNAL	 _____	2024-12-11
Dr. Xavier Soria Poma. DIRECTOR DEL TRABAJO DE TITULACIÓN	 _____	2024-12-11
Mgtr. Alberto Alzola Tamayo. ASESOR DEL TRABAJO DE TITULACIÓN	 _____	2024-12-11

DEDICATORIA

A Dios, fuente de fortaleza y guía en cada paso de mi trayectoria académica. A mi querida madre, cuyo apoyo incondicional ha sido mi mayor motivación y sostén en este arduo camino. A mi familia, por su constante aliento y comprensión, brindándome el impulso necesario para alcanzar mis metas. A mis amigos, cuya amistad y compañerismo han iluminado los días más difíciles y compartido las alegrías más grandes. A Kanye West, cuya música ha sido mi banda sonora durante este proceso, inspirándome con sus letras profundas y su creatividad sin límites, recordándome que la pasión y la innovación son clave en cualquier proyecto. Y a los miembros del gimnasio StronGym, por su colaboración y apoyo en el desarrollo de este trabajo, demostrando que el trabajo en equipo es fundamental para alcanzar el éxito.

Jaime.

AGRADECIMIENTO

Expreso mi profundo agradecimiento a la Escuela Superior Politécnica de Chimborazo y a la Carrera de Ingeniería en Tecnologías de la Información por la educación de calidad que he recibido, así como a mis estimados docentes por su invaluable enseñanza. A mi madre, mi mejor amiga y apoyo incondicional, le agradezco su amor y guía, fundamentales en mi desarrollo personal y profesional. Sin su constante respaldo, este logro no sería posible.

Jaime.

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS	ix
ÍNDICE DE ILUSTRACIONES	x
RESUMEN	xi
SUMMARY/ABSTRACT	xii
INTRODUCCIÓN	1
CAPÍTULO I	3
1 DIAGNÓSTICO DEL PROBLEMA	3
1.1 Planteamiento del Problema	3
1.2 Justificación	4
1.2.1 Justificación teórica	4
1.2.2 Justificación aplicativa	4
1.3 Objetivos	5
1.3.1 Objetivo general	5
1.3.2 Objetivos específicos	5
CAPÍTULO II	7
2 MARCO TEÓRICO	7
2.1 Antecedentes	7
2.2 Powerlifting	8
2.2.1 Metodologías de entrenamiento en el powerlifting	9
2.2.2 Lesiones en Powerlifting	10
2.2.3 Errores comunes dentro del peso muerto.	10
2.3 Detección de errores en el peso muerto	11
2.3.1 Visión Artificial	12
2.3.2 Procesamiento de imágenes	13
2.3.3 Generación de dataset	13
2.3.4 Clasificaciones de imágenes con Deep Learning	15
2.3.5 Hyerparametros para la detección de errores de peso muerto	16
2.4 Desarrollo de software	16
2.4.1 Ciclo de vida del Software	17
2.4.2 Metodología de desarrollo	17
2.4.3 Herramientas para el desarrollo	18
CAPÍTULO III	19
3 MARCO METODOLÓGICO	19
3.1 Alcance	19

3.2	Tipo de investigación	19
3.3	Estructura de la solución	19
3.4	Técnicas e instrumentos de investigación	20
3.5	Desarrollo del sistema	21
3.5.1	Análisis	22
3.5.2	Diseño	24
3.5.3	Implementación	27
3.5.4	Pruebas	31
3.5.5	Mantenimiento	31

CAPÍTULO IV 33

4	ANÁLISIS E INTERPRETACIÓN DE RESULTADOS 33
4.1	Detalles de implementación 33
4.2	Aplicación web 33
4.3	Resultados de la validación del clasificador 34
4.4	Evaluación en tiempo real de la aplicación 37
4.5	Discusión 39

CAPÍTULO V 41

5	CONCLUSIONES Y RECOMENDACIONES 41
5.1	Conclusiones 41
5.2	Recomendaciones 41

BIBLIOGRAFÍA

ÍNDICE DE TABLAS

Tabla 2-1:	Comparación entre metodologías ágiles y metodologías tradicionales.	17
Tabla 3-1:	Técnicas y herramientas	21
Tabla 3-2:	Requisitos funcionales.	22
Tabla 3-3:	Requisitos no funcionales.	22
Tabla 3-4:	Descripción del caso de uso.	23
Tabla 3-5:	Flujo de trabajo de la aplicación.	25
Tabla 4.1:	Resultados de la predicción de errores en el peso muerto. Entrenador 1 (E1) y Entrenador 2 (E2).	34

ÍNDICE DE ILUSTRACIONES

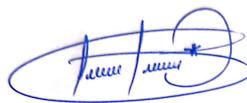
Ilustración 2.2-1: Movimiento del peso muerto.	9
Ilustración 2.2-2: Errores comunes dentro del peso muerto.	11
Ilustración 2.3-3: Fases del proceso de funcionamiento de la visión artificial al determinar una imagen.	13
Ilustración 2.3-4: Proceso de restauración de color en imágenes mediante Deep Learning.	13
Ilustración 2.3-5: Proceso de generación de un dataset.	15
Ilustración 3.3-1: Diseño de la evaluación del software.	20
Ilustración 3.5-2: Fases modelo cascada.	21
Ilustración 3.5-3: Diagrama de caso de uso: uso de la aplicación.	23
Ilustración 3.5-4: Diagrama de secuencia de la aplicación.	24
Ilustración 3.5-5: Flujo de trabajo de la aplicación.	24
Ilustración 3.5-6: Diseño de interfaz de la aplicación 1.	25
Ilustración 3.5-7: Diseño de interfaz de la aplicación 2.	26
Ilustración 3.5-8: Diseño de interfaz de la aplicación 3.	26
Ilustración 3.5-9: Arquitectura de MobileNet.	27
Ilustración 3.5-10: Clasificación por clases de imágenes para el entrenamiento del modelo.	28
Ilustración 3.5-11: Codificación de la actualización de las etiquetas.	30
Ilustración 3.5-12: Codificación de la interfaz de inicio.	30
Ilustración 4.3-1: Resultados obtenidos.	37
Ilustración 4.4-2: Rendimiento de la aplicación web en escritorio.	38
Ilustración 4.4-3: Rendimiento de la aplicación web en dispositivos móviles.	39

RESUMEN

Con el aumento de usuarios en los gimnasios se incrementa la necesidad de supervisión de ejercicios compuestos como el peso muerto, ya que la ejecución de una técnica incorrecta puede causar lesiones graves en el deportista. En este proyecto se desarrolló un prototipo de aplicación web que emplea visión artificial para detectar en tiempo real errores como la colocación incorrecta de la cadera, el redondeo excesivo de la espalda y la hiperextensión al final del levantamiento dentro del peso muerto (powerlifting). La metodología de desarrollo utilizada siguió un enfoque en cascada, estructurándose en distintas fases: una revisión documental para seleccionar algoritmos inteligentes; el desarrollo e integración de módulos y el modelo, y la evaluación del prototipo en un entorno controlado. Se capturaron y etiquetaron imágenes de distintas variaciones en la técnica del levantamiento de peso muerto para entrenar un modelo que pudiera ejecutarse en navegadores web. Al evaluar la precisión global del detector, arrojó un porcentaje de detección del 73.36%, es un desempeño satisfactorio en la identificación de errores críticos. Este sistema ofrece una solución versátil y robusta para la corrección de la técnica y prevención de lesiones en el peso muerto; al ser una alternativa de bajo costo y de fácil implementación, su uso y aplicación en tiempo real permitirá que más negocios puedan usarlo. Como futuras mejoras, se recomienda optimizar el modelo para aumentar la precisión en la detección de variaciones menos evidentes.

Palabras clave: <APRENDIZAJE PROFUNDO >, <TEACHABLE MACHINE>, <APLICACIÓN WEB >, <DETECCIÓN DE ERRORES>, <PESO MUERTO>.

0010-DBRAI-UPT-2025

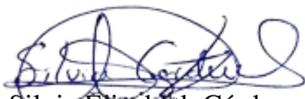


ABSTRACT

With the increase of users in gyms, the need for supervision of compound exercises such as powerlifts is increasing, since the execution of an incorrect technique can cause serious injuries to the athlete. In this project, a prototype web application was developed that uses computer vision to detect in real time errors such as incorrect hip placement, excessive back rounding and hyperextension at the end of the deadlift in powerlifting. The development methodology used followed a waterfall approach, structured in distinct phases: a document review to select intelligent algorithms; the development and integration of modules and the model; and the evaluation of the prototype in a controlled environment. Images of different variations of the power weight lifting technique were captured and labeled to train a model that could be run in web browsers. When evaluating the overall accuracy of the detector, it yielded a detection rate of 73.36%, a satisfactory performance in identifying critical errors. This system offers a versatile and robust solution for technique correction and power weight injury prevention; being a low cost and easy to implement alternative, its use and application in real time will allow more businesses to use it. As future improvements, it is recommended to optimize the model to increase accuracy in the detection of less obvious variations.

Keywords: <DEEP LEARNING >, <TEACHABLE MACHINE >, <WEB APPLICATION>, <ERROR DETECTION >, <DEADLIFT >.

0010-DBRAI-UPT-2025



Lic. Silvia Elizabeth Cárdenas Sánchez Mgs.

Docente Sede Morona Santiago

Cédula: 0603927351

INTRODUCCIÓN

Introducción

En los últimos años, la afluencia a los gimnasios en Ecuador ha crecido notablemente (62), debido a un mayor interés en mantener una vida saludable en la población. Sin embargo, este crecimiento ha traído consigo nuevos retos, como la supervisión en el entrenamiento de ejercicios complejos como el peso muerto, ya que la falta de una técnica correcta puede derivar en lesiones en la zona lumbar, desgarros musculares, deterioro de las articulaciones y la disminución de la movilidad a largo plazo en el caso de aficionados (5). De igual manera, la ausencia de supervisión por parte de un entrenador puede llevar a la repetición de errores que afectan al progreso del atleta.

En este contexto, al incorporar la tecnología en los gimnasios podría ser un aliado en el monitoreo de atletas y así reducir los problemas mencionados anteriormente. Una de las opciones para supervisar la técnica del peso muerto es la incorporación de sistemas basados en Deep Learning (DL) y Visión Artificial (VA) que automaticen la detección de errores en el peso muerto (56). Precisamente, esta trabajo con el propósito de contribuir a la solución de este problema se ha desarrollado un sistema web donde incorpora DL y VA para automatizar el reconocimiento en tiempo real de errores comunes en el peso muerto, como la colocación incorrecta de la cadera, el redondeo excesivo de la espalda y la extensión excesiva al final del levantamiento. El desarrollo de la aplicación web fue basado en el modelo en cascada, que organiza el proceso en etapas secuenciales y garantiza un avance sistemático (46).

El informe del proyecto técnico está organizado de la siguiente manera:

CAPÍTULO I: Describe los antecedentes del problema, la justificación teórica y aplicativa, así como los objetivos generales y específicos.

CAPÍTULO II: Presenta la conceptualización teórica de temas relacionados con el powerlifting, algoritmos de visión artificial, y metodologías de desarrollo de software.

CAPÍTULO III: Contiene el marco metodológico, dentro del cual se describe el alcance, tipo de investigación, métodos y técnicas para el desarrollo del sistema y el diseño de la interfaz.

CAPÍTULO IV: Presenta los resultados obtenidos mediante la evaluación del software, así como el análisis de los datos recopilados.

CAPÍTULO V: Expone las conclusiones obtenidas de acuerdo con los objetivos planteados, así

como las recomendaciones para futuros trabajos.

CAPÍTULO I

1. DIAGNÓSTICO DEL PROBLEMA

1.1. Planteamiento del Problema

En Ecuador, el uso de gimnasios ha aumentado en los últimos años, convirtiéndose en espacios importantes para la práctica de ejercicio físico (62), lo cual refleja una mayor concienciación sobre la importancia del deporte. Sin embargo, este aumento en la afluencia también ha revelado diversos problemas relacionados con el entrenamiento y la supervisión en estos entornos, ya que la falta de personal capacitado y la necesidad de monitoreo continuo son desafíos que enfrentan tanto los gimnasios grandes como los pequeños centros de entrenamiento. La supervisión constante de un entrenador para todos los deportistas no siempre es posible debido a limitaciones de tiempo y recursos por parte de los gimnasios, lo que afecta especialmente a los nuevos deportistas aficionados que se integran a estos espacios y son más propensos a sufrir accidentes leves o graves (48).

En este contexto, el riesgo de lesiones derivadas de una mala técnica se vuelve más latente en los atletas que están comenzando en deportes multiarticulares como el powerlifting (14). Estas lesiones comunes pueden incluir problemas en la zona lumbar debido a una colocación incorrecta de la cadera, lo que aumenta la presión sobre la espalda baja; esguinces y desgarros musculares, especialmente en la espalda y los isquiotibiales; y un deterioro de las articulaciones provocado por el hábito de realizar el ejercicio de manera incorrecta. Como consecuencia, a largo plazo, estas lesiones pueden resultar en una disminución permanente de la movilidad en los atletas (42).

En cuanto al ámbito del peso muerto dentro del powerlifting, actualmente, los análisis técnicos en este deporte se basan mayormente en observaciones humanas ya sea durante la ejecución de este deporte o a través de análisis de vídeo realizados por los entrenadores (25). Por otro lado, en los últimos años han aplicado diferentes tecnologías en el monitoreo de este deporte como dispositivos de Unidades de Medición Inercial (IMUs) (20). Sin embargo, estas nuevas tecnologías son complejas o tienen un alto costo en su adquisición. Además, a pesar del uso de la inteligencia artificial (IA) por varios autores dentro del powerlifting, este enfoque todavía no ha sido completamente explorado sobre todo en términos de facilidad de uso y bajo costo de implementación (59). En otros deportes, la aplicación de IA y VA ha sido utilizada para mejorar el rendimiento deportivo en áreas como la detección de jugadores, la predicción de trayectorias y el reconocimiento de estrategias de equipo (40). No obstante, dentro del peso muerto, sus aplicaciones aún no han sido completamente adoptadas, ya que no están enfocadas en el entrenamiento diario (50).

A pesar de que hay distintas tecnologías y métodos para detectar errores a la hora de practicar el peso muerto, estas soluciones, como se mencionó anteriormente, requieren una inversión considerable de dinero, lo cual puede ser una barrera para pequeños negocios locales o para aquellos que no disponen de espacios adecuados para el funcionamiento de dichas tecnologías. Entonces, la inclusión de técnicas de IA y VA podría optimizar el monitoreo y, al mismo tiempo, reducir el costo asociado al uso de herramientas o especialistas en el monitoreo de la actividad del peso muerto.

1.2. Justificación

1.2.1. Justificación teórica

Con el aumento en la práctica del peso muerto dentro del powerlifting por parte de atletas (17), se vuelve imperioso el uso de tecnologías que agilicen y optimicen el monitoreo de la técnica de los deportistas sin la necesidad de una supervisión constante ya que debido a un aumento en la aglomeración en los gimnasios, los entrenadores no pueden llevar un control constante en la técnica de los usuarios (62) y, con ello, reducir de manera indirecta lesiones comunes en la zona lumbar como hernias discales, desgarros musculares, esguinces, dolores en la zona lumbar y otros problemas en la columna vertebral (35), debido a la ejecución de una mala técnica.

Esta aplicación permite detectar en tiempo real los errores comunes en el peso muerto. La capacidad de funcionamiento en computadoras personales con alto grado de eficiencia y productividad es gracias al uso optimizado de los modelos de inteligencia artificial apropiados para dispositivos que tengan acceso a un navegador web. El detector de errores previamente fue entrenado con una de las arquitecturas de redes neuronales convolucionales más rápida, como lo es MobileNet (38). La app al ser una basada en la web permitirá que más usuarios puedan usarlo e instalarlo con mayor facilidad. Además, el modelo de clasificación, desarrollado con algoritmos de deep learning, proporcionará mayor robustez en la detección de errores durante la ejecución del peso muerto.

1.2.2. Justificación aplicativa

La creación de un sistema web que permita el reconocimiento en tiempo real de los errores más comunes en el peso muerto, como la colocación incorrecta de la cadera, el redondeo excesivo de la espalda y la extensión excesiva al final del levantamiento, es crucial ya que son causantes de la mayor parte de lesiones en la zona lumbar (42), consiguiendo su aplicación a través de la incorporación de un modelo basado en VA y DL, puesto que estas tecnologías se caracterizan por resolver los problemas con mayor grado de precisión (4). De esta manera, con una cámara web, una computadora personal y un navegador para ejecutar la aplicación web, se puede automatizar el monitoreo de los errores más comunes en el peso muerto realizado por los deportistas.

La propuesta de aplicación web al ser desarrollado con Django y el modelo de detección implementado con una librería más usada en la industria como lo es TensorFlow (51), estará al alcance de cualquier usuario con un navegador y acceso a internet, lo que eliminaría la necesidad de equipos adicionales. Esta accesibilidad permite que tanto deportistas nuevos como aficionados que tengan acceso a un dispositivo con navegador web puedan beneficiarse de la detección de sus errores al realizar los ejercicios del peso muerto. La facilidad de uso y la disponibilidad en línea también fomentan una comunidad más amplia de usuarios que pueden compartir experiencias y consejos, enriqueciendo el proceso de aprendizaje de la técnica.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar un software de reconocimiento en tiempo real para determinar los errores comunes de peso muerto dentro del powerlifting en gimnasios usando visión artificial.

1.3.2. Objetivos específicos

- Determinar los algoritmos de visión artificial para detectar errores comunes en el peso muerto dentro del powerlifting.
- Desarrollar un prototipo de software que reconozca en tiempo real la técnica de peso muerto dentro del powerlifting usando algoritmos de visión artificial.
- Evaluar la función en tiempo real del prototipo de software desarrollado para corrección de errores de peso muerto dentro del powerlifting, considerando la postura, la alineación y el rango de movimiento.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Antecedentes

En la búsqueda de proyectos de software inteligente para la detección de errores en el ejercicio de peso muerto mediante visión artificial, se llevó a cabo una revisión de la literatura siguiendo los parámetros que se exponen a continuación. Este proceso implicó la consulta de diversas fuentes académicas, como Google Scholar, IEEE Xplore y PubMed. Durante la revisión, se utilizó el filtro "Powerlifting errors detection through artificial intelligence", pero solo se pudo acceder a recursos académicos de libre acceso. A lo largo de esta revisión, se identificaron varios trabajos similares que han sido aplicados en el ámbito del powerlifting y en otras disciplinas relacionadas con la corrección de técnicas deportivas mediante inteligencia artificial y Deep Learning.

- El proyecto realizado por Singh & Taqa [52], desarrolló un modelo basado en redes neuronales convolucionales (CNN) con el fin de detectar errores en la técnica del peso muerto, el cual fue entrenado utilizando un conjunto de 500 imágenes, organizadas en dos clases: "buena técnica" y "mala técnica". Alcanzando una precisión de prueba del 80.5%. El presente trabajo de titulación se diferencia de la propuesta realizada por Singh & Taqa [52] en que, además del desarrollo de un modelo para la detección de errores en el peso muerto, este se integrará en un sistema web.
- El trabajo desarrollado por Rosenhaim [50], realizó un modelo de posturas utilizando OpenPose para clasificar la ejecución de ejercicios como la sentadilla, press banca y peso muerto, logrando una precisión del 97.1% en la identificación de estos ejercicios. A diferencia de dicha propuesta, el presente trabajo de titulación se enfoca en la identificación de errores en el peso muerto, integrando además el modelo a un sistema web.
- Durante el trabajo de Sun **et al.** [56], se hizo uso de redes neuronales recurrentes (RNN) junto con una cámara Kinect y varios sensores inerciales colocados en el cuerpo del usuario para capturar para el reconocimiento de errores en el peso muerto, consiguiendo una precisión promedio del 79.99%. A diferencia de esa propuesta, el presente trabajo de titulación prescinde del uso de sensores adicionales y se centra en el reconocimiento de errores en la técnica mediante visión artificial, integrando el modelo en un sistema web accesible desde cualquier dispositivo con cámara.
- El trabajo realizado por Militaru **et al.** [37], presenta una solución para dispositivos móviles

basada en redes neuronales convolucionales (CNN) para corregir la postura durante ejercicios de calistenia, como el plank y el squat sostenido. El trabajo consistió en la creación de un conjunto de datos compuesto por 2400 imágenes etiquetadas, clasificadas en tres categorías: postura correcta, caderas demasiado bajas y caderas demasiado altas, alcanzando una precisión promedio del 95%. Si bien el estudio previo aborda el uso de CNN para la corrección de la técnica en ejercicios de calistenia, el presente trabajo de titulación tiene un enfoque en el peso muerto, integrándolo a un sistema web.

- Jiunn [22] diseñó un sistema que reconozca la postura correcta e incorrecta de ejercicios como curl de bíceps, levantamiento frontal y press de hombros, utilizando el algoritmo K-Nearest Neighbors (KNN) para su creación, consiguiendo una precisión del 78.60%. A diferencia de esa propuesta, el presente trabajo de titulación se enfoca en la detección de errores en el peso muerto haciendo uso de redes neuronales convolucionales (CNN).
- Parisi **et al.** [43] hizo uso de un sensor de profundidad (Kinect) para la creación de un sistema para proporcionar retroalimentación automática sobre la ejecución de sentadilla con barra alta, levantamiento de peso muerto y elevaciones laterales con mancuernas, consiguiendo una precisión del 95.6% en estos ejercicios. A diferencia de esa propuesta, el presente trabajo de titulación no hace uso de sensores adicionales, e integra el sistema en una plataforma web para facilitar su uso.
- Y por último, Um **et al.** [60], desarrolló un sistema para la clasificación de movimientos de 50 ejercicios de gimnasio utilizando datos de sensores y redes neuronales convolucionales (CNN). Los datos, obtenidos del sensor de muñeca, consistió en mediciones de aceleración y orientación, las cuales se transforman en imágenes para que las CNN puedan extraer características automáticamente, con el objetivo de desarrollar un sistema automático de rehabilitación y entrenamiento deportivo, consiguiendo con la mejor configuración una precisión del modelo del 92.14%. El presente trabajo de titulación se diferencia de la propuesta realizada por Um **et al.** [60] en que se enfoca solamente en la detección de errores en la técnica del peso muerto, sin recurrir a sensores adicionales.

2.2. Powerlifting

El Powerlifting es un deporte que implica levantar la mayor cantidad de peso posible utilizando una barra equipada con discos de hierro de dimensiones específicas, donde se encuentran tres ejercicios fundamentales que son el press de banca, el peso muerto y la sentadilla [49]. También, se menciona que el peso máximo levantado en cada ejercicio se suma para determinar la puntuación total.

Peso Muerto convencional: Este ejercicio forma parte de los movimientos básicos en el ámbito

del powerlifting, el cual se emplea en el desarrollo de la aptitud neuromuscular, además de involucrar varias articulaciones, dirigido al fortalecimiento de los músculos de la cadera, piernas y espalda [16]. El ejercicio inicia con el individuo posicionándose detrás de la barra, adoptando una postura similar a la de una sentadilla, con una flexión de aproximadamente 80°-100° en las rodillas y caderas, mientras extiende los brazos para sujetar la barra (ver Figura 2.2-1). Una vez agarrada, se aplica fuerza con el propósito de erguirse en posición vertical [32].



Ilustración 2.2-1: Movimiento del peso muerto.
Realizado por: Australian Fitness Academy, 2017.

2.2.1. Metodologías de entrenamiento en el powerlifting

Dentro del powerlifting, existen diversas metodologías que se utilizan para llevar a cabo el entrenamiento. Entre las más usadas se encuentran:

- **Entrenamiento de fuerza máxima:** Este tipo de entrenamiento se centra en levantar la carga más pesada a 1 RM (repetición máxima). Esto implica realizar una contracción voluntaria máxima de un músculo o grupo de músculos específicos contra una resistencia externa, buscando la mayor tensión posible, su objetivo es mejorar la capacidad de generar fuerza bajo condiciones específicas, optimizando la motivación y control neuromuscular [45].
- **Entrenamiento de Volumen:** Según Colquhoun et al. [7] consiste en realizar un mayor número de series y repeticiones con cargas moderadas, está diseñado para aumentar la hipertrofia muscular y la resistencia.
- **Entrenamiento de Potencia:** Este tipo de entrenamiento se centra en mejorar la capacidad de generar fuerza rápidamente, combinando fuerza y velocidad [41].

2.2.2. Lesiones en Powerlifting

De acuerdo a la revisión de Aasa **et al.** [1] sobre el deporte del powerlifting, se obtuvo una tasa de lesiones relativamente baja, oscilando entre 1 y 4.4 lesiones por cada 1000 horas de entrenamiento. No obstante, la definición de lesión no aborda la principal preocupación de los levantadores: los trastornos que no requieren una interrupción total del entrenamiento ni están asociados con lesiones graves, conocidos como síndromes de sobre uso. Por ende, se ha reformulado el concepto de lesión como cualquier condición dolorosa que obligue al levantador a modificar parte de su entrenamiento, sin necesidad de interrumpirlo por completo.

Considerando esta conceptualización de lesión, Strömbäck **et al.** [55] llevaron a cabo una evaluación de los 100 mejores levantadores de pesas de la Federación Sueca en 2014, teniendo como resultado que el (73/104) de los participantes presentaban lesiones en el momento del estudio, y un (83/95) había experimentado alguna lesión en los últimos 12 meses. Las áreas más comúnmente afectadas eran la región lumbopélvica, el hombro y la cadera, tanto en hombres como en mujeres. Aunque las lesiones parecían ocurrir durante el entrenamiento, solo el (11/70) de aquellos lesionados tuvieron que suspender completamente su rutina de entrenamiento. Factores como la frecuencia de entrenamiento, el máximo logrado en el levantamiento de peso muerto, el uso de straps, el consumo de alcohol y problemas dietéticos se vincularon con las lesiones actuales.

Además, se pudo observar que las lesiones asociadas al ejercicio de levantamiento de peso muerto son principalmente de la parte baja de la espalda o las extremidades inferiores. Estas incluyen fracturas por estrés acetabular, rupturas del músculo isquiotibial, avulsiones del espinazo ilíaco anterior superior, lesiones del menisco de la rodilla y lesiones lumbosacras [5].

2.2.3. Errores comunes dentro del peso muerto.

A continuación, se detallan los errores comunes al practicar el peso muerto. En la figura 2.2-2 se muestran estos errores.

- **Caderas colocadas demasiado baja o alta:** Estas practicas pueden resultar en una distribución desequilibrada del peso corporal. Si las caderas están demasiado bajas, esto puede provocar un desequilibrio en la carga, mientras que si están demasiado altas, el movimiento puede convertirse en una simple bisagra en lugar de implicar los músculos necesarios para el levantamiento adecuado [6].
- **Redondeo excesivo de la espalda:** Esta practica puede llevar a lesiones en levantadores de peso. Estos errores, evidenciados durante el peso muerto, indican una postura inadecuada y pueden ser resultado de desequilibrios musculares o restricciones en la flexibilidad [23].
- **Excesiva extensión al final del levantamiento:** Puede causar hiperextensión lumbar, comprometiendo la estabilidad y aumentando el riesgo de lesiones en la columna vertebral du-

rante el levantamiento de peso [54].



Ilustración 2.2-2: Errores comunes dentro del peso muerto.

Realizado por: Jiménez J. 2024.

Además, es crucial destacar que una postura incorrecta al realizar el levantamiento puede aumentar significativamente el riesgo de lesiones [42]. La flexión excesiva en la espalda baja o torácica, causada por un control deficiente del tronco, falta de flexibilidad en los isquiotibiales y malos hábitos de levantamiento, puede provocar estrés adicional en los discos espinales y nervios, resultando en hernias discales, lumbago, ciática y otros problemas. Es esencial mantener una postura adecuada para prevenir lesiones y mejorar el rendimiento deportivo a largo plazo [44].

2.3. Detección de errores en el peso muerto

En el campo deportivo del peso muerto, existen múltiples herramientas tecnológicas que permiten la identificación de errores en esta disciplina. Entre las más importantes encontramos:

- **Sensores Inerciales:** Son dispositivos electrónicos que permiten medir la orientación y posición de un objeto utilizando los datos obtenidos de un acelerómetro, un giroscopio y un magnetómetro, dentro del peso muerto se usa para registrar el rango de movimiento del ejercicio y detectar anomalías durante el levantamiento [15].
- **Plataformas de Fuerza:** Son dispositivos diseñados para medir las fuerzas que se ejercen sobre su superficie durante la realización de acciones dinámicas, dentro del peso muerto suele

usarse para detectar el desequilibrio en la distribución de la fuerza o la incorrecta alineación del cuerpo durante el levantamiento [34].

- **Electromiografía:** Es una técnica utilizada para registrar la actividad eléctrica de los músculos durante la contracción, mediante el uso de sensores, se capturan las señales eléctricas generadas por las células musculares cuando estas se activan en respuesta a impulsos nerviosos, dentro del peso muerto se usa para monitorear la activación muscular durante el levantamiento, permitiendo identificar desequilibrios o deficiencias en la técnica [39].

Estas tecnologías, aunque no están basadas en Deep Learning o visión artificial, proporcionan métodos viables para la detección de errores y la mejora del rendimiento en el peso muerto. Sin embargo, su implementación implica una inversión considerable en equipos, cuyos precios oscilan entre 250 USD para las plataformas de fuerza y 180 USD en adelante para los sensores inerciales.

2.3.1. Visión Artificial

La visión artificial, también conocida como visión por computadora, es una disciplina de la informática que permite a los sistemas interpretar y comprender imágenes del mundo real, mediante el uso de algoritmos y técnicas de procesamiento, para realizar tareas como el reconocimiento, inspección y análisis de imágenes [8].

Según el autor Lozano [27] el objetivo principal de la visión artificial es dotar a las máquinas de un sistema de reconocimiento visual que les permita comprender los eventos del mundo real, facultándolas para tomar decisiones y automatizar diversos procesos.

Entre las aplicaciones distintivas de la visión artificial, según el autor Malpartida [31], se encuentran: la medición o calibración, que garantiza que las mediciones cumplan con las especificaciones establecidas; la detección de fallas, que identifica defectos o artefactos no deseados; el reconocimiento, que identifica objetos mediante descriptores o símbolos; y el análisis de localización, que evalúa la posición de un objeto.

La Figura 2.3-3 ilustra el proceso de funcionamiento de la visión artificial, delineando el orden de cada fase, que abarca desde la adquisición de la imagen y su procesamiento hasta la ejecución de una acción específica basada en los datos obtenidos.

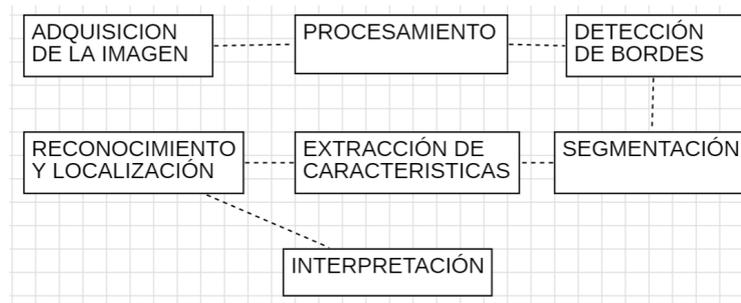


Ilustración 2.3-3: Fases del proceso de funcionamiento de la visión artificial al determinar una imagen.
Realizado por: Puertas et al. [47].

Desde el siglo pasado la visión artificial se ha utilizado en diversas áreas y disciplinas. Pero desde hace una década, con el abaratamiento del hardware de alto rendimiento ha impulsado su adopción en nuevos campos, como el deporte, donde se utiliza para automatizar ciertas tareas.

2.3.2. *Procesamiento de imágenes*

El procesamiento de imágenes se refiere a la manipulación de datos visuales que son bidimensionales o tridimensionales, como fotografías o vídeos. Sus técnicas se usan en varios campos, principalmente para mejorar la calidad de las imágenes como se puede ver en la Figura 2.3-4, reconocer patrones en ellas y codificarlas para su transmisión o almacenamiento [19].

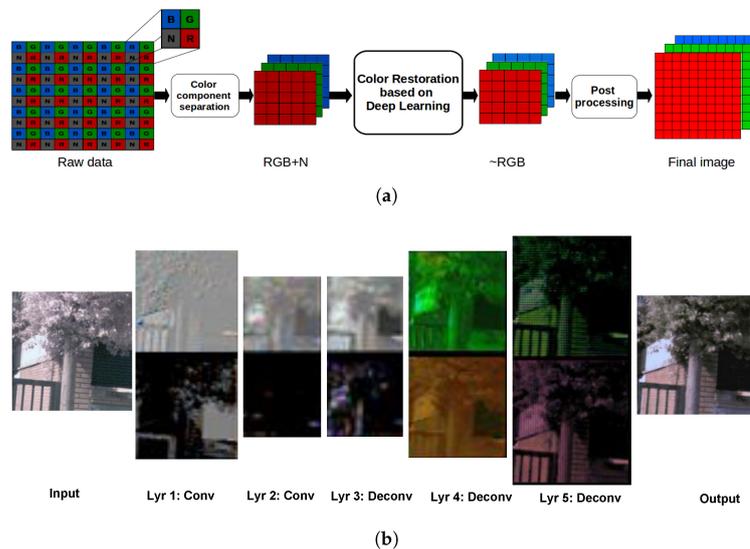


Ilustración 2.3-4: Proceso de restauración de color en imágenes mediante Deep Learning.
Realizado por: Soria et al. [53].

2.3.3. *Generación de dataset*

Para proyectos de visión artificial con IA la generación de datasets es un proceso importante para el desarrollo y entrenamiento de modelos de aprendizaje automático. Las siguientes son las formas

más comunes de obtener datasets:

Bases de Datos Públicas: Existen numerosas bases de datos públicas ampliamente utilizadas en la investigación y desarrollo de visión artificial [29]. En cuanto a las bases de datos públicas relacionadas con el levantamiento de pesas más populares encontramos:

- **Roboflow:** Contiene una variedad de datasets enfocados en visión artificial, incluyendo conjuntos de datos para reconocimiento de objetos, detección de rostros, segmentación de imágenes y clasificación de imágenes, contando con alrededor de 500,000 datasets, entre los cuales se pueden encontrar relacionados con el levantamiento de pesas.
- **Kaggle:** Es una plataforma muy popular para encontrar datasets de todo tipo, incluyendo aquellos relacionados con ejercicios de levantamiento de pesas.

Recolección Propia: En ocasiones, es necesario recolectar imágenes específicas para una aplicación particular. Esto puede implicar:

- **Captura Manual:** Utilización de cámaras para tomar fotos o vídeos.
- **Web Scraping:** Extracción de imágenes de internet mediante técnicas manuales o automatizadas.

El proceso de generación de un dataset generalmente comprende los siguientes pasos, como se ilustra en la Figura 2.3-5:

- **Recolección de Imágenes:** Captura de imágenes desde diversas fuentes.
- **Preprocesamiento:** Ajuste y limpieza de las imágenes.
- **Anotación:** Etiquetado de imágenes con información relevante, como la identificación de objetos.
- **División en Conjuntos:** Separación del dataset en conjuntos de entrenamiento, validación y prueba.
- **Entrenamiento y Validación:** Utilización de los conjuntos generados para entrenar y validar modelos de visión artificial.

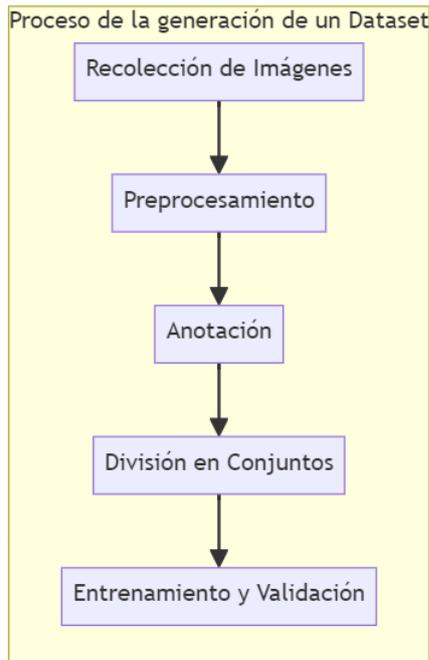


Ilustración 2.3-5: Proceso de generación de un dataset.

Realizado por: Jiménez J. 2024.

2.3.4. Clasificaciones de imágenes con Deep Learning

La clasificación de imágenes es un proceso dentro del campo de la visión por computadora que consiste en asignar una etiqueta o categoría a una imagen basándose en su contenido visual, implicando el análisis y procesamiento de las características visuales presentes en la imagen para identificar patrones u objetos específicos que correspondan a una categoría en particular [28].

Según Bengio **et al.** [4], el deep learning es un campo dentro de la inteligencia artificial que permite a los sistemas informáticos mejorar su desempeño mediante la experiencia y los datos. Este campo se basa en la utilización de redes neuronales artificiales de múltiples capas, que permiten aprender representaciones complejas construidas a partir de la combinación de conceptos más simples.

La clasificación de imágenes con deep learning se refiere al uso de redes neuronales profundas para categorizar imágenes en distintas clases basadas en su contenido. Este proceso implica la utilización de arquitecturas de deep learning, donde se destacan dos algoritmos:

Redes Neuronales Convolucionales (CNN): Las CNN son un tipo de red neuronal artificial que utilizan la operación matemática de convolución en al menos una de sus capas [3]. Su uso principal es la de reconocimiento de patrones visuales, empleándose en tareas de visión por computador y procesamiento de imágenes, ya que permiten la extracción de características de las imágenes y

su posterior clasificación [26].

MobileNet es una arquitectura de CNN diseñada para dispositivos con recursos limitados, ya que utiliza una técnica que reduce significativamente el número de parámetros y la carga computacional en comparación con modelos más complejos. Esto la hace ideal para aplicaciones en la web al integrar modelos de visión artificial directamente en el navegador sin depender de un hardware potente [38].

Redes Neuronales Recurrentes (RNN): Este tipo de red neuronal artificial se destaca por procesar secuencias de datos manteniendo la información a lo largo del tiempo, crucial para aplicaciones como predicción de texto y traducción de lenguajes, gracias a su capacidad de recordar información pasada debido a su estructura de bucles [2]. Así mismo, en otros trabajos se menciona que en la clasificación de imágenes, las RNN se suelen usar combinadas con CNN tanto para análisis de secuencias de datos derivados de imágenes y clasificación de imágenes [10].

Las RNN pueden adaptarse para la clasificación de imágenes, aunque generalmente se prefieren en aplicaciones donde el contexto es secuencial, como el análisis de texto y la predicción de series temporales [33]. Por otro lado, las Redes Neuronales Convolucionales son la elección predilecta para tareas de visión por computadora debido a su arquitectura especializada en la detección de características ya que permite el aprendizaje de patrones complejos en imágenes, lo que las hace ideal en la clasificación de imágenes, la detección de objetos y la segmentación de imágenes [60].

2.3.5. *Hyerparametros para la detección de errores de peso muerto*

- **Optimizadores:** Un optimizador es un método utilizado en el entrenamiento de redes neuronales para ajustar los parámetros del modelo con el objetivo de minimizar una función de pérdida [24].
- **Función de pérdida:** Es una forma de cuantificar qué tan buena es la imagen procesada comparada con la original, considerando no solo diferencias de píxeles, sino también aspectos de percepción visual humana [21].
- **Tasa de aprendizaje:** Es un parámetro clave en algoritmos que determina cuán rápido se ajustan las estimaciones de la función basadas en los errores calculados, influenciando la velocidad y estabilidad de la convergencia del modelo durante el entrenamiento [13].

2.4. Desarrollo de software

El desarrollo de software es el proceso de diseñar, crear y mantener aplicaciones y sistemas informáticos, este proceso incluye la definición de requisitos, la planificación, el diseño, la codificación, las pruebas y la implementación del software, para ello se suele utilizar diferentes

metodologías como puede ser el desarrollo en cascada, el desarrollo en espiral, etc. Adicionalmente, el objetivo principal es la de satisfacer las necesidades del usuario, cumplir con los plazos y presupuestos establecidos, y asegurar la adaptabilidad del software ante cambios [9].

2.4.1. *Ciclo de vida del Software*

El ciclo de vida del software, es un marco que abarca los procesos, actividades y tareas involucradas en el desarrollo y mantenimiento de un producto de software, desde la definición de los requisitos hasta su finalización, incluyendo etapas como la expresión de necesidades, análisis, diseño, implementación, pruebas y mantenimiento, cada una de estas etapas tiene características y alcances específicos, y la correcta ejecución de todas ellas es esencial para asegurar la calidad del software a lo largo de su vida útil [11].

2.4.2. *Metodología de desarrollo*

La metodología de desarrollo de software es un enfoque sistemático para planificar, gestionar y llevar a cabo un proyecto de software con el objetivo de maximizar sus posibilidades de éxito. Se basa en un conjunto integrado de técnicas y métodos que cubren todas las actividades del ciclo de vida del software, desde la definición de necesidades hasta el mantenimiento [30].

Entre las dos principales clases de metodologías de desarrollo se destacan:

Tabla 2-1 Comparación entre metodologías ágiles y metodologías tradicionales.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos
Muchos ciclos de entrega	Pocos ciclos de entrega

Realizado por: Maida & Pacienza [30].

El uso de una metodología tradicional es ideal para proyectos donde los requisitos están bien definidos y es poco probable que cambien a lo largo del ciclo de vida del proyecto. Al tener un enfoque lineal, estructurado en fases secuenciales como análisis, diseño, desarrollo, pruebas y mantenimiento, al garantizar un control y una planificación de cada fase.

2.4.3. *Herramientas para el desarrollo*

Visual Studio Code: Es un editor de código fuente, combina simplicidad y herramientas avanzadas, estando disponible para macOS, Linux y Windows, ofrece un ciclo de edición-compilación-depuración eficiente. Así mismo, cuenta con soporte para numerosos lenguajes, presenta un editor ultra rápido con resaltado de sintaxis y atajos intuitivos. Además, facilita flujos de trabajo diarios al integrarse con herramientas de compilación y Git, permitiendo trabajar con control de origen directamente desde el editor [36].

Teachable Machine: Es una plataforma en línea desarrollada por la compañía Google que facilita la creación de modelos a reconocer imágenes, sonidos y gestos, siendo una herramienta accesible y fácil de utilizar, lo que posibilita a los usuarios la creación, prueba y exportación de sus modelos personalizados para emplearlos en diversos proyectos, ya sean web, de aplicaciones o de hardware [58].

Django: Es un framework de desarrollo web de código abierto escrito en Python, su principal objetivo es permitir un desarrollo rápido y limpio. Django incluye características integradas como un sistema de autenticación, y herramientas de administración, lo que reduce significativamente el tiempo de desarrollo. Además, está respaldado por la Django Software Foundation (DSF), una organización sin fines de lucro que apoya su desarrollo continuo [12].

CAPÍTULO III

3. MARCO METODOLÓGICO

En este capítulo, se describen los procesos y las metodologías implementadas para el desarrollo tanto del modelo como de la aplicación web destinados a la detección de errores durante la ejecución del peso muerto. Se explican las fases de desarrollo de la aplicación web y de las herramientas utilizadas para su creación y ejecución.

3.1. Alcance

El trabajo de titulación se limitó al desarrollo de un prototipo de una aplicación web utilizando el framework Django solamente para la detección en tiempo real de errores comunes en el ejercicio de peso muerto convencional con barra olímpica en el contexto del powerlifting. Estos errores solamente incluyen la colocación incorrecta de la cadera (demasiado alta o baja), el redondeo excesivo de la espalda y la extensión excesiva al final del levantamiento. Para la generación del modelo de clasificación de imágenes, se utilizó la herramienta Teachable Machine para entrenar el modelo en la arquitectura de CNN MobileNet [38]. El modelo de deep learning entrenado fue exportado en formato TensorFlow.js, permitiendo su integración en la aplicación web. En cuanto a los resultados obtenidos por el software, este produce un registro de los errores detectados durante la sesión de entrenamiento tanto en formato .csv como .pdf.

3.2. Tipo de investigación

Investigación Documental: Para el desarrollo del proyecto, la investigación documental constituyó la base fundamental. A través de buscadores académicos y repositorios como PubMed, Google Scholar e IEEE Xplore, se llevó a cabo un estudio de diferentes algoritmos, librerías y herramientas relevantes para el desarrollo del prototipo de aplicación web. Las fuentes consultadas en la revisión de la literatura incluyeron tesis, artículos científicos y libros. De esta manera, la aplicación web se fue consolidando hasta llegar a la versión actual, ver Sec. 3.1, a base de literatura existente en detección de errores de peso muerto, modelos de detección basados en DL y metodologías de desarrollo de software.

3.3. Estructura de la solución

En cuanto a la estructura de solución, se dividió en las siguientes fases:

- **Fase 1 (Revisión de pre-requisitos):** En esta etapa se revisó fuentes que incluyen páginas web especializadas, artículos científicos, revistas y otras publicaciones con el fin de identificar tecnologías para detectar el movimiento en el peso muerto, así como los errores comunes asociados con este deporte. Además, se realizaron consultas con entrenadores para identificar cuáles son los errores más comunes dentro de esta disciplina.
- **Fase 2 (Diseño de Software):** Se llevó a cabo el diseño del software, definiendo la arquitectura y la interfaz de usuario, estableciendo los requisitos funcionales.
- **Fase 3 (Desarrollo del Software):** Se desarrolló la aplicación web en el framework Django, en el que el sistema empieza con el usuario dando clic a un botón para empezar el cual permite que la cámara capture la imagen de la repetición del ejercicio, el cual será procesado por el sistema utilizando un modelo previamente entrenado en formato TensorFlow.js para detectar la técnica, finalmente, una vez que la sesión de entrenamiento termine, el software permite guardar un registro de los errores ocurridos en formato .csv y .pdf.
- **Fase 4 (Evaluación del Desempeño):** Se evaluó el desempeño del software en la detección de la técnica de peso muerto en el powerlifting, con la ayuda de dos entrenadores, los cuales documentaron los errores cometidos por el deportista en una ficha, para luego ser comparado con los resultados obtenidos por el software. Para una representación visual de este proceso, se puede observar en la figura 3.3-1.

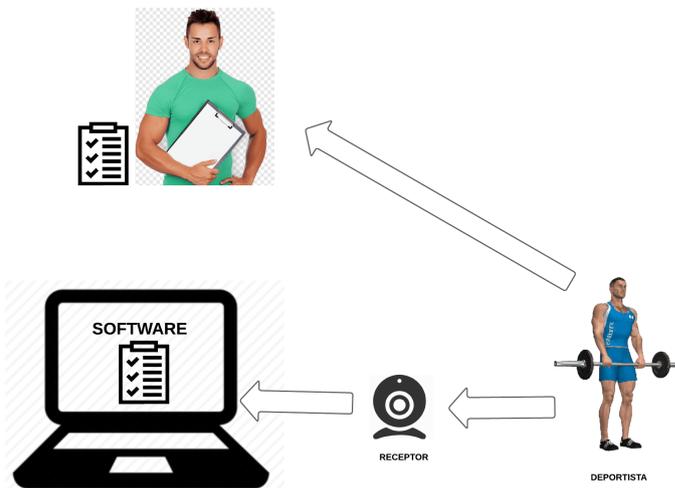


Ilustración 3.3-1: Diseño de la evaluación del software.

Realizado por: Jiménez J. 2024.

3.4. Técnicas e instrumentos de investigación

En la tabla 3-1 se redactan los métodos y técnicas para lograr cumplir los objetivos específicos propuestos en el proyecto técnico:

Tabla 3-1 Técnicas y herramientas

OBJETIVOS	MÉTODOS	DESCRIPCIÓN	TÉCNICAS	FUENTES
Determinar los algoritmos que permitirían identificar los errores en el peso muerto.	Revisión sistemática de la literatura.	Un estudio de diversas fuentes bibliográficas respecto a la visión artificial.	Revisión de artículos científicos, tesis e investigaciones.	Bases de datos del Google académico, repositorios de tesis.
Desarrollar el prototipo de la aplicación web para el reconocimiento de errores en el peso muerto.	Metodología Cascada.	Permite el desarrollo del aplicativo, generando avances en cada etapa para las revisiones pertinentes al proyecto.	Análisis documental.	Artículos científicos y tesis.
Evaluar la funcionalidad del aplicativo web.	Evaluación experta.	Permite corroborar el funcionamiento correcto del software con un experto.	Ficha técnica de evaluación y entrevistas	Opiniones de expertos.

Realizado por: Jiménez J. 2024.

3.5. Desarrollo del sistema

Para el desarrollo de la aplicación web, para el reconocimiento de errores en el peso muerto, se utilizó la metodología en cascada. Esta metodología organiza por etapas el proceso de desarrollo de software, de modo que cada etapa debe completarse secuencialmente como se puede ver en la figura 3.5-2, esta metodología es especialmente beneficiosa cuando no hay fechas inmediatas de implementación, lo que permite un desarrollo adecuado de cada fase [46].

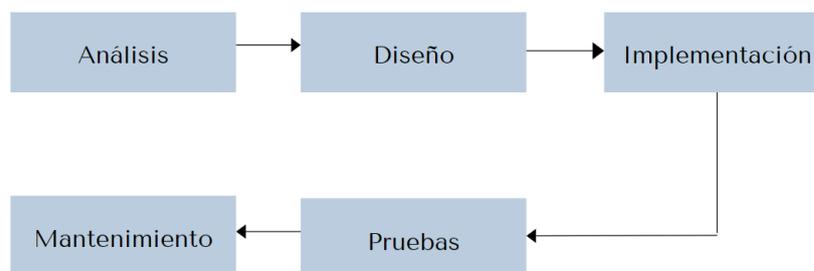


Ilustración 3.5-2: Fases modelo cascada.

Realizado por: Jiménez J. 2024

3.5.1. Análisis

- **Requisitos Funcionales**

Tabla 3-2 Requisitos funcionales.

Actor	Requisito	Descripción
Usuario	Iniciar sesión de entrenamiento	El usuario podrá abrir la aplicación web, la cual mostrará una interfaz que le permitirá capturar los errores en el peso muerto a través de un botón.
Sistema	Predicción de errores	La aplicación preprocesará la imagen para aplicar el modelo CNN entrenado previamente para detectar los errores en el peso muerto.
Usuario	Gestión de resultados	La aplicación web permitirá visualizar los resultados obtenidos, el cual incluirá la cantidad de veces que se ha cometido los errores y la opción de guardar los resultados obtenidos.

Realizado por: Jiménez J. 2024.

- **Requisitos No Funcionales.**

Tabla 3-3 Requisitos no funcionales.

Requisito	Descripción
Usabilidad	El sistema tendrá una interfaz de usuario fácil de manejar, donde el usuario podrá acceder a todo el contenido de manera rápida.
Portabilidad	El sistema será compatible con cualquier navegador que el usuario disponga en donde se ejecutará de manera correcta con todas sus funciones y manteniendo un diseño responsivo.
Funcionalidad	El sistema identificará en tiempo real los errores detectados en la técnica del peso muerto.

Realizado por: Jiménez J. 2024.

• Diagrama de Caso de Uso:

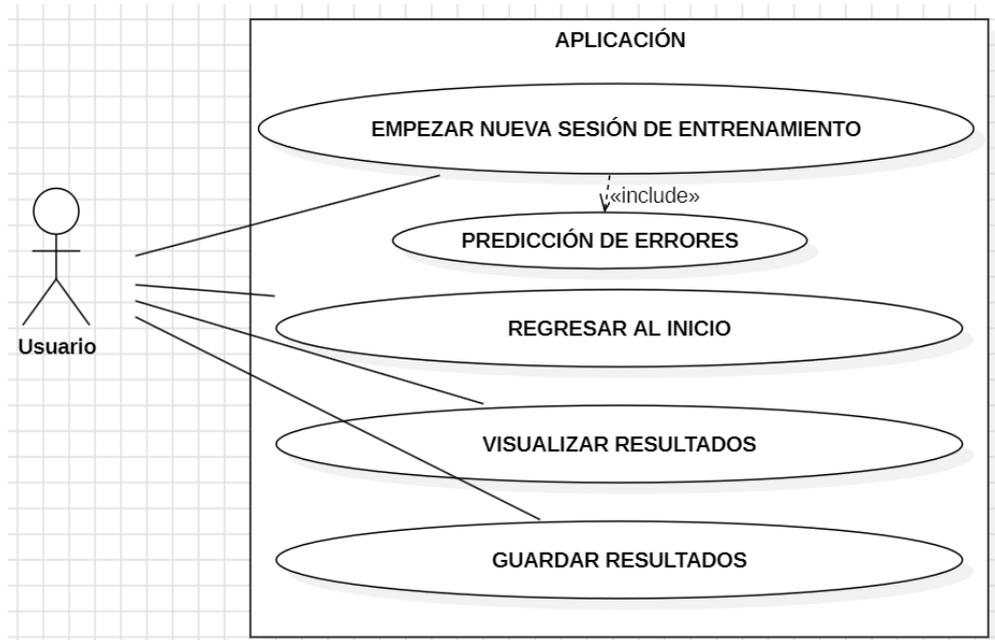


Ilustración 3.5-3: Diagrama de caso de uso: uso de la aplicación.

Realizado por: Jiménez J. 2024

Tabla 3-4 Descripción del caso de uso.

Nombre	Uso de la Aplicación
Actores	Usuario
Propósito	Funcionalidades de la aplicación.
Pre-condiciones	Acceder al Sistema
Flujo de Eventos	<ol style="list-style-type: none"> 1. El usuario presiona el botón de “Empezar” 2. El usuario realiza el entrenamiento correspondiente 3. En caso de querer volver al Inicio, lo puede hacer a través del botón ”Salir” 4. Podrá acceder a los Resultados de su entrenamiento a través del botón ”Resultados” 5. El usuario puede guardar los resultados obtenidos a través del botón ”Guardar”
Post-condiciones	El usuario obtiene un archivo con los resultados del entrenamiento.

Realizado por: Jiménez J. 2024.

3.5.2. Diseño

Diagrama de secuencia

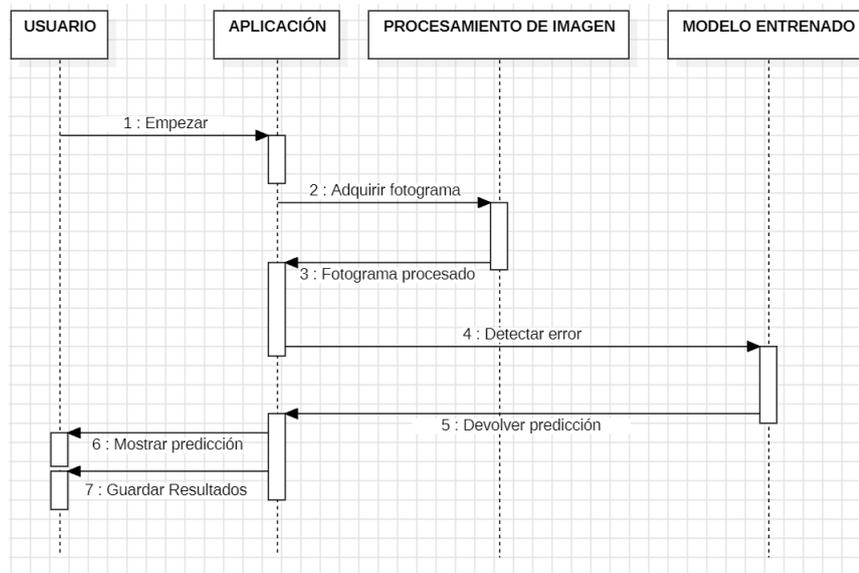


Ilustración 3.5-4: Diagrama de secuencia de la aplicación.
Realizado por: Jiménez J. 2024

Flujo de trabajo de la aplicación

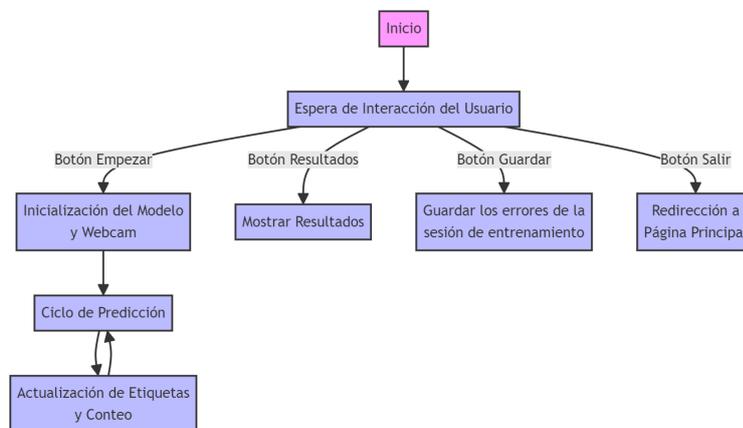


Ilustración 3.5-5: Flujo de trabajo de la aplicación.
Realizado por: Jiménez J. 2024

Tabla 3-5 Flujo de trabajo de la aplicación.

Proceso	Descripción
Inicio	El usuario abre la aplicación web.
Espera de Interacción del Usuario	La aplicación espera a que el usuario interactúe con los botones disponibles.
Inicialización del Modelo y Webcam	Al hacer clic en "Empezar", se carga el modelo de TensorFlow.js, se configura y activa la webcam, y se inicializan los contenedores para las etiquetas y los resultados.
Ciclo de Predicción	La webcam captura imágenes continuamente, y el modelo de predicción analiza estas imágenes para detectar errores.
Actualización de Etiquetas y Conteo	Las etiquetas se actualizan con las probabilidades de las predicciones y se cuenta la frecuencia de los errores detectados.
Mostrar Resultados	Al hacer clic en "Resultados", se detiene la webcam y se muestran los resultados resumidos de los errores detectados y su frecuencia.
Guardar los Resultados	Al hacer clic en el botón "Guardar", se descarga los resultados obtenidos de los errores durante el entrenamiento.
Redirección a Página Principal	Al hacer clic en "Salir", el usuario es redirigido a la página principal de la aplicación.

Realizado por: Jiménez J. 2024.

Diseño de Interfaz de la aplicación

El diseño de los prototipos de las interfaces se realizó con la utilización del software Figma, basándose en un dashboard general. En el cual se proporciona una interfaz fácil de usar para la detección de errores en el ejercicio de peso muerto. Las figuras 3.5-6, 3.5-7 y 3.5-8 muestran los distintos estados de la aplicación durante su uso.

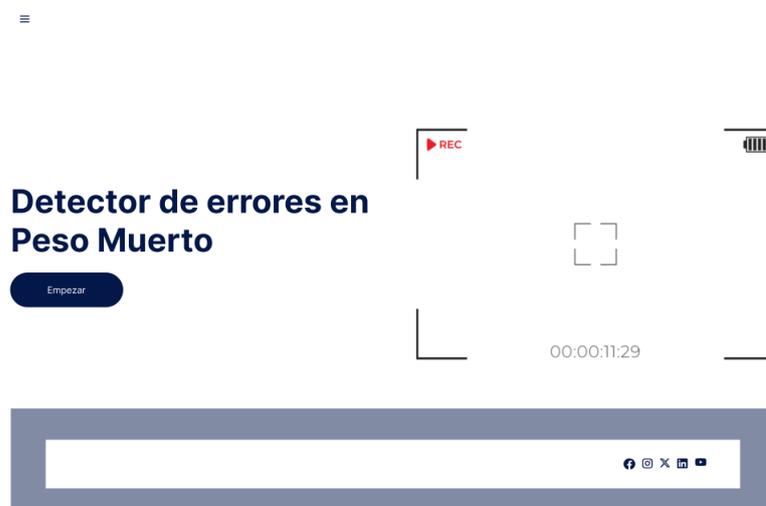


Ilustración 3.5-6: Diseño de interfaz de la aplicación 1.

Realizado por: Jiménez J. 2024

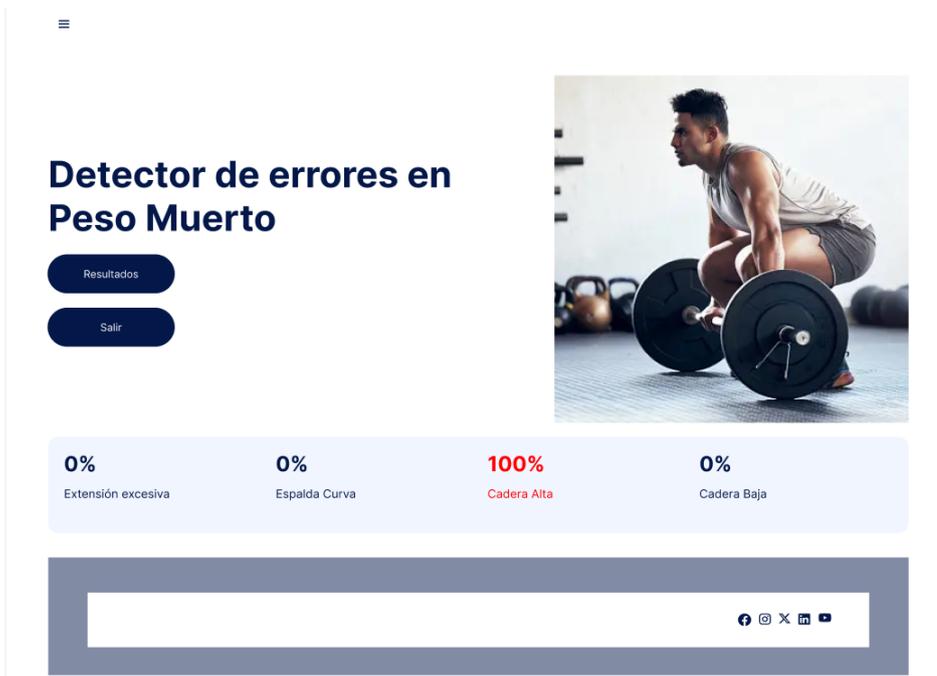


Ilustración 3.5-7: Diseño de interfaz de la aplicación 2.
 Realizado por: Jiménez J. 2024



Ilustración 3.5-8: Diseño de interfaz de la aplicación 3.
 Realizado por: Jiménez J. 2024

Algoritmo de detección de errores del peso muerto

Para la detección de errores en el levantamiento de peso muerto, existen diversos algoritmos de visión artificial, como se describe en el marco teórico (Ver sección 2.3.4). Sin embargo, considerando la naturaleza del proyecto y su objetivo, que es implementar una solución para un

entorno de computación limitada. Bajo esta premisa, se han revisado diferentes algoritmos inteligentes como por ejemplo las redes neuronales convolucionales.

Tras revisar la literatura, la arquitectura MobileNet basado en CNN, ha destacado como uno de los modelos de deep learning más eficientes y adaptables en diferentes proyectos donde se exige un equilibrio entre precisión y eficiencia [18]. MobileNet fue desarrollado por investigadores de Google, y está diseñado bajo un enfoque modular y eficiente [18]. Esta arquitectura consta de 29 capas (Ver figura 3.5-9), las cuales se distribuyen de la siguiente manera:

- 14 Capas de Convoluciones estándar (Conv).
- 13 Capas Convoluciones separables por profundidad (Conv dw).
- 1 Capa Average pooling (Avg Pool).
- 1 Capa Multi-layer perceptron (Fully Connected layer).

Adicionalmente, MobileNet incluye una capa de operación Softmax, que se encarga de convertir los valores de salida en probabilidades normalizadas, permitiendo clasificar las imágenes.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Ilustración 3.5-9: Arquitectura de MobileNet.

Realizado por: Howard et al. [18]

3.5.3. Implementación

En esta sección se describe el proceso de creación e implementación del modelo para la detección de errores durante la ejecución del peso muerto, tanto de la recopilación y clasificación de mues-

tras, así como en el entrenamiento del modelo utilizando Teachable Machine. Además, de los materiales utilizados, la configuración de la cámara para la captura de datos, y los parámetros aplicados durante el entrenamiento. Asimismo, se presenta parte de la codificación realizada para la creación del sistema web, tanto en el Front-End como en la lógica para la actualización de etiquetas necesaria para mostrar la predicción en tiempo real, incluyendo un enlace al **repositorio público de GitHub**.

Generación del modelo

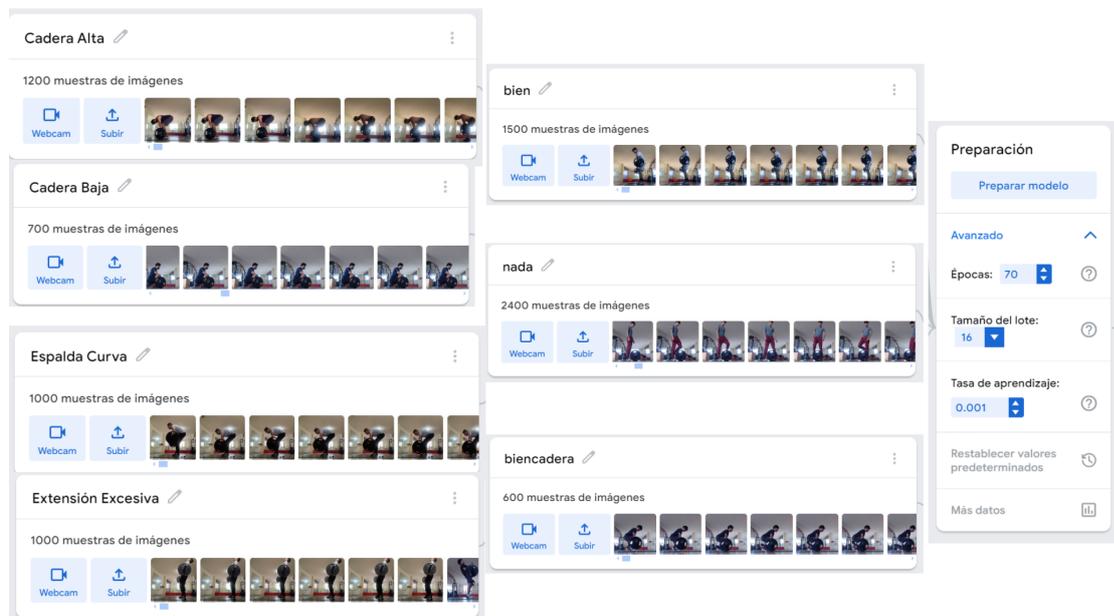


Ilustración 3.5-10: Clasificación por clases de imágenes para el entrenamiento del modelo.
Realizado por: Jiménez J. 2024

Materiales necesarios

- Teléfono móvil
- Deportista
- Barra Olímpica y discos.
- Teachable Machine

Procedimiento

a) Posicionamiento de la cámara

- La cámara debe de apuntar al lado izquierdo y estar alineado con el deportista.

- Asegurarse de que la distancia entre el deportista y la cámara sea de alrededor de 2 metros para una correcta captura de imagen.

b) **Captura de las muestras**

- Se graba un vídeo de los siguientes errores, como se puede ver en la figura 2.2-2 y diferentes posiciones del deportista:
 - Cadera alta: El deportista tiene que mantener la cadera a más de 100° con respecto a la rodilla y pies.
 - Cadera baja: El deportista tiene que mantener la cadera a menos de 80° con respecto a la rodilla y pies.
 - Redondeo de la espalda: El deportista tiene que tener su espalda redondeada durante el levantamiento.
 - Extensión excesiva al final del levantamiento: El deportista tiene que hiper extender la espalda al final del levantamiento.
 - Bien: El deportista tiene que ejecutar de manera correcta el ejercicio.
 - BienCadera: El deportista tiene que mantener la cadera a 90° con respecto a la rodilla.
 - Nada: Se captura diferentes poses e imágenes que corresponderían cuando el deportista no ejecuta el ejercicio.
- Los vídeos capturados se transforman a un conjunto de imágenes.

c) **Entrenamiento del modelo**

- Subir las imágenes a Teachable Machine en las clases como se lo ve la figura 3.5-10 y con el siguiente número de muestras.
 - Nada: 2400 muestras.
 - Bien: 1500 muestras.
 - BienCadera: 600 muestras.
 - Cadera Alta: 1200 muestras.
 - Cadera Baja: 700 muestras.
 - Espalda Curva: 1000 muestras.
 - Extensión Excesiva: 1000 muestras.
- Configuración: Épocas: 70, Tamaño de Lote: 16, Tasa de aprendizaje: 0.001

- El entrenamiento en Teachable Machine dura unos 8 minutos, tras lo cual el modelo se puede exportar en formato Tensorflow.js.

Codificación

En esta etapa se procede con la codificación de la aplicación, siguiendo los requerimientos planteados. La figura 3.5-11 muestra la actualización de etiquetas en tiempo real para la predicción de errores en el peso muerto, mientras que la figura 3.5-12 presenta la interfaz de inicio.

```
// Actualizar los labels
const validPredictions = predictions.filter(pred => pred.className !== 'nada' && pred.className !== 'bien');
validPredictions.forEach((prediction, i) => {
  const labelDiv = labelContainer.childNodes[i];
  labelDiv.querySelector('.label-percentage').textContent = `${(prediction.probability * 100).toFixed(0)}%`;
  labelDiv.querySelector('.label-name').textContent = prediction.className;

  // Aplicar estilos basado en la probabilidad más alta
  if (prediction.className === predictions[maxIndex].className) {
    labelDiv.querySelector('.label-percentage').style.color = 'red';
    labelDiv.querySelector('.label-name').style.color = 'red';
  } else {
    labelDiv.querySelector('.label-percentage').style.color = '#031749';
    labelDiv.querySelector('.label-name').style.color = '#031749';
  }
});
```

Ilustración 3.5-11: Codificación de la actualización de las etiquetas.

Realizado por: Jiménez J. 2024

```
<main>
<br>
<div class="horizontal-container" id="hz">
  <div class="titulos_btn">
    <h1>Detector de errores en Peso Muerto</h1>
    <div class="botones_rd" id="btns">
      <button class="btn" id="startButton" type="button" onclick="init()">Empezar</button>
      <button class="btn" id="resultadosButton" type="button" style="display: none;" onclick="resultados()">Resultados</button>
      <button class="btn" id="salirButton" style="display: none;" onclick="salir()">Salir</button>
    </div>
  </div>
  <div class="video", id="container" >
    <div class="webcam" id="webcam-container">
      
    </div>
  </div>
</div>
<br>
<div id="label-container" style="display: none;" ></div>
<div id="container_results" style="display: none;">
  <h2>Resultados</h2>
  <h3>Detector de errores en peso muerto</h3>
  <br><br>
  <div id="results-content"></div>
  <button class="btn" onclick="guardar()"> Guardar </button>
  <button class="btn" id="salirButton2" onclick="Salir()" >Salir</button>
</div>
</main>
```

Ilustración 3.5-12: Codificación de la interfaz de inicio.

Realizado por: Jiménez J. 2024

El resto del código fuente se puede encontrar en el siguiente **repositorio de GitHub**.

3.5.4. Pruebas

En esta fase se procede con las pruebas del sistema, las mismas que se llevaron a cabo como parte de la propuesta del trabajo, siendo realizadas en el gimnasio StronGym en la ciudad de Macas, durante el mes de Junio hasta Septiembre del 2024. Donde se realizaron las siguientes pruebas:

- Pruebas de funcionalidad del FrontEnd: Verificar que la aplicación se inicie correctamente y que todos los elementos de la interfaz se carguen sin errores.
- Pruebas de Navegación: Verificar que los botones redirijan a las paginas correspondientes.
- Pruebas de Funcionalidad del Modelo: Verificar que el modelo de TensorFlow.js se cargue correctamente, la webcam se active sin errores y prediga los errores dentro del peso muerto.
- Pruebas de Resultados: Verificar que se muestren los resultados de los errores detectados.

3.5.5. Mantenimiento

En esta etapa, de acuerdo a los resultados obtenidos de las pruebas, se ajustó el modelo editando las muestras obtenidas para obtener mejores resultados en las predicciones. Además, se corrigieron errores en el apartado visual del sitio web para mejorar la experiencia del usuario.

CAPÍTULO IV

4. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

En este capítulo se presentan los resultados del objetivo general del proyecto técnico, enfocado en el desarrollo de un software de reconocimiento en tiempo real para la detección de errores comunes en la ejecución del peso muerto en el powerlifting, utilizando visión artificial e inteligencia artificial. Primero, se detallan los aspectos generales de la implementación de la aplicación. A continuación, se describe la aplicación web y se exponen los resultados del modelo seleccionado para la clasificación de errores comunes en el powerlifting. Posteriormente, se aborda la evaluación en tiempo real de la aplicación, finalizando con una discusión de la aplicación para la detección de errores comunes en el peso muerto.

4.1. Detalles de implementación

La implementación de la aplicación web se realizó de manera local, utilizando una laptop Asus con un procesador Ryzen 7, 16 GB de memoria RAM y un disco SSD de 500 GB. Para el desarrollo y las pruebas del sistema, se emplearon diversas herramientas de software. Visual Studio Code fue utilizado como entorno de desarrollo integrado (IDE) para el desarrollo del sistema web. El navegador Brave sirvió como plataforma de despliegue durante las pruebas. Conda se utilizó para la gestión de entornos de desarrollo, usado para la instalación y configuración de las dependencias necesarias. Además, para la creación del modelo de reconocimiento de errores en la ejecución del peso muerto, se utilizó Teachable Machine.

4.2. Aplicación web

La aplicación web ha sido denominada DLCHECKER y está diseñada para detectar errores comunes en la ejecución del peso muerto. Entre las funcionalidades destacadas se encuentran la identificación de fallos como la incorrecta colocación de la cadera (demasiado alta o baja), el redondeo excesivo de la espalda y la sobreextensión al finalizar el levantamiento. El análisis puede realizarse en tiempo real mediante la cámara web del dispositivo, o bien, el usuario tiene la opción de cargar un video para su análisis posterior.

Además, la aplicación incluye una guía de instalación detallada, que especifica todas las dependencias necesarias para su correcta ejecución. Esta guía, junto con el código fuente del proyecto, está disponible en el siguiente **repositorio público de GitHub**.

4.3. Resultados de la validación del clasificador

Para validar el modelo de clasificación de errores en el peso muerto, se utilizaron 15 vídeos que documentan sesiones de entrenamiento, de los cuales 10 corresponden a varones y 5 a mujeres. Estos vídeos, disponibles en el **dataset de validación**, fueron evaluados para determinar la efectividad del algoritmo de clasificación implementado en la aplicación web.

La evaluación fue realizada en colaboración con dos entrenadores del gimnasio StronGym, ubicado en la ciudad de Macas, el 7 de agosto de 2024. Los entrenadores emplearon los siguientes criterios para identificar los errores durante la ejecución del peso muerto:

- Cadera Alta: La cadera se eleva a más de 100 grados con respecto a la rodilla.
- Cadera Baja: La cadera desciende a menos de 80 grados con respecto a la rodilla.
- Redondeo Excesivo de la Espalda: La espalda se redondea de manera notable durante el levantamiento.
- Excesiva Extensión al Final del Levantamiento: El levantador se inclina exageradamente hacia atrás al finalizar el levantamiento.

En cuanto a la implementación en la aplicación web, para que un movimiento sea considerado como un error en el reporte final, se estableció que la probabilidad de detección debe superar el umbral del 70%. Además, al menos 34 de las últimas 35 predicciones deben coincidir para clasificar el movimiento como un error consistente, evitando así que la detección sea un caso aislado y asegurar que se trata de un patrón recurrente en la ejecución del movimiento. La tabla 4.1 muestra los resultados obtenidos tanto de los entrenadores como el de la aplicación web.

Tabla 4.1 Resultados de la predicción de errores en el peso muerto. Entrenador 1 (E1) y Entrenador 2 (E2).

SE- SI- ÓN	CADERA ALTA				CADERA BAJA				REDONDEO EXCESIVO DE LA ESPALDA				EXCESIVA EXTENSIÓN AL FINAL DEL LEVANTAMIENTO			
	E1	E2	App	Dif	E1	E2	App	Dif	E1	E2	App	Dif	E1	E2	App	Dif
1	0:00	0:00	0:00	1					0:17	0:17	0:15	1	0:31	0:31	0:29	1
	0:33	0:33	0:33	1					0:22	0:22	0:22	1				
2													0:03	0:04	0:04	1
													0:11	0:10	0:10	1

Continúa en la siguiente página

Tabla 4.1 (Continuación)

SE- SI- ÓN	CADERA ALTA				CADERA BAJA				REDONDEO EXCESIVO DE LA ESPALDA				EXCESIVA EXTENSIÓN AL FINAL DEL LEVAN-TAMIENTO			
	E1	E2	App	Dif	E1	E2	App	Dif	E1	E2	App	Dif	E1	E2	App	Dif
3	0:02 0:05	0:00 0:04	0:02 0:04	1 1	0:12	0:12	0:12	1								
4	0:08 0:19 0:24	0:08 0:20 0:24	0:12 0:23	0 1 0									0:21	0:21	0:21	1
5	0:01 0:05 0:09 0:16 0:19	0:05 0:09 0:16 0:19	0:05 0:08 0:15 0:19 0:35	1 1 1 1 0	0:02	0:02	0:00	1	0:26	0:26	0:25	1	0:17 0:32	0:17 0:32	0:17	1 0
6										0:00	0:00	1	0:11 0:15	0:11 0:15	0:15	1 0
7	0:01	0:01		0					0:07 0:10 0:17	0:07 0:10 0:17	0:00 0:04 0:07 0:10 0:17	0 0 1 1 1	0:15	0:15	0:14	1
8	0:04 0:10 0:12 0:16 0:35	0:03 0:10 0:12 0:16 0:35	0:00 0:03 0:06 0:12 0:15 0:18 0:34	0 1 0 1 1 0 1					0:06 0:39	0:06 0:39	0:06 0:39	1 1				
9	0:16 0:24 0:26	0:16 0:24 0:26	0:15 0:22 0:27	1 1 1												
10	0:03 0:20 0:26	0:03 0:20 0:26	0:02 0:08 0:17 0:26 0:30	1 0 0 1 0									0:06 0:15 0:21 0:28	0:06 0:15 0:21 0:28	0:06 0:15 0:21 0:28	1 1 1 1

Continúa en la siguiente página

Tabla 4.1 (Continuación)

SE- SI- ÓN	CADERA ALTA				CADERA BAJA				REDONDEO EXCESIVO DE LA ESPALDA				EXCESIVA EXTENSIÓN AL FINAL DEL LEVANTAMIENTO			
	E1	E2	App	Dif	E1	E2	App	Dif	E1	E2	App	Dif	E1	E2	App	Dif
11	0:00	0:00	0:00	1									0:07	0:07		0
	0:06	0:05	0:03	1												
	0:11	0:11	0:09	1												
			0:13	1												
12	0:06	0:06	0:06	1	0:02	0:02		0					0:04	0:04		0
	0:10	0:10	0:08	1												
	0:16	0:16	0:16	1												
13													0:14	0:14	0:16	1
													0:17	0:17		0
14																
15	0:08	0:08	0:09	1												
	0:12	0:12	0:18	1												
	0:15	0:15	0:20	1												
	0:17	0:18		1												

Realizado por: Jiménez J. 2024.

Para considerar una predicción como correcta, se tomó en cuenta que el tiempo de respuesta no superara los 2 segundos con respecto a los resultados obtenidos de los entrenadores; de lo contrario, se clasificó como predicción incorrecta. Como se puede observar en la figura 4.3-1, los resultados obtenidos en cuanto a los errores capturados muestran que el desempeño del sistema presenta variabilidad según el tipo de error detectado. En el caso de la Cadera Alta, el sistema logró identificar correctamente el 74.36% de las instancias, mientras que el 25.64% correspondieron a errores de identificación. Para Cadera Baja, la precisión fue del 66.67%, lo que indica que un tercio de las predicciones fueron incorrectas en esta categoría.

Por otro lado, el error de Redondeo Excesivo fue identificado correctamente en un 81.82% de las ocasiones, con un 18.18% de predicciones incorrectas. Finalmente, para el error de Extensión Final, la precisión alcanzó un 70.59% de correctas, dejando un 29.41% de errores en la detección.

En conjunto, el sistema presenta una precisión global del 73.36%, lo que demuestra un desempeño general aceptable en la identificación de errores en las diferentes pruebas que se hizo. Con los datos obtenidos se puede observar que existen mayores inconsistencias en la predicción de los errores en personas de contextura ancha.

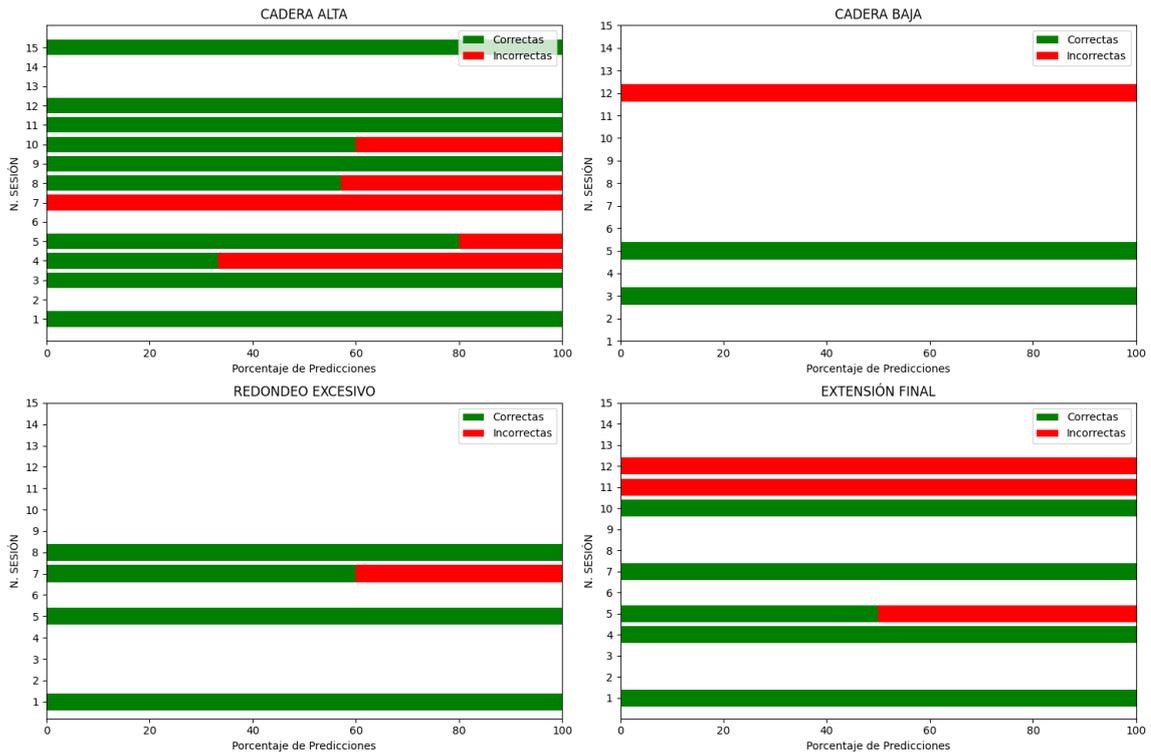


Ilustración 4.3-1: Resultados obtenidos.

Realizado por: Jiménez J. 2024

4.4. Evaluación en tiempo real de la aplicación

Para la evaluación en tiempo real de la aplicación web, se utilizaron dos dispositivos: un iPhone XS Max, accediendo a través del navegador Safari, y una laptop HP Pavilion, utilizando el navegador Brave. La aplicación fue alojada en un servicio gratuito de hosting: PythonAnywhere, permitiendo su acceso a través del siguiente enlace: <https://alex37.pythonanywhere.com/>. Durante las pruebas realizadas en ambos dispositivos, no se encontraron inconvenientes de uso. Además, se evaluó el rendimiento de la aplicación utilizando la herramienta PageSpeed Insights, con el fin de medir la experiencia del usuario en sitios web.

En el análisis de rendimiento, en entornos de escritorio (ver Figura 4.4-2), la aplicación mostró un desempeño excelente, obteniendo una puntuación de 98 sobre 100. Los tiempos de carga de contenido visual fueron notablemente rápidos, con un First Contentful Paint de 0.7 segundos y un Largest Contentful Paint de 1.1 segundos, proporcionando a los usuarios una experiencia visual casi instantánea. La interactividad del sitio fue eficiente, con un Total Blocking Time de solo 60 mili segundos. Además, la estabilidad visual durante la carga alcanzó un nivel óptimo, como lo demuestra el Cumulative Layout Shift de 0.035. El Speed Index de 0.7 segundos confirma la rapidez general del sitio en dispositivos de escritorio, ofreciendo una experiencia fluida para el usuario.

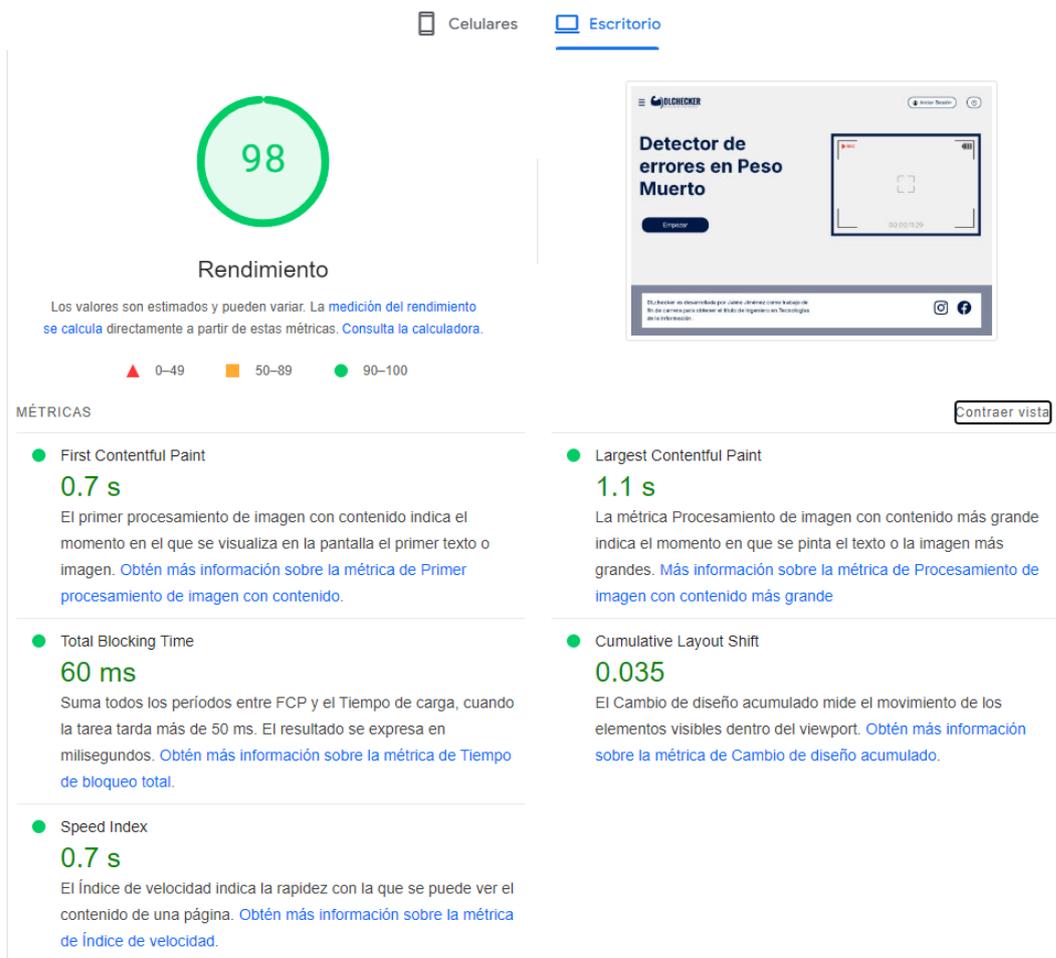


Ilustración 4.4-2: Rendimiento de la aplicación web en escritorio.
Realizado por: Jiménez J. 2024

En contraste, la versión móvil del sitio presentó un rendimiento más moderado, obteniendo una puntuación de 76 sobre 100, debido en parte a las limitaciones del hosting utilizado (ver Figura 4.4-3). Los tiempos de carga en dispositivos móviles fueron algo más lentos, con un First Contentful Paint de 2.7 segundos y un Largest Contentful Paint de 3.1 segundos. Aunque estos valores son aceptables, indican un margen considerable para mejorar la velocidad de carga en plataformas móviles. La interactividad presentó cierta latencia, con un Total Blocking Time de 430 milisegundos, mientras que la estabilidad visual, medida por el Cumulative Layout Shift, fue de 0.108, lo que muestra un rendimiento adecuado, pero con posibilidades de optimización.

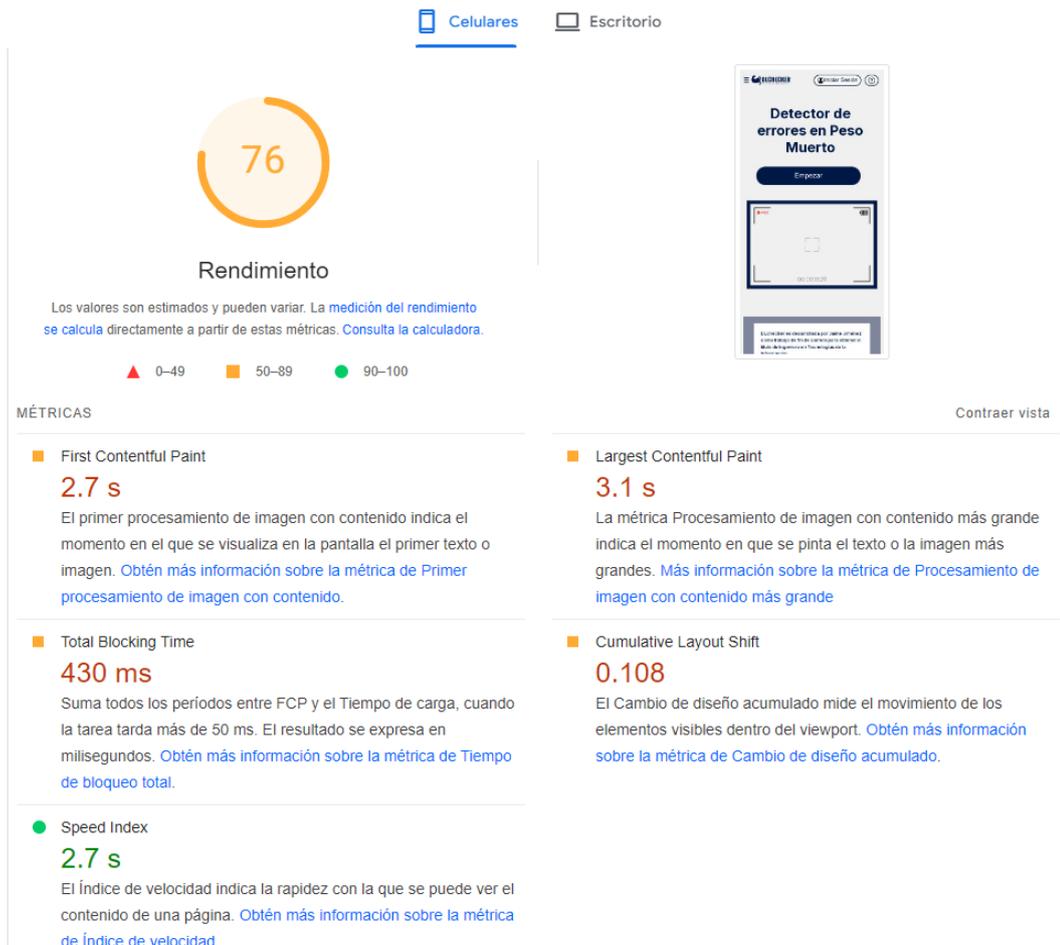


Ilustración 4.4-3: Rendimiento de la aplicación web en dispositivos móviles.
Realizado por: Jiménez J. 2024

4.5. Discusión

En esta sección se realiza un análisis comparativo entre el software desarrollado y dos aplicaciones similares que también se enfocan en la detección de errores posturales en el levantamiento de peso muerto. A través de esta comparación, se busca destacar las características diferenciadoras del presente trabajo.

Uno de los trabajos más relevantes es el presentado por Sutthiprapa **et al.** [57], quienes desarrollaron un sistema de análisis de la forma del peso muerto, que utiliza la herramienta Microsoft Kinect para la captura de movimiento, la aplicación web propuesta ofrece una alternativa más accesible y viable para una mayor cantidad de usuarios. Mientras que el sistema basado en Kinect es efectivo en la corrección postural, su uso se ve limitado por la necesidad de un dispositivo costoso para el usuario promedio. En contraste, DLCHECKER no depende de hardware específico, lo que la hace mucho más accesible al emplear cámaras convencionales y algoritmos de visión artificial. Además, al ser una aplicación web, puede ser utilizada desde cualquier dispositivo con conexión a internet, facilitando su implementación en gimnasios. A pesar de que el software propuesto no calcula fuerzas biomecánicas como en el caso de Kinect, se enfoca en corregir errores posturales

comunes con un nivel de precisión suficiente para prevenir lesiones, lo cual es su principal objetivo, haciéndolo una opción más accesible para el público general.

De manera similar, el trabajo de Yasser **et al.** [61] propone un sistema basado en una cámara de infrarrojos para la detección de errores durante el levantamiento de pesas, usando también Kinect junto con el algoritmo FastDTW para evaluar la técnica del usuario. Aunque su enfoque logra una alta precisión en la detección de movimientos incorrectos, enfrenta las mismas limitaciones que otros sistemas basados en Kinect, como la necesidad de hardware especializado y la correcta calibración de los sensores. En cambio, esta aplicación se distingue por eliminar la dependencia de equipos adicionales, lo que permite su uso en una mayor variedad de contextos, sin comprometer la precisión del análisis. Al no depender de una configuración específica de cámaras o sensores, DLCHECKER es adaptable a diferentes escenarios de entrenamiento, brindando a los usuarios una solución más práctica, especialmente en entornos donde no se dispone de herramientas tecnológicas avanzadas.

En la revisión de la literatura de este trabajo se puede encontrar proyectos similares a Sutthiprapa **et al.** [57] y Yasser **et al.** [61]. Sin embargo, la propuesta en este proyecto técnico intenta utilizar la menor cantidad de hardware especial posible, pero garantiza resultados de buena calidad. Luego de realizar toda la parte de ingeniería para la implementación de la App web se pudo observar que para ganar mayor fiabilidad en la detección se necesita invertir mayor tiempo en investigación de modelos de inteligencia artificial. Es evidente que esta aplicación web es práctica y fácil de implementar en cualquier gym, se espera que en trabajos futuros se pueda mejorar el clasificador para detectar los errores de manera óptima o con mayor precisión.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Luego de realizar la revisión de literatura en inteligencia artificial y visión artificial aplicables a la detección de errores en el peso muerto, se concluyó que las Redes Neuronales Convolucionales (CNN) son más efectivas que otros modelos de extracción de características para el análisis de patrones en imágenes. Esto se debe a la capacidad de este tipo de algoritmos de aprender patrones de comportamiento de grandes conjuntos de datos.
- El prototipo de aplicación web, DLChecker reconoce en tiempo real errores de la técnica de peso muerto dentro del powerlifting. El algoritmo de detección de errores utiliza un modelo de DL basado en CNN, que fue entrenado a través de Teachable Machine para luego exportar los pesos aprendidos en formato TensorFlow.js. Posteriormente se implementó en Django para su despliegue en la web.
- Durante la evaluación del prototipo de software para la detección de errores en el peso muerto, se obtuvo un desempeño satisfactorio, con una precisión general del 73.36% al evaluarlo en un dataset creado para este propósito. El sistema mostró una mayor exactitud en la identificación de errores como la cadera alta y el redondeo excesivo de la espalda, aunque presentó una leve disminución en la precisión al detectar errores como cadera baja y extensión excesiva.

5.2. Recomendaciones

- Se sugiere utilizar una cámara de mayor calidad para la captura de muestras, lo cual permitiría obtener imágenes más nítidas y mejorar la precisión de las predicciones realizadas por el modelo.
- Se recomienda aumentar la cantidad de muestras, incluyendo personas de diferentes edades, contexturas, géneros y estaturas, con el objetivo de ampliar la base de datos del modelo y mejorar su capacidad para reconocer variaciones en distintos deportistas.
- Se aconseja ampliar el dataset para abarcar otros deportes multiarticulares de gran popularidad, como la halterofilia, lo que permitiría extender el alcance de la app.

BIBLIOGRAFÍA

1. **AASA, Ulrika et al.** Injuries among weightlifters and powerlifters: a systematic review. *PubMed*. 2016, pp. 211–219. Available from DOI: 10.1136/bjsports-2016-096037.
2. **ARANA, Carlos.** *Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales*. Buenos Aires: Universidad del Centro de Estudios Macroeconómicos de Argentina (UCEMA), 2021. Serie Documentos de Trabajo, 797. Available also from: <https://hdl.handle.net/10419/238422>.
3. **BEINAT, Natalia Casado.** *Clasificación automática de tumores de ovario y citologías cervicovaginales a partir de imágenes ecográficas y microscópicas mediante su análisis con técnicas de aprendizaje automático*. 2022. Universidad Complutense de Madrid.
4. **BENGIO, Yoshua et al.** Deep Learning. In: London, England: Massachusetts Institute of Technology, 2016.
5. **BENGTSSON, Victor et al.** Narrative review of injuries in powerlifting with special reference to their association to the squat, bench press and deadlift. *BMJ Open Sport & Exercise Medicine*. 2018, vol. 4, no. 1. Available from DOI: 10.1136/bmjsem-2018-000382.
6. **BUGERA, Eric.** *9 Deadlift Mistakes and How to Fix Them for Stronger Pulls* [<https://barbend.com/deadlift-mistakes/>]. 2023. En línea; último acceso 26 de febrero de 2024.
7. **COLQUHOUN, Ryan J. et al.** Training Volume, not frequency, indicative of maximal strength adaptations to resistance training. *Journal of Strength and Conditioning Research*. 2018, vol. 32, no. 5, pp. 1207–1213. Available from DOI: 10.1519/jsc.0000000000002414.
8. **CULJAK, Ivan et al.** A brief introduction to OpenCV. In: *2012 Proceedings of the 35th International Convention MIPRO*. 2012, pp. 1725–1730.
9. **DELGADO OLIVERA, Lisdania de la Caridad & DIAZ ALONSO, Lexys Manuel.** Modelos de Desarrollo de Software. *Revista Cubana de Ciencias Informaticas*. 2021, vol. 15, pp. 37–51. ISSN 2227-1899. Available also from: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992021000100037&nrm=iso.
10. **DHRUV, Patel & NASKAR, Subham.** Image Classification Using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN): A Review. In: *Machine Learning and Information Processing*. Singapore: Springer Singapore, 2020, pp. 367–381. ISBN 978-981-15-1884-3.

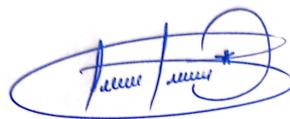
11. **DIÉGUEZ, Mauricio & CARES, Carlos.** De la Gestión de Seguridad en el Ciclo de Vida del Software. In: *International Workshop on Advanced Software Engineering*. 2012.
12. **DJANGO, Team.** *About the Django Software Foundation* [<https://n9.cl/reznv>]. 2024. En línea; último acceso 4 de Noviembre de 2024.
13. **EVEN-DAR, Eyal & MANSOUR, Yishay.** Learning Rates for Q-Learning. In: *Computational Learning Theory*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001.
14. **FAIREN, Mariano Fernández & VILLARREA, José María Busto.** Prevención de lesiones deportivas. *Medigraphic*. 2009, vol. 5, no. 1. Available also from: <https://n9.cl/ad551>.
15. **GONZALO, Alarcón Hellín.** *Aplicación de algoritmos Machine Learning en ejercicios deportivos realizados en series de repeticiones*. 2022. Universidad Politécnica de Cartagena.
16. **HIDALGO, Alex.** *Análisis comparativo del peso muerto sumo con el peso muerto convencionales* [<https://mundoentrenamiento.com/peso-muerto-sumo/>]. 2023. En línea; último acceso 1 de marzo de 2024.
17. **HOEK, Daniel J van den et al.** Powerlifting participation and engagement across all ages: A retrospective, longitudinal, population analysis with comparison to community strength norms. *International Journal of Sports Science & Coaching*. 2024, vol. 19, no. 5, pp. 2199–2209. Available from DOI: 10.1177/17479541241244481.
18. **HOWARD, Andrew G. et al.** *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017. Available from arXiv: 1704.04861 [cs.CV].
19. **HUANG, T.S. et al.** Image processing. *Proceedings of the IEEE*. 1971, vol. 59, no. 11, pp. 1586–1609. Available from DOI: 10.1109/PROC.1971.8491.
20. **IBARRA, Francisco Javier Rosas et al.** Design of a Biomechanical Model and a Set of Neural Networks for Monitoring of Weightlifting. *Res. Comput. Sci*. 2014, vol. 80, pp. 31–42. Available also from: <https://n9.cl/25gk3>.
21. **JANOCHA, Katarzyna & CZARNECKI, Wojciech Marian.** On Loss Functions for Deep Neural Networks Classification. *Cornell University*. 2017, vol. abs/1702.05659. Available from arXiv: 1702.05659.
22. **JIUNN, Ng.** *Posture Evaluation for Variants of Weight-Lifting Workouts Recognition*. 2020. University Tunku Abdul Rahman.
23. **KIRSTEN, Spencer & MATHEW, Croiss.** The effect of increasing loading on powerlifting movement form during the squat and deadlift. *RUA*. 2015. Available from DOI: 10.14198/jhse.2015.103.02.

24. **LE, Quoc V. et al.** On optimization methods for deep learning. In: *International Conference on Machine Learning*. Bellevue, Washington, USA: Omnipress, 2011, pp. 265–272. ICML'11. ISBN 9781450306195.
25. **LEWIS, Bryce.** *Progressions of video analysis* [<https://www.thestrengthathlete.com/blog/progressions-of-video-analysis>]. 2023. En línea; último acceso 27 de diciembre de 2023.
26. **LI, Shutao et al.** Deep Learning for Hyperspectral Image Classification: An Overview. *IEEE Transactions on Geoscience and Remote Sensing*. 2019, vol. 57, no. 9, pp. 6690–6709. Available from DOI: 10.1109/TGRS.2019.2907932.
27. **LOZANO, Ricardo Alfonso Casanova.** *Procesamiento de algoritmos de visión artificial en la nube – Amazon AWS*. 2021. Universidad de Alcalá.
28. **LU, D. & WENG, Q.** A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*. 2007, vol. 28, no. 5, pp. 823–870. Available from DOI: 10.1080/01431160600746456.
29. **LU, Yuzhen & YOUNG, Sierra.** A survey of public datasets for computer vision tasks in precision agriculture. *Computers and Electronics in Agriculture*. 2020, vol. 178, p. 105760. ISSN 0168-1699. Available from DOI: <https://doi.org/10.1016/j.compag.2020.105760>.
30. **MAIDA, Esteban Gabriel & PACIENZIA, Julián.** *Metodologías de desarrollo de software*. 2015. Universidad Católica Argentina.
31. **MALPARTIDA, Eddie Angel Sobrado.** *Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot*. 2003. Available also from: <https://www.iiis.org/cds2008/cd2008csc/cisci2008/paperspdf/c336gi.pdf>. PhD thesis. Pontificia Universidad Católica del Perú. Language: English.
32. **MEDRANO, Ivan Chulvi.** *Inclusión del peso muerto y sus variantes dentro de los programas de acondicionamiento neuromuscular saludables* [<https://www.efdeportes.com/efd115/inclusion-del-peso-muerto.htm>]. 2007. En línea; último acceso 1 de marzo de 2024.
33. **MEDSKER, Larry R; JAIN, Lakhmi, et al.** Recurrent neural networks. *Design and Applications*. 2001, vol. 5, no. 64-67, p. 2.
34. **MELGAREJO CHICCHON, Yalhy Rosario.** *Diseño E Implementación De Un Sistema De Evaluación Dinámica Para El Apoyo De La Técnica De Recepción Del Voleibol Basado En Una Configuración Óptima De Plataformas De Fuerza*. 2020. Pontificia Universidad Católica del Perú.

35. **MICHAUD, Florian et al.** Lower Back Injury Prevention and Sensitization of Hip Hinge with Neutral Spine Using Wearable Sensors during Lifting Exercises. *Sensors*. 2021, vol. 21, no. 16. ISSN 1424-8220. Available from DOI: 10.3390/s21165487.
36. **MICROSOFT, Team.** *Why did we build Visual Studio Code?* [<https://n9.cl/jkth36>]. 2023. En línea; último acceso 17 de enero de 2024.
37. **MILITARU, Cristian et al.** Physical Exercise Form Correction Using Neural Networks. In: *Companion Publication of the 2020 International Conference on Multimodal Interaction*. Virtual Event, Netherlands: Association for Computing Machinery, 2021, pp. 240–244. ICMI '20 Companion. ISBN 9781450380027. Available from DOI: 10.1145/3395035.3425302.
38. **MOHAPATRA, Subasish et al.** Comparison of MobileNet and ResNet CNN Architectures in the CNN-Based Skin Cancer Classifier Model. In: *Machine Learning for Healthcare Applications*. John Wiley & Sons, Ltd, 2021, chap. 11, pp. 169–186. ISBN 9781119792611.
39. **MORENO, Adriana Villa et al.** Consideraciones para el análisis de la marcha humana. Técnicas de videogrametría, electromiografía y dinamometría. In: 2008. Available also from: <https://n9.cl/0cy02>.
40. **NAIK, Banoth Thulasya et al.** A Comprehensive Review of Computer Vision in Sports: Open Issues, Future Trends and Research Directions. *Applied Sciences*. 2022, vol. 12, no. 9. ISSN 2076-3417. Available from DOI: 10.3390/app12094429.
41. **ORR, Rhonda et al.** Power Training Improves Balance in Healthy Older Adults. *The Journals of Gerontology: Series A*. 2006, vol. 61, no. 1, pp. 78–85. ISSN 1079-5006. Available from DOI: 10.1093/gerona/61.1.78.
42. **OSORIO CIRO, Jorge Alberto et al.** Lesiones deportivas. *Iatreia*. 2007, vol. 20, pp. 167–177. ISSN 0121-0793. Available also from: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-079320070002000006&nrm=iso.
43. **PARISI, German I. et al.** Learning human motion feedback with neural self-organization. In: *2015 International Joint Conference on Neural Networks (IJCNN)*. 2015, pp. 1–6. Available from DOI: 10.1109/IJCNN.2015.7280701.
44. **PARKER, David.** *The Deadlift* [<https://n9.cl/7t2fk>]. 2008. En línea; último acceso 1 de marzo de 2024.
45. **PINTO, Víctor Manuel Melgarejo et al.** Powerlifting: entrenamiento de fuerza, 1 RM contra ecuación Brzycki, en deportista con discapacidad física (Powerlifting: strength training, 1 RM vs Brzycki equation in an athlete with physical impairment). *Retos*. 2020. Available also from: <https://api.semanticscholar.org/CorpusID:219108568>.

46. **PRIETO, Carlos Gerardo.** *Adaptación de las Metodologías Tradicionales Cascada y Espiral para la Inclusión de Evaluación Inicial de Usabilidad en el Desarrollo de Productos de Software en México.* 2015. Universidad Tecnológica de la Mixteca.
47. **PUERTAS, Vanessa Alvear et al.** Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura. *SciELO.* 2017, vol. 8, pp. 244–256. Available from DOI: 10.29019/enfoqueute.v8n1.121.
48. **QUILLUPANGUI, Soraya.** *Los gimnasios son espacios para la intensa actividad y la amistad* [<https://www.elcomercio.com/actualidad/quito/gimnasios-actividad-amistad-salud-entrenamiento.html>]. 2022. En línea; último acceso 17 de julio de 2024.
49. **RECAS, Lucio Doncel.** *Manual de powerlifting y otras modalidades de levantamiento de peso.* Vision Libros, 2010.
50. **ROSENHAIM, Argus Luconi.** *Human Action Evaluation applied to Weightlifting.* 2023. Available also from: <https://repositorio-aberto.up.pt/bitstream/10216/152213/2/637071.pdf>. PhD thesis. Universidade do Porto. Language: English.
51. **SANCHEZ, S A et al.** A review: Comparison of performance metrics of pretrained models for object detection using the TensorFlow framework. *IOP Conference Series: Materials Science and Engineering.* 2020, vol. 844, no. 1, p. 012024. Available from DOI: 10.1088/1757-899X/844/1/012024.
52. **SINGH, Arush & TAQA, Amer.** Application of Machine Learning with Convolutional Neural Networks to Prevent Injuries Caused Due to Compound Movements and Exercises. *International Journal of All Research Education Scientific Methods.* 2022, vol. 10, pp. 2455–6211.
53. **SORIA, Xavier et al.** Wide-Band Color Imagery Restoration for RGB-NIR Single Sensor Images. *Sensors.* 2018, vol. 18, no. 7. ISSN 1424-8220. Available from DOI: 10.3390/s18072059.
54. **STEPHEN, Bird & BENJAMIN, Barrington Higgs.** Exploring the Deadlift. *Strength and Conditioning Journal.* 2010. Available from DOI: 10.1519/SSC.0b013e3181d59582.
55. **STRÖMBÄCK, Edit et al.** Prevalence and Consequences of Injuries in Powerlifting: A Cross-sectional Study. *PubMed.* 2018, vol. 6, no. 5. Available from DOI: 10.1177/2325967118771016.
56. **SUN, Shih-Wei et al.** Deadlift Recognition and Application based on Multiple Modalities using Recurrent Neural Network. *Electronic Imaging.* 2020, vol. 2020, pp. 2–1. Available from DOI: 10.2352/ISSN.2470-1173.2020.17.3DMP-A17.

57. **SUTTHIPRAPA, Suputra et al.** The deadlift form analysis system using Microsoft Kinect. *Procedia Computer Science*. 2017, vol. 111, pp. 174–182. ISSN 1877-0509. Available from DOI: <https://doi.org/10.1016/j.procs.2017.06.025>. The 8th International Conference on Advances in Information Technology.
58. **TEACHABLE MACHINE, Team.** *FAQ* [<https://teachablemachine.withgoogle.com/faq>]. 2024. En línea; último acceso 27 de abril de 2024.
59. **TEIKARI, Petteri & PIETRUSZ, Aleksandra.** Precision strength training: Data-driven artificial intelligence approach to strength and conditioning. *SportRxiv*. 2021. Available from DOI: [10.31236/osf.io/w734a](https://doi.org/10.31236/osf.io/w734a).
60. **UM, Terry Taewoong et al.** Exercise motion classification from large-scale wearable sensor data using convolutional neural networks. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 2385–2390. Available from DOI: [10.1109/IROS.2017.8206051](https://doi.org/10.1109/IROS.2017.8206051).
61. **YASSER, Ammar et al.** Smart Coaching: Enhancing Weightlifting and Preventing Injuries. *International Journal of Advanced Computer Science and Applications*. 2019, vol. 10. Available from DOI: [10.14569/IJACSA.2019.0100789](https://doi.org/10.14569/IJACSA.2019.0100789).
62. **ZAMBRANO, Lina.** *Los gimnasios se preparan para tener más clientes en el 2023* [<https://www.expreso.ec/actualidad/economia/gimnasios-preparan-clientes-2023-143381.html>]. 2022. En línea; último acceso 17 de julio de 2024.





ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
CERTIFICADO DE CUMPLIMIENTO DE LA GUÍA PARA
NORMALIZACIÓN DE TRABAJOS DE FIN DE GRADO

Fecha de entrega: 17/1/2025

INFORMACIÓN DEL AUTOR
Nombres – Apellidos: Jaime Alexander Jiménez Peralta
INFORMACIÓN INSTITUCIONAL
Facultad: Facultad de Informática y Electrónica (Sede Morona Santiago)
Carrera: Tecnologías de la Información (Sede Morona Santiago)
Título a optar: Ingeniero en Tecnologías de la Información
<div style="text-align: center;"> Xavier Soria Poma Director del Trabajo de Titulación</div> <div style="text-align: center;"> Alberto Alzola Tamayo Asesor del Trabajo de Titulación</div>