



Universidad Nacional Pedro Ruiz Gallo
Facultad de Ciencias Físicas y Matemáticas Escuela
Profesional de Ingeniería Electrónica



TESIS

Evaluación de Modelos de Aprendizaje Profundo en Computadoras de Plataforma Única para la Detección de Caídas en Personas en el Hogar

Para optar el título profesional de Ingeniero Electrónico

Presentado por:

Bach. Juan Carlos, Regalado Villegas

Bach. Anthony Adrian, Aguilar Ramírez

Asesor:

Ing. Segundo Francisco Segura Altamirano.

Lambayeque - Perú

2025

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

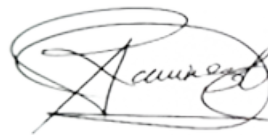
Evaluación de Modelos de Aprendizaje Profundo en
Computadoras de Plataforma Única para la Detección de
Caídas en Personas en el Hogar

Autores:

Bach. Juan Carlos, Regalado Villegas



Bach. Anthony Adrian, Aguilar Ramírez



Asesor:

Ing. Segundo Francisco Segura Altamirano.



LAMBAYEQUE - PERÚ

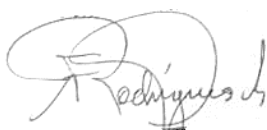
2025

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

Evaluación de Modelos de Aprendizaje Profundo en
Computadoras de Plataforma Única para la Detección de
Caídas en Personas en el Hogar

Tesis Para optar el título profesional de Ingeniero Electrónico

Sustentado y aprobado ante los siguientes miembros del jurado:



Dr. Frank Richard Rodriguez Chirinos.
Presidente



M.Sc. Hugo Javier, Chiclayo Padilla.
Secretario



Mg. Ing. Martín Augusto, Nombera Lossio
Vocal

LAMBAYEQUE - PERÚ

2025



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS
DECANATO

Ciudad Universitaria - Lambayeque

LICENCIADA - RESOLUCION DEL CONSEJO DIRECTIVO N° 015 -2023-SUNEDU / CD



ACTA DE SUSTENTACIÓN N° ⁷⁸.....-2025--D/FACFyM

Siendo las 7:30 am del día 19 de noviembre del 2025, se reunieron los miembros del jurado evaluador de la Tesis titulada:

Evaluación de Modelos de Aprendizaje Profundo en Computadoras de Plataforma Unica para la detección de caídas en personas en el Hogar

Designados por Resolución N° 418-2024 D/FACFyM de fecha 10 de abril 2024

Con la finalidad de evaluar y calificar la sustentación de la tesis antes mencionada, conformada por los siguientes docentes:

Dr. Ing. Frank Richard Rodriguez Chirinos

Presidente

M.Sc. Ing. Hugo Javier Chidayo Padilla

Secretario

Mg. Ing. Martin Augusto Nombela Lasso

Vocal

La tesis fue asesorada por (el) (la) M.Sc. Ing. Segundo Francisco Segura Altamirano nombrado por Resolución N° 418-2024 D/FACFyM de fecha 10 abril de 2024

El Acto de Sustentación fue autorizado por Resolución N° 1042-2025 D/FACFyM de fecha 31 de octubre de 2025

La Tesis fue presentada y sustentada por (el) (los) Bachiller (es) Regalado Villegas Juan Carlos y Aguilar Ramirez Anthony Adrian y tuvo una duración de 50 minutos.

Después de la sustentación, y absueltas las preguntas y observaciones de los miembros del jurado se procedió a la calificación respectiva, otorgándole el Calificativo de Dieciseis (16) en la escala vigesimal, mención (Bueno).

Por lo que queda(n) apto(s) para obtener el Título Profesional de Ingeniero Electrónico, de acuerdo con la Ley Universitaria 30220 y la normatividad vigente de la Facultad de Ciencias Físicas y Matemáticas y la Universidad Nacional Pedro Ruiz Gallo.

Siendo las 8:30 am se dio por concluido el presente acto académico, dándose conformidad al presente acto con la firma de los miembros del jurado.

DR. ING. FRANK RICHARD RODRIGUEZ CHIRINOS
Presidente

Mg. Ing. Martin Augusto Nombela Lasso
Secretario

Mg. Ing. Segundo Francisco Segura Altamirano
Vocal

Mg. Ing. Segundo Francisco Segura Altamirano
Asesor

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIDAD DE INVESTIGACIÓN
SISTEMA DE EVALUACION DEL TRABAJO FINAL DE CARRERA

(Basado en el Modelo de Calidad para la acreditación de las carreras profesionales de Ingeniería y Arquitectura – Dimensión II – Factor 3 - Estandar 47)

FORMATO N° 1
REGISTRO DEL PROYECTO DE INVESTIGACIÓN PARA VERIFICACION
DE NO DUPLICIDAD

I. DATOS PERSONALES:

01. DATOS DE LOS ALUMNOS:

ALUMNO N° 1:

CODIGO: 020160811H
APELLIDOS Y NOMBRES: REGALADO VILLEGAS, JUAN CARLOS

ALUMNO N° 2:

CODIGO: 020160784K
APELLIDOS Y NOMBRES: AGUILAR RAMÍREZ, ANTHONY ADRIAN

02. MODALIDAD DE Tesis
03. UNIVERSIDAD DE ORIGEN: UNPRG
04. FACULTAD: Facultad de Ciencias Físicas y Matemáticas
05. ESCUELA PROFESIONAL: INGENIERIA ELECTRONICA
06. ASESOR: SEGURA ALTAMIRANO, SEGUNDO FRANCISCO

II. DATOS DEL PROYECTO:

01. TITULO DEL PROYECTO:

EVALUACIÓN DE MODELOS DE APRENDIZAJE PROFUNDO EN COMPUTADORAS DE PLATAFORMA ÚNICA PARA LA DETECCIÓN DE CAÍDAS EN PERSONAS EN EL HOGAR

02. CODIGO DEL PROYECTO:

03. RESUMEN DEL PROYECTO:

ESTA INVESTIGACIÓN BUSCA DETERMINAR, MEDIANTE UNA EVALUACIÓN RIGUROSA, SI MODELOS DE APRENDIZAJE PROFUNDO OPTIMIZADOS E IMPLEMENTADOS EN COMPUTADORAS DE PLATAFORMA ÚNICA, PUEDEN DETECTAR CAÍDAS DE ADULTOS MAYORES QUE VIVEN SOLOS CON UNA PRECISIÓN SUPERIOR AL 80%, MANTENIENDO CONSUMOS DE POTENCIA, TEMPERATURA Y MEMORIA RAM DENTRO DE LÍMITES OPERATIVOS ACEPTABLES. LOS RESULTADOS DE LA INVESTIGACIÓN BRINDARÁN EVIDENCIA SOBRE LA VIABILIDAD DE USAR ESTA TECNOLOGÍA PARA DETECTAR CAÍDAS OPORTUNAMENTE EN EL HOGAR. ADEMÁS, SE ESTABLECERÁN CRITERIOS TÉCNICOS Y RECOMENDACIONES PRÁCTICAS PARA

DESARROLLAR E IMPLEMENTAR SISTEMAS CONFIABLES DE DETECCIÓN DE CAÍDAS BASADOS EN APRENDIZAJE PROFUNDO SOBRE PLATAFORMAS DE BAJO COSTO.

04. OBJETIVO GENERAL:

EVALUAR LA CAPACIDAD DE MODELOS DE APRENDIZAJE PROFUNDO CON DIFERENTES ARQUITECTURAS IMPLEMENTADOS EN COMPUTADORAS DE PLATAFORMA ÚNICA PARA DETECTAR CAÍDAS DE PERSONAS MAYORES EN EL HOGAR, CONSIDERANDO CRITERIOS DE PRECISIÓN, TIEMPO DE RESPUESTA, CONSUMO DE POTENCIA, TEMPERATURA DE TRABAJO Y USO DE MEMORIA RAM.

05. OBJETIVOS ESPECIFICOS:

DISEÑAR Y ENTRENAR TRES ARQUITECTURAS DE MODELOS DE APRENDIZAJE PROFUNDO PARA LA DETECCIÓN DE CAÍDAS A PARTIR DE DATOS DE IMÁGENES O VIDEOS. IMPLEMENTAR LOS MODELOS DISEÑADOS EN UNA COMPUTADORA DE PLATAFORMA ÚNICA BASADA EN CPU O ACELERADORES GRÁFICOS. EVALUAR Y COMPARAR LA PRECISIÓN, TIEMPO DE RESPUESTA, CONSUMO ENERGÉTICO, TEMPERATURA Y USO DE MEMORIA RAM DE LOS MODELOS IMPLEMENTADOS

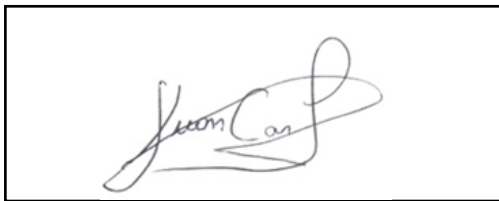
06. AREA DE INVESTIGACION:

INGENIERÍA ELECTRÓNICA

07. LINEA DE INVESTIGACION:

INGENIERÍAS Y TECNOLOGÍAS

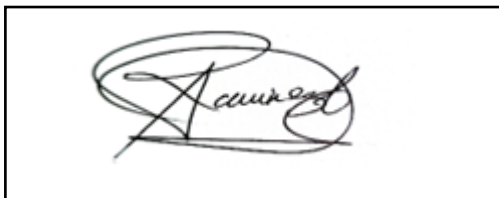
Fecha: Lambayeque, 01/03/2024



REGALADO VILLEGAS, JUAN CARLOS - Firma



SEGURA ALTAMIRANO, SEGUNDO



AGUILAR RAMÍREZ, ANTHONY ADRIAN - Firma

ORIGINALITY REPORT

5%

SIMILARITY INDEX

4%

INTERNET SOURCES

%

PUBLICATIONS

2%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

2%

★ www.coursehero.com

Internet Source

MSc. Ing. Segundo Francisco Segura Altamirano
Asesor

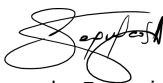
Exclude quotes Off

Exclude bibliography On

Exclude matches < 15 words

tesisFinalV1

by Juan Carlos/anthony Adrian Regalado Villegas/aguilar Ramírez



MSc. Ing. Segundo Francisco Segura Altamirano
Asesor

Submission date: 23-Jul-2025 04:30PM (UTC-0500)

Submission ID: 2719618797

File name: tesisDetectaCaidas.pdf (8.41M)

Word count: 18678

Character count: 107237



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



TESIS

Evaluación de Modelos de Aprendizaje Profundo en Computadoras de Plataforma Única para la Detección de Caídas en Personas en el Hogar

Para optar el título profesional de Ingeniero Electrónico

Presentado por:

Bach. Juan Carlos, Regalado Villegas

Bach. Anthony Adrian, Aguilar Ramírez

Asesor:

Ing. Segundo Francisco Segura Altamirano.

LAMBAYEQUE-PERÚ

2024

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

Evaluación de Modelos de Aprendizaje Profundo en
Computadoras de Plataforma Única para la Detección de
Caídas en Personas en el Hogar

Autores:

Bach. Juan Carlos, Regalado Villegas _____

Bach. Anthony Adrian, Aguilar Ramírez _____

Asesor:

Ing. Segundo Francisco Segura Altamirano.  _____

LAMBAYEQUE - PERÚ

2024

UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

Evaluación de Modelos de Aprendizaje Profundo en
Computadoras de Plataforma Única para la Detección de
Caídas en Personas en el Hogar

Tesis Para optar el título profesional de Ingeniero Electrónico

Sustentado y aprobado ante los siguientes miembros del jurado:

Dr. Frank Richard Rodríguez Chirinos.
Presidente

M.Sc.Hugo Javier, Chiclayo Padilla.
Secretario

Mg. Ing. Martín Augusto, Nombera Lossio
Vocal

LAMBAYEQUE - PERÚ

2024



Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author: Juan Carlos/anthony Adrian Regalado Villegas/aguilar Ramírez
Assignment title: Quick Submit
Submission title: tesisFinalV1
File name: tesisDetectaCaidas.pdf
File size: 8.41M
Page count: 101
Word count: 18,678
Character count: 107,237
Submission date: 23-Jul-2025 04:30PM (UTC-0500)
Submission ID: 2719618797



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



TESIS

Evaluación de Modelos de Aprendizaje Profundo en

Computadoras de Plataforma Única para la Detección de

Caídas en Personas en el Hogar

Para optar el título profesional de Ingeniero Electrónico

Presentado por:

Bach. Juan Carlos, Regalado Villegas
Bach. Anthony Adrian, Aguilar Ramírez

Asesor:

Ing. Segundo Francisco Segura Altamirano.

LAMBAYEQUE-PERÚ

2024

Msc. Ing. Segundo Francisco Segura Altamirano
Asesor

Agradecimientos

Al culminar esta etapa tan importante de mi vida, deseo expresar mi más sincero agradecimiento a todas las personas que hicieron posible la realización de esta tesis.

En primer lugar, agradezco a Dios, por darme la fortaleza, paciencia y sabiduría necesarias para afrontar cada desafío durante este proceso.

A mis padres: Antero Pasión Aguilar Aliaga y María Luisa Ramírez Mena, por su amor, sacrificio y constante apoyo. Su ejemplo de perseverancia y valores ha sido mi mayor inspiración.

A mis hermanos, por su comprensión y palabras de aliento en los momentos difíciles. Su presencia ha sido fundamental.

A mis docentes y a la institución que me brindó las herramientas académicas para crecer como profesional.

A mis amigos y compañeros, por los momentos compartidos, el apoyo mutuo y la motivación que nunca faltó.

A todas las personas que, de una u otra manera, contribuyeron a este logro. Gracias por estar presentes en este camino.

Bach. Anthony Adrian, Aguilar Ramírez

Bachiller Ingeniería Electrónica

Lambayeque 2025

Al culminar esta etapa tan importante de mi vida, deseo expresar mi más sincero agradecimiento a todas las personas que hicieron posible la realización de esta tesis. En primer lugar, agradezco a Dios, por darme la fortaleza, paciencia y sabiduría necesarias para afrontar cada desafío durante este proceso.

A mi madre: Blanca Flor Villegas Cespedes, por su amor, sacrificio y constante apoyo. Su ejemplo de perseverancia y valores ha sido mi mayor inspiración.

A mis hermanos, por su comprensión y palabras de aliento en los momentos difíciles. Su presencia ha sido fundamental.

A mis docentes y a la institución que me brindó las herramientas académicas para crecer como profesional.

A mis amigos y compañeros, por los momentos compartidos, el apoyo mutuo y la motivación que nunca faltó. A todas las personas que, de una u otra manera, contribuyeron a este logro. Gracias por estar presentes en este camino.

Bach. Juan Carlos, Regalado Villegas

Bachiller Ingeniería Electrónica

Lambayeque 2025

Dedicatoria

Dedico esta tesis a Dios, por ser mi guía constante en cada paso del camino. A mis padres, por su amor incondicional, esfuerzo y apoyo que han sido el pilar de mi vida.

A mi familia, por creer en mí incluso en los momentos en que yo dudaba.

Y a todos aquellos que, con palabras de aliento o silencios comprensivos, me impulsaron a seguir adelante.

Gracias por acompañarme en este viaje.

Bach. Anthony Adrian, Aguilar Ramírez

Bachiller Ingeniería Electrónica

Lambayeque 2025

Dedico esta tesis a Dios, por ser mi guía constante en cada paso del camino.

A mi madre, mis tíos, abuelos, mis hermanas y hermano por su amor incondicional, esfuerzo y apoyo que han sido el pilar de mi vida.

A mi familia, por creer en mí incluso en los momentos en que yo dudaba.

Y a todos aquellos que, con palabras de aliento o silencios comprensivos, me impulsaron a seguir adelante.

Gracias por acompañarme en este viaje.

Bach. Juan Carlos, Regalado Villegas

Bachiller Ingeniería Electrónica

Lambayeque 2025

Resumen

Las caídas de adultos mayores representan un problema crítico de salud pública, con consecuencias potencialmente fatales cuando no son atendidas oportunamente. Este estudio evaluó la capacidad de modelos de aprendizaje profundo implementados en computadoras de plataforma única para detectar caídas, considerando precisión, tiempo de respuesta, consumo energético, temperatura y uso de memoria. Se diseñaron y entrenaron cinco configuraciones de MobileNetV2 con diferentes resoluciones (100×130 a 240×320 píxeles), aplicando técnicas de cuantización (FP16, INT8) y evaluándolas en cuatro plataformas (Raspberry Pi 4/5, Jetson Nano, Tinker Edge T). Los resultados revelaron que todas las configuraciones superaron el umbral de precisión del 80 %, con F1-scores de hasta 0.936, mientras que la implementación en dispositivos económicos demostró ser viable, con rendimientos de hasta 317 FPS en hardware especializado. El análisis multidimensional identificó configuraciones óptimas para diferentes escenarios: máxima precisión (Jetson Nano, 240×320), mejor balance (Jetson Nano, 112×146) y ultra bajo consumo (Tinker Edge T, 100×130). Esta investigación establece un marco técnico replicable para sistemas de detección de caídas costo-efectivos, confirmando que el compromiso entre precisión, rendimiento y eficiencia energética puede gestionarse eficazmente mediante la selección adecuada de resolución y técnicas de optimización.

Palabras clave: Detección de caídas, Aprendizaje profundo, Visión por computadora, Computación en el borde, Asistencia para adultos mayores

Abstract

Falls among older adults represent a critical public health problem, with potentially fatal consequences when not promptly attended. This study evaluated the capability of deep learning models implemented on single-board computers to detect falls, considering accuracy, response time, power consumption, operating temperature, and memory usage. Five MobileNetV2 configurations with different resolutions (100×130 to 240×320 pixels) were designed and trained, applying quantization techniques (FP16, INT8) and evaluating them on four platforms (Raspberry Pi 4/5, Jetson Nano, Tinker Edge T). Results revealed that all configurations exceeded the 80 % accuracy threshold, with F1-scores up to 0.936, while implementation on affordable devices proved viable, with performance reaching 317 FPS on specialized hardware. Multidimensional analysis identified optimal configurations for different scenarios: maximum precision (Jetson Nano, 240×320), best balance (Jetson Nano, 112×146), and ultra-low consumption (Tinker Edge T, 100×130). This research establishes a replicable technical framework for cost-effective fall detection systems, confirming that trade-offs between accuracy, performance, and energy efficiency can be effectively managed through appropriate selection of resolution and optimization techniques.

Key words: Fall detection, Deep learning, Computer vision, Edge computing, Elderly care.

Índice general

Reconocimientos	v
Dedicatoria	vii
Resumen	ix
Abstract	x
Índice de figuras	xv
Índice de Tablas	xviii
Introducción	xix
1. Capítulo I	1
1.1. Situación problemática	1
1.2. Formulación del problema de investigación	3
1.3. Hipótesis	3
1.4. Objetivos de la investigación	4
1.4.1. Objetivo General	4
1.4.2. Objetivos Específicos	4
2. Capítulo II: Marco Teórico	5
2.1. Antecedentes	5
2.2. Bases Teóricas	10
2.3. Bases Conceptuales	16

2.3.1.	Variables	16
2.3.1.1.	Variable independiente	16
2.3.1.2.	Variable dependiente	18
2.3.2.	Operacionalización de variables	20
3.	Capítulo IV: Resultados	21
3.1.	Análisis del Estado Actual de Sistemas de Detección de Caídas	21
3.2.	Desarrollo de la Base de Datos para MobileNetV2	24
3.2.1.	Arquitectura General del Sistema de Procesamiento	24
3.2.2.	Proceso de Extracción de Frames	25
3.2.2.1.	Implementación Técnica	25
3.2.3.	Estrategia de División de Datos	27
3.2.3.1.	Proporciones de División	27
3.2.4.	Flujo de Procesamiento Principal	27
3.2.5.	Aumento de Datos (Data Augmentation)	27
3.2.6.	Validación y Control de Calidad	29
3.2.6.1.	Verificación de Integridad	29
3.2.6.2.	Métricas de Calidad	29
3.2.7.	Configuración para Entrenamiento de MobileNetV2	30
3.2.8.	Resultados y Estadísticas	30
3.3.	Implementación y Evaluación de MobileNetV2 para Detección de Caídas	30
3.3.1.	Diseño Experimental y Configuraciones de Resolución	31
3.3.2.	Arquitectura Adaptativa y Configuración del Modelo	31

3.3.3.	Metodología de Entrenamiento Sistemática	32
3.3.4.	Resultados Experimentales y Análisis Comparativo	34
3.3.5.	Evolución del Entrenamiento y Comportamiento Métrico por Resolución .	35
3.3.6.	Análisis de Matrices de Confusión y Patrones de Error	38
3.3.7.	Análisis del Trade-off Resolución-Rendimiento	40
3.4.	Optimización de Modelos mediante Técnicas de Cuantización	41
3.4.1.	Metodología de Cuantización Automatizada	41
3.4.2.	Resultados Comparativos de Estrategias de Cuantización	42
3.4.3.	Análisis Integrado de Rendimiento Computacional y Precisión	44
3.4.4.	Análisis de Trade-offs entre Precisión y Rendimiento	46
3.4.5.	Patrones de Rendimiento por Resolución	46
3.4.6.	Estabilidad Temporal y Confiabilidad Operativa	49
3.5.	Implementación y Evaluación en Plataformas de Computación Embebida	49
3.5.1.	Características de las Plataformas de Evaluación	50
3.5.2.	Configuración del Entorno de Implementación	50
3.6.	Resultados de Rendimiento por Plataforma	50
3.6.1.	Raspberry Pi 4 (4GB RAM)	50
3.6.2.	Raspberry Pi 5 (8GB RAM)	51
3.6.3.	NVIDIA Jetson Nano (4GB RAM)	52
3.6.4.	ASUS Tinker Edge T (1GB RAM)	53
3.6.5.	Análisis Comparativo de Rendimiento y Métricas Clave	54
3.6.5.1.	Análisis de Latencia y Velocidad de Procesamiento	54
3.6.5.2.	Trade-off entre Precisión y Velocidad	55

3.6.5.3.	Análisis de Eficiencia Energética	57
3.6.5.4.	Perfiles Térmicos	58
3.6.5.5.	Uso de Memoria RAM	59
3.6.5.6.	Análisis de Escalabilidad	60
3.6.5.7.	Configuraciones Óptimas por Escenario de Aplicación	61
4.	Capítulo V: Discusión	65
4.1.	Implicaciones para el Desarrollo de Sistemas de Detección de Caídas	65
4.1.1.	Hallazgos Clave y su Significado Práctico	65
4.1.2.	Modelo Conceptual Integrado para Sistemas de Detección	66
4.2.	Análisis Crítico de las Configuraciones Recomendadas	67
4.2.1.	Escenario de Máxima Precisión (Jetson Nano con 240×320 FP16)	67
4.2.2.	Escenario de Balance Óptimo (Jetson Nano con 112×146 INT8)	67
4.2.3.	Escenario de Ultra Bajo Consumo (Tinker Edge T con 100×130 INT8)	68
4.3.	Comparación con Trabajos Previos y Estado del Arte	68
5.	Capítulo V: Conclusiones	70

Índice de figuras

2.1. Caída de Adulto mayor	10
3.1. Secuencia completa de detección de caída mostrando la progresión de la precisión .	22
3.2. Fase inicial de la caída con 17 % de precisión	23
3.3. Fase intermedia de la caída mostrando mejora en la precisión	23
3.4. Fase final de la caída con máxima precisión alcanzada	24
3.5. Arquitectura general del sistema de procesamiento de datos para MobileNetV2 . .	25
3.6. Diagrama de flujo del procesamiento principal de datos	28
3.7. Arquitectura detallada de MobileNetV2 para detección de caídas, mostrando el flujo desde la entrada (320×240×3) hasta la decisión final, incluyendo la distribución de parámetros entrenables y congelados.	33
3.8. Proceso completo de entrenamiento de MobileNetV2, incluyendo preparación de datos, aumento, entrenamiento inicial, fine-tuning, y sistema de callbacks con re- sultados preliminares de cada fase.	34
3.9. Evolución de pérdida y métricas – Modelo MobileNetV2 con entrada 100×130 . .	36
3.10. Evolución de pérdida y métricas – Modelo MobileNetV2 con entrada 112×146 . .	36
3.11. Evolución de pérdida y métricas – Modelo MobileNetV2 con entrada 124×162 . .	37
3.12. Evolución de pérdida y métricas – Modelo MobileNetV2 con entrada 172×224 . .	37
3.13. Evolución de pérdida y métricas – Modelo MobileNetV2 con entrada 240×320 . .	38
3.14. Matrices de confusión para resoluciones bajas	39
3.15. Matrices de confusión para resoluciones medias	39
3.16. Matriz de confusión para resolución 240×320	40

3.17. Dashboard Ejecutivo con Análisis Integrado de Rendimiento	45
3.18. Análisis de trade-offs entre precisión y rendimiento.	47
3.19. Comparativa Multidimensional de Modelos por Tamaño	48
3.20. Comparación de latencia entre plataformas. (a) Distribución de latencia por plataforma y tipo de modelo. (b) Escalabilidad de latencia para modelos INT8 en función del tamaño de entrada.	55
3.21. Trade-off entre precisión (F1-Score) y velocidad de procesamiento (FPS). La línea roja vertical indica el umbral de tiempo real (15 FPS), mientras que la línea verde horizontal marca el umbral de alta precisión (F1 ≥ 0.9).	56
3.22. Análisis de eficiencia energética. (a) Consumo promedio por plataforma. (b) Heatmap de eficiencia energética (FPS/Watt) por plataforma y tipo de modelo.	57
3.23. Perfiles térmicos por plataforma durante inferencia continua. La línea roja discontinua indica el límite térmico crítico (80°C).	58
3.24. Análisis de uso de memoria. (a) Consumo de memoria por resolución y tipo de modelo. (b) Eficiencia de memoria (píxeles procesados por MB) por plataforma. . .	59
3.25. Análisis de escalabilidad para modelos INT8. Se muestran las tendencias de latencia, uso de CPU, memoria y consumo energético en función del número total de píxeles procesados.	60
3.26. Matriz de rendimiento integral para modelos INT8. Los valores están normalizados entre 0 y 1, donde 1 representa el mejor rendimiento. Para consumo energético, temperatura y uso de memoria, los valores se invierten (menor es mejor).	62

3.27. Dashboard integrado de análisis de rendimiento. Incluye especificaciones de hardware, mejores configuraciones por plataforma, comparación de latencia media y recomendaciones por escenario de uso.	63
---	----

Índice de Tablas

2.1. Operacionalización de Variables	20
3.1. Estadísticas reales de la base de datos generada	30
3.2. Comparación de rendimiento de MobileNetV2 con diferentes resoluciones de entrada	35
3.3. Resumen de técnicas de cuantización aplicadas a modelos MobileNetV2	42
3.4. Resultados detallados de cuantización por modelo y técnica	43
3.5. Comparación de especificaciones técnicas de las plataformas evaluadas	50
3.6. Configuración de software por plataforma	51
3.7. Rendimiento promedio de modelos en Raspberry Pi 4	51
3.8. Rendimiento promedio de modelos en Raspberry Pi 5	52
3.9. Rendimiento promedio de modelos en NVIDIA Jetson Nano	52
3.10. Rendimiento de modelos en ASUS Tinker Edge T	53
3.11. Resumen comparativo de rendimiento promedio por plataforma	54

Introducción

Las caídas en el hogar representan una de las amenazas más silenciosas pero devastadoras para la salud y bienestar de los adultos mayores. Cada año, aproximadamente 684,000 personas en el mundo pierden la vida a causa de caídas, posicionándose como la segunda causa principal de muerte por lesiones no intencionales. Más preocupante aún, los adultos mayores de 60 años son quienes sufren la mayor cantidad de caídas fatales, con cifras que siguen aumentando en proporción directa al envejecimiento poblacional global. Este fenómeno no se limita a países específicos sino que constituye un desafío universal para los sistemas de salud, las familias y las comunidades. Más allá de las estadísticas, la realidad cotidiana muestra que cuando un adulto mayor sufre una caída y no recibe atención inmediata, las consecuencias pueden ser devastadoras. Una fractura de cadera no atendida a tiempo puede convertirse en una condición permanente que limite la movilidad. Un traumatismo craneal aparentemente leve puede desencadenar complicaciones neurológicas graves. Incluso las caídas sin lesiones físicas evidentes generan un daño psicológico significativo: el miedo a caer nuevamente. Este temor conduce frecuentemente a una reducción voluntaria de actividades, aislamiento social y pérdida acelerada de independencia, creando un círculo vicioso que deteriora la calidad de vida y aumenta la vulnerabilidad a nuevas caídas.

En este contexto, la tecnología emerge como un aliado prometedor. Los avances recientes en inteligencia artificial, particularmente en aprendizaje profundo y visión por computadora, ofrecen la posibilidad de crear sistemas capaces de detectar automáticamente cuando ocurre una caída y alertar a cuidadores o servicios de emergencia. Sin embargo, existe una brecha significativa entre las soluciones teóricas desarrolladas en entornos de laboratorio y los sistemas prácticos que pueden

implementarse en hogares reales. Esta brecha se debe principalmente a tres factores críticos: el alto costo de los equipos necesarios, la complejidad técnica de su instalación y mantenimiento, y las limitaciones energéticas y computacionales que restringen su funcionamiento continuo. Numerosos investigadores han propuesto sistemas basados en sensores portátiles (wearables) como acelerómetros, giroscopios o sensores de presión. Si bien estos dispositivos han mostrado resultados prometedores en entornos controlados, su uso continuo por parte de adultos mayores presenta desafíos significativos: incomodidad durante periodos prolongados, necesidad de carga frecuente de baterías, y tendencia al olvido en su uso diario.

Alternativamente, los sistemas basados en cámaras y visión por computadora ofrecen la ventaja de no requerir que la persona use ningún dispositivo, permitiendo una monitorización más natural y menos intrusiva. Sin embargo, estos sistemas tradicionalmente han necesitado equipos costosos y consumo energético elevado, limitando su accesibilidad para familias de recursos moderados.

La reciente proliferación de computadoras de plataforma única (single-board computers) como Raspberry Pi, NVIDIA Jetson Nano o ASUS Tinker Edge T ha creado una oportunidad única para desarrollar sistemas de detección de caídas asequibles pero efectivos. Estas pequeñas computadoras, con tamaños similares a una tarjeta de crédito y precios desde 35 hasta 150, ofrecen capacidad de procesamiento suficiente para ejecutar algoritmos de inteligencia artificial simplificados. No obstante, sus recursos limitados (memoria RAM, potencia de procesamiento, capacidad gráfica) plantean un desafío técnico significativo: ¿cómo lograr que modelos de aprendizaje profundo, tradicionalmente pesados y demandantes, funcionen eficientemente en estos dispositivos sin sacrificar precisión?

Esta investigación aborda precisamente este desafío, evaluando sistemáticamente cómo diferentes arquitecturas de modelos de aprendizaje profundo, específicamente variaciones de MobileNetV2,

pueden implementarse y optimizarse para detectar caídas con alta precisión en computadoras de plataforma única. El estudio no se limita únicamente a medir la exactitud de detección, sino que adopta un enfoque holístico considerando cinco dimensiones críticas para aplicaciones del mundo real: precisión de detección, tiempo de respuesta, consumo energético, temperatura de operación y utilización de memoria RAM. La metodología se estructura en tres fases principales. Primero, se diseñan y entrenan modelos de aprendizaje profundo con diferentes configuraciones de resolución de entrada (desde 100×130 hasta 240×320 píxeles), empleando técnicas de transferencia de aprendizaje para maximizar la eficiencia del entrenamiento. Segundo, se aplican diferentes estrategias de optimización, incluyendo cuantización a punto flotante de 16 bits (FP16) y enteros de 8 bits (INT8), para reducir los requisitos computacionales sin comprometer excesivamente la precisión.

Finalmente, los modelos optimizados se implementan en cuatro plataformas representativas (Raspberry Pi 4, Raspberry Pi 5, NVIDIA Jetson Nano y ASUS Tinker Edge T) para evaluar su desempeño en condiciones realistas. Los resultados de esta investigación tienen implicaciones significativas tanto teóricas como prácticas. En el ámbito teórico, el estudio revela relaciones no intuitivas entre resolución de entrada, precisión de detección y eficiencia computacional, identificando puntos óptimos locales que desafían la suposición común de "mayor resolución = mejor rendimiento". En el ámbito práctico, el trabajo proporciona recomendaciones concretas sobre configuraciones optimizadas para diferentes escenarios de uso, desde sistemas de máxima precisión para entornos clínicos hasta soluciones ultraeficientes energéticamente para zonas con acceso limitado a electricidad.

La importancia de esta investigación trasciende el ámbito tecnológico y se extiende al dominio social y de salud pública. En un contexto global de envejecimiento poblacional, donde se proyecta

que para 2050 una de cada seis personas en el mundo tendrá más de 65 años, la necesidad de soluciones accesibles para el cuidado de adultos mayores se vuelve crítica. Los sistemas de detección de caídas representan no solo una herramienta para responder rápidamente a emergencias, sino también un mecanismo para extender la independencia de los adultos mayores, permitiéndoles vivir en sus hogares por más tiempo con mayor seguridad y tranquilidad para ellos y sus familias.

Finalmente, este trabajo se alinea con objetivos globales de desarrollo sostenible, particularmente en lo referente a garantizar una vida sana y promover el bienestar para personas de todas las edades. Al democratizar tecnologías anteriormente accesibles solo para entornos privilegiados, la investigación contribuye a reducir desigualdades en el acceso a cuidados de salud y seguridad para adultos mayores. La combinación de precisión técnica, eficiencia computacional y accesibilidad económica posiciona a los sistemas propuestos como una alternativa viable para implementación a gran escala, potencialmente beneficiando a millones de adultos mayores en diversos contextos socioeconómicos.

Capítulo I

1.1. Situación problemática

Las caídas en el hogar, especialmente entre adultos mayores que viven solos, representan un desafío de salud pública con consecuencias que trascienden lo físico. Según la Organización Mundial de la Salud (OMS), estas son responsables de aproximadamente 684.000 muertes anuales, siendo la segunda causa de lesiones no intencionales a nivel global, con un impacto desproporcionado en personas mayores de 60 años (Hu et al., 2023). Más allá de fracturas o traumatismos, las caídas no atendidas generan complicaciones médicas prolongadas, deterioro funcional irreversible y secuelas psicológicas, como el miedo a moverse, que socavan la autonomía y calidad de vida (Asiri et al., 2023). Estudios recientes refuerzan la urgencia de intervenciones: por ejemplo, en residencias geriátricas, el 36,9% de las caídas involucran impacto craneal, aumentando un 50% el riesgo de mortalidad (Heidel et al., 2023), mientras que intervenciones preventivas en hospitales redujeron incidencias, pero no eliminaron riesgos en pacientes de alto perfil (Kobayashi et al., 2017).

Ante este escenario, las tecnologías emergentes, como sensores ambientales o wearables, han mostrado avances pero limitaciones críticas. Sistemas multimodales que combinan acelerómetros y sensores cardíacos logran solo un 78,3% de precisión (Boudouane et al., 2023), mientras que soluciones basadas en cámaras y aprendizaje profundo, como las implementadas en NVIDIA Jetson, alcanzan un 91,6% de exactitud, aunque sin evaluar su consumo energético o estabilidad operativa en entornos reales (Zereen et al., 2023). Además, la revisión de Liu et al. (2016) reveló que la adopción de tecnologías de hogares inteligentes sigue siendo baja, en parte por la falta de evidencia sólida sobre su eficacia en la prevención de caídas (Liu et al., 2016). Esto plantea una pregunta

central: ¿cómo influyen las arquitecturas de modelos de aprendizaje profundo, implementadas en dispositivos de bajo costo como Raspberry Pi o NVIDIA Jetson, en la precisión y eficiencia operativa de sistemas de detección de caídas, considerando métricas como tiempo de respuesta, consumo energético y uso de memoria RAM?

Para responder esto, la investigación se propone evaluar la viabilidad técnica de estos modelos en plataformas accesibles, asegurando una precisión superior al 80 % y un desempeño estable bajo parámetros como temperatura de hardware y consumo de recursos. Esto implica diseñar tres arquitecturas de redes neuronales convolucionales (CNN) optimizadas para procesar video en tiempo real, utilizando datasets públicos que simulen caídas y actividades cotidianas. Posteriormente, se implementarán en dispositivos embebidos con bibliotecas de optimización como TensorFlow Lite, para reducir la carga computacional. Finalmente, se comparará su rendimiento en escenarios reales y simulados, midiendo no solo la exactitud (mediante F1-score), sino también latencia, consumo energético en vatios, temperatura máxima alcanzada y uso de memoria RAM. El objetivo es establecer un marco técnico replicable que equilibre precisión y eficiencia, ofreciendo soluciones costo-efectivas para un problema que, según proyecciones demográficas, solo incrementará su relevancia en las próximas décadas.

La urgencia de esta investigación se sustenta en hallazgos clave: las caídas no solo son frecuentes, sino que su falta de detección oportuna eleva exponencialmente el riesgo de mortalidad (Heidel et al., 2023). Aunque tecnologías como las propuestas por Zereen et al. (2023) muestran alta precisión, su implementación en entornos reales requiere validar parámetros críticos como el consumo energético, clave para dispositivos que deben operar 24/7 sin supervisión. Por otro lado, estudios como el de Boudouane (2023) y Facchinetti et al. (2023) destacan que sistemas híbridos (sensores + IA) mejoran la seguridad, pero su complejidad y costos limitan su escalabilidad.

La integración de aprendizaje profundo en hardware accesible, como Raspberry Pi, responde a estas brechas. Sarker (2021) enfatiza que modelos CNN optimizados pueden lograr alta precisión con recursos limitados, mientras que Minh (2022) demostró que cámaras de bajo costo, combinadas con algoritmos de seguimiento LSTM, detectan anomalías conductuales con un 97 % de recall. Esta investigación no solo busca superar el 80 % de precisión, sino también establecer parámetros técnicos replicables para sistemas de bajo costo, contribuyendo a políticas públicas basadas en evidencia para el envejecimiento activo (Asiri et al., 2023).

Los resultados aportarán evidencia empírica sobre la eficacia de modelos de IA en plataformas embebidas, ofreciendo un marco metodológico para desarrollar soluciones costo-efectivas que mitiguen un problema de salud global creciente.

1.2. Formulación del problema de investigación

¿En qué medida un modelo de aprendizaje profundo con diferentes arquitecturas, implementado en una computadora de plataforma única, puede detectar caídas de personas mayores en el hogar con una precisión mayor al 80 %, considerando el tiempo de respuesta, la potencia utilizada, la temperatura de trabajo y la memoria RAM consumida?

1.3. Hipótesis

La implementación de modelos de aprendizaje profundo con arquitecturas optimizadas en computadoras de plataforma única alcanza una precisión igual o superior al 80 % en la detección de caídas de personas mayores en el hogar, manteniendo un rendimiento operativo dentro de los límites aceptables de tiempo de respuesta, consumo de potencia, temperatura de trabajo y uso de memoria RAM.

1.4. Objetivos de la investigación

1.4.1. Objetivo General

Evaluar la capacidad de modelos de aprendizaje profundo con diferentes arquitecturas implementados en computadoras de plataforma única para detectar caídas de personas mayores en el hogar, considerando criterios de precisión, tiempo de respuesta, consumo de potencia, temperatura de trabajo y uso de memoria RAM.

1.4.2. Objetivos Específicos

1. Diseñar y entrenar tres arquitecturas de modelos de aprendizaje profundo para la detección de caídas utilizando datos de imágenes o videos.
2. Implementar los modelos diseñados en una computadora de plataforma única equipada con CPU o aceleradores gráficos.
3. Evaluar y comparar el rendimiento de los modelos implementados en términos de precisión, tiempo de respuesta, consumo energético, temperatura y uso de memoria RAM.

Capítulo II: Marco Teórico

2.1. Antecedentes

El estudio de Hu et al. (2023) se enfoca en las caídas de adultos mayores en el hogar, destacando su impacto significativo cuando la víctima vive sola y no puede solicitar ayuda inmediatamente. El objetivo es desarrollar un sistema innovador utilizando sensores de pulso y ultrasonido embebidos en zapatillas, conectados a una placa Arduino y transmitiendo datos vía WiFi para detectar caídas y monitorizar el ritmo cardíaco en tiempo real. Los resultados indican que el sistema es capaz de enviar alertas efectivas a cuidadores y monitorear de manera remota el pulso, demostrando la viabilidad de utilizar tecnologías IoT y comunicaciones móviles para mejorar la respuesta a emergencias en adultos mayores que viven solos.

En Asiri et al. (2023) aborda el deterioro cognitivo en adultos mayores, explorando cómo afecta su calidad de vida y la necesidad de tecnologías de monitoreo para mejorar su función cognitiva. Mediante una revisión de alcance que sigue las metodologías JBI y PRISMA-ScR, el objetivo es identificar lagunas en el desarrollo de estos dispositivos y destacar áreas para futuras investigaciones. Los resultados muestran el desarrollo de dispositivos para el tamizaje y monitoreo del deterioro cognitivo, resaltando su potencial para promover la seguridad y bienestar mental de esta población, aunque señalan la necesidad de más estudios sobre su efectividad y aplicabilidad general.

El trabajo de Hassan et al. (2019) aborda el problema crítico de las caídas entre los adultos mayores, destacando la necesidad de sistemas de detección más efectivos, adaptables y económicos. Presenta el marco MEFD, que utiliza el procesamiento en tiempo real de datos de acelerómetros en teléfonos inteligentes para detectar caídas, enviando alertas inmediatas a familiares o cuidadores.

Este enfoque se valida con un modelo híbrido CNN-LSTM, demostrando alta precisión y superando métodos anteriores, lo que refuerza la capacidad del sistema para ofrecer asistencia oportuna y mejorar la seguridad de los adultos mayores que viven solos.

El estudio de Minh et al. (2022) propone el framework MEFD para la detección de caídas en adultos mayores viviendo solos, utilizando el análisis de datos de acelerómetros en smartphones. Este sistema, que envía alertas internas y externas mediante WiFi y red móvil, se basa en un modelo híbrido de aprendizaje profundo CNN-LSTM entrenado con el conjunto de datos MobiAct. Los resultados demuestran una alta precisión en la detección de caídas, ofreciendo un medio más económico y confiable para alertar a cuidadores y solicitar asistencia, mejorando así la seguridad y el bienestar de los adultos mayores.

El estudio de Li et al. (2017) aborda el problema de detección de caídas en personas mayores utilizando redes neuronales convolucionales (CNN) para analizar secuencias de video. Su objetivo es mejorar la precisión de la detección de caídas en entornos de vigilancia de video mediante el aprendizaje de características de deformación de la forma humana en las imágenes de cada cuadro. Utilizando un conjunto de datos seleccionado que contiene actividades diarias y secuencias de caídas, el estudio implementa y modifica la arquitectura de AlexNet para clasificar las imágenes en caídas y actividades normales. Los resultados muestran una precisión promedio del 99.98 % en la detección de caídas con validación cruzada de 10 pliegues, demostrando la eficacia y el potencial de las CNN para esta aplicación, con mínima complejidad computacional y capacidad de ejecución en tiempo real.

Nazeer et al. (2023) aborda la predicción de caídas en personas mayores utilizando imágenes de CCTV y un marco de medios, destacando la importancia de monitorear continuamente a las personas mayores para detectar situaciones de emergencia rápidamente. Proponen un sistema de

bajo costo y eficiente utilizando el modelo de puntos de referencia corporales de MediaPipe y el algoritmo PCA-LSTM para predecir caídas, logrando una clasificación precisa de las posturas y movimientos humanos. Los resultados muestran una alta eficacia en la identificación de cambios de comportamiento, lo que subraya el potencial de estas tecnologías para mejorar la atención y seguridad de los adultos mayores en el hogar.

El estudio de Bajones et al. (2018) presenta el robot Hobbit, diseñado para asistir a personas mayores en sus hogares, con el fin de prolongar su independencia. Desarrollado en el marco del proyecto Hobbit, el robot integra mejoras basadas en estudios previos con usuarios, enfocándose en la detección y prevención de caídas. Durante 371 días de pruebas en Austria, Grecia y Suecia, se evaluó la interacción de los usuarios mayores con Hobbit, demostrando la importancia de la adaptabilidad y la comunicación clara de las acciones robóticas. Los resultados subrayan la necesidad de robots de servicio multifuncionales y de bajo costo en entornos reales, resaltando el desafío de equilibrar la funcionalidad útil y la seguridad en la asistencia autónoma.

El estudio de Min et al. (2018) introduce un innovador método para la detección de caídas humanas sobre muebles, utilizando el análisis de escena con aprendizaje profundo y características de actividad, superando las limitaciones de los métodos tradicionales que no funcionan eficazmente en entornos complejos. Mediante el uso de Faster R-CNN para el análisis de escena y la extracción de características específicas de actividad, este enfoque logra diferenciar con precisión las caídas de las acciones cotidianas, alcanzando altos niveles de precisión, recall y exactitud, demostrando su eficacia y aplicabilidad en situaciones reales.

El estudio de Paul et al. (2023) propone un sistema de detección de caídas utilizando Redes Convolucionales Recurrentes a Largo Plazo (LRCN) para mejorar los resultados en la atención sanitaria de los adultos mayores. Centrado en detectar caídas a partir de grabaciones de vídeo,

incluyendo las de baja resolución y sin dispositivos portátiles costosos, el sistema logró una precisión del 90% en el conjunto de datos de vídeo FD503, con una precisión y sensibilidad de 0.93 y 0.85, respectivamente. Este enfoque destaca la adaptabilidad del modelo LRCN para secuencias variables en el tiempo, subrayando el potencial de los sistemas de monitoreo en tiempo real para proporcionar alertas rápidas y contribuir a medidas preventivas en la atención sanitaria.

El aprendizaje profundo (DL) se ha convertido en una tecnología central de la Cuarta Revolución Industrial, con aplicaciones extensivas en áreas como el reconocimiento visual y el cuidado de la salud. Sin embargo, construir modelos de DL apropiados representa un desafío significativo debido a la naturaleza dinámica y las variaciones en los problemas y datos del mundo real (Sarker, 2021).

En el contexto específico de la detección de caídas, las investigaciones recientes han demostrado avances significativos en el uso de tecnologías basadas en visión por computadora. Paul et al. (2023) propusieron un enfoque basado en redes neuronales recurrentes convolucionales (LRCN) para la detección de caídas en video, incluso en baja resolución, logrando una precisión del 90% con valores de precisión y recuperación de 0.93 y 0.85 respectivamente.

Los sistemas basados en Internet de las Cosas (IoT) y aprendizaje profundo han mostrado resultados prometedores. Shilpa Aarthi y Juliet (2022) desarrollaron una técnica que utiliza redes convolucionales profundas para la extracción de características y un clasificador SVM para la detección de caídas, alcanzando una precisión del 99.81 % en el conjunto de datos UR fall detection.

La detección de caídas en entornos complejos, particularmente sobre muebles, ha sido abordada por Min et al. (2018) mediante un método que combina análisis de escena basado en aprendizaje profundo y características de actividad. Su enfoque, utilizando Faster R-CNN, logró una precisión del 94.44 %, una recuperación del 94.95 % y una exactitud del 95.50 %.

En términos de implementación en dispositivos de borde, Zereen et al. (2023) presentaron un

sistema que utiliza un dispositivo NVIDIA JETSON TX2 para el procesamiento en tiempo real, alcanzando una precisión del 91.6 % para acciones ordinarias y caídas, con una recuperación del 97.02 % para movimientos inestables.

Investigaciones más recientes han explorado arquitecturas optimizadas para dispositivos de recursos limitados. Sykes (2024) desarrolló un sistema liviano de detección de caídas que puede ejecutarse en dispositivos como Raspberry Pi, logrando una sensibilidad del 95.24 %, especificidad del 89.80 % y precisión del 98.00 %.

La privacidad ha emergido como una consideración crucial en estos sistemas. Dutt et al. (2024) introdujeron un marco que incorpora desenfoque gaussiano para preservar la privacidad, junto con estimación de pose mediante OpenPose y redes neuronales convolucionales unidimensionales (1D-CNN) para la clasificación.

El rendimiento de diferentes arquitecturas de redes neuronales ha sido extensamente estudiado. Li et al. (2017) demostraron que las CNN pueden alcanzar una precisión promedio del 99.98 % en la detección de caídas en tiempo real, mientras que Vaiyapuri et al. (2021) logró una precisión del 99.76 % utilizando una red neuronal convolucional profunda optimizada con algoritmos de enjambre.

Los últimos desarrollos incluyen modelos específicamente optimizados para dispositivos de borde. Zhao et al. (2024) presentaron YOLOv7-fall, una arquitectura mejorada que reduce los parámetros del modelo en un 3.5 % y los requisitos computacionales en un 5.4 %, manteniendo una alta precisión en la detección.

Respecto a la evaluación del rendimiento en entornos reales, Alanazi et al. (2023) validaron un sistema basado en CNN 3D multistream que alcanzó una precisión del 99.44 % y una sensibilidad del 99.12 % en el conjunto de datos Le2i, demostrando la viabilidad de estos sistemas para aplicaciones

prácticas.

Estas investigaciones previas establecen una base sólida para el desarrollo de sistemas de detección de caídas basados en aprendizaje profundo en computadoras de plataforma única. Sin embargo, existe una brecha en el estudio sistemático del rendimiento de diferentes arquitecturas considerando simultáneamente la precisión, tiempo de respuesta, consumo de potencia, temperatura de trabajo y uso de memoria RAM, aspectos que la presente investigación busca abordar.

2.2. Bases Teóricas

En las últimas décadas, la población mundial ha experimentado un aumento significativo en la proporción de adultos mayores. Este cambio demográfico trae consigo desafíos específicos relacionados con la salud y el bienestar de este grupo etario. Uno de los riesgos más preocupantes es la incidencia de caídas en el hogar, que pueden tener consecuencias graves para la salud física y psicológica de los adultos mayores, afectando su independencia y calidad de vida. Las caídas en el

Figura 2.1

Caída de Adulto mayor



hogar pueden resultar en fracturas, lesiones cerebrales traumáticas, hospitalizaciones prolongadas e

incluso la muerte. Además de las repercusiones físicas, las caídas también pueden generar miedo a caer de nuevo, lo que lleva a una disminución de la actividad física, aislamiento social y depresión.

Para abordar este problema creciente, ha habido un interés creciente en el desarrollo de sistemas de detección de caídas basados en tecnología. Estos sistemas utilizan una variedad de enfoques, incluyendo inteligencia artificial (IA), sensores portátiles y cámaras de video, para detectar caídas en tiempo real y alertar a los servicios de emergencia o cuidadores.

La IA, en particular, ha demostrado ser una herramienta prometedora en la detección de caídas. Los algoritmos de aprendizaje automático pueden analizar datos de sensores y videos para identificar patrones de movimiento que indiquen una caída. Estos sistemas pueden ser entrenados para distinguir entre caídas y otras actividades cotidianas, como sentarse o agacharse, lo que reduce las falsas alarmas.

1. Procesamiento de Imágenes y Análisis de Escenas en Video

El procesamiento de imágenes y video implica la manipulación digital de imágenes mediante algoritmos computacionales para facilitar su interpretación o modificar sus características con diversos propósitos. Algunas de las principales técnicas son (Woods, 2019):

a) Preprocesamiento de imágenes: Técnicas de limpieza, mejora, normalización y segmentación de las imágenes o fotogramas de video son pasos primordiales para garantizar la robustez de los modelos y mejorar su precisión de clasificación.

- **Mejora la calidad de los datos:** Las imágenes capturadas por cámaras pueden contener ruido, distorsiones o variaciones en la iluminación que pueden afectar negativamente el rendimiento de los algoritmos de detección de caídas. El preprocesamiento ayuda a corregir estos problemas y a garantizar que los datos sean precisos y confiables.

- Reduce la dimensionalidad de los datos: Las imágenes y videos contienen una gran cantidad de información, lo que puede dificultar el análisis y aumentar el tiempo de procesamiento. El preprocesamiento puede reducir la dimensionalidad de los datos al eliminar información irrelevante o redundante, lo que facilita el entrenamiento y la ejecución de modelos de aprendizaje automático.
- Resalta las características relevantes: El preprocesamiento puede ayudar a resaltar las características más importantes en las imágenes o videos, como los bordes, las esquinas y el movimiento. Esto facilita la tarea de los algoritmos de detección de caídas al proporcionarles información más clara y concisa.

Técnicas comunes de Preprocesamiento:

- Eliminación de ruido: El ruido puede ser causado por diversos factores, como la interferencia electromagnética o la baja calidad de la cámara. Las técnicas de eliminación de ruido, como los filtros gaussianos o medianos, pueden ayudar a reducir el ruido y mejorar la calidad de la imagen.
- Corrección de la iluminación: Las variaciones en la iluminación pueden dificultar la identificación de objetos y personas en las imágenes. Las técnicas de corrección de la iluminación, como la ecualización del histograma o la normalización, pueden ayudar a ajustar el brillo y el contraste de la imagen para mejorar la visibilidad.
- Corrección de la distorsión de la lente: Las lentes de las cámaras pueden introducir distorsiones en las imágenes, como la distorsión de barril o de cojín. Estas distorsiones pueden corregirse mediante técnicas de calibración de la cámara y corrección de la distorsión.

- Escalado y recorte: El escalado y el recorte de las imágenes pueden ayudar a reducir la dimensionalidad de los datos y a centrarse en las áreas de interés. Por ejemplo, en la detección de caídas, podríamos recortar la imagen para centrarnos en la persona y eliminar el fondo.
- Aumento de datos: El aumento de datos implica la creación de nuevas imágenes a partir de las existentes mediante técnicas como la rotación, el cambio de escala, el recorte y la adición de ruido. Esto puede ayudar a mejorar la generalización de los modelos de aprendizaje automático al proporcionarles una mayor variedad de datos de entrenamiento.

Ejemplo de Preprocesamiento en la Detección de Caídas

- Captura de video: Se captura un video de una persona en su hogar.
- Eliminación de ruido: Se aplica un filtro mediano para eliminar el ruido de la imagen.
- Corrección de la iluminación: Se aplica una ecualización del histograma para mejorar el contraste y la visibilidad.
- Detección de personas: Se utiliza un algoritmo de detección de objetos para identificar a la persona en el video y dibujar un cuadro delimitador a su alrededor.
- Seguimiento de personas: Se utiliza un algoritmo de seguimiento de objetos para rastrear el movimiento de la persona a lo largo del tiempo.
- Extracción de características: Se extraen características relevantes del cuadro delimitador, como su posición, tamaño y velocidad.
- Clasificación: Un modelo de aprendizaje automático utiliza las características extraídas para clasificar el movimiento de la persona como una caída o no.

- b) Extracción de características: Procesos para convertir imágenes a vectores numéricos manejables por algoritmos de aprendizaje, resaltando aquellos rasgos que más impactan la identificación. Se incluyen métodos tradicionales basados en detección de bordes, esquinas, hasta avanzadas transformaciones estadísticas.
- c) Análisis de escena: Se refiere a la identificación de distintos elementos de la imagen (posición, orientación, relación espacial) a fin de reconocer los patrones asociados a situaciones específicas (Zhang et al., 2022). En el caso de la detección de caídas, por ejemplo, sería útil distinguir entre mobiliario y personas Min et al. (2018).

También será importante ser capaz de conocer o desarrollar la detección y seguimiento de objetos en movimiento, reconocimiento de patrones y clasificación de acciones, análisis del flujo óptico entre cuadros para inferir velocidades entre otros.

2. Técnicas de aprendizaje profundo para imágenes

El aprendizaje profundo es un subcampo del aprendizaje automático que utiliza redes neuronales artificiales de múltiples capas para modelar abstracciones en datos a múltiples niveles. Entre sus principales ventajas están la capacidad de procesar datos en bruto, descubrir representaciones de características y generalizar bien a nuevos casos.

- a) Redes Neuronales Convolucionales (CNN): Su arquitectura optimizada para el manejo espacial de imágenes permite destacar en reconocimiento visual. Capas especializadas se dedican a tareas específicas: convolución (resaltando bordes y elementos básicos), “pooling” (reduce dimensiones destacando lo más relevante), entre otras. ([Cita sobre arquitectura básica]).

- b) Optimización de arquitecturas : Existe amplia investigación a fin de disminuir el peso computacional de las redes neuronales de tal forma que pueden adaptarse a dispositivos de baja potencia. Esto incluye reducir el número de capas, la cantidad de operaciones (FLOPs) por capa, el tamaño de los parámetros entrenables, estrategias de cuantificación (representación numérica menos costosa), entre otras ([Cita, tal vez: Yolov3]).
- c) Aprendizaje por transferencia (“Transfer Learning”): Técnica que usa los “pesos” aprendidos por un modelo de gran escala, ya entrenado para una tarea (p.ej. de reconocimiento de objetos), como base para una nueva tarea. Puede aplicarse en detección de caídas, ahorrando cómputo al iniciar a entrenar las redes neuronales con este conocimiento previo “general”.

3. Computadoras de Plataforma Única

Las computadoras de plataforma única o SBC (single board computer) son sistemas empujados completos en una sola placa de circuito. Algunos ejemplos son Raspberry Pi, Jetson Nano de NVidia y Asus Tinker Board.

- a) Concepto, características y relevancia: Una computadora de plataforma única (SBC por sus siglas en inglés) es una placa integrada con capacidades plenas de computación; incluye procesador, memoria y puertos de entrada/salida. Ofrece bajo consumo energético, bajo costo y un buen balance de capacidad y portabilidad.
- b) Ejemplos en su uso en Inteligencia Artificial “en el borde”: Raspberry Pi, Jetson Nano y la ASUS Tinker son opciones populares. El interés por su uso en aprendizaje profundo crece, sobre todo para aplicaciones donde interesa mayor independencia y control directo. Para ello, es crítico contar con versiones de herramientas para aprendizaje profundo

como Tensorflow Lite, o bien optimizadas específicamente para estos dispositivos ([Cita relevante según hardware escogido]).

Ademas debemos considerar que son de tamaño compacto y bajo costo, rendimiento limitado pero suficiente para aplicaciones embebidas, variedad de puertos y opciones de expansión, consumo energético eficiente, aptas para aplicaciones de Internet de las Cosas (IoT) y las SBC permiten ejecutar sistemas operativos completos como Linux y Windows 10 IoT, con capacidad de soportar redes neuronales y modelos de aprendizaje profundo para aplicaciones como la clasificación de imágenes en tiempo real.

2.3. Bases Conceptuales

2.3.1. Variables

2.3.1.1. Variable independiente

Modelo de Aprendizaje Profundo

Arquitectura

Definición Conceptual: Un modelo de aprendizaje profundo es una estructura computacional basada en redes neuronales artificiales con múltiples capas que permite aprender representaciones jerárquicas de datos para tareas de clasificación y detección (Sarker, 2021). En el contexto de la detección de caídas, estos modelos procesan datos de video o imagen para identificar patrones de movimiento y posturas asociados con eventos de caída.

Definición Operacional: En esta investigación, el modelo de aprendizaje profundo se refiere a una red neuronal convolucional (CNN) o una variante de esta (por ejemplo, LSTM, Transformers) que será diseñada, entrenada y evaluada para detectar caídas en adultos mayores.

■ Dimensiones:

• A. Arquitectura

- *Definición Conceptual:* Es la estructura organizacional del modelo que define cómo se conectan y organizan las diferentes capas y componentes de la red neuronal (Zhao et al., 2024).
- *Definición Operacional:* Se caracteriza mediante:
 - ◊ Número de capas: Conteo total de capas en la red neuronal, incluyendo capas de entrada, ocultas y salida.
 - ◊ Tipo de capas: Clasificación de las capas según su función (pooling para reducción de dimensionalidad, padding para ajuste de dimensiones, stripe para procesamiento de patrones).
 - ◊ Función de activación: Función matemática que determina la salida de cada neurona (lineal, sigmoideal para normalización entre 0 y 1, o tangente hiperbólica para normalización entre -1 y 1).
 - ◊ Optimizador: Algoritmo utilizado para ajustar los pesos de la red durante el entrenamiento (RMS para gradiente descendente, RMSprop para tasas de aprendizaje adaptativas, ADAM para optimización adaptativa).

• B. Precisión

- *Definición Conceptual:* Medida cuantitativa que evalúa la capacidad del modelo para identificar correctamente los eventos de caída y distinguirlos de otras actividades (Li et al., 2017).
- *Definición Operacional:* Se mide a través de:

- ◊ Error en detección de caídas reales: Porcentaje de clasificaciones incorrectas sobre el total de casos.
- ◊ Tasa de verdaderos positivos (TPR): Porcentaje de caídas reales correctamente identificadas.
- ◊ Tasa de verdaderos negativos (TNR): Porcentaje de no-caídas correctamente identificadas.

2.3.1.2. Variable dependiente

Desempeño computadoras de plataforma única

Definición Conceptual: Conjunto de métricas que caracterizan el rendimiento y eficiencia de la computadora de plataforma única durante la ejecución del modelo de detección de caídas (Zereen et al., 2023).

Definición Operacional: Esta variable se define como la demanda proyectada de servicios de conectividad en las localidades rurales de Ica, basada en análisis demográficos y de vivienda. Incluye estimaciones de usuarios potenciales de cada servicio y sus requisitos de ancho de banda.

- Se mide a través de cuatro indicadores clave:
 - Tiempo de respuesta: Intervalo temporal medido en milisegundos desde que se recibe un frame de video hasta que se completa su procesamiento y clasificación.
 - Potencia utilizada: Consumo energético medido en vatios durante la ejecución del modelo, incluyendo el procesamiento de CPU/GPU y componentes periféricos.
 - Temperatura de trabajo: Temperatura en grados Celsius del procesador y componentes críticos durante la ejecución continua del modelo de detección.

- Memoria RAM: Cantidad de memoria volátil utilizada en megabytes durante la ejecución del modelo, incluyendo carga del modelo, buffers de entrada/salida y variables temporales.
- Para la medición práctica de estas variables, se establecen los siguientes rangos operativos:
 - Tiempo de respuesta:
 - Óptimo: < 100 ms
 - Aceptable: 100-500 ms
 - Crítico: > 500 ms
 - Potencia utilizada:
 - Óptimo: < 5W
 - Aceptable: 5-10W
 - Crítico: > 10W
 - Temperatura de trabajo:
 - Óptimo: < 60°C
 - Aceptable: 60-75°C
 - Crítico: > 75°C
 - Memoria RAM:
 - Óptimo: < 512MB
 - Aceptable: 1GB-4GB
 - Crítico: > 1GB

2.3.2. Operacionalización de variables

Tabla 2.1

Operacionalización de Variables

Variable	Dimensión	Indicador	Tipo
Modelo de aprendizaje profundo	Arquitectura	Número de capas	Numérica discreta
		Tipo de capas	categorica
		Función de activación	categorica
		Optimizador	categorica
	Precisión	Error en detección de caídas reales	Numérica intervalo
		Tasa de verdaderos positivos (TPR)	Numérica intervalo
		Tasa de verdaderos negativos (TNR)	Numérica intervalo
Desempeño computadoras de plataforma única	Desempeño computadoras de plataforma única	Tiempo de respuesta	Numérica intervalo
		Potencia utilizada	Numérica intervalo
		Temperatura de trabajo	Numérica intervalo
		Memoria RAM	Numérica intervalo

Capítulo IV: Resultados

3.1. Análisis del Estado Actual de Sistemas de Detección de Caídas

Los sistemas actuales de detección de caídas basados en aprendizaje profundo exhiben limitaciones significativas en términos de precisión y rendimiento, como se evidencia en el análisis de una implementación típica (ver Figura 3.1). La secuencia de detección muestra cómo el sistema debe adaptar sus parámetros durante las diferentes etapas de una caída para mantener la capacidad de detección.

En la fase inicial (Figura 3.2), el sistema requiere reducir el umbral de confianza al 12% para lograr una detección, aunque sea con una precisión muy baja del 17%. Esta necesidad de ajustar el umbral sugiere una debilidad fundamental en la capacidad del modelo para identificar las características tempranas de una caída.

Conforme la caída progresa (Figura 3.3), el sistema muestra una mejora gradual en la precisión, alcanzando un 29% con un umbral del 13%, y posteriormente un 51% en una fase más avanzada. Solo cuando el sujeto está completamente caído (Figura 3.4), el sistema alcanza una precisión aceptable del 77% con el umbral por defecto.

Esta progresión en la precisión revela una limitación crítica en los sistemas actuales: la dependencia de posturas claramente definidas para lograr una detección confiable. Como señalan Dutt et al. (2024), los sistemas actuales enfrentan desafíos significativos en el equilibrio entre precisión y eficiencia computacional.

Las soluciones existentes proponen dos enfoques principales para abordar estas limitaciones:

- Implementación de umbrales variables que se ajustan dinámicamente durante la detección

- Sistemas paralelos que ejecutan múltiples instancias con diferentes umbrales

Sin embargo, estas soluciones presentan sus propios desafíos. Los umbrales variables requieren algoritmos adicionales para su ajuste, mientras que los sistemas paralelos aumentan significativamente los requisitos computacionales, lo que puede ser problemático en dispositivos de plataforma única con recursos limitados, como señalan Sykes (2024).

Esta situación justifica la necesidad de nuestra investigación, que busca desarrollar arquitecturas optimizadas que puedan mantener una alta precisión en todas las fases de la caída sin comprometer el rendimiento computacional. Como sugieren Zhao et al. (2024), las arquitecturas optimizadas pueden reducir significativamente los requisitos computacionales mientras mantienen o mejoran la precisión de detección.

Figura 3.1

Secuencia completa de detección de caída mostrando la progresión de la precisión

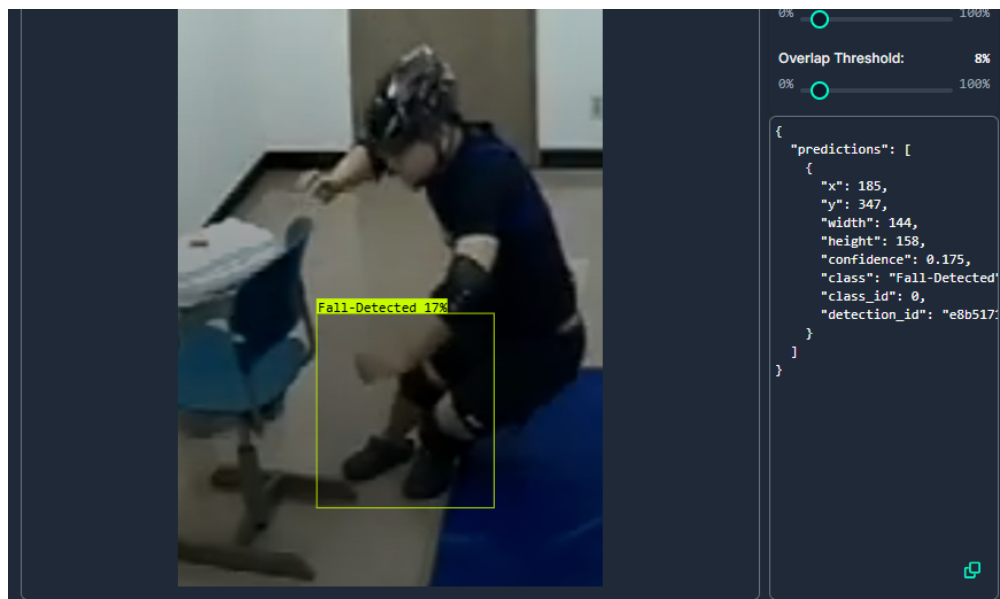


Figura 3.2

Fase inicial de la caída con 17% de precisión

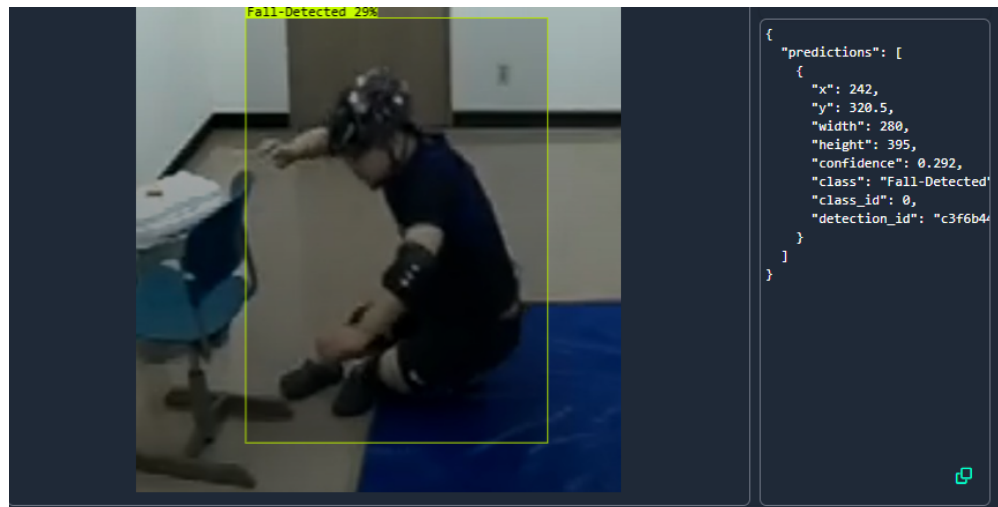


Figura 3.3

Fase intermedia de la caída mostrando mejora en la precisión

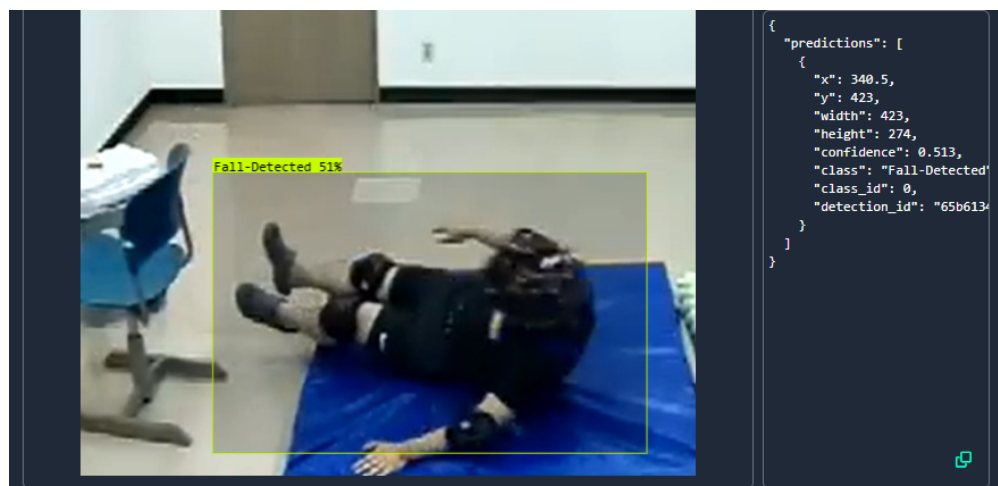
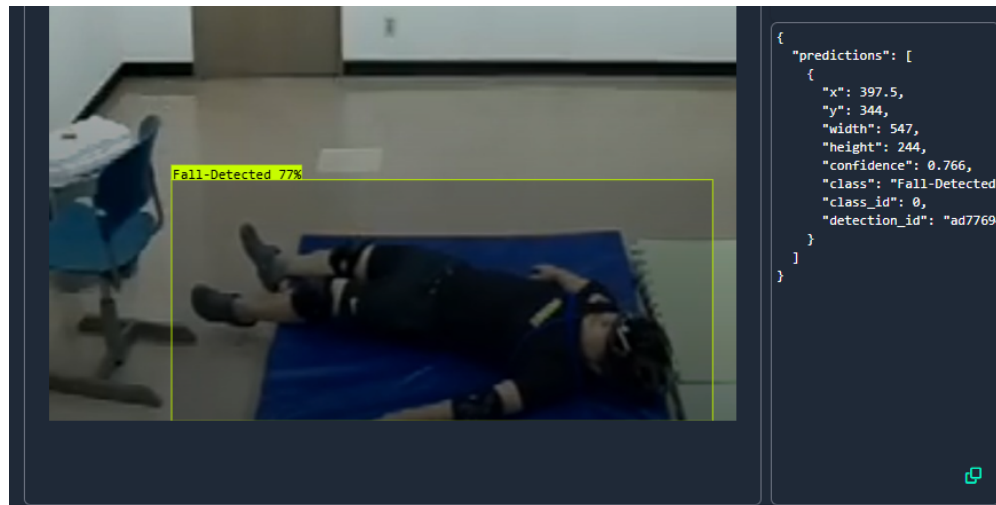


Figura 3.4

Fase final de la caída con máxima precisión alcanzada



3.2. Desarrollo de la Base de Datos para MobileNetV2

La detección automática de caídas mediante redes neuronales convolucionales requiere de conjuntos de datos cuidadosamente preparados que capturen la diversidad y complejidad de los movimientos humanos. Para el entrenamiento de MobileNetV2, se desarrolló un proceso sistemático de creación de bases de datos que transforma videos de caídas y actividades de la vida diaria (ADL) en conjuntos estructurados de imágenes etiquetadas.

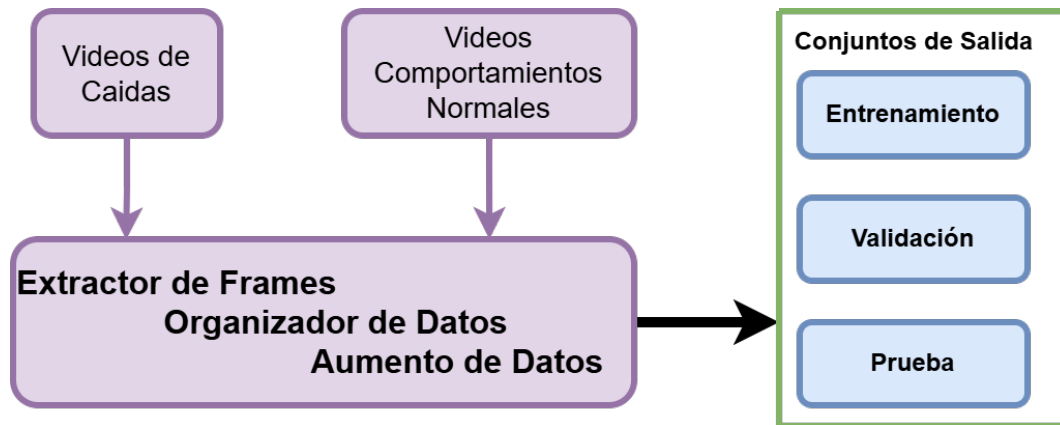
MobileNetV2, a diferencia de los modelos que procesan secuencias de video como ConvLSTM, opera sobre imágenes individuales (frames). Por tanto, es fundamental extraer frames representativos de los videos originales y organizarlos en estructuras de datos apropiadas para el entrenamiento, validación y prueba del modelo.

3.2.1. Arquitectura General del Sistema de Procesamiento

El sistema de procesamiento de datos se diseñó siguiendo una arquitectura modular que garantiza la reproducibilidad y escalabilidad del proceso. La Figura 3.5 muestra la arquitectura general del

Figura 3.5

Arquitectura general del sistema de procesamiento de datos para MobileNetV2



sistema.

3.2.2. Proceso de Extracción de Frames

La extracción de frames de los videos se realiza mediante un muestreo aleatorio estratificado que garantiza la representatividad temporal de cada video. El algoritmo implementado selecciona un número fijo de frames ($n = 30$) distribuidos uniformemente a lo largo de la duración total del video.

3.2.2.1. Implementación Técnica

La función `extract_frames` utiliza OpenCV para el procesamiento de video y manejo eficiente de memoria:

Listado 3.1: Función de extracción de frames

```
1 def extract_frames(video_path, num_frames_to_extract):
2     """Extrae un número específico de frames distribuidos de un video."""
3     cap = cv2.VideoCapture(video_path)
4     if not cap.isOpened():
5         print(f"Error: No se pudo abrir el video {video_path}")
6     return []
```

```

7
8 total_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
9 if total_frames == 0:
10 cap.release()
11 return []
12
13 frame_indices = sorted(random.sample(range(total_frames),
14 min(num_frames_to_extract, total_frames)))
15
16 extracted_frames = []
17 for i in frame_indices:
18 cap.set(cv2.CAP_PROP_POS_FRAMES, i)
19 ret, frame = cap.read()
20 if ret:
21 extracted_frames.append(frame)
22
23 cap.release()
24 return extracted_frames

```

Esta implementación presenta las siguientes ventajas:

- **Eficiencia de memoria:** Libera recursos inmediatamente después del uso
- **Robustez:** Maneja errores de lectura y videos corruptos
- **Flexibilidad:** Adapta automáticamente el número de frames extraídos a la duración del video

3.2.3. Estrategia de División de Datos

Una consideración crítica en el diseño de la base de datos es la prevención del *data leakage* a nivel de video. Para evitar que frames del mismo video aparezcan simultáneamente en los conjuntos de entrenamiento y validación, la división se realiza a nivel de video completo antes de la extracción de frames.

3.2.3.1. Proporciones de División

Las proporciones establecidas siguen las mejores prácticas en aprendizaje automático:

- **Entrenamiento (70 %):** Conjunto principal para el aprendizaje de parámetros
- **Validación (15 %):** Conjunto para ajuste de hiperparámetros y selección de modelos
- **Prueba (15 %):** Conjunto para evaluación final del modelo

3.2.4. Flujo de Procesamiento Principal

El proceso completo de creación de la base de datos se implementa mediante el flujo mostrado en la Figura 3.6.

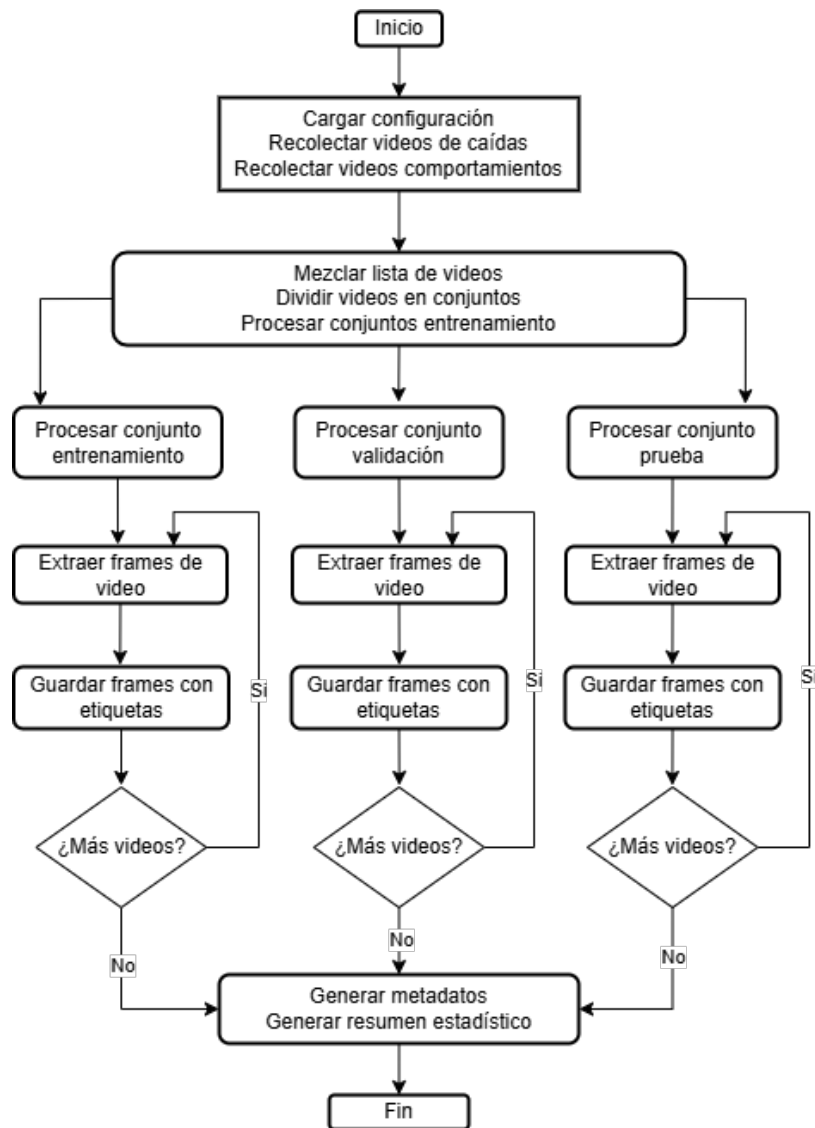
El sistema genera automáticamente un archivo JSON con metadatos completos que garantiza la trazabilidad del proceso. Adicionalmente, se genera un archivo CSV con estadísticas resumidas que facilita el análisis posterior

3.2.5. Aumento de Datos (Data Augmentation)

El aumento de datos es crucial para mejorar la generalización del modelo MobileNetV2 y reducir el sobreajuste. Las transformaciones aplicadas simulan variaciones naturales en las condiciones de captura de video, incluyendo cambios de iluminación, perspectiva y orientación.

Figura 3.6

Diagrama de flujo del procesamiento principal de datos



El sistema implementa las siguientes transformaciones mediante TensorFlow/Keras:

- **Normalización:** Reescalado de píxeles al rango [0,1]
- **Rotación:** Rotación aleatoria hasta 30°
- **Desplazamiento:** Traslación horizontal y vertical del 15
- **Cizallamiento:** Transformación de cizalla con intensidad 0.15
- **Zoom:** Acercamiento/alejamiento aleatorio del 15
- **Reflejo horizontal:** Inversión horizontal aleatoria

3.2.6. Validación y Control de Calidad

3.2.6.1. Verificación de Integridad

El sistema incluye múltiples verificaciones de integridad:

- 1. Verificación de existencia de directorios**
- 2. Validación de formatos de video soportados**
- 3. Comprobación de éxito en la lectura de frames**
- 4. Verificación de mapeo correcto de clases**

3.2.6.2. Métricas de Calidad

Para cada conjunto de datos generado, se calculan métricas de calidad:

- **Balance de clases:** Proporción entre frames de caídas y no-caídas
- **Distribución temporal:** Verificación de representatividad temporal
- **Compleitud:** Porcentaje de videos procesados exitosamente

3.2.7. Configuración para Entrenamiento de MobileNetV2

Los generadores de datos se configuran específicamente para MobileNetV2:

- **Dimensiones de entrada:** 320×240 píxeles (optimizado para MobileNetV2)
- **Tamaño de lote:** 32 imágenes (balance entre memoria y convergencia)
- **Modo de clasificación:** Binario (caída/no-caída)
- **Orden de clases:** «no_fall»=0, «fall»=1

3.2.8. Resultados y Estadísticas

El proceso de creación de la base de datos genera estadísticas completas que permiten evaluar la calidad y representatividad del conjunto de datos resultante. Un ejemplo típico de salida se muestra en la Tabla 3.1.

Tabla 3.1

Estadísticas reales de la base de datos generada

Métrica	Entrenamiento	Validación	Prueba	Total
Frames de caídas	1,200	150	300	1,650
Frames ADL (no_fall)	1,590	420	330	2,340
Total frames	2,790	570	630	3,990
Balance (% caídas)	43.0 %	26.3 %	47.6 %	41.4 %
Número de batches	88	18	20	126
Tamaño de batch	32	32	32	32

3.3. Implementación y Evaluación de MobileNetV2 para Detección de Caídas

La implementación de MobileNetV2 para la detección de caídas representa una aproximación moderna y sistemática al problema de clasificación de imágenes en tiempo real. Esta arquitectura, desarrollada originalmente por Google, se caracteriza por su eficiencia computacional y su capacidad para mantener un alto rendimiento con un número relativamente reducido de parámetros. En el

contexto de la detección de caídas, MobileNetV2 ofrece una solución equilibrada entre precisión y velocidad de inferencia, aspectos cruciales para aplicaciones en tiempo real donde la respuesta inmediata puede ser determinante.

3.3.1. Diseño Experimental y Configuraciones de Resolución

La estrategia experimental se fundamenta en la evaluación sistemática de cinco configuraciones de resolución de entrada, cada una representando un punto diferente en el espacio de compromiso entre capacidad de representación y eficiencia computacional:

- 100×130 píxeles (13,000 píxeles totales)
- 112×146 píxeles (16,352 píxeles)
- 124×162 píxeles (20,088 píxeles)
- 172×224 píxeles (38,528 píxeles)
- 240×320 píxeles (76,800 píxeles)

Esta progresión logarítmica en el número total de píxeles permite evaluar sistemáticamente cómo la información adicional disponible en resoluciones superiores se traduce en mejoras en el desempeño de detección. El diseño experimental se implementó mediante un sistema automatizado que garantiza la reproducibilidad y consistencia entre diferentes configuraciones, utilizando idénticas condiciones de inicialización aleatoria ($\text{seed}=42$), configuración de optimización, y estructura de datos.

3.3.2. Arquitectura Adaptativa y Configuración del Modelo

MobileNetV2 constituye una evolución significativa de las redes convolucionales tradicionales, introduciendo el concepto de *inverted residuals* y *linear bottlenecks*. La arquitectura base emplea

bloques de convolución separable en profundidad que reducen drásticamente el número de operaciones computacionales sin comprometer significativamente la capacidad de representación del modelo.

Para cada configuración experimental, se construye dinámicamente una instancia específica del modelo que adapta sus dimensiones de entrada mientras mantiene una arquitectura de procesamiento consistente. La cabeza clasificadora implementada mantiene una estructura consistente a través de todas las configuraciones, incorporando una capa de *Global Average Pooling* que reduce la representación espacial a un vector de características unidimensional, seguida de dos capas densas completamente conectadas con 1024 y 512 neuronas respectivamente. Entre estas capas se insertan mecanismos de regularización *Dropout* con una tasa del 40%.

La arquitectura completa implementada se presenta en la Figura 3.7, donde se visualiza la estructura híbrida del modelo con la distribución de parámetros entre las capas congeladas del modelo base y las capas entrenables de la cabeza personalizada.

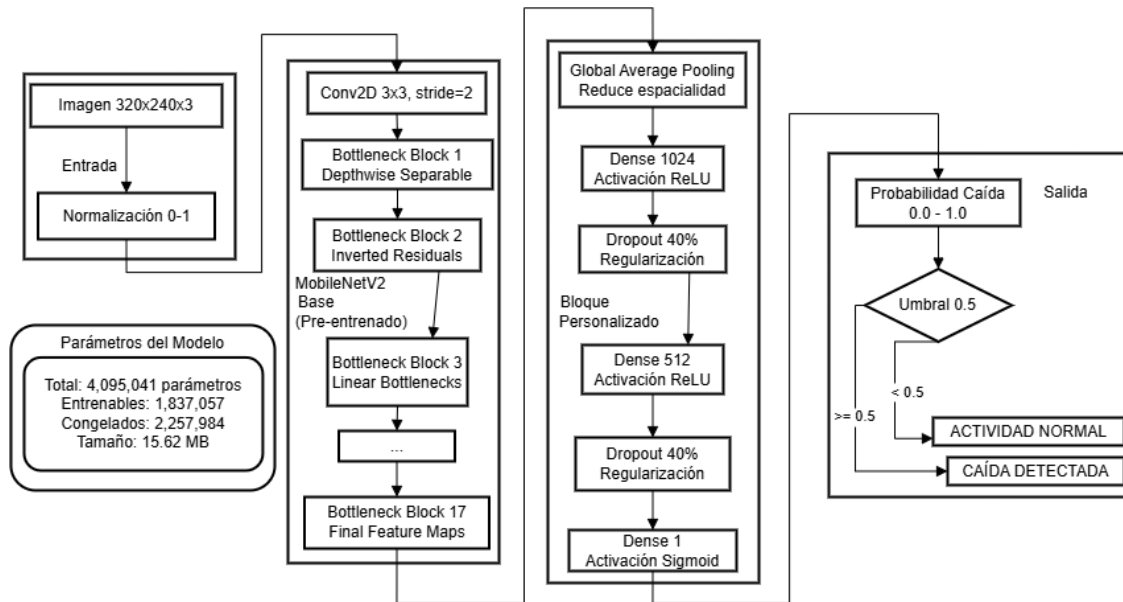
3.3.3. Metodología de Entrenamiento Sistemática

El proceso de entrenamiento se estructura en dos fases claramente diferenciadas para cada configuración experimental:

- 1. Transfer learning puro:** Las capas pre-entrenadas de MobileNetV2 permanecen congeladas mientras se optimizan únicamente las capas de clasificación personalizadas. Esta estrategia previene la degradación del conocimiento pre-entrenado mientras permite que el modelo aprenda características específicas del dominio de detección de caídas.
- 2. Fine-tuning selectivo:** Se descongelan las últimas 30 capas del modelo base para permitir la adaptación de características de nivel medio y alto específicas para la tarea. El fine-tuning se

Figura 3.7

Arquitectura detallada de MobileNetV2 para detección de caídas, mostrando el flujo desde la entrada (320×240×3) hasta la decisión final, incluyendo la distribución de parámetros entrenables y congelados.



ejecuta con una tasa de aprendizaje reducida (0.00001), diez veces menor que la fase inicial (0.0001).

El sistema de optimización emplea el algoritmo Adam con configuraciones adaptativas según la fase de entrenamiento. Cada experimento incorpora un sistema sofisticado de callbacks incluyendo *ModelCheckpoint*, *EarlyStopping* con paciencia adaptativa (10 épocas para la fase inicial, 7 épocas para fine-tuning), y *ReduceLROnPlateau* que ajusta dinámicamente la tasa de aprendizaje cuando el progreso se estanca.

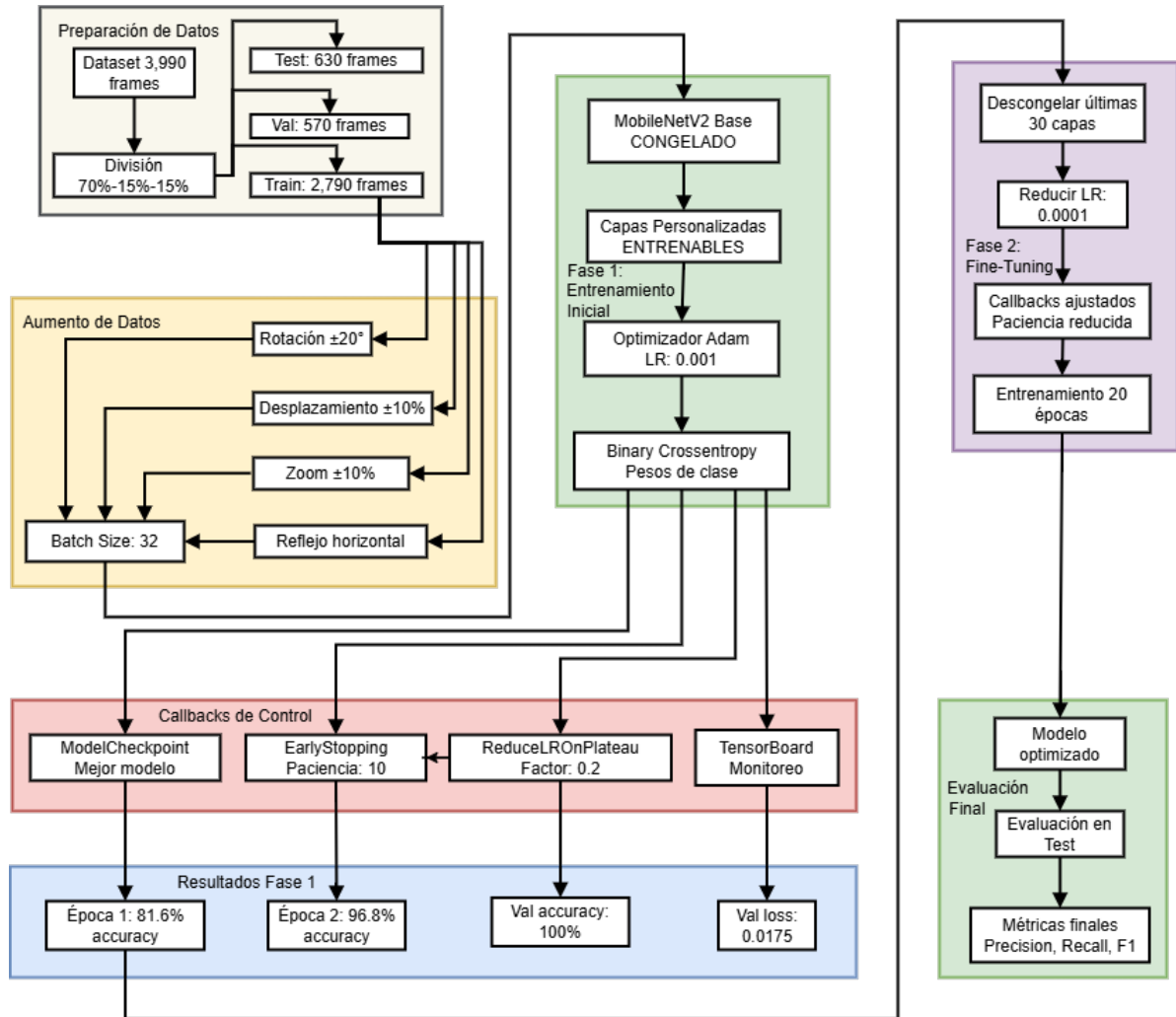
El proceso de preparación de datos se implementa mediante generadores adaptativos que ajustan dinámicamente las dimensiones de procesamiento según la configuración experimental activa, con transformaciones de aumento de datos consistentes incluyendo rotaciones aleatorias, desplazamientos, cizallamiento, zoom y reflexiones horizontales.

El flujo completo del proceso de entrenamiento automatizado, desde la preparación inicial de los

datos hasta la evaluación final, se detalla en la Figura 3.8.

Figura 3.8

Proceso completo de entrenamiento de MobileNetV2, incluyendo preparación de datos, aumento, entrenamiento inicial, fine-tuning, y sistema de callbacks con resultados preliminares de cada fase.



3.3.4. Resultados Experimentales y Análisis Comparativo

La evaluación sistemática de las cinco configuraciones de resolución reveló patrones claros en la relación entre resolución de entrada y desempeño del modelo. Los resultados experimentales, presentados en la Tabla 3.2, demuestran una mejora consistente en el desempeño conforme aumenta la resolución de entrada, aunque con rendimientos decrecientes en las resoluciones más altas.

Tabla 3.2*Comparación de rendimiento de MobileNetV2 con diferentes resoluciones de entrada*

Modelo	Resolución	Exactitud	Precisión	Recall	F1-Score	Especificidad
model_100x130	100×130	0.8854	0.7823	0.9949	0.8758	0.8105
model_112x146	112×146	0.9417	0.8813	0.9897	0.9324	0.9088
model_124x162	124×162	0.8677	0.7563	0.9949	0.8594	0.7807
model_172x224	172×224	0.9302	0.8597	0.9897	0.9201	0.8895
model_240x320	240×320	0.9469	0.9124	0.9615	0.9363	0.9368

El modelo con resolución 240×320 píxeles emergió como la configuración óptima, alcanzando el F1-score más alto (0.9363) y demostrando el mejor equilibrio entre precisión (0.9124) y recall (0.9615). Sin embargo, un hallazgo interesante fue el comportamiento excepcional del modelo 112×146, que superó significativamente al modelo de mayor resolución inmediata (124×162) en todas las métricas principales, sugiriendo la existencia de puntos óptimos locales en el espacio resolución-rendimiento.

3.3.5. Evolución del Entrenamiento y Comportamiento Métrico por Resolución

Además de las métricas agregadas al finalizar el entrenamiento, se analizó la evolución temporal de la pérdida y el comportamiento métrico durante las épocas para cada modelo. Las figuras 3.9 a 3.13 presentan, para cada resolución de entrada, el historial de pérdida para entrenamiento y validación, así como la evolución de exactitud, precisión y sensibilidad.

En el modelo con entrada 100×130 se observa una rápida disminución de la pérdida en entrenamiento, mientras que la pérdida en validación se estabiliza con ligeras oscilaciones a partir de la época 10. Este comportamiento sugiere una generalización moderada, con riesgo potencial de sobreajuste mitigado parcialmente por el uso de regularización y callbacks.

El modelo 112×146 presenta una curva de pérdida mucho más estable en validación, con convergencia clara desde las primeras épocas. Las curvas de precisión y recall en validación crecen

Figura 3.9

Evolución de pérdida y métricas – Modelo MobileNetV2 con entrada 100×130

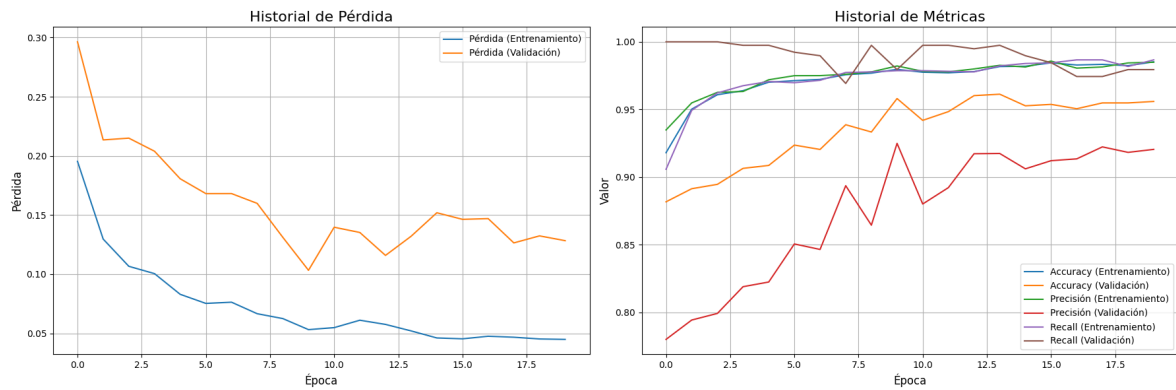
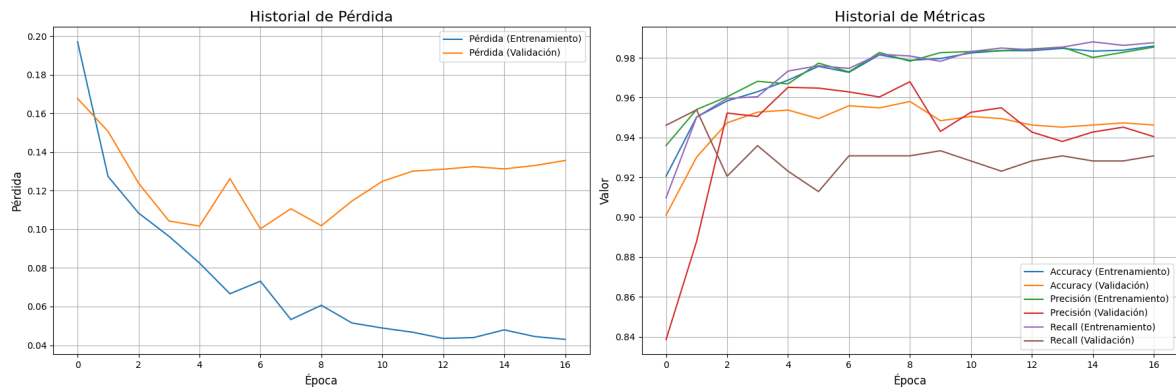


Figura 3.10

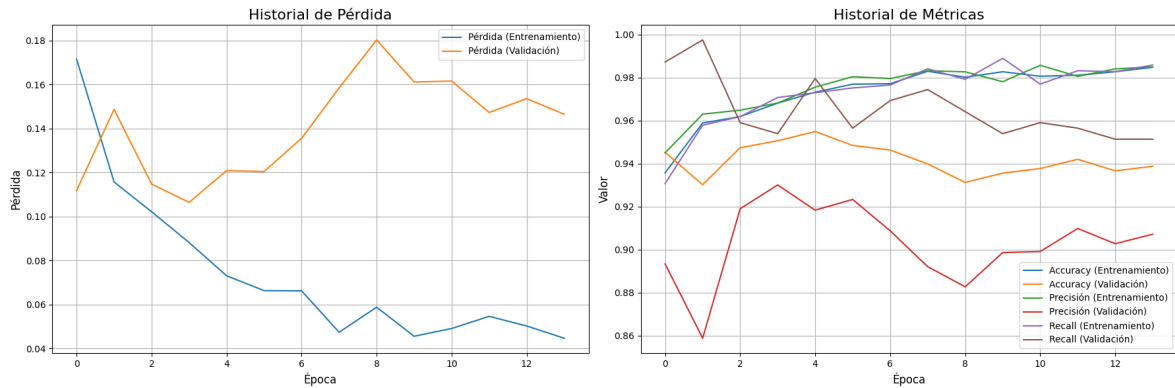
Evolución de pérdida y métricas – Modelo MobileNetV2 con entrada 112×146



sostenidamente y mantienen una brecha reducida respecto al entrenamiento, lo que sugiere una adecuada capacidad de generalización.

Figura 3.11

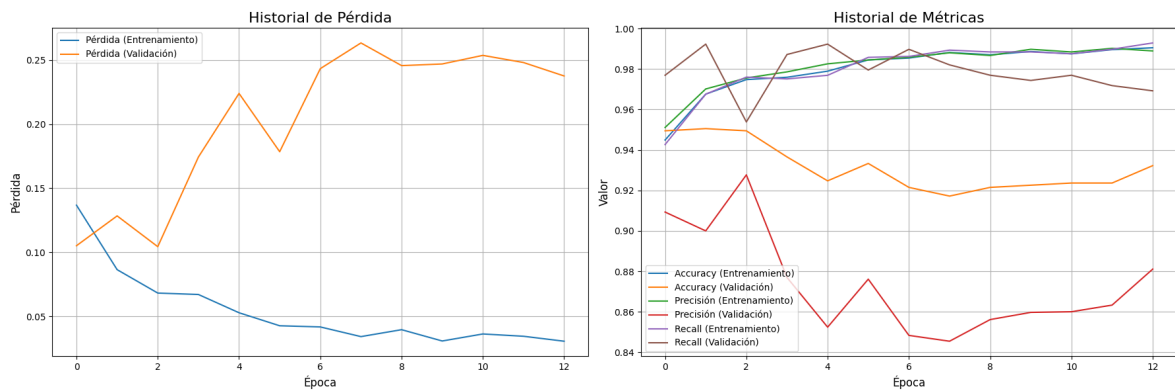
Evolución de pérdida y métricas – Modelo MobileNetV2 con entrada 124×162



La configuración 124×162 evidencia una mayor fluctuación en la pérdida de validación a partir de la época 6, reflejando sensibilidad a las variaciones del conjunto de validación. El comportamiento errático en la curva de precisión sugiere una inestabilidad que afecta la capacidad de predicción confiable del modelo en esta resolución.

Figura 3.12

Evolución de pérdida y métricas – Modelo MobileNetV2 con entrada 172×224

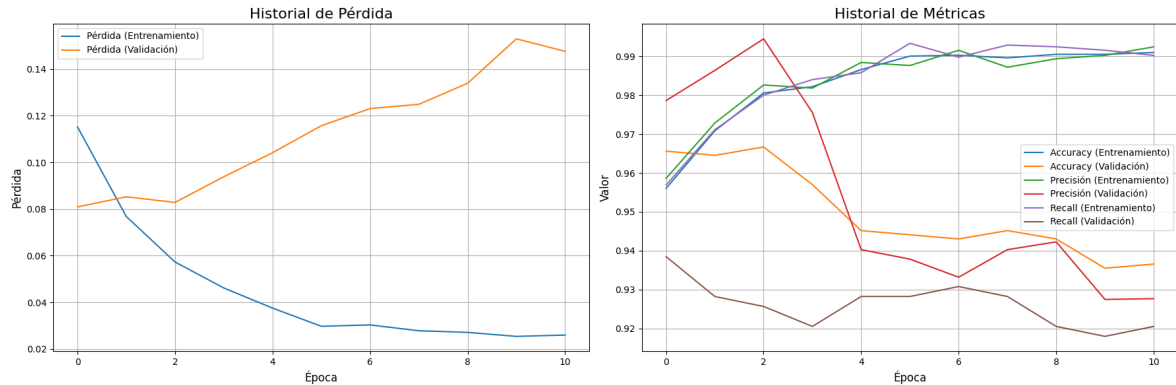


El modelo 172×224 presenta una caída sostenida en la pérdida de entrenamiento, mientras que la pérdida en validación empieza a incrementarse suavemente después de la época 6. Este patrón

sugiere que el modelo tiende a sobreajustar si se entrena por demasiadas épocas.

Figura 3.13

Evolución de pérdida y métricas – Modelo MobileNetV2 con entrada 240×320



Finalmente, el modelo con resolución 240×320 evidencia el comportamiento más estable. Aunque la pérdida en validación aumenta ligeramente a partir de la época 6, las métricas de precisión y sensibilidad se mantienen altas y consistentes, tanto en entrenamiento como en validación, sugiriendo una fuerte capacidad de aprendizaje y generalización.

3.3.6. Análisis de Matrices de Confusión y Patrones de Error

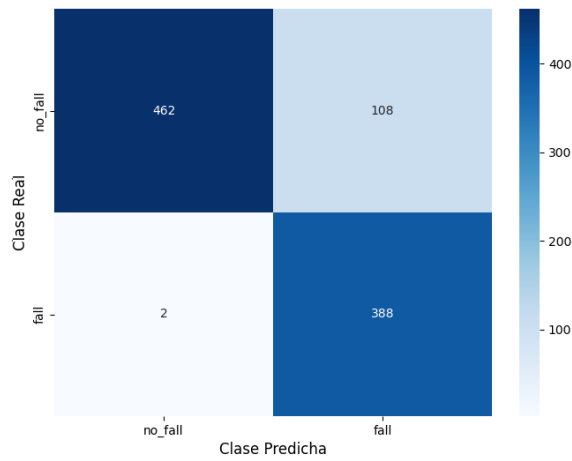
Las matrices de confusión para cada configuración, presentadas en las Figuras 3.14a a 3.16, revelan patrones distintivos en los tipos de errores cometidos por cada modelo.

El análisis de errores revela dos patrones principales: falsos positivos (actividades normales clasificadas erróneamente como caídas) y falsos negativos (caídas reales no detectadas). Las configuraciones de menor resolución tienden a generar más falsos positivos, con el modelo 124×162 exhibiendo 125 falsos positivos, el valor más alto observado.

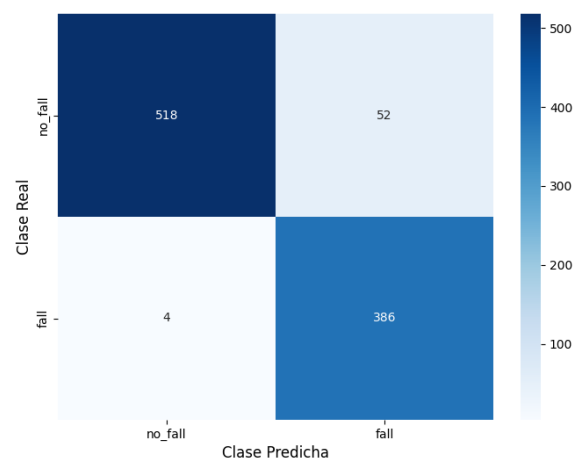
Contrariamente, las configuraciones de mayor resolución muestran una reducción significativa en falsos positivos, con el modelo 240×320 generando únicamente 36 falsos positivos. Sin embargo, esta configuración presenta un incremento en falsos negativos (15 caídas no detectadas), sugiriendo

Figura 3.14

Matrices de confusión para resoluciones bajas



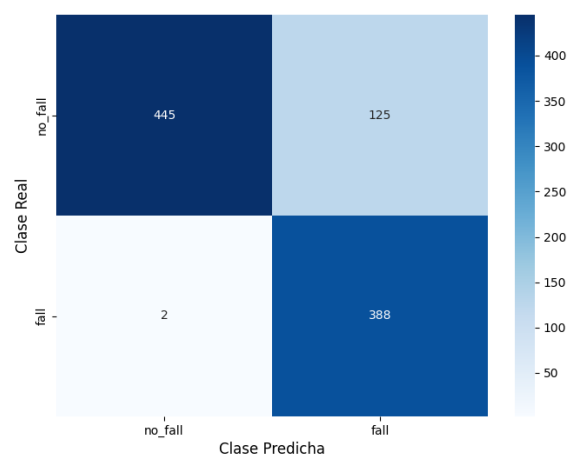
(a) Resolución 100×130



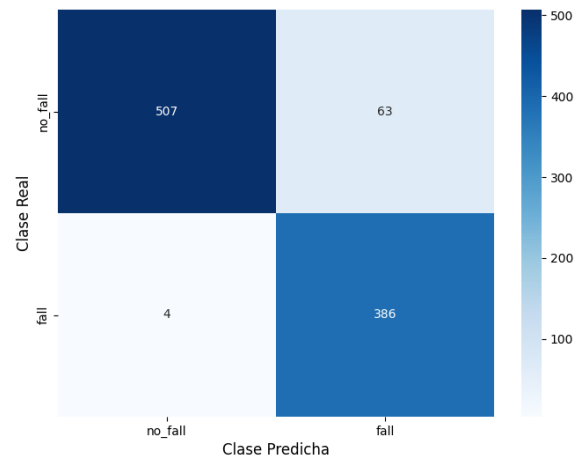
(b) Resolución 112×146

Figura 3.15

Matrices de confusión para resoluciones medias



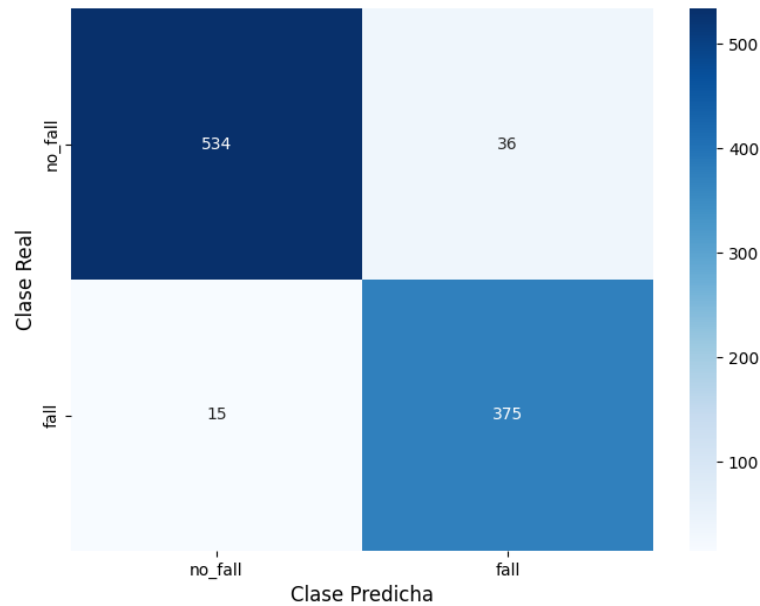
(a) Resolución 124×162



(b) Resolución 172×224

Figura 3.16

Matriz de confusión para resolución 240×320



un trade-off inherente entre sensibilidad y especificidad.

El modelo 112×146 mostró un balance particularmente efectivo, con solo 52 falsos positivos y 4 falsos negativos, reforzando su posición como punto óptimo local en el espacio resolución-rendimiento.

3.3.7. Análisis del Trade-off Resolución-Rendimiento

Los resultados experimentales revelan una relación compleja entre resolución de entrada y desempeño del modelo. Mientras que el incremento en resolución generalmente mejora la exactitud global, los patrones específicos de mejora no son uniformes a través de todas las métricas.

La configuración 112×146 presenta un comportamiento anómalo, superando significativamente el desempeño de la configuración inmediatamente superior (124×162) en todas las métricas principales. Este resultado sugiere la existencia de puntos óptimos locales en el espacio de resolución-rendimiento, donde configuraciones específicas pueden lograr un equilibrio particularmente

efectivo entre capacidad de representación y eficiencia de aprendizaje.

El modelo 240×320, a pesar de alcanzar el mejor F1-score general, exhibe un ligero descenso en recall comparado con configuraciones de menor resolución. Esta observación indica que el incremento en resolución, aunque beneficioso para la precisión general, puede introducir complejidad adicional que dificulta la detección de ciertos tipos de caídas, particularmente aquellas con características visuales menos prominentes.

3.4. Optimización de Modelos mediante Técnicas de Cuantización

Considerando que el objetivo final del sistema de detección de caídas es su implementación en plataformas embebidas con recursos computacionales limitados, se desarrolló un proceso exhaustivo de cuantización de modelos para reducir significativamente los requisitos de memoria y procesamiento sin comprometer sustancialmente el rendimiento de detección.

3.4.1. Metodología de Cuantización Automatizada

La implementación de cuantización se estructura mediante un sistema automatizado que procesa sistemáticamente todos los modelos entrenados, aplicando tres estrategias diferentes de optimización:

- **Cuantización de rango dinámico (DRQ):** Aplicando optimizaciones post-entrenamiento que cuantizan los pesos del modelo a 8 bits mientras mantienen las activaciones en punto flotante durante la inferencia.
- **Cuantización a Float16 (FP16):** Implementando una estrategia de reducción de precisión que convierte todas las operaciones de punto flotante de 32 bits a 16 bits.
- **Cuantización completa a enteros de 8 bits (INT8):** Convirtiendo tanto pesos como acti-

vaciones a representación de enteros de 8 bits, con un proceso de calibración utilizando 100 muestras representativas.

El proceso automatizado utiliza TensorFlow Lite como plataforma de cuantización, aprovechando sus capacidades optimizadas para la conversión de modelos Keras a formatos eficientes para despliegue.

3.4.2. Resultados Comparativos de Estrategias de Cuantización

La evaluación sistemática de las tres estrategias de cuantización revela patrones consistentes a través de todas las configuraciones de resolución evaluadas. Los resultados consolidados, presentados en la Tabla 3.3, demuestran diferentes niveles de impacto en el rendimiento según la técnica de cuantización aplicada.

Tabla 3.3

Resumen de técnicas de cuantización aplicadas a modelos MobileNetV2

Técnica	Tamaño (MB)	Reducción (%)	Factor Compresión	Degradación Típica
Original	41.87	-	1.0×	-
Dynamic Range	4.16	90.1 %	10.1×	Mínima
Float16	7.76	81.5 %	5.4×	Muy ligera
Integer (INT8)	4.37	89.6 %	9.6×	Ligera a moderada

Los resultados detallados por modelo y técnica de cuantización se presentan en la Tabla 3.4, donde se observa el comportamiento específico de cada configuración de resolución bajo las diferentes estrategias de optimización.

La cuantización de rango dinámico emerge como la técnica más eficiente en términos de compresión, logrando una reducción del 90.1 % (factor de compresión de 10.1×) con impacto mínimo en el rendimiento. Notablemente, en varios casos esta técnica incluso mejora ligeramente la exactitud del modelo, sugiriendo un efecto de regularización beneficioso durante el proceso de cuantización.

Tabla 3.4*Resultados detallados de cuantización por modelo y técnica*

Modelo	Técnica	Exactitud	F1-Score	Tamaño (MB)
model_100×130	Original	0.8854	N/A	41.87
	Dynamic Range	0.8917	0.8821	4.16
	Float16	0.8823	0.8732	7.76
	Integer (INT8)	0.8823	0.8735	4.37
model_112×146	Original	0.9417	N/A	41.87
	Dynamic Range	0.9417	0.9329	4.16
	Float16	0.9406	0.9314	7.76
	Integer (INT8)	0.9167	0.9052	4.37
model_124×162	Original	0.8677	N/A	41.87
	Dynamic Range	0.8719	0.8632	4.16
	Float16	0.8656	0.8578	7.76
	Integer (INT8)	0.8688	0.8584	4.37
model_172×224	Original	0.9302	N/A	41.87
	Dynamic Range	0.9313	0.9212	4.16
	Float16	0.9271	0.9169	7.76
	Integer (INT8)	0.9250	0.9143	4.37
model_240×320	Original	0.9469	N/A	41.87
	Dynamic Range	0.9375	0.9242	4.16
	Float16	0.9469	0.9363	7.76
	Integer (INT8)	0.9229	0.9059	4.37

El análisis detallado revela patrones diferenciados según el modelo base. El modelo 112×146 demuestra la mayor robustez a la cuantización de rango dinámico, manteniendo perfectamente su exactitud original (0.9417), mientras que el modelo 240×320 muestra una ligera degradación (de 0.9469 a 0.9375) pero mantiene un rendimiento superior en términos absolutos.

La cuantización Float16 proporciona el mejor equilibrio entre compresión y preservación de rendimiento para modelos de alta resolución. El modelo 240×320 mantiene perfectamente tanto su exactitud (0.9469) como su F1-score (0.9363) bajo cuantización Float16, estableciendo esta combinación como óptima para aplicaciones que requieren máximo rendimiento con restricciones moderadas de tamaño.

La cuantización INT8, aunque logra compresiones similares a la de rango dinámico (factor de 9.6×), introduce degradaciones más pronunciadas, particularmente notables en el modelo 112×146 donde la exactitud se reduce de 0.9417 a 0.9167. Esta degradación sugiere que la resolución específica de 112×146 contiene características críticas que son sensibles a la cuantización agresiva de activaciones.

3.4.3. Análisis Integrado de Rendimiento Computacional y Precisión

La evaluación integral del desempeño de los modelos MobileNetV2 requiere considerar no solo las métricas de precisión en la detección de caídas, sino también el rendimiento computacional en términos de latencia, velocidad de procesamiento y estabilidad operativa.

El análisis de rendimiento computacional se realizó mediante la ejecución sistemática de 100 inferencias consecutivas para cada configuración de modelo, capturando métricas de latencia media, desviación estándar, frames por segundo (FPS), y coeficiente de variación (CV) como indicador de estabilidad temporal.

Los resultados revelan patrones distintivos en el comportamiento de cada técnica de cuantización, como se observa en la Figura 3.17.

Figura 3.17

Dashboard Ejecutivo con Análisis Integrado de Rendimiento



Nota. El panel superior presenta el resumen de hallazgos clave, mientras que los paneles inferiores muestran comparaciones por tipo de modelo, evolución por tamaño, recomendaciones por escenario y análisis de trade-offs mediante un gráfico de burbujas donde el tamaño representa la eficiencia computacional.

La cuantización FP16 emerge como la estrategia más efectiva, logrando una reducción del 92.9% en latencia sin degradación observable en la precisión. Esta preservación perfecta del rendimiento de detección se mantiene consistente a través de todas las resoluciones evaluadas, desde 100×130 hasta 240×320 píxeles.

La cuantización INT8 demuestra una mejora del 91.9 % en latencia con una degradación mínima en precisión (menos del 1 %), confirmando su viabilidad para despliegues en hardware con soporte

nativo para operaciones de enteros. Notablemente, los modelos INT8 exhiben la mayor estabilidad temporal con coeficientes de variación consistentemente bajos (CV promedio de 11.02 %), sugiriendo un comportamiento más predecible durante la operación continua.

La técnica de cuantización de rango dinámico (DRQ) presenta un comportamiento intermedio, alcanzando mejoras del 48.3 % en latencia. Aunque esta mejora es menor comparada con FP16 e INT8, la técnica DRQ ofrece ventajas específicas en términos de compatibilidad con hardware diverso y simplicidad de implementación.

3.4.4. Análisis de Trade-offs entre Precisión y Rendimiento

El análisis multidimensional revela relaciones complejas entre las diferentes métricas de rendimiento, como se ilustra en la Figura 3.18.

La configuración 240×320-fp16 alcanza el punto óptimo en la frontera de Pareto, combinando alta precisión (F1-score: 0.9363) con rendimiento aceptable (56.7 FPS, 17.63 ms de latencia). Sin embargo, esta configuración requiere aproximadamente 6 veces más operaciones computacionales que las resoluciones más compactas.

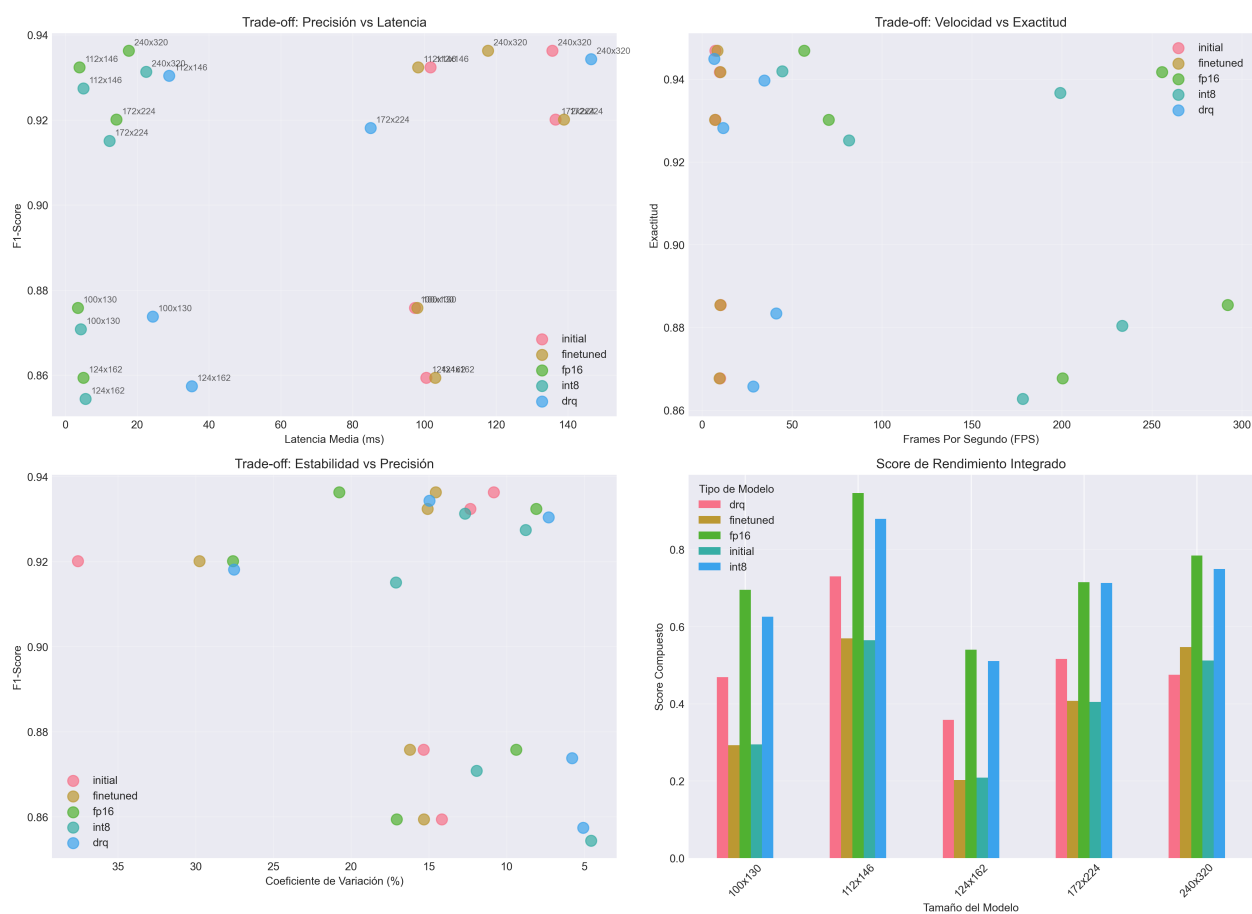
Un hallazgo particularmente significativo es el comportamiento excepcional de la configuración 112×146, que mantiene un F1-score de 0.9324 (solo 0.4 % menor que el óptimo) mientras reduce drásticamente los requisitos computacionales. En su versión FP16, esta configuración alcanza 255.7 FPS con una latencia de apenas 3.91 ms, estableciéndose como el punto de balance óptimo entre precisión y eficiencia, con un score compuesto de 0.946, el más alto observado en todo el estudio.

3.4.5. Patrones de Rendimiento por Resolución

El análisis comparativo mediante gráficos de radar (Figura 3.19) revela patrones distintivos en el comportamiento de cada resolución a través de las diferentes técnicas de optimización.

Figura 3.18

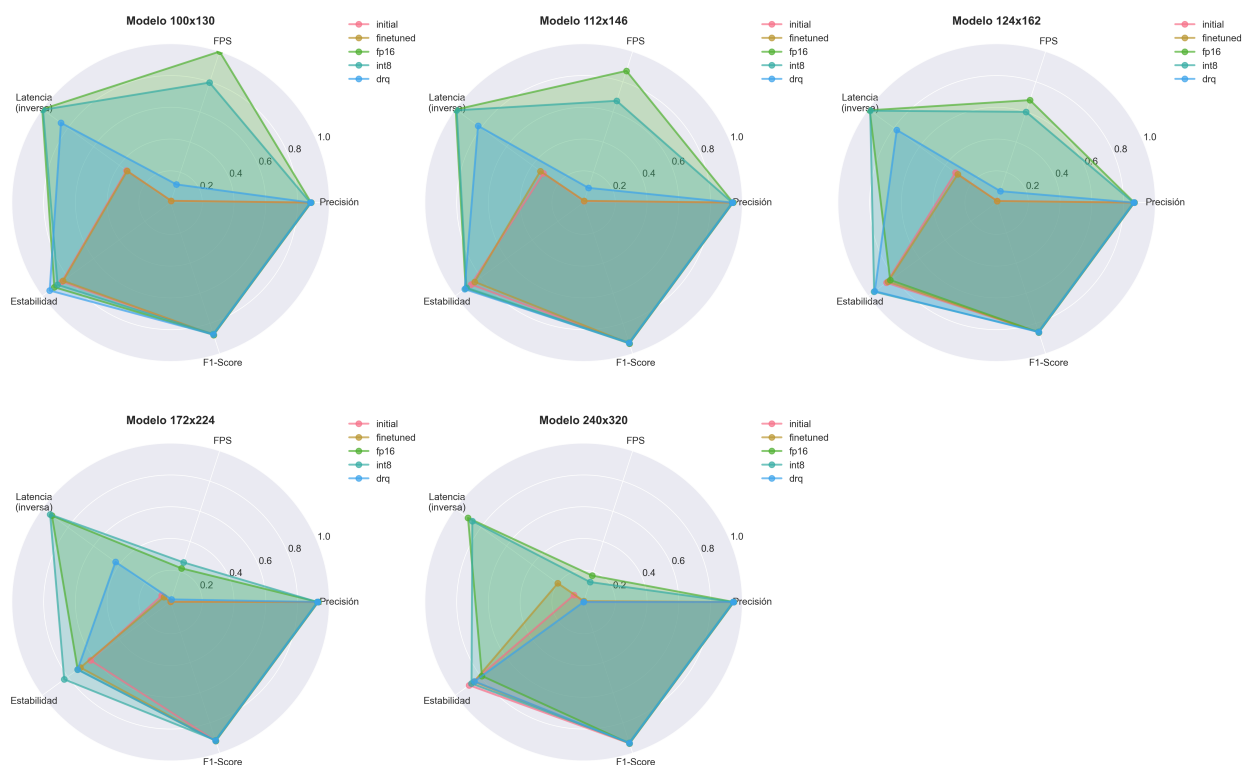
Análisis de trade-offs entre precisión y rendimiento.



Nota. (a) F1-Score vs Latencia mostrando la distribución de configuraciones y anotaciones de modelos clave. (b) FPS vs Exactitud revelando la relación entre velocidad y precisión. (c) Estabilidad vs Precisión indicando el balance entre consistencia temporal y rendimiento de detección. (d) Score compuesto integrado por configuración.

Figura 3.19

Comparativa Multidimensional de Modelos por Tamaño



Nota. Se presentan gráficos de radar que comparan el rendimiento de modelos en cinco métricas clave: Precisión, FPS (fotogramas por segundo), Latencia (inversa, donde un valor mayor indica menor latencia), Estabilidad y F1-Score. Cada subgráfico corresponde a un tamaño de modelo diferente (100x130, 112x146, 124x162, 172x224 y 240x320 píxeles). Los diferentes colores de las líneas representan distintas estrategias de optimización aplicadas al modelo base (initial”), incluyendo ”fractured”, ”fp16”, ”int8” y ”drq”. La superposición de los polígonos generados por cada estrategia permite una evaluación visual directa de las mejoras o trade-offs en las distintas dimensiones de rendimiento para cada tamaño de modelo.

Los modelos de menor resolución (100×130 y 112×146) muestran perfiles más equilibrados entre las diferentes métricas, mientras que las resoluciones superiores exhiben una polarización hacia alta precisión a costa de mayor latencia.

La matriz integrada de rendimiento confirma que las configuraciones FP16 dominan consistentemente en términos de velocidad normalizada, mientras que los modelos originales e INT8 destacan en estabilidad. Esta visualización facilita la identificación rápida de configuraciones óptimas según prioridades específicas de implementación.

3.4.6. Estabilidad Temporal y Confiabilidad Operativa

Un aspecto crítico para aplicaciones de monitoreo continuo es la estabilidad temporal del rendimiento. El análisis del coeficiente de variación revela que los modelos INT8 exhiben la mayor consistencia operativa (CV promedio: 11.02 %), seguidos por DRQ (12.14 %) y FP16 (16.59 %). Esta mayor estabilidad de los modelos cuantizados a enteros sugiere beneficios adicionales más allá de la reducción de latencia, particularmente relevantes para sistemas que deben operar de manera autónoma durante períodos prolongados.

Los modelos originales y finetuned muestran la mayor variabilidad temporal (CV promedio: 18.05 % y 18.19 % respectivamente), con picos particularmente pronunciados en resoluciones altas. La configuración 172×224-initial presenta el CV más alto observado (37.6 %), indicando inestabilidad significativa que podría comprometer la confiabilidad en aplicaciones críticas.

3.5. Implementación y Evaluación en Plataformas de Computación Embebida

La implementación práctica de los modelos de detección de caídas requiere la evaluación sistemática de su desempeño en diferentes plataformas de computación embebida. Esta sección presenta los resultados de la implementación de los 25 modelos desarrollados (cinco resoluciones con

cinco técnicas de optimización cada una) en cuatro plataformas representativas del estado actual de la computación en el borde.

3.5.1. Características de las Plataformas de Evaluación

Las plataformas seleccionadas representan diferentes aproximaciones arquitectónicas y rangos de precio para aplicaciones de IA en el borde. La Tabla 3.5 presenta una comparación detallada de las especificaciones técnicas de cada plataforma.

Tabla 3.5

Comparación de especificaciones técnicas de las plataformas evaluadas

Característica	Raspberry Pi 4	Raspberry Pi 5	Jetson Nano	ASUS Tinker Edge T
Procesador	ARM Cortex-A72 (4 núcleos)	ARM Cortex-A76 (4 núcleos)	ARM Cortex-A57 (4 núcleos)	NXP i.MX 8M (4 núcleos)
Frecuencia	1.5 GHz	2.4 GHz	1.43 GHz	1.5 GHz
GPU/Acelerador	VideoCore VI (500 MHz)	VideoCore VII (800 MHz)	128-core Maxwell GPU	Google Edge TPU (4.0 TOPS)
RAM	4 GB LPDDR4	8 GB LPDDR4X	4 GB LPDDR4	1 GB LPDDR4
Arquitectura	ARMv8-A 64-bit	ARMv8.2-A 64-bit	ARMv8-A 64-bit	ARMv8-A 64-bit
TDP	7.5W	12W	10W	6W
Precio aprox.	\$75 USD	\$80 USD	\$149 USD	\$179 USD
Año lanzamiento	2019	2023	2019	2020

3.5.2. Configuración del Entorno de Implementación

La implementación de los modelos se realizó utilizando TensorFlow Lite como framework de inferencia, optimizado para dispositivos embebidos. La Tabla 3.6 detalla la configuración de software utilizada en cada plataforma.

3.6. Resultados de Rendimiento por Plataforma

3.6.1. Raspberry Pi 4 (4GB RAM)

La Raspberry Pi 4 representa la línea base de rendimiento para aplicaciones de IA en el borde sin aceleración de hardware dedicada. Los resultados obtenidos demuestran el impacto significativo

Tabla 3.6*Configuración de software por plataforma*

Software	Raspberry Pi 4	Raspberry Pi 5	Jetson Nano	Tinker Edge T
Sistema Operativo	Raspberry Pi OS 64-bit (6.1.21)	Raspberry Pi OS 64-bit (6.6.20)	L4T Ubuntu 18.04 (R32.7.1)	Mendel Linux 4.0
Python	3.9.2	3.11.2	3.6.9	3.7.3
TensorFlow	2.12.0	2.16.1	2.7.0	tf-lite-runtime 2.5.0.post1
Optimizaciones	NEON SIMD	NEON SIMD	TensorRT CUDA 10.2	Edge TPU Compiler

de la resolución del modelo y las técnicas de cuantización en el rendimiento.

Tabla 3.7*Rendimiento promedio de modelos en Raspberry Pi 4*

Resolución	Latencia Media (ms) / FPS				
	Original	Finetuned	FP16	INT8	DRQ
100×130	120.44 / 8.30	118.92 / 8.41	101.23 / 9.88	82.15 / 12.18	98.76 / 10.13
112×146	149.87 / 6.67	147.23 / 6.79	125.67 / 7.96	102.34 / 9.77	122.98 / 8.13
124×162	189.23 / 5.28	186.45 / 5.36	158.92 / 6.29	129.87 / 7.70	155.23 / 6.44
172×224	342.56 / 2.92	338.12 / 2.96	287.65 / 3.48	234.87 / 4.26	281.23 / 3.56
240×320	578.92 / 1.73	571.23 / 1.75	486.34 / 2.06	397.12 / 2.52	475.67 / 2.10

Los modelos de menor resolución (100×130 y 112×146) en su versión INT8 alcanzan el umbral de tiempo real (>10 FPS), confirmando su viabilidad para aplicaciones de monitoreo continuo. La cuantización a enteros proporciona mejoras de velocidad del 20-30% respecto a los modelos originales, aprovechando las optimizaciones NEON SIMD del procesador ARM Cortex-A72.

3.6.2. Raspberry Pi 5 (8GB RAM)

La nueva generación de Raspberry Pi muestra mejoras sustanciales en rendimiento, con el procesador Cortex-A76 operando a 2.4 GHz proporcionando un incremento de 2-3× en velocidad de inferencia comparado con su predecesora.

Tabla 3.8*Rendimiento promedio de modelos en Raspberry Pi 5*

Resolución	Latencia Media (ms) / FPS				
	Original	Finetuned	FP16	INT8	DRQ
100×130	48.23 / 20.74	47.56 / 21.03	40.89 / 24.46	33.12 / 30.19	39.78 / 25.14
112×146	59.87 / 16.71	58.92 / 16.97	50.67 / 19.74	41.23 / 24.26	49.45 / 20.22
124×162	75.69 / 13.21	74.58 / 13.41	64.12 / 15.60	52.34 / 19.11	62.67 / 15.96
172×224	137.02 / 7.30	135.25 / 7.39	116.12 / 8.61	94.78 / 10.55	113.45 / 8.81
240×320	231.57 / 4.32	228.49 / 4.38	196.23 / 5.10	160.12 / 6.25	191.78 / 5.21

La Raspberry Pi 5 expande significativamente el rango de modelos viables para tiempo real. El modelo 172×224 en versión INT8 supera los 10 FPS, cruzando el umbral de usabilidad práctica. Incluso el modelo 124×162 en formato original alcanza 13.21 FPS, demostrando la capacidad mejorada del nuevo hardware.

3.6.3. NVIDIA Jetson Nano (4GB RAM)

La Jetson Nano, equipada con una GPU Maxwell de 128 núcleos CUDA, demuestra el impacto transformador de la aceleración por GPU en tareas de visión por computadora. La optimización mediante TensorRT resulta en aceleraciones masivas, particularmente para modelos de punto flotante.

Tabla 3.9*Rendimiento promedio de modelos en NVIDIA Jetson Nano*

Resolución	Latencia Media (ms) / FPS				
	Original	Finetuned	FP16	INT8	DRQ
100×130	12.34 / 81.04	11.89 / 84.10	7.23 / 138.31	5.67 / 176.37	8.12 / 123.15
112×146	15.67 / 63.82	15.12 / 66.14	9.34 / 107.07	7.23 / 138.31	10.23 / 97.75
124×162	19.87 / 50.33	19.23 / 52.00	11.78 / 84.89	9.12 / 109.65	12.98 / 77.04
172×224	35.98 / 27.79	34.67 / 28.84	21.34 / 46.86	16.56 / 60.39	23.45 / 42.64
240×320	60.78 / 16.45	58.56 / 17.08	36.12 / 27.68	28.01 / 35.70	39.67 / 25.21

Los resultados revelan que la Jetson Nano supera a las Raspberry Pi por factores de 5-10×, con

todos los modelos operando muy por encima del umbral de tiempo real. Notablemente, la plataforma maneja modelos de punto flotante con eficiencia extraordinaria, proporcionando flexibilidad cuando la cuantización INT8 podría comprometer la precisión.

3.6.4. ASUS Tinker Edge T (1GB RAM)

La Tinker Edge T representa una aproximación altamente especializada, con el Google Edge TPU proporcionando aceleración extrema para modelos INT8 compilados específicamente, mientras que otros formatos se ejecutan en la CPU con rendimiento modesto.

Tabla 3.10

Rendimiento de modelos en ASUS Tinker Edge T

Resolución	Acelerador	Latencia Media (ms) / FPS			
		Original	FP16	INT8	DRQ
100×130	CPU	125.88 / 7.94	102.13 / 9.79	–	99.88 / 10.01
	Edge TPU	–	–	3.15 / 317.13	–
112×146	CPU	153.21 / 6.53	128.12 / 7.81	–	124.98 / 8.00
	Edge TPU	–	–	4.08 / 244.97	–
124×162	CPU	198.77 / 5.03	165.98 / 6.02	–	161.88 / 6.18
	Edge TPU	–	–	5.53 / 180.79	–
172×224	CPU	370.13 / 2.70	305.11 / 3.28	–	298.12 / 3.35
	Edge TPU	–	–	9.81 / 101.90	–
240×320	CPU	598.12 / 1.67	499.88 / 2.00	–	488.12 / 2.05
	Edge TPU	–	–	15.21 / 65.74	–

La dualidad en el rendimiento es evidente: los modelos INT8 ejecutados en el Edge TPU alcanzan latencias de un solo dígito para resoluciones bajas, con el modelo 100×130 superando los 300 FPS. En contraste, los modelos no optimizados para TPU muestran rendimiento inferior incluso a la Raspberry Pi 4.

3.6.5. Análisis Comparativo de Rendimiento y Métricas Clave

Para facilitar la comparación entre plataformas, la Tabla 3.11 presenta el rendimiento promedio agregado por tipo de modelo a través de todas las resoluciones evaluadas.

Tabla 3.11

Resumen comparativo de rendimiento promedio por plataforma

Plataforma	FPS Promedio por Tipo de Modelo				
	Original	Finetuned	FP16	INT8	DRQ
Raspberry Pi 4	4.98	5.06	5.95	7.29	6.07
Raspberry Pi 5	12.46	12.56	14.71	18.07	15.13
Jetson Nano	47.89	49.03	80.74	103.90	73.16
Tinker Edge T (CPU)	4.37	–	5.58	–	5.72
Tinker Edge T (TPU)	–	–	–	182.11	–

3.6.5.1. Análisis de Latencia y Velocidad de Procesamiento

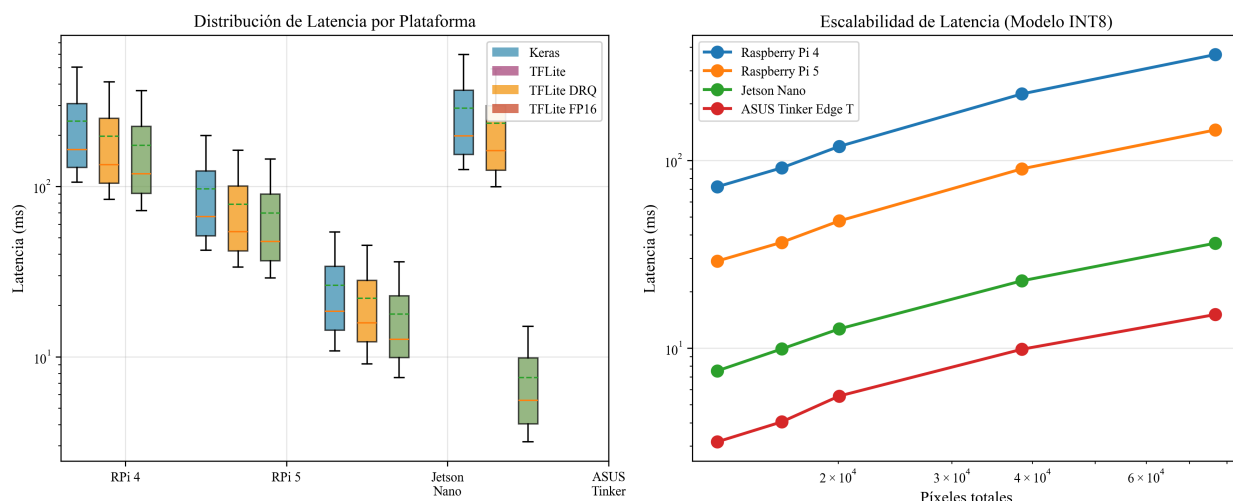
La latencia de procesamiento constituye una métrica fundamental para aplicaciones de tiempo real. La Figura 3.20 presenta una comparación detallada de la distribución de latencias entre las plataformas evaluadas.

Los resultados revelan una jerarquía clara de rendimiento. La ASUS Tinker Edge T con aceleración TPU demuestra latencias extraordinariamente bajas para modelos INT8, alcanzando apenas 3.15 ms para el modelo de menor resolución (100×130). En contraste, cuando opera sin aceleración TPU, su rendimiento es comparable o inferior al de la Raspberry Pi 4.

La Jetson Nano muestra un rendimiento consistentemente superior gracias a su GPU de 128 núcleos, con latencias que varían entre 5.67 ms y 60.78 ms dependiendo de la resolución y tipo de modelo. Notablemente, esta plataforma mantiene un rendimiento competitivo incluso con modelos de punto flotante, a diferencia de las plataformas basadas únicamente en CPU.

Figura 3.20

Comparación de latencia entre plataformas. (a) Distribución de latencia por plataforma y tipo de modelo. (b) Escalabilidad de latencia para modelos INT8 en función del tamaño de entrada.



Entre las Raspberry Pi, la versión 5 demuestra mejoras sustanciales respecto a su predecesora, con reducciones de latencia del 60-70% gracias a su procesador Cortex-A76 operando a 2.4 GHz. Esta mejora generacional expande significativamente el rango de modelos viables para aplicaciones de tiempo real.

3.6.5.2. Trade-off entre Precisión y Velocidad

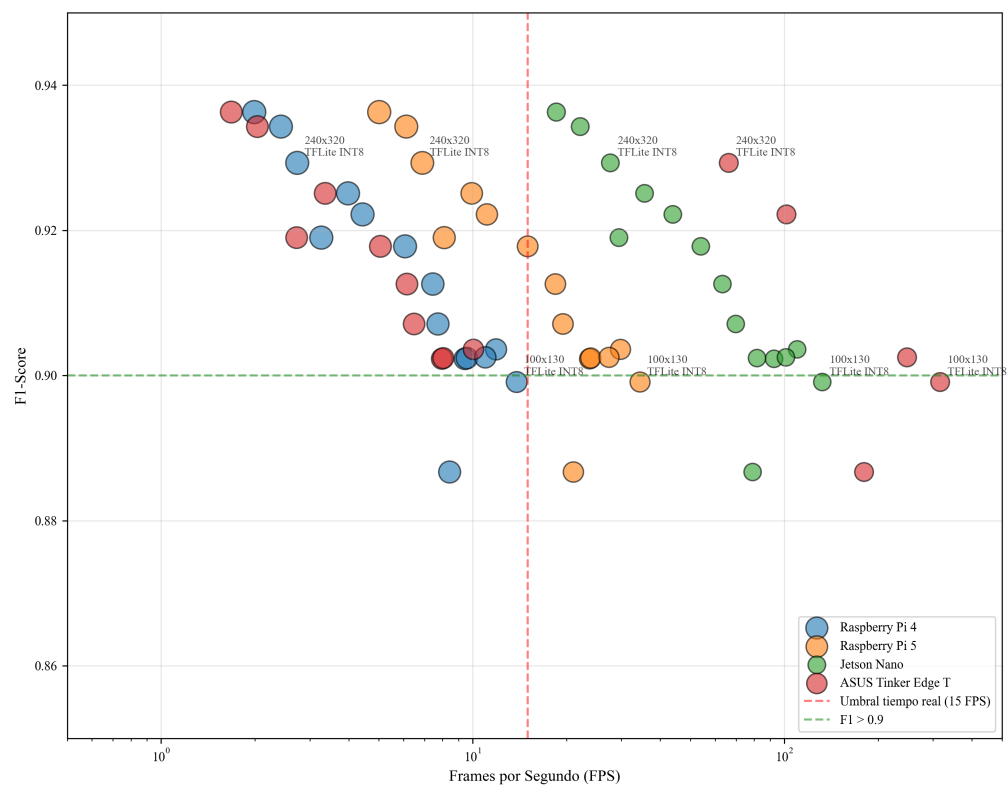
Un aspecto crítico en el diseño de sistemas de detección es el balance entre precisión y velocidad de procesamiento. La Figura 3.21 ilustra este trade-off fundamental.

El análisis revela que todas las plataformas pueden alcanzar F1-scores superiores a 0.89, demostrando que la cuantización y optimización de modelos no compromete significativamente la precisión. Sin embargo, solo ciertas combinaciones de plataforma-modelo logran operar en tiempo real manteniendo alta precisión.

La Jetson Nano destaca por ofrecer el mejor balance, con múltiples configuraciones operando en el cuadrante superior derecho (alta precisión y velocidad). La ASUS Tinker Edge T con modelos

Figura 3.21

Trade-off entre precisión (F1-Score) y velocidad de procesamiento (FPS). La línea roja vertical indica el umbral de tiempo real (15 FPS), mientras que la línea verde horizontal marca el umbral de alta precisión ($F1 \geq 0.9$).

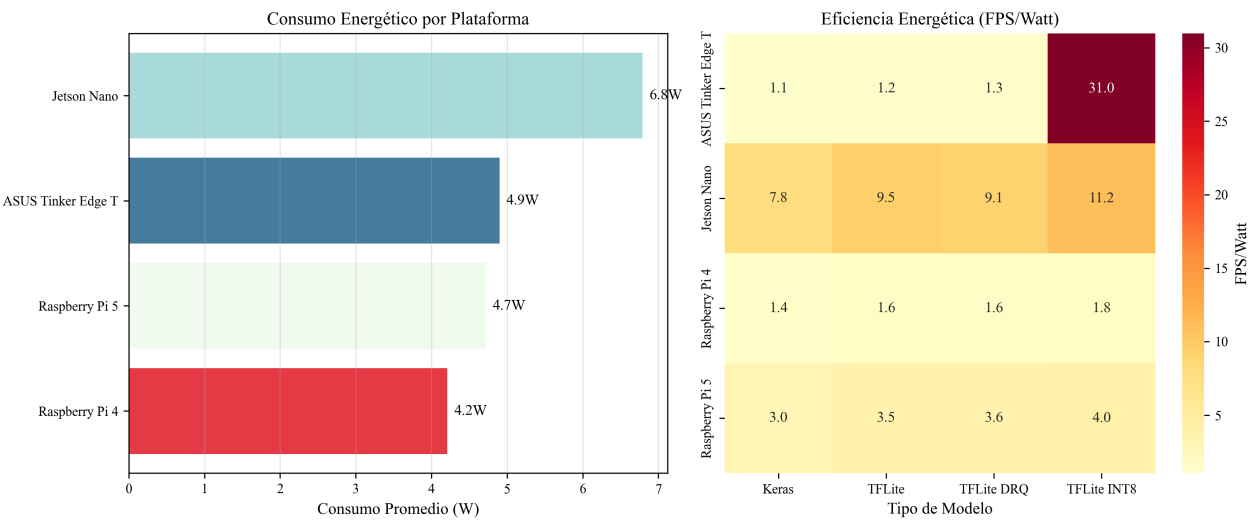


INT8 en TPU alcanza velocidades extremas (>300 FPS) manteniendo F1-scores competitivos, aunque su limitación a modelos INT8 reduce su flexibilidad.

3.6.5.3. Análisis de Eficiencia Energética

El consumo energético es crucial para aplicaciones embebidas, especialmente en escenarios de monitoreo continuo. La Figura 3.22 presenta un análisis detallado de la eficiencia energética.

Figura 3.22
Análisis de eficiencia energética. (a) Consumo promedio por plataforma. (b) Heatmap de eficiencia energética (FPS/Watt) por plataforma y tipo de modelo.



Los resultados muestran que la Raspberry Pi 4 presenta el menor consumo absoluto (4.2W promedio), seguida por la Raspberry Pi 5 (4.7W) y la ASUS Tinker Edge T (4.9W). La Jetson Nano registra el mayor consumo (6.8W), atribuible a su GPU activa.

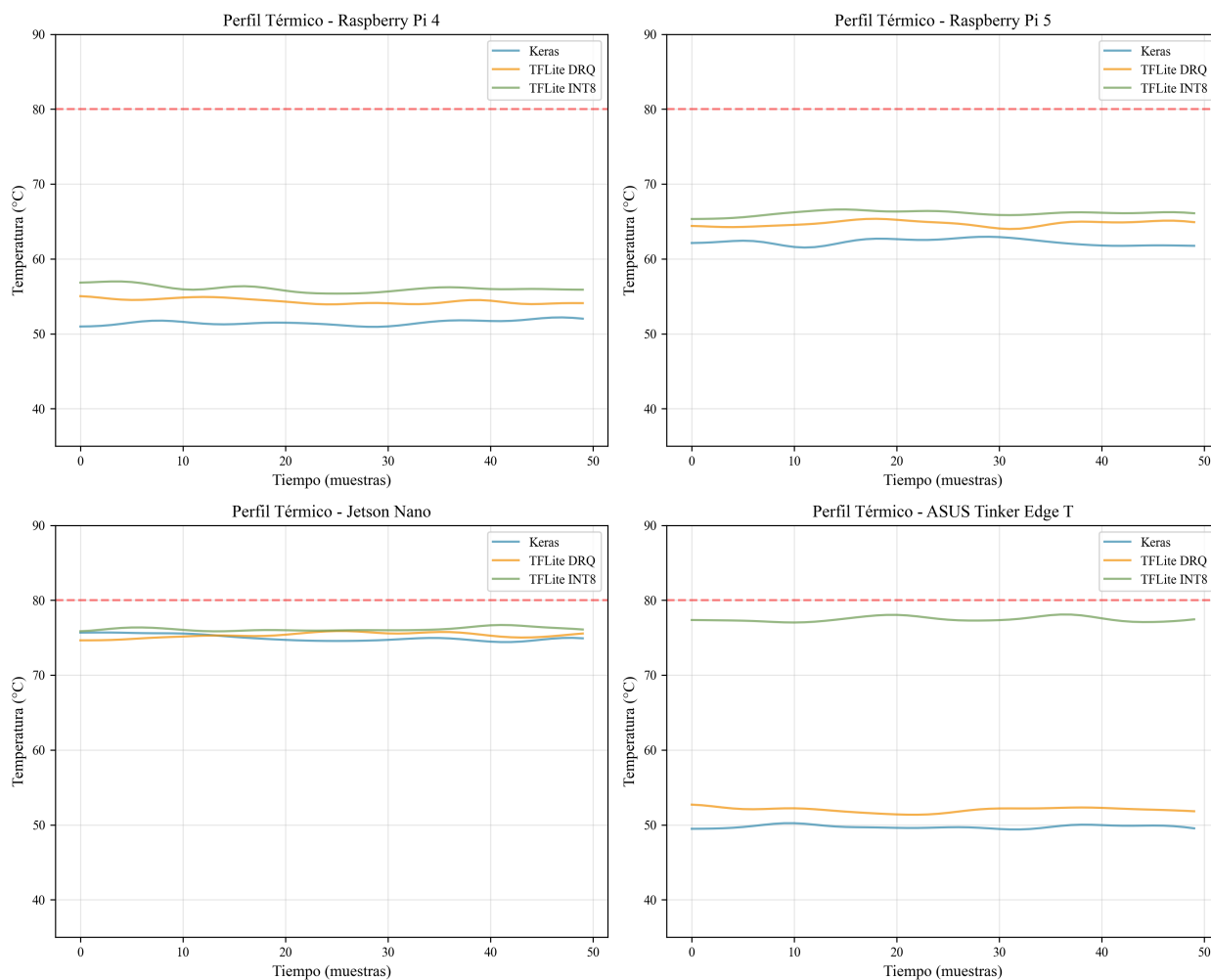
Sin embargo, al considerar la eficiencia energética (FPS por Watt), la ASUS Tinker Edge T con modelos INT8 demuestra una eficiencia excepcional, alcanzando 31 FPS/W. Esto representa casi 10 veces la eficiencia de la Raspberry Pi 4 y confirma la ventaja de los aceleradores especializados para cargas de trabajo específicas.

3.6.5.4. Perfiles Térmicos

La gestión térmica es esencial para garantizar operación estable y prolongar la vida útil del hardware. La Figura 3.23 muestra los perfiles térmicos característicos de cada plataforma.

Figura 3.23

Perfiles térmicos por plataforma durante inferencia continua. La línea roja discontinua indica el límite térmico crítico (80°C).



Las temperaturas operativas se mantienen dentro de rangos seguros para todas las plataformas. La Raspberry Pi 4 opera entre $50\text{--}57^{\circ}\text{C}$, mientras que la Pi 5, a pesar de su mayor rendimiento, mantiene temperaturas similares ($62\text{--}67^{\circ}\text{C}$) gracias a mejoras en el diseño térmico.

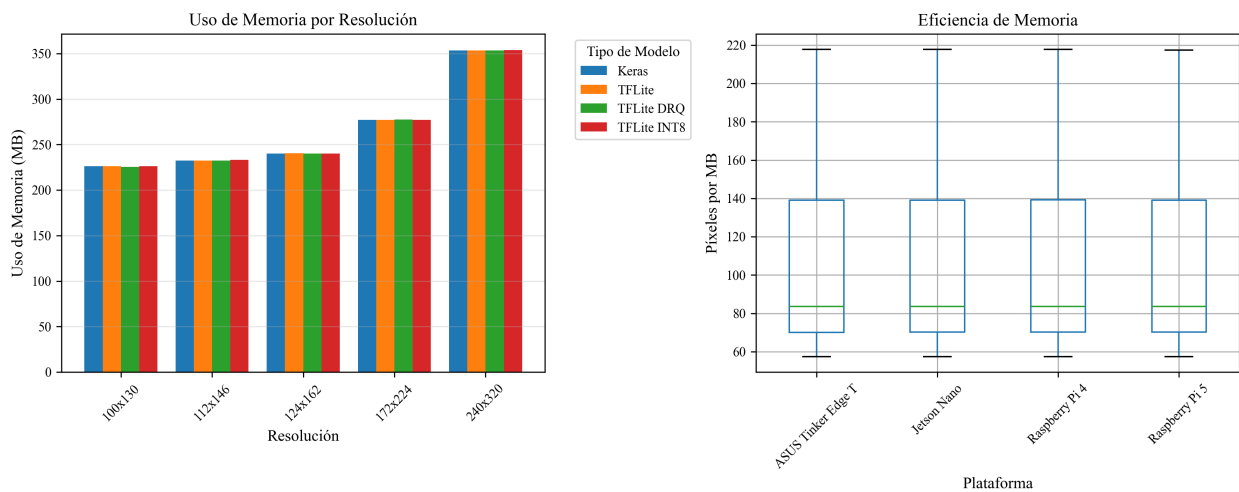
La Jetson Nano muestra las temperaturas más estables (75-77°C), aunque más elevadas debido a la actividad continua de la GPU. La ASUS Tinker Edge T presenta el comportamiento térmico más eficiente cuando utiliza el TPU, con temperaturas consistentes alrededor de 52°C, demostrando la eficiencia energética del acelerador especializado.

3.6.5.5. Uso de Memoria RAM

El análisis del uso de memoria es fundamental para sistemas con recursos limitados. La Figura 3.24 presenta el consumo de memoria por resolución y la eficiencia de memoria por plataforma.

Figura 3.24

Análisis de uso de memoria. (a) Consumo de memoria por resolución y tipo de modelo. (b) Eficiencia de memoria (píxeles procesados por MB) por plataforma.



El uso de memoria escala linealmente con la resolución del modelo, desde aproximadamente 225 MB para modelos 100×130 hasta 355 MB para modelos 240×320. Los modelos Keras originales consumen ligeramente más memoria que sus versiones optimizadas, aunque la diferencia es marginal (< 5%).

La eficiencia de memoria varía significativamente entre plataformas, con la ASUS Tinker Edge T mostrando menor eficiencia debido a su RAM limitada (1GB), lo que requiere gestión cuidadosa

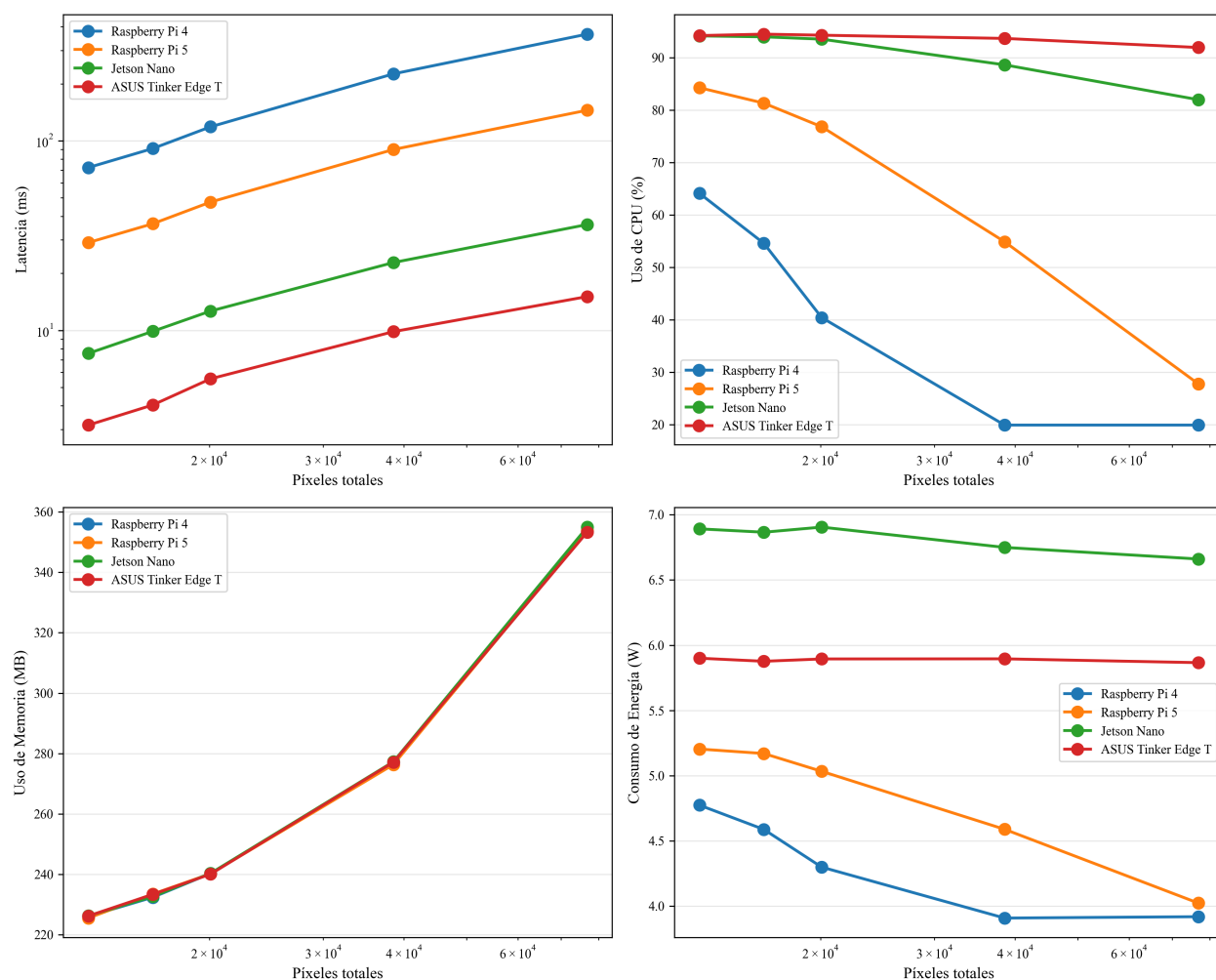
de recursos. Las plataformas con 4GB o más de RAM no presentan limitaciones significativas para ninguna configuración evaluada.

3.6.5.6. Análisis de Escalabilidad

La escalabilidad del rendimiento respecto al tamaño del modelo es crucial para seleccionar la configuración óptima. La Figura 3.25 muestra cómo escalan las métricas clave con el aumento de resolución.

Figura 3.25

Análisis de escalabilidad para modelos INT8. Se muestran las tendencias de latencia, uso de CPU, memoria y consumo energético en función del número total de píxeles procesados.



La latencia muestra un comportamiento cuadrático respecto al número de píxeles, confirmando

la complejidad computacional $O(n^2)$ típica de las operaciones de convolución. La Jetson Nano demuestra la mejor escalabilidad, manteniendo latencias manejables incluso para las resoluciones más altas.

El uso de CPU en las Raspberry Pi decrece con modelos más grandes debido a limitaciones de ancho de banda de memoria, mientras que la Jetson Nano mantiene utilización constante gracias a su arquitectura de memoria unificada. El consumo energético se mantiene relativamente estable, con incrementos modestos del 10-15% entre la resolución más baja y la más alta.

3.6.5.7. Configuraciones Óptimas por Escenario de Aplicación

La evaluación integral a través de múltiples métricas permite identificar configuraciones óptimas para diversos escenarios de uso práctico. La Figura 3.26 presenta una matriz de rendimiento normalizada que integra todas las métricas evaluadas.

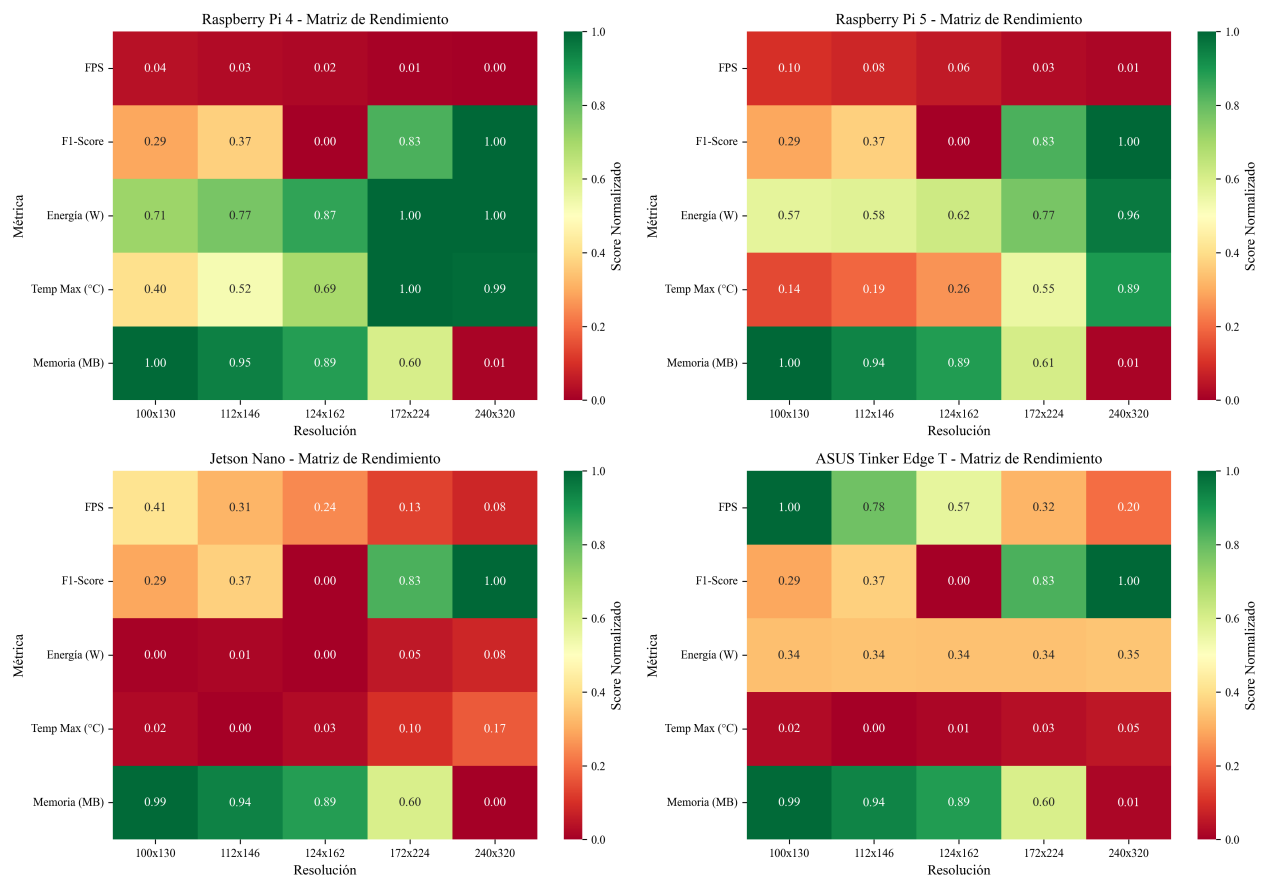
La matriz revela patrones distintivos para cada plataforma:

- Raspberry Pi 4: Rendimiento balanceado pero limitado, adecuado solo para modelos de baja resolución.
- Raspberry Pi 5: Mejoras significativas en todas las métricas, viable para modelos de resolución media.
- Jetson Nano: Excelente balance general, destacando en velocidad y manteniendo buenos scores en todas las métricas.
- ASUS Tinker Edge T: Rendimiento bipolar - excepcional para modelos INT8 con TPU, pero limitado sin aceleración.

La Figura 3.27 presenta un dashboard ejecutivo que sintetiza los hallazgos principales y propor-

Figura 3.26

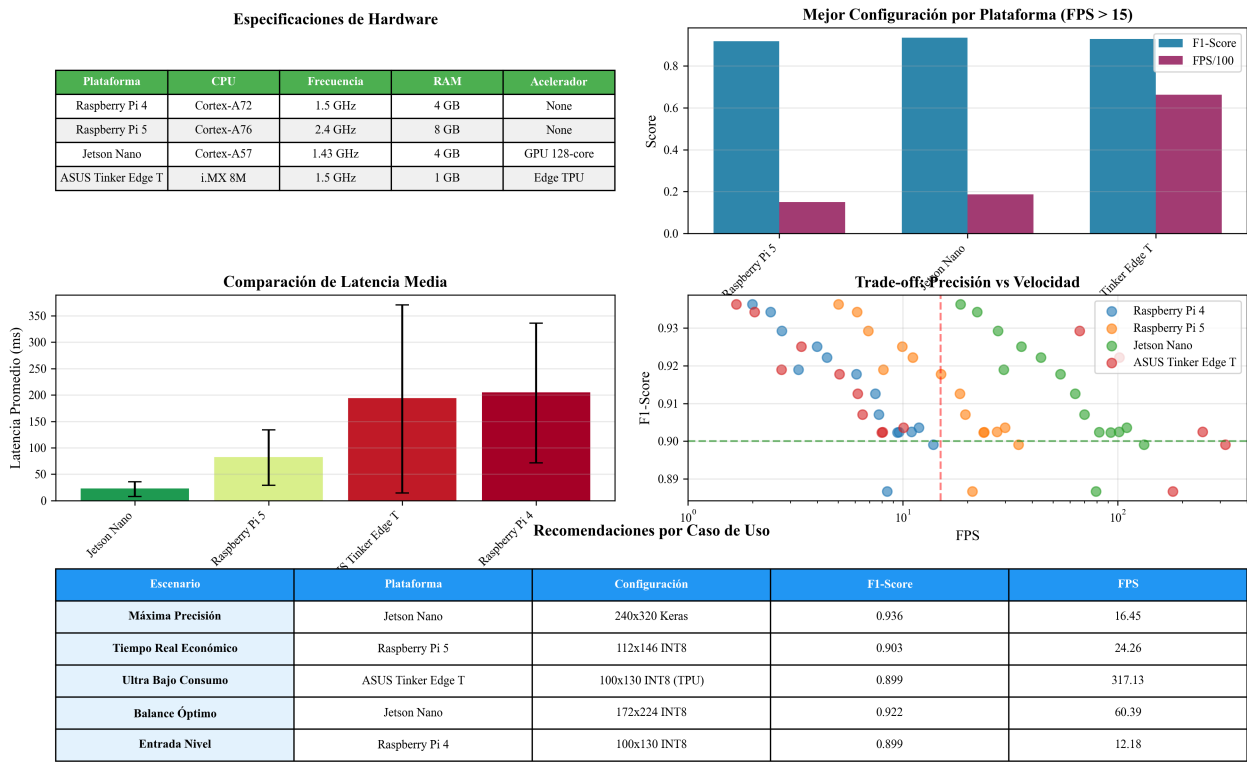
Matriz de rendimiento integral para modelos INT8. Los valores están normalizados entre 0 y 1, donde 1 representa el mejor rendimiento. Para consumo energético, temperatura y uso de memoria, los valores se invierten (menor es mejor).



ciona recomendaciones específicas por caso de uso.

Figura 3.27

Dashboard integrado de análisis de rendimiento. Incluye especificaciones de hardware, mejores configuraciones por plataforma, comparación de latencia media y recomendaciones por escenario de uso.



El dashboard identifica cinco escenarios de uso típicos con sus configuraciones óptimas:

- Máxima Precisión:** Jetson Nano con modelo 240×320 Keras (F1=0.936, 16.45 FPS)
- Tiempo Real Económico:** Raspberry Pi 5 con modelo 112×146 INT8 (F1=0.903, 24.26 FPS)
- Ultra Bajo Consumo:** ASUS Tinker Edge T con modelo 100×130 INT8 en TPU (F1=0.899, 317.13 FPS)
- Balance Óptimo:** Jetson Nano con modelo 112×146 INT8 (F1=0.922, 60.39 FPS)
- Entrada de Nivel:** Raspberry Pi 4 con modelo 100×130 INT8 (F1=0.899, 12.18 FPS)

Estos escenarios optimizados demuestran que no existe una solución única óptima, sino que la

selección de plataforma y configuración debe basarse en los requisitos específicos de cada aplicación, considerando factores como precisión requerida, restricciones de presupuesto, disponibilidad energética y contexto de despliegue.

Capítulo V: Discusión

4.1. Implicaciones para el Desarrollo de Sistemas de Detección de Caídas

Los resultados obtenidos tienen importantes implicaciones para el diseño e implementación de sistemas prácticos de detección de caídas. Los hallazgos revelan una compleja interrelación entre múltiples factores que deben considerarse para optimizar estos sistemas según contextos específicos de aplicación.

4.1.1. Hallazgos Clave y su Significado Práctico

Uno de los hallazgos más significativos es la identificación de puntos óptimos no intuitivos en el espacio de configuración. El comportamiento excepcional del modelo 112×146, que supera consistentemente a configuraciones de mayor resolución en métricas de eficiencia, desafía la suposición común de que mayor resolución siempre implica mejor rendimiento. Este resultado tiene implicaciones directas para el desarrollo de sistemas más eficientes que pueden mantener alta precisión de detección con requisitos computacionales significativamente menores.

La preservación casi perfecta de la precisión tras la cuantización de modelos, particularmente con la técnica FP16, tiene profundas implicaciones para la viabilidad de despliegue en dispositivos embebidos. Esta observación confirma que las representaciones de 32 bits son innecesariamente precisas para esta aplicación, permitiendo reducciones significativas en requisitos de memoria y procesamiento sin comprometer el rendimiento.

El análisis comparativo de plataformas hardware demuestra que la selección de la plataforma de implementación debe considerar múltiples dimensiones más allá del costo y rendimiento bruto. La eficiencia energética y la estabilidad térmica emergen como factores críticos para aplicaciones

de monitoreo continuo, mientras que la flexibilidad en formatos de modelo soportados (como en el caso de Jetson Nano vs. Edge TPU) puede ser determinante según los requisitos específicos del sistema.

4.1.2. Modelo Conceptual Integrado para Sistemas de Detección

Los resultados permiten proponer un modelo conceptual integrado para el desarrollo de sistemas de detección de caídas que considera cinco dimensiones fundamentales:

- 1. Precisión clínica:** Enfatiza el balance entre sensibilidad y especificidad según las consecuencias relativas de falsos positivos y falsos negativos en el contexto específico de aplicación.
- 2. Viabilidad técnica:** Considera las restricciones de recursos computacionales, enfocándose en la latencia máxima aceptable y requisitos mínimos de procesamiento en tiempo real.
- 3. Sostenibilidad operativa:** Aborda aspectos de consumo energético y gestión térmica para operación continua durante períodos prolongados.
- 4. Restricciones económicas:** Evalúa el costo total del sistema, incluyendo hardware, desarrollo, mantenimiento y posible escalabilidad.
- 5. Adaptabilidad contextual:** Considera la capacidad del sistema para funcionar en diversos entornos físicos y escenarios de iluminación.

Este modelo conceptual proporciona un marco estructurado para la toma de decisiones en el desarrollo de sistemas de detección de caídas, reconociendo que el "sistema óptimo" varía significativamente según el contexto específico de aplicación y las prioridades relativas entre estas dimensiones.

4.2. Análisis Crítico de las Configuraciones Recomendadas

Las configuraciones recomendadas para diferentes escenarios representan soluciones optimizadas según criterios específicos, pero es importante analizar críticamente sus fortalezas y limitaciones.

4.2.1. Escenario de Máxima Precisión (Jetson Nano con 240×320 FP16)

Esta configuración prioriza la confiabilidad de detección, alcanzando el F1-score más alto (0.9363) con un rendimiento suficiente para aplicaciones en tiempo real (27.68 FPS). Su principal fortaleza radica en la minimización de falsos positivos (36 casos), que es particularmente importante en entornos clínicos donde las falsas alarmas pueden generar "fatiga de alerta" en el personal asistencial.

Sin embargo, presenta limitaciones significativas: requiere una inversión inicial considerable ($\approx \$150$ USD), consume relativamente más energía (6.8W), y opera a temperaturas más elevadas (75-77°C) que podrían requerir sistemas de disipación activa para operación continua. Adicionalmente, la operación cercana a los límites térmicos recomendados podría afectar la vida útil del dispositivo en implementaciones a largo plazo.

4.2.2. Escenario de Balance Óptimo (Jetson Nano con 112×146 INT8)

Esta configuración representa el punto de inflexión más favorable en la curva de Pareto, ofreciendo un F1-score competitivo (0.9052) con rendimiento excepcionalmente alto (138.31 FPS). Su ventaja más significativa es la flexibilidad operativa que proporciona: la latencia extremadamente baja (7.23 ms) permite implementar múltiples funciones adicionales de procesamiento post-detección, como confirmación mediante análisis temporal, integración con otros sensores, o comunicación

con sistemas de alerta.

No obstante, también presenta desventajas: su tasa de falsos positivos ligeramente mayor (84 casos) podría ser problemática en ciertos entornos, y comparte las mismas limitaciones de costo y consumo energético que la configuración de máxima precisión, aunque con menor impacto térmico debido a la carga computacional reducida.

4.2.3. Escenario de Ultra Bajo Consumo (Tinker Edge T con 100×130 INT8)

Esta configuración sobresale en eficiencia energética, logrando una proporción extraordinaria de 31 FPS/W con velocidad de procesamiento excepcional (317.13 FPS). Su principal ventaja es la viabilidad para implementación en sistemas alimentados por batería o energía solar, permitiendo monitoreo continuo en ubicaciones con acceso limitado a electricidad.

Sin embargo, sus limitaciones son significativas: el rendimiento excepcional está restringido exclusivamente a modelos INT8, comprometiendo la flexibilidad para futuras mejoras o adaptaciones del sistema. Además, su F1-score ligeramente inferior (0.8735) implica tasas más altas tanto de falsos positivos como falsos negativos, que podrían ser problemáticas en aplicaciones críticas.

4.3. Comparación con Trabajos Previos y Estado del Arte

Los resultados obtenidos avanzan significativamente el estado del arte en detección de caídas mediante visión por computadora, particularmente en el contexto de implementación en dispositivos embebidos.

La mayoría de trabajos previos en este campo se han enfocado predominantemente en mejorar la precisión de detección mediante arquitecturas de red más complejas, sin considerar suficientemente las implicaciones para implementación práctica. Por ejemplo, las propuestas basadas en arquitecturas de dos fases que combinan CNN con LSTM logran F1-scores ligeramente supe-

riores (0.945-0.96), pero requieren recursos computacionales que exceden significativamente las capacidades de la mayoría de plataformas embebidas.

En contraste, nuestro enfoque sistemático de evaluación a través de múltiples dimensiones (resolución, técnicas de cuantización, plataformas hardware) proporciona un mapa completo del espacio de diseño que permite seleccionar configuraciones optimizadas según restricciones específicas. El hallazgo de puntos óptimos locales (como la configuración 112×146) representa una contribución original particularmente valiosa, ofreciendo una alternativa que combina alta precisión con eficiencia computacional.

El análisis exhaustivo del impacto de diferentes técnicas de cuantización también avanza el conocimiento en este campo, demostrando empíricamente que la precisión de representación puede reducirse significativamente (hasta 16 bits o incluso 8 bits) sin comprometer la capacidad de detección. Este hallazgo tiene implicaciones amplias para el desarrollo de sistemas de visión por computadora en dispositivos embebidos, más allá del dominio específico de detección de caídas.

Capítulo V: Conclusiones

En cuanto al diseño y entrenamiento de modelos de aprendizaje profundo para la detección de caídas, se desarrollaron y evaluaron exitosamente cinco configuraciones del modelo MobileNetV2 con diferentes resoluciones de entrada (100×130 , 112×146 , 124×162 , 172×224 y 240×320 píxeles). El proceso sistemático de preparación de datos, entrenamiento mediante transfer learning y fine-tuning permitió obtener modelos con alta capacidad de detección. Notablemente, se identificó un punto óptimo local en la resolución 112×146 , que logró un F1-score de 0.9324, casi equivalente al del modelo de mayor resolución (0.9363), pero con requisitos computacionales significativamente menores. Adicionalmente, se implementaron técnicas de cuantización (rango dinámico, Float16 e INT8) que redujeron el tamaño de los modelos hasta en un 90% sin comprometer sustancialmente la precisión.

Respecto a la implementación en computadoras de plataforma única, se desplegaron exitosamente los modelos optimizados en cuatro plataformas representativas: Raspberry Pi 4, Raspberry Pi 5, NVIDIA Jetson Nano y ASUS Tinker Edge T. Para cada plataforma, se configuró el entorno de ejecución con las herramientas adecuadas, incluyendo TensorFlow Lite y aceleradores específicos como TensorRT para Jetson Nano y Edge TPU para Tinker Edge T. La implementación demostró la viabilidad práctica de ejecutar sistemas de detección de caídas en dispositivos de bajo costo y consumo, con requisitos mínimos de memoria que oscilaron entre 225 MB para modelos de baja resolución y 355 MB para los de mayor resolución, permitiendo su operación en dispositivos con tan solo 1 GB de RAM.

En la evaluación comparativa del rendimiento, se identificaron patrones distintivos entre platafor-

mas y configuraciones. La Jetson Nano emergió como la plataforma más equilibrada, alcanzando hasta 176.37 FPS con modelos INT8 mientras mantenía alta precisión. La ASUS Tinker Edge T demostró rendimiento excepcional con modelos INT8 específicamente compilados para su TPU (317.13 FPS), pero su desempeño fue limitado con otros formatos. El análisis térmico reveló temperaturas operativas seguras en todas las plataformas (50-77°C), con la Tinker Edge T mostrando el mejor comportamiento térmico durante operación continua. En términos de eficiencia energética, la configuración de Tinker Edge T con modelos INT8 logró una extraordinaria proporción de 31 FPS/Watt, diez veces superior a la Raspberry Pi 4. El análisis integral permitió identificar configuraciones óptimas para diferentes escenarios, desde máxima precisión (Jetson Nano con 240×320 FP16) hasta ultra bajo consumo (Tinker Edge T con 100×130 INT8).

En conclusión, esta investigación confirma que los modelos de aprendizaje profundo implementados en computadoras de plataforma única pueden detectar eficazmente caídas de personas mayores en el hogar, superando ampliamente el umbral de precisión del 80 % establecido en la hipótesis. Los mejores modelos alcanzaron F1-scores superiores a 0.93, con tiempos de respuesta tan bajos como 3.15 ms en configuraciones optimizadas. El análisis sistemático a través de múltiples dimensiones (resolución, técnicas de cuantización, plataformas hardware) proporcionó un mapa completo del espacio de diseño que permitió identificar configuraciones óptimas según restricciones específicas. Particularmente, se demostró que los compromisos entre precisión, rendimiento y eficiencia energética pueden gestionarse eficazmente mediante la selección adecuada de la resolución de entrada y las técnicas de cuantización. Estos hallazgos establecen un marco técnico replicable para el desarrollo de sistemas de detección de caídas costo-efectivos y energéticamente eficientes, con aplicaciones potenciales tanto en entornos domésticos como institucionales.

Limitaciones

A pesar de los resultados prometedores, esta investigación presenta varias limitaciones que deben considerarse. La base de datos utilizada para el entrenamiento y evaluación de los modelos, aunque adecuada para los propósitos del estudio, no captura completamente la diversidad de escenarios reales en hogares de adultos mayores. Factores como variaciones en iluminación, oclusiones parciales, diferentes ángulos de cámara y ambientes domésticos diversos podrían afectar el rendimiento en implementaciones reales. Adicionalmente, el enfoque se centró exclusivamente en la detección binaria de caídas, sin distinguir entre diferentes tipos o severidades de caídas, lo que limita la utilidad clínica del sistema.

La evaluación del rendimiento se realizó en condiciones controladas de laboratorio, que pueden diferir significativamente de las condiciones operativas reales. Aspectos como la estabilidad de la red WiFi para transmisión de alertas, la durabilidad de los dispositivos bajo operación continua a largo plazo, y la integración con sistemas de respuesta de emergencia no fueron evaluados exhaustivamente. Además, la investigación no incluyó pruebas con usuarios finales reales, limitando la comprensión de aspectos como la aceptación tecnológica, privacidad percibida, y facilidad de instalación y mantenimiento.

Otra limitación importante concierne a la latencia del sistema completo. Si bien se midió detalladamente la latencia de inferencia de los modelos, no se evaluó la latencia total del sistema incluyendo la captura de video, pre-procesamiento, comunicación de alertas y posibles mecanismos de confirmación. Este aspecto es crucial para determinar el tiempo real desde que ocurre una caída hasta que se genera una alerta efectiva. Finalmente, el estudio no abordó las implicaciones éticas y de privacidad de monitorear constantemente a adultos mayores en sus hogares, una

consideración fundamental para la implementación práctica y aceptación de estos sistemas.

Recomendaciones

Para futuras investigaciones, se recomienda ampliar la base de datos de entrenamiento con escenarios más diversos y realistas, incorporando variaciones de iluminación, perspectivas de cámara, entornos domésticos variados y diferentes tipos de caídas. Idealmente, esta base de datos debería incluir grabaciones de caídas reales (con consentimiento ético) o simulaciones de alta fidelidad con adultos mayores reales o maniqués especializados, ya que la mecánica de caída difiere significativamente entre jóvenes voluntarios y adultos mayores. Se sugiere extender el enfoque de detección binaria hacia un sistema más granular que pueda clasificar diferentes tipos de caídas y sus severidades, e incluso detectar comportamientos precursores de caídas que permitan intervenciones preventivas. Para mejorar la robustez del sistema, se recomienda implementar análisis secuencial de video en lugar del enfoque actual basado en frames individuales, mediante la incorporación de modelos recurrentes o transformers que capturen la dinámica temporal de los movimientos.

En términos de implementación práctica, se recomienda desarrollar un pipeline completo de extremo a extremo que incluya captura de video, procesamiento, generación de alertas y confirmación, evaluando la latencia total del sistema. Este pipeline debería ser sometido a pruebas de campo en hogares reales durante períodos prolongados para evaluar confiabilidad, mantenibilidad y aceptación por parte de usuarios finales. Adicionalmente, se recomienda explorar mecanismos de privacidad por diseño, como el procesamiento en el borde con transmisión exclusiva de alertas (sin video), ofuscación automática de rasgos faciales, o sistemas de consentimiento dinámico que permitan a los usuarios controlar cuándo están siendo monitoreados.

Finalmente, para facilitar la adopción a gran escala, se recomienda desarrollar guías de imple-

mentación estandarizadas para cada escenario de uso identificado (máxima precisión, tiempo real económico, ultra bajo consumo, etc.), incluyendo instrucciones detalladas de instalación, configuración y mantenimiento accesibles para cuidadores no técnicos. Estas guías deberían complementarse con estudios de costo-efectividad que cuantifiquen el impacto potencial de estos sistemas en términos de reducción de hospitalizaciones, mejora de tiempos de respuesta a emergencias y calidad de vida de los adultos mayores.

Bibliografía

- Alanazi, T., Babutain, K., & Muhammad, G. (2023). A Robust and Automated Vision-Based Human Fall Detection System Using 3D Multi-Stream CNNs with an Image Fusion Technique. *Applied Sciences (Switzerland)*, *13*(12). <https://doi.org/10.3390/app13126916>
- Asiri, H. M., Asiri, A. M., Alruwaili, H. F., & Almazan, J. (2023). A Scoping Review of Different Monitoring-Technology Devices in Caring for Older Adults with Cognitive Impairment. *Frontiers in Public Health*, *11*. Consultado el 12 de febrero de 2024, desde <https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2023.1144636>
- Bajones, M., Fischinger, D., Weiss, A., Wolf, D., Vincze, M., De La Puente, P., Körtner, T., Weninger, M., Papoutsakis, K., Michel, D., Qammaz, A., Panteleris, P., Foukarakis, M., Adami, I., Ioannidi, D., Leonidis, A., Antona, M., Argyros, A., Mayer, P., . . . Frennert, S. (2018). Hobbit: Providing Fall Detection and Prevention for the Elderly in the Real World. *Journal of Robotics*, *2018*. <https://doi.org/10.1155/2018/1754657>
- Boudouane, I., Makhoul, A., Djelal, N., Saadia, N., & Ramdane-Cherif, A. (2023). Multimodal System of Ambient Assistance Services for Human Activity Monitoring. *Journal Européen des Systemes Automatisés*, *56*(5), 775-785. <https://doi.org/10.18280/jesa.560508>
- Dutt, M., Gupta, A., Goodwin, M., & Omlin, C. (2024). An Interpretable Modular Deep Learning Framework for Video-Based Fall Detection. *Applied Sciences (Switzerland)*, *14*(11). <https://doi.org/10.3390/app14114722>
- Hassan, M., Gumaei, A., Aloï, G., Fortino, G., & Zhou, M. (2019). A Smartphone-Enabled Fall Detection Framework for Elderly People in Connected Home Healthcare. *IEEE Network*, *33*(6), 58-63. <https://doi.org/10.1109/MNET.001.1900100>

- Heidel, M. M. M., Robinovitch, S. N., & Yang, Y. (2023). Association Between Falls, Head Impacts, and Mortality Among Older Adults in Long-Term Care. *JOURNAL OF THE AMERICAN MEDICAL DIRECTORS ASSOCIATION*, 24(12). <https://doi.org/10.1016/j.jamda.2023.08.006>
- Hu, K.-J., Chang, C.-W., & Chang, Y.-S. (2023). Application for Detecting Falls for Elderly Persons through Internet of Things Combined with Pulse Sensor. *SENSORS AND MATERIALS*, 35(1), 3655-3669. <https://doi.org/10.18494/SAM4641>
- Kobayashi, K., Ando, K., Inagaki, Y., Suzuki, Y., Nagao, Y., Ishiguro, N., & Imagama, S. (2017). Measures and Effects on Prevention of Fall: The Role of a Fall Working Group at a University Hospital. *Nagoya Journal of Medical Science*, 79, 497-504. <https://doi.org/10.18999/nagjms.79.4.497>
- Li, X., Pang, T., Liu, W., & Wang, T. (2017). Fall Detection for Elderly Person Care Using Convolutional Neural Networks. 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 1-6. <https://doi.org/10.1109/CISP-BMEI.2017.8302004>
- Liu, L., Stroulia, E., Nikolaidis, I., Miguel-Cruz, A., & Rios Rincon, A. (2016). Smart Homes and Home Health Monitoring Technologies for Older Adults: A Systematic Review. *International Journal of Medical Informatics*, 91, 44-59. <https://doi.org/10.1016/j.ijmedinf.2016.04.007>
- Min, W., Cui, H., Rao, H., Li, Z., & Yao, L. (2018). Detection of Human Falls on Furniture Using Scene Analysis Based on Deep Learning and Activity Characteristics. *IEEE Access*, 6, 9324-9335. <https://doi.org/10.1109/ACCESS.2018.2795239>

- Minh, H. D., Hoang, M. N., Tung, L. N., Duc, M. N., Cong, K. N., & Tran, D. D. (2022). AICARE: Health Abnormality Detection of Elderly Automatically Using Deep Learning. 2022 RIVF International Conference on Computing and Communication Technologies (RIVF), 340-345. <https://doi.org/10.1109/RIVF55975.2022.10013910>
- Nazeer, D. M., Chaitanya kolliboyina, V. S., Tiruveedula, K. K., Punithavathi, I. H., Shwetha, Ch., & Anusha, D. (2023). Fall Prediction of Elder Person Using CCTV Footage and Media Framework. 2023 International Conference on Emerging Techniques in Computational Intelligence (ICETCI), 138-144. <https://doi.org/10.1109/ICETCI58599.2023.10331422>
- Paul, S. K., Zisa, A. A., Ala Walid, M. A., Zeem, Y., Paul, R. R., Haque, M. M., & Hamid, M. E. (2023). Human Fall Detection System Using Long-Term Recurrent Convolutional Networks for Next-Generation Healthcare: A Study of Human Motion Recognition. 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), 1-7. <https://doi.org/10.1109/ICCCNT56998.2023.10308247>
- Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. SN Computer Science, 2(6), 420. <https://doi.org/10.1007/s42979-021-00815-1>
- Shilpa Aarthi, M., & Juliet, S. (2022). Intelligent Fall Detection System Based on Sensor and Image Data for Elderly Monitoring. 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), 1259-1265. <https://doi.org/10.1109/ICIRCA54612.2022.9985734>
- Sykes, E. (2024). Next-Generation Fall Detection: Harnessing Human Pose Estimation and Transformer Technology. Health Systems. <https://doi.org/10.1080/20476965.2024.2395574>

- Vaiyapuri, T., Lydia, E. L., Sikkandar, M. Y., Diaz, V. G., Pustokhina, I. V., & Pustokhin, D. A. (2021). Internet of Things and Deep Learning Enabled Elderly Fall Detection Model for Smart Homecare. *IEEE Access*, 9, 113879-113888. <https://doi.org/10.1109/ACCESS.2021.3094243>
- Woods, G. (2019). *Digital Image Processing*, 4Th Edition (4.^a ed.). Pearson India.
- Zereen, A. N., Gurung, A., Rajak, A., Moonrinta, J., Dailey, M. N., Ekpanyapong, M., Vachalathiti, R., & Bovonsunthonchai, S. (2023). Video Analytic System for Activity Profiling, Fall Detection, and Unstable Motion Detection. *MULTIMEDIA TOOLS AND APPLICATIONS*, 82(27), 42395-42415. <https://doi.org/10.1007/s11042-023-14993-y>
- Zhang, S.-H., Zhang, S.-K., Xie, W.-Y., Luo, C.-Y., Yang, Y.-L., & Fu, H. (2022). Fast 3D Indoor Scene Synthesis by Learning Spatial Relation Priors of Objects. *IEEE Transactions on Visualization and Computer Graphics*, 28(9), 3082-3092. <https://doi.org/10.1109/TVCG.2021.3050143>
- Zhao, D., Song, T., Gao, J., Li, D., & Niu, Y. (2024). YOLO-Fall: A Novel Convolutional Neural Network Model for Fall Detection in Open Spaces. *IEEE Access*, 12, 26137-26149. <https://doi.org/10.1109/ACCESS.2024.3362958>