



UNIVERSIDAD DE GRANADA

Tesis Doctoral

Programa de Doctorado en **Tecnologías de la Información y la Comunicación**

Departamento de Ingeniería de Computadores, Automática y Robótica



Línea de investigación: Ingeniería Neuronal y Sistemas Integrados Bioinspirados

Sistemas ciberfísicos (CPS) reconfigurables y su aplicación a técnicas de aprendizaje automático para la monitorización de actividades

José Daniel Déniz Cerpa

Directores:

Francisco Barranco Expósito

Eduardo Ros Vidal

Editor: Universidad de Granada. Tesis Doctorales
Autor: José Daniel Déniz Cerpa
ISBN: 978-84-1195-018-3
URI: <https://hdl.handle.net/10481/84503>

El doctorando **José Daniel Déniz Cerpa** y los directores de la tesis: **Francisco Barranco Expósito** y **Eduardo Ros Vidal**.

Garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Lugar y fecha

Granada, mayo de 2023

Directores de la Tesis

Francisco Barranco Expósito

Eduardo Ros Vidal

Firma

Firma

Doctorando

José Daniel Déniz Cerpa

Firma

Agradecimientos

Esta tesis ha sido posible gracias al apoyo y esfuerzo de muchas personas.

En primer lugar, quiero expresar mi profundo agradecimiento a mis directores de tesis, Dr. Francisco Barranco y Dr. Eduardo Ros, por darme la oportunidad de formar parte de su equipo de investigación y por guiarme con su experiencia y sabiduría a lo largo de este proceso. Su apoyo, motivación, generosidad, flexibilidad y confianza han sido esenciales para mi crecimiento profesional y la culminación de esta tesis.

De igual forma, quiero extender mi gratitud por el apoyo de mis compañeros y amigos del grupo de investigación, en especial a Juan y Gabriel, que me han acompañado, animado y ayudado durante estos años. Su colaboración y amistad han sido esenciales para superar los obstáculos y celebrar los logros, haciendo más llevadero este camino.

A mi familia y amigos, por su amor y apoyo incondicional, que han sido mi mayor fuente de inspiración y motivación, creyendo en mí y animándome a seguir adelante. Sin ellos, nada de esto hubiera sido posible.

Tabla de contenidos

RESUMEN	9
ÍNDICE DE FIGURAS	11
ÍNDICE DE TABLAS	13
GLOSARIO.....	15
1 INTRODUCCIÓN	17
1.1 MOTIVACIÓN.....	19
1.2 HIPÓTESIS	20
1.3 OBJETIVOS	22
1.4 FINANCIACIÓN DEL TRABAJO	23
2 MARCO TEÓRICO	24
2.1 SISTEMAS CIBERFÍSICOS RECONFIGURABLES	24
2.1.1 Adaptabilidad de los sistemas ciberfísicos	25
2.1.2 Gestión de calidades y recursos	26
2.1.3 Ejecución distribuida.....	28
2.2 MONITORIZACIÓN DE HÁBITOS HUMANOS.....	28
2.2.1 Reconocimiento de Acciones Humanas	29
2.2.2 Conjuntos de datos para clasificación de acciones	33
2.2.3 Módulos de Telemedicina	35
2.3 OPTIMIZACIÓN DE SOLUCIONES DE APRENDIZAJE PROFUNDO.....	35
2.3.1 Técnicas de compresión y aceleración de modelos	36
2.3.1.1 Cuantización.....	37
2.3.1.2 Poda de conexiones	37
2.3.1.3 Destilación de conocimiento	38
2.3.2 Herramientas de optimización de modelos.....	39
3 METODOLOGÍA	41
3.1 ARQUITECTURAS HARDWARE.....	42
3.1.1 Servidor central	42
3.1.2 Nodos de procesamiento local.....	43
3.1.2.1 Nvidia Jetson TX2.....	43
3.1.2.2 Nvidia Jetson Nano	43
3.1.3 Módulo de telemedicina.....	44
3.2 MÓDULO DE RECONOCIMIENTO DE ACCIONES	45
3.2.1 Construcción de conjunto de datos.....	46
3.2.1.1 CIAV Dataset.....	47
3.2.1.2 Conjunto de datos multimodal (IAPV).....	49
3.2.1.3 Indoor Action Dataset	51
3.2.1.4 Métricas de evaluación.....	51
3.2.2 Selección de arquitecturas.....	52
3.2.3 Entrenamiento de modelos.....	54
3.2.3.1 Procedimiento de entrenamiento	55
3.2.3.2 Transfer Learning.....	56
3.2.3.3 Fine Tuning.....	57
3.2.4 Mejora de reconocimiento de situaciones críticas	57
3.2.4.1 Integración de estimación de movimiento (TwoStream I3D).....	58
3.2.4.2 Reconocimiento de acciones a partir de pulso cardíaco y vídeo.....	59

3.2.5	<i>Optimización de arquitecturas de aprendizaje</i>	63
3.2.5.1	Reducción de operaciones en coma flotante	64
3.2.5.2	Destilación de conocimiento.....	65
3.2.6	<i>Despliegue de modelos en el nodo local</i>	67
3.2.6.1	Asignación eficiente de tareas a recursos	68
3.3	SISTEMA CIBERFÍSICO RECONFIGURABLE	70
3.3.1	<i>Herramienta de monitorización y Gestión de Calidades y Recursos</i>	72
3.3.1.1	Monitorización	73
3.3.1.2	Análisis y Planificación.....	74
3.3.1.3	Ejecución.....	77
4	RESULTADOS Y DISCUSIÓN	78
4.1	MÓDULO DE RECONOCIMIENTO DE ACCIONES	78
4.1.1	<i>Primeras arquitecturas seleccionadas</i>	78
4.1.2	<i>Reducción de operaciones en coma flotante</i>	83
4.1.3	<i>Mejora de reconocimiento de situaciones críticas</i>	86
4.1.3.1	Integración de información de movimiento	87
4.1.3.2	Reconocimiento de acciones a partir de pulso cardíaco y vídeo	90
4.1.4	<i>Optimización de arquitecturas de aprendizaje profundo</i>	96
4.1.4.1	Destilación de conocimiento.....	97
4.2	SISTEMA CIBERFÍSICO RECONFIGURABLE	103
4.2.1	<i>Política de reconfiguración del sistema</i>	105
4.2.2	<i>Evaluación del sistema en escenarios reales</i>	108
4.2.2.1	Escenario 1: Un día de una persona de avanzada edad.....	108
4.2.2.2	Escenario 2: Una persona de avanzada edad sufre un accidente.....	112
4.2.2.3	Escenario 3: Despliegue del CPS en residencia de mayores.....	113
5	CONCLUSIONES	119
	PUBLICACIONES	123
	BIBLIOGRAFÍA	126

Resumen

El envejecimiento de la población supone un reto en términos sociales y económicos debido a la necesidad de prestar asistencia de calidad a personas de avanzada edad (o dependientes), más vulnerables. En los últimos años, la evolución de las Tecnologías de la Información y la Comunicación (TIC), el desarrollo del Internet de las Cosas y los dispositivos *System-on-Chip* (SoC) han fomentado la implementación de sistemas ciberfísicos (*Cyber-Physical Systems* - CPS). Afortunadamente, estos CPS han facilitado la introducción de soluciones de bajo coste, eficientes y reconfigurables que buscan mitigar los costes del cuidado de la salud y ofrecer una atención personalizada a través del uso de la tecnología.

En esta tesis, se propone el diseño y desarrollo de sistemas ciberfísicos reconfigurables que permitan la monitorización de las actividades cotidianas gracias a técnicas de aprendizaje automático optimizadas. Mediante el uso de estos sistemas, se pretende fomentar un estilo de vida saludable y autónomo en personas mayores o con necesidades especiales, que viven solas en sus hogares o en residencias, pudiendo ofrecer además una rápida respuesta ante situaciones de emergencia, por ejemplo, cuando sufren un accidente. El sistema ciberfísico distribuido se adapta automáticamente en tiempo de ejecución para optimizar el rendimiento global del sistema, monitorizando de forma continua diversas calidades a nivel de aplicación y sistema como el rendimiento (fotogramas por segundo - fps), la precisión en el reconocimiento de las acciones o el consumo de energía. La adaptación en tiempo de ejecución se realiza mediante el despliegue automático de diferentes módulos basados en Aprendizaje Profundo en los nodos de procesamiento local. El procesamiento local se realiza en dispositivos empujados que operan en tiempo real, a pesar de su limitada capacidad de cálculo en comparación con los servidores comunes de alta gama de procesamiento en la nube, garantizando así inherentemente la privacidad de los datos de los sujetos.

El diseño de este sistema se compone de dos etapas. Por una parte, se efectúa el desarrollo del módulo de reconocimiento de acciones para monitorizar las rutinas de las personas en su hogar, integrando modelos de Aprendizaje Profundo precisos y eficientes. Para ello, se proponen dos nuevos conjuntos de datos de actividades cotidianas para el marco de aplicación de la asistencia en el hogar, con la finalidad de diseñar soluciones precisas sobre contextos reales. Se evalúa cómo el análisis de información multimodal (vídeo e información de movimiento o pulso cardíaco) contribuye a mejorar la capacidad de reconocimiento general del sistema hasta en un 4% con respecto a solo procesar vídeo, impulsando la identificación de situaciones críticas como las caídas y reduciendo posibles falsas alarmas. Finalmente, para permitir el despliegue de modelos de procesamiento basados en Aprendizaje Profundo precisos en dispositivos empujados de bajos recursos, se optimizan estos modelos a través de la técnica de destilación de conocimiento. Se estudia cómo es posible mejorar la precisión de arquitecturas eficientes destilando conocimiento a

partir de arquitecturas de redes neuronales multimodales (que conjugan vídeo y estimación de movimiento). En concreto, se mejora en un 8,3% el F1-Score de uno de los modelos computacionalmente más eficientes diseñados en esta tesis, alcanzando un rendimiento superior a los 250 fps en el nodo de procesamiento Jetson Nano. Esta optimización permite el análisis en tiempo real de varios sujetos en un mismo nodo de procesamiento de forma concurrente.

Los modelos diseñados se integran en el CPS distribuido y reconfigurable con una arquitectura distribuida nodos-nube. Para una gestión eficiente de este CPS distribuido, es necesaria la implementación de una herramienta de Gestión de Calidades y Recursos (*Quality and Resource Management - QRM*), que tiene como función principal monitorizar y analizar la información generada por el sistema para tomar decisiones de reconfiguración inteligentes. Para esta toma de decisiones, es crítico el diseño de una política de reconfiguración que permita realizar reajustes y modificar el comportamiento del sistema en tiempo de ejecución, para hacer un uso optimizado de los escasos recursos disponibles y a su vez, ofrecer una elevada precisión en la identificación de situaciones críticas. La adaptación automática dota de gran flexibilidad al sistema, permitiendo extender hasta en un 63% el tiempo de operación de la solución con nodos alimentados por baterías y triplicar la confianza de la predicción de situaciones críticas sobre los escenarios evaluados. Además, la asignación eficiente de tareas a recursos y el despliegue de modelos destilados puede llegar a reducir hasta en un 75% los recursos de cómputo necesarios (nodos de procesamiento) para ejecutar la solución.

Índice de Figuras

Figura 2.1. Sistema ciberfísico para la monitorización de actividades.....	25
Figura 2.2. Arquitectura del sistema de visualización y análisis de datos en FIVIS [60].....	27
Figura 2.3. Esquema de alto nivel de PoCL-R [60].	28
Figura 2.4. Aproximaciones más populares para el reconocimiento de actividades en función de la fuente de datos analizada.	30
Figura 2.5. Secuencia de fotogramas que representa dos acciones.	30
Figura 2.6. Arquitecturas más populares de clasificación de acciones.....	32
Figura 2.7. Comparativa de número de parámetros y precisión de modelos de clasificación de acciones en vídeos sobre los conjuntos de datos UCF-101 (azul) y HMDB51 (amarillo).	33
Figura 2.8. Ejemplo de cuantización de vector en coma flotante a un entero de 8 bits [133].	37
Figura 2.9. Ejemplo de podado (pruning) de conexiones en la red [136].....	38
Figura 2.10. Destilación de conocimiento [123].	38
Figura 3.1. Visión general de relación entre principales componentes del sistema ciberfísico de monitorización de actividades.....	41
Figura 3.2 Fotografía Jetson TX2.	43
Figura 3.3. Fotografía de Jetson Nano.	44
Figura 3.4. Metodología para desarrollo del módulo de reconocimiento de acciones.....	46
Figura 3.5. Muestras del CIAV dataset.	47
Figura 3.6. Ejemplos del conjunto de datos IAPV [151].....	50
Figura 3.7. Arquitectura de modelo RGBI3D.	53
Figura 3.8. Arquitectura de modelo RGBI3D reduced.....	54
Figura 3.9. Procedimiento de entrenamiento en dos etapas.	55
Figura 3.10. Aproximaciones propuestas para la mejora del reconocimiento de acciones críticas.	58
Figura 3.11. Arquitectura de Red Neuronal Profunda Multimodal para el reconocimiento de acciones humanas [151].....	60
Figura 3.12. Modelo 1DCNN + LSTM para reconocimiento de acciones a partir de información de pulso cardíaco [151].....	61
Figura 3.13. Enfoques introducidos para la optimización de arquitecturas.....	64
Figura 3.14. Ejemplo de destilación [156].	67
Figura 3.15. Diagrama de ejemplo de funcionamiento del balanceo de carga adaptativo basado en recursos..	69
Figura 3.16. Visión general de la arquitectura del sistema ciberfísico reconfigurable.....	70
Figura 3.17. Ciclo MAPE-K.	71
Figura 3.18. Optimización de calidades a nivel de sistema: precisión, rendimiento y consumo de recursos.....	72
Figura 3.19. Consumo energético y frecuencia de funcionamiento de los componentes del sistema.	73
Figura 3.20. Rendimiento del reconocimiento de acciones en FPS y diagrama de líneas apiladas de la confianza de las acciones identificadas.	74
Figura 3.21. Gráfico que indica el modo de operación activo en el sistema y los eventos de reconfiguración.	77

Figura 4.1. Curvas Macro Precision-Recall de los modelos propuestos [35].....	80
Figura 4.2. Ejemplos cualitativos de vídeo de test sobre CIAV dataset [35]..	80
Figura 4.3. Izquierda y derecha: Rendimiento en fps vs. consumo energético en GPU de Jetson TX2 y RTX 2080Ti respectivamente. [35].....	81
Figura 4.4. Macro F1-Score vs. Consumo energético en GPU de Jetson TX2 y tamaño de los modelos (en millones de parámetros) [35].	82
Figura 4.5. Rendimiento de los modelos propuestos en términos de FPS.....	85
Figura 4.6. F1-Score vs. GFLOPs.	86
Figura 4.7. F1-Score vs. GFLOPs con red TwoStream [60].	88
Figura 4.8. Comparativa de rendimiento entre usar GPU local en el nodo o un servidor remoto a través de PoCL-R para la inferencia de la red TwoStream I3D [60].....	89
Figura 4.9. Comparativa de rendimiento de las arquitecturas de DL seleccionadas en términos de F1-Score, fps y consumo de energía [60]..	90
Figura 4.10. Matriz de confusión que evalúa el modelo TwoStream con la capa WeighPerClass sobre el conjunto de datos IAPV [151].....	92
Figura 4.11. Curvas de macro Precision-Recall para la red TwoStream y cada uno de los streams.	94
Figura 4.12. Evaluación del rendimiento de modelos para el reconocimiento de acciones [156].....	101
Figura 4.13. Compromiso de F1-Score vs. tiempo de inferencia de los modelos destilados [156].....	102
Figura 4.14. Confianza de las predicciones de los modelos destilados seleccionados sobre muestras de test del conjunto de datos.....	103
Figura 4.15. Arquitectura de sistema ciberfísico reconfigurable.....	104
Figura 4.16. Diagrama de transición de estados de la política de reconfiguración [60].....	106
Figura 4.17. Media de horas al día dedicadas a distintas actividades por una persona de avanzada edad (mayor de 65) [171].....	109
Figura 4.18. Tiempos de funcionamiento estimados de los nodos locales alimentados por baterías de 25V 25000mAh [60].	109
Figura 4.19. Escenario 1: Reducción del consumo energético por medio de la reconfiguración [60].	110
Figura 4.20. Panel de visualizaciones del sistema de monitorización de hábitos con información agregada de tres personas.....	111
Figura 4.21. Panel de visualizaciones del sistema de monitorización de hábitos de vida.	111
Figura 4.22. Escenario 2: Confirmación de situaciones críticas por medio de la reconfiguración [60].	112
Figura 4.23. Visión general del sistema. Sistema ciberfísico para la monitorización eficiente de actividades en espacios interiores [156].....	113
Figura 4.24. Carga computacional de los nodos locales al analizar los residentes de la residencia mostrada en la Figura 4.23.	116

Índice de Tablas

Tabla 2.1. Precisión de arquitecturas en los datasets UCF-101 y HMDB51.....	32
Tabla 3.1. Muestras por clase del CIAV dataset creado a medida	48
Tabla 3.2. Número de clips del conjunto de datos IAPV	49
Tabla 3.3. Número de clips grabados por cada sujeto.....	51
Tabla 4.1. Métricas de evaluación de los modelos propuestos.....	79
Tabla 4.2. Rendimiento de las arquitecturas de Deep Learning	84
Tabla 4.3. Rendimiento de arquitecturas de Flujo Óptico	87
Tabla 4.4. TwoStream I3D, alternativas de estimación de flujo óptico.....	88
Tabla 4.5. Evaluación de modelo TwoStream de vídeo y pulso cardíaco	91
Tabla 4.6. Evaluación de adaptación a los usuarios	93
Tabla 4.7. Evaluación por clase de cada una de las redes individuales.....	94
Tabla 4.8. Rendimiento arquitectura Multimodal en la Nvidia Jetson Nano	96
Tabla 4.9. Mejora aportada por la destilación en las arquitecturas de DL	99
Tabla 4.10. Modos de operación de los nodos locales	106
Tabla 4.11. Carga computacional en los nodos Jetson Nano	115
Tabla 4.12. Análisis de integración de asignación eficiente de tareas a recursos con diferentes escenarios	117

Glosario

3D-ConvNet	Red Convolucional 3D
AAL	<i>Ambient Assisted Living</i>
AUC	Área bajo la curva (<i>Area Under the Curve</i>)
CIAV	<i>Combined Indoor Action Video dataset</i>
CNN	Red Neuronal Convolucional (<i>Convolutional Neural Network</i>)
CPS	Sistemas Ciberfísico (<i>Cyber-Physical System</i>)
DL	Aprendizaje Profundo (<i>Deep Learning</i>)
FLOPs	Operaciones en coma flotante (<i>Floating Point Operations</i>)
FPS	Fotogramas por segundo
GPU	Unidad de Procesamiento Gráfico (<i>Graphic Processing Unit</i>)
GRU	<i>Gated Recurrent Unit</i>
HAR	Reconocimiento de Actividades Humanas (<i>Human Activity Recognition</i>)
HT	Monitorización de Hábitos (<i>Habit Tracking</i>)
IA	Inteligencia Artificial
IAPV	<i>Indoor Action Pulses and Video dataset</i>
IoT	Internet de las Cosas (<i>Internet of Things</i>)
LSTM	<i>Long-short Term Memory</i>
PPG	Fotopleletismograma (señal de pulso cardíaco)
PR	<i>Precision-Recall</i>
QRM	Gestión de Calidades y Recursos (<i>Quality and Resource Management</i>)
RCN	Red Convolucional Recurrente (<i>Recurrent Convolutional Network</i>)
RNN	Red Neuronal Recurrente (<i>Recurrent Neural Network</i>)
SoC	<i>System-on-Chip</i>
SpO ₂	Pulsioxímetro

1 Introducción

La evolución de las Tecnologías de la Información y la Comunicación (TIC), unido al desarrollo de sistemas del Internet de las Cosas (*Internet of Things* - IoT) y dispositivos *System-on-Chip* (SoC) han sido claves para la realización efectiva de los sistemas ciberfísicos (*Cyber-Physical System* - CPS). Los sistemas ciberfísicos se componen de elementos *software*, *hardware* y de comunicaciones, que se integran para dar respuesta a entornos cambiantes [1, 2]. Estos sistemas son cada vez más autónomos, formados por arquitecturas distribuidas y provistos de elementos inteligentes que pueden tomar decisiones de forma independiente [3]. Las nuevas arquitecturas distribuidas del estado del arte consisten en configuraciones nodo-nube (*edge-cloud*), que permiten interconectar nodos inteligentes de procesamiento local y de altas prestaciones. Esta tecnología está presente en áreas de distinta índole como puede ser la industria automovilística para el desarrollo de vehículos autónomos [4], la seguridad en infraestructuras críticas [5], la computación en la nube [6] o en la gestión de sistemas de ciber salud o *Smart-Health* (*S-Health*) [7].

Los CPS requieren tener disponibilidad de diversidad de dispositivos, sensores, topologías de red o configuraciones para adaptarse a los distintos escenarios del mundo real que están en constante cambio, especialmente cuando hay personas que interactúan con ellos. Recientemente, en lugar de llevar a cabo modificaciones a nivel de los elementos físicos del sistema para adaptarse al entorno cambiante, se ha propuesto la **reconfiguración** o adaptación de los CPS como solución a este problema. Con ella, se busca evitar el redesplicue de elementos *hardware* y sensores, ahorrando los costes y el tiempo que esto supondría.

La reconfiguración o adaptabilidad automática permite efectuar una optimización en tiempo de ejecución para ofrecer una mejora en términos de calidad, uso de recursos (uso de CPU, uso de GPU, memoria o ancho de banda) o rendimiento de los sistemas [8]. Por ello, el que los sistemas ciberfísicos dispongan de capacidad de adaptación en tiempo de ejecución es crucial para la construcción de sistemas eficientes, escalables, distribuidos y precisos. De esta forma, por medio de la gestión y monitorización de calidades y recursos, pueden ofrecer un rendimiento óptimo, dando una respuesta inteligente a cambios en el entorno en función de los objetivos definidos.

En particular, los sistemas ciberfísicos han cobrado un papel relevante en el diseño de soluciones en el campo de *Smart-Health*. Estos sistemas de ciber salud acercan las TIC al ecosistema médico, incluyendo la gestión de la información, el uso de dispositivos como pulseras de actividad (*wearables*) o el IoT. Los sistemas ciberfísicos han facilitado la introducción de soluciones de bajo coste, eficientes y adaptables que buscan mitigar los costes del cuidado de la salud y ofrecer una atención personalizada a través del uso de la tecnología [9]. Un ejemplo de esto es la introducción de trabajos que realizan la monitorización en tiempo real de la salud del paciente para la detección temprana de

posibles patologías [10], o que fomentan la independencia de las personas mayores por medio de la monitorización de los pacientes de forma remota [11, 12].

En los últimos años, los nodos de procesamiento locales están teniendo cada vez más relevancia en este tipo de sistemas distribuidos [13]. Esto es debido a que el **procesamiento local** contribuye a la preservación de la privacidad de los usuarios, al no enviar datos sensibles a través de la red. A su vez, facilitan la reducción del uso del ancho de banda de red, así como la latencia de respuesta. Los dispositivos empotrados suelen ser utilizados como nodos de procesamiento local, ya que a pesar de su limitada capacidad de cómputo ofrecen una alta eficiencia energética.

Los avances en el campo de la Inteligencia Artificial (IA) y el aprendizaje automático han propiciado la aplicación de los sistemas de Reconocimiento de Acciones Humanas (*Human Activity Recognition* - **HAR**) en este campo de la ciber salud. El reconocimiento de acciones tiene un papel muy relevante en los sistemas de *S-Health* y de monitorización de hábitos de personas. Los sistemas HAR recopilan datos e información del comportamiento de las personas e identifican automáticamente, a partir de estos datos, qué acción se está llevando a cabo [14]. Esta aplicación se ha vuelto muy popular, dando lugar a sistemas que identifican acciones humanas a partir de diversas fuentes de datos y utilizando distintas tecnologías, como modelos de aprendizaje profundo [15].

Los modelos de aprendizaje profundo o de *Deep Learning* (*DL*) son grandes arquitecturas multi-capas de redes neuronales. Recientemente, el *DL* ha contribuido notablemente al avance del estado de la técnica en el campo de la Visión por Computador. En particular, se ha popularizado la integración de modelos de aprendizaje profundo para el desarrollo de soluciones de reconocimiento de actividades debido a su buen rendimiento en comparación con las alternativas basadas en el uso de descriptores o de extracción de características [16]. Sin embargo, aunque estas arquitecturas ofrezcan una alta capacidad de reconocimiento en la identificación de acciones a nivel general, también tienen un alto coste computacional, lo que dificulta la ejecución en tiempo real de estos modelos altamente complejos en dispositivos empotrados de bajos recursos. Por ello, la aplicación de estrategias de optimización de modelos es esencial para hacer posible y también acelerar la inferencia de arquitecturas complejas en el nodo [17].

Es por ello que, plantear el desarrollo de **sistemas autónomos adaptables** orientados a la monitorización precisa de las actividades que realizan personas en espacios interiores es uno de los desafíos que supone el desarrollo de esta tesis. Al mismo tiempo, el **reconocimiento de acciones** relevantes para estudiar los hábitos de vida, así como la identificación de situaciones potencialmente peligrosas o de emergencia para estas personas, supone otro de los desafíos de este trabajo. Adicionalmente, se presenta el reto de lograr el funcionamiento en **tiempo real** de estos sistemas en dispositivos de recursos limitados e interconectados entre sí, y a un nivel superior, optimizando también el sistema distribuido completo.

1.1 Motivación

La esperanza de vida ha aumentado significativamente a lo largo del último siglo, y al mismo tiempo, ha habido un descenso de la tasa de natalidad, lo que ha tenido como consecuencia el envejecimiento de la población mundial. Se estima que en el año 2050 un 25% de la población de Norte América y Europa será mayor de 65 años [18], por lo que la sociedad estará más expuesta a enfermedades asociadas a la avanzada edad. Este hecho supone un importante reto en términos sociales y económicos debido a los recursos que serán necesarios para prestar una asistencia de calidad a las personas de avanzada edad (o dependientes) más vulnerables que viven en residencias o solas en sus hogares [19].

Esta situación ya ha dado lugar a un aumento de las necesidades de cuidados de mayores [20]. En concreto, en países europeos como España o Alemania, el número de asistentes y trabajadores en residencias de mayores ha crecido un 30% de media desde el 2010 [21]. Una de las consecuencias directas de la evolución de esta situación es la demanda de sistemas que sean capaces de monitorizar y mejorar la calidad de vida de estas personas [22].

Diversos trabajos señalan que, la monitorización de los hábitos y rutinas contribuye a la mejora del bienestar, lo que favorece un envejecimiento saludable para las personas de avanzada edad [23, 24]. Además de todo esto, una de cada cuatro personas mayores de 65 años sufre al menos una caída al año [25]. Dar una rápida respuesta y asistencia ante este tipo de situaciones es crucial para evitar que se agraven los problemas de salud derivados de la caída [26]. Los sistemas enmarcados en el área del *Ambient Assisted Living (AAL)* son ideales para la monitorización de los hábitos de los pacientes y sus rutinas diarias para así fomentar un estilo de vida activo [27]; a su vez, estos sistemas ayudan a dar una rápida respuesta cuando se dan situaciones potencialmente peligrosas, como los desmayos o las caídas [28].

Los sistemas del área de la ciber salud o *Smart-Health* relacionan el campo de la salud con las TIC para abordar algunos de los retos más relevantes a los que se enfrenta nuestra sociedad en la actualidad [29, 30]. En particular, el *S-Health* contribuye a la reducción de los costes del cuidado de la salud, ofreciendo nuevas soluciones [31, 32]. La evolución del campo del *IoT* y la introducción de dispositivos *System-on-Chip (SoC)* para sistemas empotrados, han servido como catalizadores para el desarrollo de sistemas ciberfísicos (CPS) que ofrecen soluciones de ciber salud de bajo coste, distribuidas y escalables para asistir a estas personas con necesidades [9, 33, 34]. En relación con el campo de la ciber salud o *Smart-Health*, los sistemas ciberfísicos proporcionan soluciones distribuidas para el cuidado remoto de las personas. Por lo tanto, los CPS son una de las alternativas más apropiadas para el diseño de sistemas de monitorización de estilo de vida [35]. Sin embargo, es fundamental resaltar que la interacción y asistencia humanas siguen siendo indispensables para ofrecer una atención integral y adecuada a las personas. Los sistemas

de ciber salud son complementarios y están destinados a mejorar la calidad de vida, pero siempre bajo la supervisión y atención de profesionales y cuidadores.

1.2 Hipótesis

Ante el problema de la monitorización de personas de avanzada edad o con necesidades especiales que viven solas en el hogar, y considerando la importancia de la protección de la privacidad, la eficiencia de la solución y su robustez en la identificación de situaciones peligrosas como las caídas, surge la siguiente hipótesis: ¿podemos llevar a cabo la monitorización de actividades de la vida cotidiana, de forma precisa y eficiente, utilizando para ello CPS adaptables dotados de inteligencia mediante el uso de aprendizaje automático?

En este trabajo se plantea el diseño y desarrollo de estos **sistemas ciberfísicos reconfigurables** para la monitorización de las actividades cotidianas a través de la utilización de técnicas de **aprendizaje automático** optimizadas. Mediante el uso de estos sistemas, se pretende fomentar un estilo de vida saludable y autónomo en personas mayores o con necesidades especiales que viven solas en sus hogares o en residencias. Esto permitirá aumentar la eficiencia del personal asociado a este tipo de tareas y facilitará abordar el reto del envejecimiento con mejor calidad de vida.

El análisis de los sistemas de monitorización actuales permite identificar los aspectos a mejorar de este tipo de aplicaciones: 1) protección de la privacidad de los datos, clave en el campo de la salud; 2) alto consumo energético debido a su operación de forma continuada, requiriendo recargas de baterías de manera frecuente en el caso de los sistemas independientes; 3) generación de falsas alarmas, lo cual reduce la confianza en el sistema e incurre en mayores costes económicos; 4) problemas de escalabilidad y falta de adaptación a los entornos cambiantes del mundo real.

En este trabajo, estos problemas se abordan a través del **procesamiento distribuido** y la integración de **reconfiguraciones** automáticas del sistema nodo-nube (*edge-cloud*). Por un lado, el procesamiento local en el nodo permitirá hacer un análisis eficiente de las actividades y contribuirá a evitar la transmisión de datos a través de internet, protegiendo inherentemente la información sensible y mejorando de forma transparente la escalabilidad de la solución. Por otro lado, reconfiguración automática tendrá como objetivo la adaptación al cambio y la optimización del uso de recursos o consumo energético, así como la reducción de falsas alarmas, especialmente para el caso de las acciones críticas, como son las caídas.

La introducción de los CPS ha permitido el desarrollo de sistemas de limitado coste, escalables y distribuidos. Sin embargo, este tipo de dispositivos empujados encargados de realizar procesamiento local presentan limitaciones en términos de recursos computacionales. Por ello, la adaptabilidad del sistema juega un papel muy importante para

aprovechar al máximo los recursos limitados disponibles. En el caso del sistema que se propone, la **adaptabilidad** automática permitirá optimizar el uso de los recursos en función de distintos objetivos como podrían ser: consumo energético, rendimiento o precisión dependiendo de diferentes escenarios, evitando así mayores costes de tiempo y económicos asociados a ajustes manuales y redesplicue de elementos *hardware* o recursos de red, necesarios para adaptar la solución a nuevas condiciones. De esta forma, por ejemplo, se podría priorizar la precisión del sistema frente a su consumo para tratar de identificar con la mayor confianza posibles situaciones críticas (por ejemplo, caídas). O bien, se podría buscar una reducción del consumo energético a expensas de una limitación en el rendimiento.

Uno de los principales componentes del sistema que se propone será el reconocedor de acciones humanas (**HAR**) en espacios interiores. En los últimos años, se han presentado trabajos que llevan a cabo HAR a través de técnicas de visión artificial utilizando aprendizaje máquina. Estas soluciones logran una buena precisión, aunque a un elevado coste computacional en términos de tiempo de ejecución y memoria. Por ello, se plantea la aplicación de nuevas arquitecturas de modelos de aprendizaje automático para el reconocimiento de acciones **eficiente**. De forma que estas nuevas alternativas permitan alcanzar un procesamiento en tiempo real, limitando el uso de recursos y ofreciendo una alta precisión.

Por lo tanto, para el diseño de este componente se tendrán en cuenta las capacidades computacionales limitadas de los dispositivos empujados donde se llevará a cabo el procesamiento. De este modo, se deberán **optimizar** los modelos de aprendizaje automático para reducir el número de cálculos necesarios, así como el espacio de memoria requerido.

Adicionalmente, los sistemas de reconocimiento de acciones están actualmente centrados en la identificación de actividades a nivel general. Concretamente, los conjuntos de datos que representan acciones humanas en vídeo que hay disponibles, no permiten entrenar y evaluar un modelo de aprendizaje automático sobre acciones de utilidad para monitorizar el estilo de vida de las personas en sus hogares y, simultáneamente, identificar situaciones potencialmente peligrosas, como pueden ser caídas. Teniendo en cuenta la relevancia de la **calidad de los datos** al entrenar modelos de aprendizaje automático, es crucial poder disponer de conjuntos de datos más específicos y de una mayor calidad que permitan lograr una mayor precisión al identificar actividades de la vida cotidiana en contextos reales.

En este trabajo, se busca validar las hipótesis introducidas mediante el diseño y desarrollo de un módulo de reconocimiento de acciones preciso y eficiente, y su integración en un sistema ciberfísico reconfigurable.

En primer lugar, los esfuerzos del trabajo se centran en el diseño del módulo de reconocimiento de acciones. Se diseñan arquitecturas de aprendizaje automático **precisas**

que puedan ejecutarse en **tiempo real** en el nodo. En el desarrollo de las soluciones se busca optimizar las prestaciones en términos de dos calidades. Por un lado, se proponen dos aproximaciones para mejorar la **precisión** de las soluciones individuales de reconocimiento de acciones, especialmente cuando se trata de situaciones críticas como las caídas, haciendo uso de información multimodal (señales vitales o información de movimiento) a expensas de un aumento de uso de recursos. Por otro lado, se presentan técnicas de optimización, compresión y aceleración de modelos de aprendizaje profundo para el análisis **eficiente** de actividades, ofreciendo una alta precisión a la vez que se minimiza el número de cálculos y uso de memoria de las arquitecturas. Se busca que esta optimización permita el análisis de acciones de varios sujetos de forma concurrente en cada nodo de procesamiento, garantizando operación en tiempo real.

Finalmente, se presenta la implementación integral de un CPS de monitorización de actividades reconfigurable, con arquitectura **nodo-nube**. El sistema se **adapta** en tiempo de ejecución para hacer un uso optimizado de los limitados **recursos** disponibles, y a su vez ofrecer una elevada **precisión** en la identificación de situaciones críticas, reduciendo las falsas alarmas. Se pretende lograr estas mejoras a nivel de sistema mediante la integración de un módulo de gestión de calidades y recursos (*Quality and Resource Management* - **QRM**), de forma que, mediante el análisis de las calidades y actividades monitorizadas por el sistema, se tomen decisiones de adaptación automática en tiempo de ejecución para la mejora de la identificación de acciones potencialmente peligrosas y se haga un uso optimizado de los escasos recursos disponibles.

1.3 Objetivos

El objetivo principal de esta tesis es el de diseñar y desarrollar un sistema ciberfísico automáticamente adaptable y reconfigurable que permita realizar de forma eficiente la monitorización actividades cotidianas a través de técnicas de aprendizaje automático para prestar asistencia y ayudar a la vida independiente de, por ejemplo, personas mayores o con necesidades especiales.

Los objetivos específicos definidos para este trabajo se describen a continuación:

- Recopilar y crear un conjunto de datos multimodal que represente actividades de la vida diaria relevantes para el marco de aplicación de asistencia en el hogar.
- Diseñar e implementar modelos de aprendizaje automático para el reconocimiento de actividades de la vida diaria.
- Optimizar y desplegar modelos de aprendizaje automático en dispositivos empujados con rendimiento en tiempo real.
- Diseñar la política de reconfiguración multi-objetivo del sistema completo, con el objetivo de optimizar distintas calidades como precisión o consumo.

- Diseñar, implementar y validar la arquitectura de sistema ciberfísico usando nodos locales y un servidor central para procesamiento distribuido, que permita su reconfiguración en tiempo real en función del análisis de diferentes calidades, como la precisión del sistema o el consumo total del mismo.

1.4 Financiación del trabajo

Esta tesis se ha elaborado bajo el marco del Proyecto: *From the cloud to the edge – smart IntegraTion and OPTimisation Technologies for highly efficient Image and Video processing Systems* (FitOPTiVis - H2020-RIA. ECSEL-2017-737451). FitOptiVis es un proyecto europeo de investigación en el que han colaborado entidades de cinco países diferentes: Países Bajos, República Checa, Finlandia, España e Italia. El objetivo de este proyecto es proponer un enfoque general para la integración inteligente de canalizaciones de procesamiento de imagen y vídeo para sistemas ciberfísicos, definiendo una arquitectura de referencia, sustentada por dispositivos de bajos recursos, de altas prestaciones e inteligentes. Así como por metodologías y herramientas para la optimización multiobjetivo en tiempo de diseño y ejecución, teniendo en cuenta las restricciones del sistema y el entorno.

Este objetivo es abordado mediante la introducción de varios casos de uso industriales de diversos ámbitos de aplicación que buscan implementar tareas de procesamiento de imagen y vídeo en sistemas ciberfísicos, donde la integración de componentes de diversa naturaleza toma un papel crucial en el diseño de las soluciones. En particular, parte del trabajo realizado en esta tesis se ha integrado en el caso de uso 3: *Habit Tracking* (Monitorización de hábitos). En este caso de uso se ha diseñado la arquitectura de un sistema nodo-nube para el análisis de las actividades en el hogar de las personas de avanzada edad que integra un componente de monitorización de calidades y gestión de recursos. Este módulo analiza la información generada por el sistema y toma decisiones de reconfiguración en tiempo de ejecución para optimizar el uso de recursos y reducir falsas alarmas.

En la última etapa de la tesis, también se ha apoyado con financiación de IMOCO4.E: *Intelligent Motion Control under Industry4.E.* (EU H2020 RIA-101007311), (PCI2021-121925) en la que se ha colaborado en la evaluación conceptual de este tipo de técnicas para ámbitos más industriales.

Asimismo, los siguientes proyectos nacionales también han apoyado la elaboración de este trabajo. 1) Desde la nube a los puntos finales. Tecnologías de integración inteligente y optimización para sistemas eficientes de procesamiento de imágenes y video (PCI2018-093184); 2) BIO-Perception. Nueva generación de sistemas de visión inteligentes para procesamiento en tiempo real con sensores bio-inspirados (PID2019.109434RA.I00).

2 Marco teórico

La posibilidad de interactuar y ampliar las capacidades del mundo físico a través de la computación, la comunicación y el control es un factor clave para desarrollos tecnológicos futuros, como puede ser el diseño y desarrollo de los próximos aviones, vehículos autónomos o prótesis médicas que permitan ejercer control sobre elementos físicos [1, 36]. A lo largo de los años, en los campos del diseño de sistemas y control, se han introducido herramientas y metodologías innovadoras para el análisis, optimización y control de los sistemas. Al mismo tiempo, en el ámbito de las ciencias de la computación se han propuesto nuevos lenguajes de programación, compiladores y nuevas arquitecturas empotradas, así como nuevos enfoques para el diseño de sistemas resilientes y tolerantes a fallos. Mediante la introducción de sistemas ciberfísicos se busca la integración del conocimiento y fundamentos de ingeniería de diversas disciplinas, para así desarrollar una nueva generación de sistemas que integran procesos físicos y de computación [36].

2.1 Sistemas ciberfísicos reconfigurables

Impulsado por la rápida evolución de la integración inteligente de dispositivos *System-on-Chip* (SoC), el Internet de las Cosas (IoT) y la computación en la nube, los sistemas ciberfísicos (CPS) cada vez están más presentes en varios campos de aplicación como la automoción [37], aviación [38], videovigilancia de infraestructuras críticas [39] o telemedicina [40]. CPS se refiere a un sistema que integra estrechamente módulos software de procesamiento, redes y procesos físicos [1, 2]. Los módulos de procesamiento y las redes monitorizan y controlan los procesos físicos, generalmente a través de ciclos de retroalimentación en los que los elementos físicos tienen influencia en los procesos del sistema, y viceversa. De esta forma, los sistemas ciberfísicos facilitan la integración del mundo físico con componentes de red y *software* que interactúan con personas [1, 36].

Los sistemas ciberfísicos más recientes se están convirtiendo en arquitecturas distribuidas de sistemas empotrados inteligentes altamente conectados entre sí que cada vez son más autónomos. La computación en la periferia de la nube (*Edge Computing*) ha cobrado especial importancia para el desarrollo de sistemas ciberfísicos distribuidos [13], especialmente en el paradigma nodos locales-nube (*edge-cloud*) [41]. La computación *edge-cloud* (nodo-nube) distribuida aprovecha los beneficios que ofrece la computación local, acercando el procesamiento a las fuentes de los datos. El diseño *edge-cloud* contribuye a la reducción de la latencia y el uso de ancho de banda mediante el procesamiento en el *edge*, a la vez que hace posible disponer de una mayor capacidad de cálculo a través de la computación en la nube [42, 43].

Los nodos de procesamiento locales son dispositivos empotrados e inteligentes que están estrechamente conectados con los sistemas físicos y su entorno [44]. Los sistemas ciberfísicos basados en visión ofrecen soluciones distribuidas donde la mayor parte del procesamiento se realiza en los nodos locales [45]. Estos nodos locales son dispositivos de limitados recursos que son capaces de ejecutar soluciones de aprendizaje automático

eficientemente, a un bajo coste energético. Los nodos locales solo transmiten los resultados de su procesamiento, por lo que permiten proteger de forma inherente la privacidad, al no transmitir ninguna información sensible a través de la red.

El mundo real está en constante cambio, especialmente cuando también hay personas que interactúan con el sistema. Los CPS no operan sobre un entorno controlado y deben ser robustos y adaptables a diversas situaciones para garantizar una elevada fiabilidad y altas prestaciones [36]. Los enfoques más recientes persiguen la reconfiguración o adaptación de los CPS en lugar de un redespliegue de elementos hardware y sensores con la finalidad de reducir tiempo y costes [46].

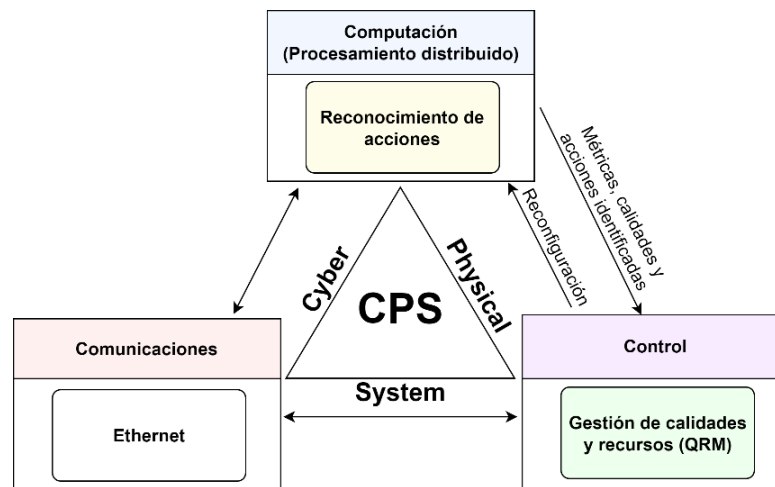


Figura 2.1. Sistema ciberfísico para la monitorización de actividades.

En la Figura 2.1 se muestran los componentes principales del sistema ciberfísico propuesto en este trabajo para la monitorización de actividades. El sistema interrelaciona elementos de comunicaciones, computación y control para integrar los componentes y dar respuesta a cambios en el mundo físico. Por un lado, la computación se efectúa a través del procesamiento distribuido en nodos locales de alta eficiencia. En estos nodos se ejecuta el **reconocimiento de acciones** para la monitorización de actividades. A su vez, como elemento de control del sistema se utiliza una herramienta de **Gestión de Calidades y Recursos (QRM)**. Este módulo recibe calidades y métricas acerca del rendimiento, consumo energético y actividades detectadas por el componente de clasificación de acciones. El módulo QRM analiza esta información para generar visualizaciones de los datos y tomar decisiones de adaptación del sistema en tiempo de ejecución con la finalidad de hacer un uso optimizado de los limitados recursos disponibles.

2.1.1 Adaptabilidad de los sistemas ciberfísicos

En los últimos años, se han introducido CPSs reconfigurables que se adaptan de forma dinámica a entornos cambiantes [47]. Los sistemas ciberfísicos reconfigurables no sólo ofrecen una mayor flexibilidad en términos de funcionalidad, sino que también buscan la optimización ciertas calidades cuando se ejecuta una tarea. Sin embargo, los sistemas

proveen distintos niveles de autonomía, con reconfiguraciones lanzadas por un usuario [48], o de forma automática generadas por el propio sistema por medio de la monitorización del entorno y su rendimiento [49]. Por ejemplo, en [50], los autores proponen un sistema ciberfísico reconfigurable para plantas de producción usando Avatares Virtuales que se adaptan automáticamente a nuevas arquitecturas de automatización y rendimiento. En [51], se propone un sistema para monitorización eficiente, integrando aproximaciones de efectividad del rendimiento global a través de reconfiguraciones automáticas. Sin embargo, algunos trabajos muestran que la intervención humana es crucial, como en [48], donde se introduce un sistema ciberfísico para robots colaborativos en la industria automovilística construido con una estrecha interacción humano-robot.

En concreto, la reconfiguración habilita una optimización escalable y modular en tiempo de ejecución para proporcionar una mejora de prestaciones en función de diversos objetivos [52]. En tiempo de ejecución, la arquitectura del sistema se debe adaptar a cambios en el entorno para conservar o mejorar su rendimiento y eficiencia [39, 52].

2.1.2 Gestión de calidades y recursos

Las técnicas de monitorización en línea son un componente crucial para habilitar la adaptación a distintos perfiles de reconfiguración [53]. Tanto los perfiles de reconfiguración como las necesidades de recursos son optimizados para dar respuesta a contextos operativos cambiantes. Finalmente, las herramientas de gestión de calidades y recursos (*Quality and Resource Management – QRM*) toman decisiones sobre la estrategia de reconfiguración óptima en función de distintos objetivos como la eficiencia energética, el ancho de banda, el rendimiento o coste del sistema [54].

La herramienta de Gestión de Calidades y Recursos (QRM) tiene un papel muy importante dentro del sistema ciberfísico debido a que definir una configuración del sistema presenta muchos grados de libertad y encontrar una configuración óptima es un problema de una alta complejidad. Esta es un área de estudio de gran interés en la que se han presentado distintos enfoques para la reconfiguración dinámica y auto-adaptación de los CPS, pero hay mucha fragmentación y no existe un conjunto completo de herramientas [55].

Desde el punto de vista práctico, la implementación de estas soluciones en sistemas reales se centra en la monitorización y visualización, limitando la automatización a acciones basadas en reglas. Un ejemplo de tal solución es Grafana [56], una plataforma de código abierto para la visualización de datos monitorizados por CPS o sistemas IoT. Grafana ofrece multitud de visualizaciones para facilitar a los usuarios entender los datos. Otros proyectos similares son Kibana [57], una herramienta de gestión y visualización de datos para Elasticsearch, o Chronograph [58], un *framework* para bases de datos de series temporales InfluxDB. Una lista más completa de este tipo de soluciones se puede encontrar en [59].

Estas herramientas están pensadas principalmente para la rápida construcción de paneles de monitorización (*dashboards*) sencillos usando complementos de visualización predefinidos, así como otros gráficos tales como histogramas, gráficos de líneas, gráficos circulares, etc. Esto hace que sea muy sencillo visualizar datos en paneles de visualización estáticos para aquellos usuarios que no son expertos. Al mismo tiempo, sin embargo, su simplicidad y su interfaz gráfica amigable los hacen menos apropiados para la gestión del sistema que se monitoriza y para la creación de visualizaciones personalizadas que tienen que adaptarse a los datos (por ejemplo, crear gráficos con límites dinámicos, o *dashboards* con un diseño o plantilla cambiante).

Por este motivo, la herramienta QRM seleccionada para este trabajo es FIVIS, la cual ofrece la flexibilidad necesaria para construir una solución QRM a medida. FIVIS es una plataforma para el almacenamiento, visualización y análisis. Como se muestra en la Figura 2.2, la arquitectura de FIVIS se compone de dos elementos diferenciados *back-end* (representado en la parte superior de la figura) que se ejecuta en el servidor, y el *front-end*, que se ejecuta en el navegador web (parte inferior de la figura).

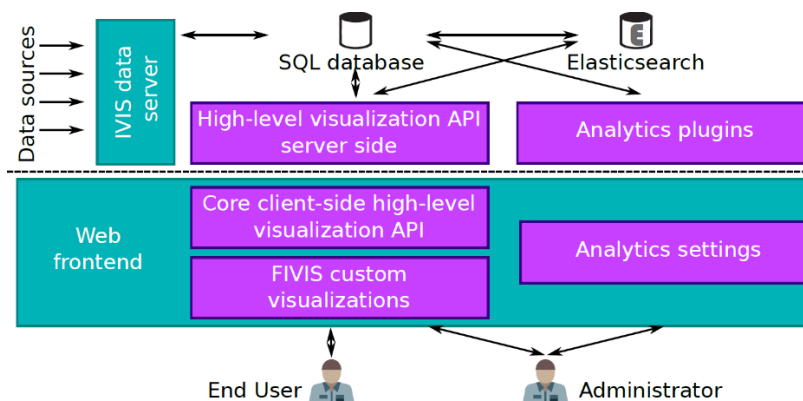


Figura 2.2. Arquitectura del sistema de visualización y análisis de datos en FIVIS [60]

El *back-end* gestiona todos los datos (una base de datos SQL se utiliza como almacenamiento principal de los datos, mientras que el motor Elasticsearch [61] permite realizar búsquedas de forma rápida y hacer agregaciones de datos sobre la marcha). El *back-end* también ofrece una API (*Application Programming Interface*) de visualización para el *front-end*, y programa la ejecución de tareas de análisis a medida que puede procesar y actuar en función de los datos entrantes. Para garantizar el procesamiento ocasional de datos entrantes, el sistema permite programar tareas de análisis en una infraestructura de computación *edge-cloud* de latencia controlada [62].

El *front-end* provee al sistema de interfaz de usuario, tanto para los usuarios finales como para los administradores. Los usuarios finales consumirán los datos principalmente en forma de visualizaciones, interactuando con *dashboards* que contienen visualizaciones de los datos generados por el sistema monitorizado o por las tareas de análisis. Los administradores son responsables de la creación de los paneles de visualización de las tareas

de análisis, e interactuarán principalmente con la parte dedicada a gestión de la interfaz de usuario. Debido a que FIVIS se centra en la flexibilidad, tanto las visualizaciones como las tareas de análisis se pueden definir y crear a medida usando un editor de código integrado en la interfaz de usuario, que permite la compilación y despliegue de los cambios sin causar interrupciones en el servicio.

2.1.3 Ejecución distribuida

Para minimizar el coste del sistema, es de gran utilidad transferir (*offloading*) la ejecución de tareas con alta carga computacional a un servidor local y solo ejecutar tareas de baja complejidad computacional en los nodos locales. En este trabajo, [63] provee una plataforma de virtualización para el *offloading* de las tareas. Esta plataforma es un controlador de ejecución distribuida construido sobre PoCL [64], un *framework* de código abierto implementado en OpenCL (*Khronos® OpenCL Working Group*).

Un esquema de alto nivel de esta configuración se muestra en la Figura 2.3. Los servidores remotos que contienen potentes GPUs u otros dispositivos de computación que soporten OpenCL ejecutan un proceso en segundo plano al que se conecta el controlador del cliente de *PoCL-R*. Este proceso informa al cliente remoto cuáles son los recursos de computación disponibles, exponiéndolos en la máquina cliente como si estuviesen conectados directamente en el dispositivo local. Esto facilita el cambio de la ejecución de una tarea entre los nodos locales y los dispositivos remotos a través de OpenCL.

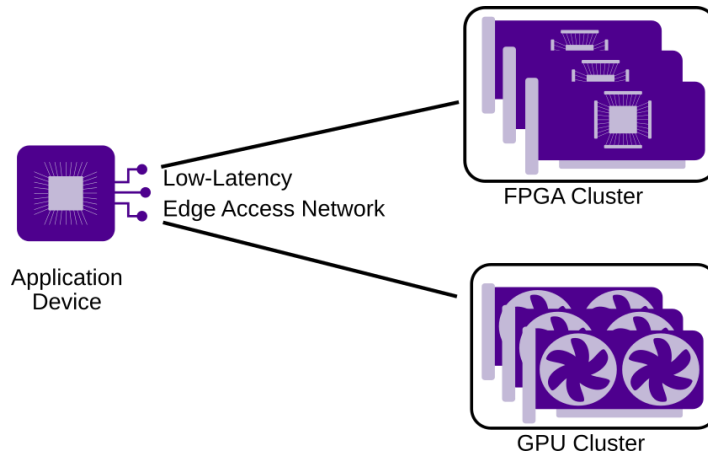


Figura 2.3. Esquema de alto nivel de *PoCL-R* [60].

2.2 Monitorización de hábitos humanos

En los últimos años, la introducción de aplicaciones de monitorización de hábitos o estilo de vida (*Habit Tracking - HT*) ha contribuido a paliar los efectos socioeconómicos derivados del envejecimiento de la población, permitiendo el cuidado remoto de la salud de personas mayores o con necesidades especiales a través del uso de la tecnología [10]. Los sistemas de *Habit Tracking* buscan mejorar la calidad de vida de las personas con

necesidades de atención y alertar a los cuidadores en caso de que se origine una situación de emergencia. En concreto, las soluciones que se apoyan en el *HT* buscan aliviar los costes del cuidado de la salud en hogares, residencias u hospitales [65]. Por un lado, los sistemas *HT* asisten en la detección de situaciones potencialmente peligrosas, con la consecuente activación de alarmas [66], ofreciendo garantías de seguridad y promoviendo la autonomía de los usuarios. Por otro lado, estos sistemas también buscan fomentar hábitos de vida saludables y son capaces de identificar el posible deterioro progresivo de la salud de los usuarios [67]. Algunas aplicaciones del estado de la técnica en el campo de *HT* son: detección de caídas, introduciendo un sistema que notifica a los cuidadores cuando se identifica una caída por medio de la monitorización de sensores inerciales [68]; en [69], el trabajo se limita a la monitorización de parámetros médicos; o en [70], donde los autores describen un sistema que usa sensores pasivos inalámbricos que monitorizan el consumo energético, la temperatura y movimiento para estudiar el comportamiento de la persona en su hogar.

2.2.1 Reconocimiento de Acciones Humanas

El objetivo principal de los sistemas de monitorización de hábitos es analizar las acciones humanas que están teniendo lugar, especialmente para personas mayores o personas que necesitan supervisión [71]. Algunos ejemplos de acciones interesantes para la monitorización del estilo de vida de las personas son: caminar, comer, ver la televisión o actividades relacionadas con la higiene personal [14]. Los sistemas de Reconocimiento de Acciones Humanas (HAR) han contribuido al desarrollo de sistemas de monitorización de estilo de vida, realizando el análisis y reconocimiento de acciones humanas a partir de diversas fuentes de datos [14]. En función de la fuente de datos analizada, los sistemas HAR se dividen en dos categorías principales (ver Figura 2.4):

- HAR basado en **sensores**: se procesa la información extraída de sensores situados o no en el individuo como giroscopios o acelerómetros para detectar el movimiento de distintas partes del cuerpo [71]. Generalmente se utilizan sensores de movimiento o dispositivos inteligentes como pulseras de actividad, relojes inteligentes o *smartphones* para recopilar la información [72, 73].
- HAR basado en **visión**: se analiza una secuencia de fotogramas que se procesan usando algoritmos de visión por computador para identificar acciones [74]. Algunos autores se han centrado en el procesamiento únicamente de vídeos RGB [75], mientras que otros autores también se apoyado en el uso de sensores de profundidad para mejorar la robustez de la identificación de las acciones [76].

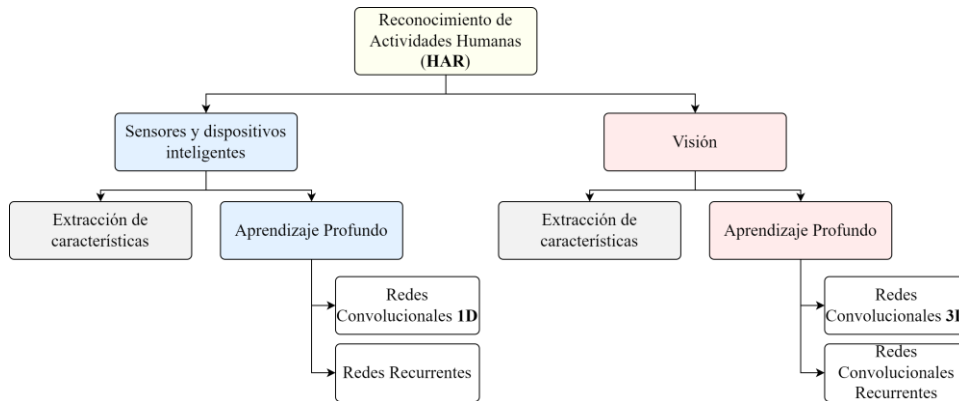


Figura 2.4. Aproximaciones más populares para el reconocimiento de actividades en función de la fuente de datos analizada.

Ambos enfoques pueden abordarse mediante técnicas de extracción de características, que combinan el uso de descriptores con clasificadores como *Random Forests* o Máquinas de Soporte Vectorial (SVM), o bien, mediante una aproximación basada en el Aprendizaje Profundo [77]. En los dos casos, los enfoques basados en Redes Neuronales Profundas ofrecen un rendimiento al nivel del estado de la técnica en términos de precisión en comparación con enfoques tradicionales basados en extracción de características [77–79].

La principal ventaja de la metodología basada en visión, frente a la basada en el uso de sensores, es que es menos intrusiva y más sencilla de implantar. El HAR basado en visión simplemente analiza un flujo de vídeo capturado a través de una cámara (ver Figura 2.5), evitando que tenga que utilizarse ningún dispositivo adicional, lo cual es un factor a tener en cuenta dado que las personas mayores son a veces reacias a utilizar aparatos inteligentes [80].



Figura 2.5. Secuencia de fotogramas que representa dos acciones: (arriba) una persona sufriendo una caída, (abajo) una persona cocinando.

Con respecto al método basado en el uso de sensores, las Redes Neuronales Convolucionales (*Convolutional Neural Networks* - CNN) y las Redes Neuronales Recurrentes (*Recurrent Neural Networks* – RNN) son dos de las soluciones más efectivas

a la hora de analizar señales [81]. Concretamente, en [82] se propone un modelo basado en capas de RNN como son las capas LSTM o *Long-short Term Memory* [83] para extraer los patrones temporales que se encuentran en las señales medidas de sensores de actividad como acelerómetros y giroscopios para identificar acciones cotidianas. Por otro lado, en [84], los autores usan CNNs para extraer patrones relevantes de señales de sensores corporales y ambientales para identificar acciones.

Con respecto al reconocimiento de acciones basado en visión, una de las arquitecturas más populares para la identificación de acciones en vídeo son las Redes Convolucionales Recurrentes (*Recurrent Convolutional Network - RCN*) [85], que hacen uso de capas convolucionales 2D para extraer información espacial de una secuencia de fotogramas, mientras que las unidades recurrentes (RNN) como las LSTM o *Gated Recurrent Units (GRU)* aprenden los patrones temporales [86]. Por otro lado, la introducción de redes Convolucionales 3D (3D-ConvNets) para el procesamiento de vídeo ha impulsado la mejora del reconocimiento de acciones en términos de exactitud [87]. Las operaciones convolucionales 3D permiten extraer de forma simultánea información temporal y espacial de los vídeos. Sin embargo, esto va ligado a un alto coste en términos de complejidad computacional en comparación con operaciones convolucionales 2D, utilizadas principalmente en el procesamiento de imágenes. En particular, el reconocimiento de acciones basado en 3D-ConvNets se ha abordado a través de trabajos como:

- **3D Residual Network** [88], en donde se propone un modelo 3D basado en la arquitectura ResNet [89] para la identificación de acciones a nivel general.
- **Temporal 3D ConvNet** [90], que incorpora filtros 3D sobre el modelo DenseNet [91], priorizando el análisis de las características temporales frente al procesamiento de características espacio-temporales.
- **$R(2+1)D$** [92], donde se propone la factorización de capas convolucionales 3D en convoluciones 2D+1D para reducir la complejidad computacional de la arquitectura final. De esta manera, las capas 2D extraen la información espacial, y las capas 1D se encargan de procesar las características temporales del vídeo. Este enfoque ofrece mejoras en términos de precisión debido al aumento de no linealidades en la red.
- **RGBI3D** [75] expande la arquitectura *Inception* [93] con operaciones 3D. Concretamente, el modelo *RGBI3D* es una aproximación muy interesante que obtiene una alta precisión a un coste computacional razonable en el reconocimiento de acciones [94].
- **S3DG** [95] es un modelo que replica la arquitectura de *RGBI3D* pero incluyendo diseños que la hacen más eficiente, como pueden ser las convoluciones temporalmente separables o los mecanismos de activación de características espaciotemporales (*feature gating*). Esto convierte a esta arquitectura en un modelo con menos parámetros y computacionalmente más eficiente, capaz de capturar dependencias entre canales de características.

- **TwoStream I3D** [75] es una de las propuestas del estado del arte que ofrece una elevada precisión. Esta red está compuesta de dos partes (ver Figura 2.6): un primer *stream* 3D-ConvNet que procesa vídeo en RGB, y un segundo *stream* 3D-ConvNet que analiza información de movimiento (estimada usando el método Optical Flow TV-L1 [96], un algoritmo muy exigente computacionalmente). La estimación de flujo óptico describe el movimiento entre dos fotogramas consecutivos en función de la velocidad y dirección de los píxeles en la escena.

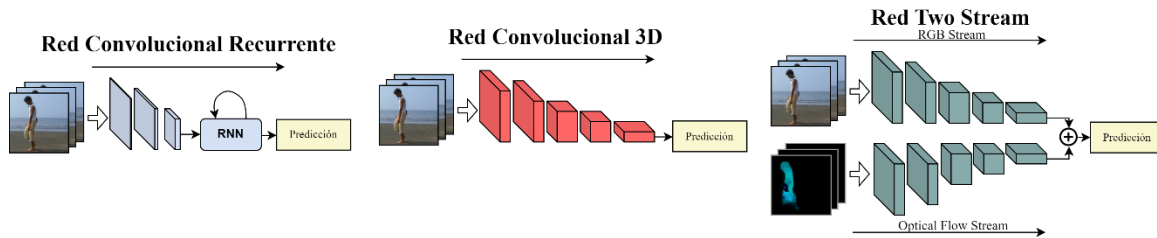


Figura 2.6. Arquitecturas más populares de clasificación de acciones.

En la Tabla 2.1 se muestra como la arquitectura *TwoStream I3D* ofrece uno de los mejores resultados en términos de precisión en dos de los conjuntos de datos más populares del estado del arte. Sin embargo, esto se consigue a expensas de un uso elevado de recursos. Recientemente, se ha popularizado la introducción de arquitecturas Transformers [97] debido a su gran rendimiento en varios campos de aplicación como el procesamiento del lenguaje natural o la visión por computador.

Tabla 2.1. Precisión de arquitecturas en los datasets UCF-101 y HMDB51.

Arquitectura	UCF-101[98]	HMDB51[99]
ConvNet+LSTM (RCN) [75]	82,1	46,4
3D Residual Network [88]	86,1	55,6
Temporal 3DConvNet [90]	93,2	63,5
R(2+1)D [92]	96,8	74,5
RGBI3D [75]	95,6	74,8
S3DG [95]	96,8	75,9
Two-Stream I3D [75]	98,0	80,7
VideoMAE V2 [100]	99,6	88,1

En los últimos años se han introducido soluciones basadas en Transformers para la clasificación de actividades como son: Video Swin Transformer [101], TimeSformer [102] o VideoMAEv2 [100]. Sin embargo, a pesar del elevado nivel de prestaciones en términos de bondad de ajuste que ofrecen estas arquitecturas, su complejidad computacional es

superior a la de las redes convolucionales 3D presentadas [101]. Por ejemplo, la red VideoMAEv2, que es actualmente el estado del arte en clasificación de acciones dispone de más de **mil** millones de parámetros, en cambio, la red *TwoStream I3D* cuenta con menos de 25 millones de pesos (ver Figura 2.7). En esta tesis no se implementan arquitecturas basadas en Transformers debido a que su alta complejidad provoca que los nodos locales de procesamiento no dispongan de recursos suficientes para ejecutar las soluciones.

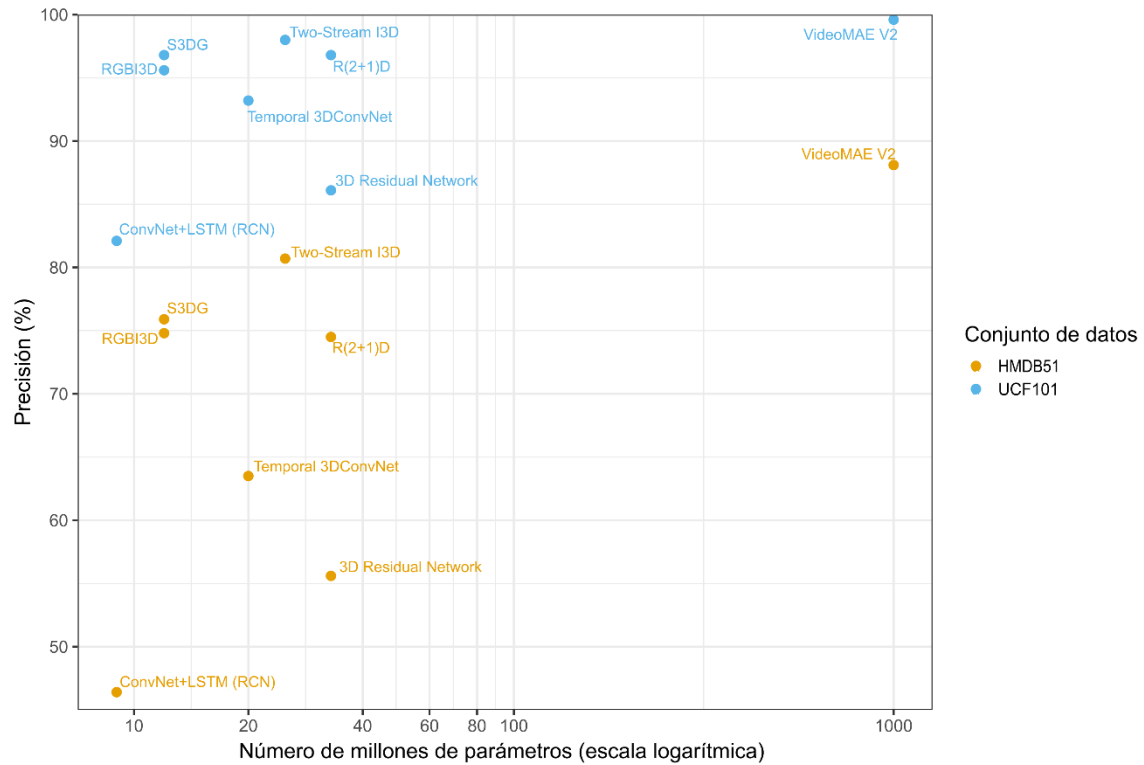


Figura 2.7. Comparativa de número de parámetros y precisión de modelos de clasificación de acciones en vídeos sobre los conjuntos de datos UCF-101 (azul) y HMDB51 (amarillo).

2.2.2 Conjuntos de datos para clasificación de acciones

Un aspecto importante en este campo de investigación es la disponibilidad de datos de entrenamiento y validación para la evaluación del sistema. Disponer de un conjunto de datos representativo y de gran diversidad es fundamental para HAR [103], ya que en general las aplicaciones de *DL* son altamente dependientes de la calidad de los datos [104]. En concreto, se han introducido varios conjuntos de datos de acciones en vídeo como *UCF-101* [98], *HMDB-51* [99], *Kinetics* [94] o *Charades* [103] para permitir el desarrollo de soluciones de reconocimiento de acciones basadas en *Deep Learning*. Entre los conjuntos de datos localizados y estudiados se encuentran los siguientes:

- **Kinetics** [94]: Es un conjunto de datos creado por *DeepMind* que contiene 600 clases de acciones humanas, con al menos 600 clips de vídeo para cada acción. Cada clip dura alrededor de 10 segundos y se toma de un vídeo diferente de YouTube.

- **UCF-101** [98]: Es un conjunto de datos compuesto por 101 acciones diferentes con más de 13 mil vídeos en total. Los vídeos han sido hechos por usuarios y contienen movimiento de cámara y un fondo desordenado al igual que sucede con Kinetics.
- **HMDB51** [99]: Este conjunto de datos ha recopilado sus vídeos de varias fuentes, principalmente de películas, y una pequeña proporción de bases de datos públicas como YouTube y Google. El conjunto de datos contiene 6.849 clips divididos en 51 categorías de acciones, cada una de las cuales contiene un mínimo de 101 vídeos.
- **Charades** [103]: Este conjunto de datos se ha construido gracias a la colaboración de cientos de personas que se han grabado videos en sus propios hogares llevando a cabo actividades cotidianas. Está compuesto por 9.848 vídeos con una duración media de 30 segundos, mostrando las actividades realizadas por 267 personas de tres continentes. Este conjunto de datos identifica 157 acciones distintas que han sido etiquetadas a lo largo de la duración del vídeo, indicándose su segundo de inicio y final.
- **Fall Detection** [105]: Es un conjunto de datos grabado en siete escenarios diferentes donde se muestran las caídas de varias personas en situaciones dispares grabado con una cámara en una posición fija. Este conjunto de datos incluye información acerca de en qué momento del tiempo exacto se produce la caída.
- **STAIR Dataset** [106]: Es un conjunto de datos que representa 100 acciones distintas que suelen ser llevadas a cabo a diario en el hogar con más de 100.000 vídeos de una duración de entre 5 y 6 segundos.

Estos grandes conjuntos de datos incluyen gran variedad de acciones humanas. Sin embargo, ninguno de estos *datasets* de forma individual contiene suficientes acciones representativas para la monitorización de hábitos. Además, presentan problemas que suponen un lastre para el reconocimiento de acciones de interior, con muestras con etiquetas erróneas, con movimientos de cámara o con un desenfoque de movimiento significativo.

En esta tesis, se ha abordado este problema siguiendo dos aproximaciones. La primera consiste en recopilar muestras de los *datasets* públicos disponibles y construir un conjunto de datos heterogéneo que incluya actividades cotidianas en el hogar, haciendo una revisión manual de los ejemplos seleccionados para paliar la baja calidad de los conjuntos. Por otro lado, otra solución es el diseño y grabación de un nuevo conjunto de datos desde cero, que incluya actividades interesantes para la monitorización de hábitos en espacios interiores en entornos similares a aquellos en los que se desplegaría el sistema. Finalmente, es importante tener presente que las arquitecturas de Aprendizaje Profundo que procesan información multimodal ofrecen un reconocimiento de acciones más robusto y preciso a través del análisis de datos heterogéneos de diversas fuentes [107]. Concretamente, hay trabajos que combinan vídeo con información de sensores inerciales [108], o audio y vídeo

[109]. En esta tesis, se propone la inclusión de información de señales vitales como es el pulso cardíaco para mejorar la identificación de situaciones críticas.

2.2.3 Módulos de Telemedicina

El uso de dispositivos de telemedicina como pueden ser módulos de electrocardiograma (EGC) o pulsioxímetros (SpO_2) contribuyen a la descentralización del cuidado de los pacientes. Mediante la monitorización remota de la salud es posible reducir los costes de hospitalizaciones y aumentar la igualdad de atención a las personas que viven en entornos aislados a través de la ciber salud [110, 111].

En esta tesis, para mejorar el reconocimiento de situaciones peligrosas se ha hecho uso de un pulsioxímetro [112], un módulo que mide el ritmo cardíaco y la saturación de oxígeno de los pacientes a través de una técnica denominada pulsioximetría [113]. La pulsioximetría es una técnica no invasiva usada para monitorizar de forma continua la saturación de oxígeno en sangre del paciente [114]. Esta medición se realiza por medio de fotoplethismografía: la sangre arterial contiene principalmente dos moléculas absorbentes de oxígeno (oxihemoglobina y la hemoglobina reducida), este método se basa en emisores de luz con dos longitudes de onda diferentes (rojo e infra-rojo) para capturar los cambios en la presión sanguínea de las arterias y para obtener información relativa a la saturación de oxígeno y ritmo cardíaco [115]. El conjunto de señales capturadas a través de este método se denomina fotoplethysmograma (*photoplethysmogram* - PPG) o señal de pulso cardíaco. En resumen, las señales PPG dan información relacionada con el sistema cardiopulmonar de las personas [114].

La información extraída por los sensores PPG puede ser analizada para una amplia variedad de aplicaciones como en: [116, 117], donde los autores analizan la señal de pulso cardíaco (PPG) para dar una estimación precisa del ritmo cardíaco; identificación biométrica [118]; reconocimiento de emociones [119]; o reconocimiento de acciones [120, 121], identificando actividades deambulatorias como: sentarse, caminar, trotar o correr.

2.3 Optimización de soluciones de aprendizaje profundo

La evolución y avances en el campo del *Deep Learning* ha llevado al desarrollo de arquitecturas precisas y complejas, comúnmente con millones de parámetros usados para procesar los datos de entrada realizando transformaciones sucesivas para dar una predicción [122]. Sin embargo, supone un gran reto implementar estas arquitecturas en dispositivos de limitados recursos como teléfonos inteligentes o sistemas empujados debido a la gran complejidad computacional y requisitos de memoria de estos grandes modelos de *Deep Learning* [123]. Por esto, la aplicación de estrategias de optimización de modelos es esencial para acelerar la inferencia de modelos muy complejos en el nodo [17].

Algunas de las ventajas de optar por el procesamiento optimizado de modelos de aprendizaje en dispositivos locales de bajos recursos son:

- Menor **latencia**, el procesamiento se realiza cerca de la fuente de datos, sin depender de servidores en la nube.
- **Privacidad**, ningún dato procesado se envía fuera de la red local.
- **Conectividad**, no se requiere de conexión a internet para funcionar.
- **Tamaño**, se reduce el tamaño del modelo para que la huella de uso de recursos, especialmente de memoria, se minimice.
- Consumo de **energía**, realizando inferencias de alta eficiencia en dispositivos de bajo consumo.

Generalmente, los modelos con un elevado número de pesos o parámetros suelen definirse como modelos altamente complejos. Sin embargo, la complejidad computacional que supone ejecutar la inferencia de un modelo de aprendizaje profundo no depende sólo del número de parámetros de la arquitectura, sino también del número de operaciones, de la resolución de los datos de entrada y de los requisitos de memoria. Una de las métricas más comunes para comparar la complejidad computacional de las arquitecturas son el número de operaciones en coma flotante (FLOPs), o número de sumas, restas y multiplicaciones que tienen que efectuarse para realizar una inferencia [124]. Por lo tanto, al comparar la complejidad de los modelos, las alternativas más eficientes serán las que consigan obtener una mayor precisión a un menor coste de FLOPs. El número de operaciones en coma flotante de los modelos depende principalmente de tres factores [60, 125]: 1) **número de parámetros** o pesos de la red, ya que los parámetros se utilizan para realizar transformaciones a los datos a través de la red; 2) tipo de operaciones o **capas de la red**, por ejemplo, a igualdad en número de parámetros, el número de operaciones en coma flotante derivadas de una capa convolucional es mayor al de una capa densa o perceptrón multicapa; 3) resolución o **tamaño de los datos de entrada** a la red, en particular, el número de sumas y multiplicaciones que tienen que llevarse a cabo por una capa convolucional con un filtro de un tamaño definido será mayor cuanto mayor sea la resolución de la imagen procesada.

En cuanto al uso de memoria, disminuir la precisión en coma flotante de los parámetros de la red contribuye a la reducción de los requisitos de memoria de los modelos. Además, en los últimos años se han introducido unidades de procesamiento, como es el caso de los TensorCores [126], para realizar un procesamiento eficiente y de altas prestaciones de los modelos en formato *half-precision* (usando tan solo 16 bits para representar cada número en coma flotante) o en precisión de enteros de 8 bits.

2.3.1 Técnicas de compresión y aceleración de modelos

Para acelerar la inferencia de modelos complejos en dispositivos integrados de bajos recursos se han introducido diversas técnicas y estrategias de compresión y aceleración de

modelos como la cuantización [127], poda (*pruning*) [128], o destilación de conocimiento [129].

2.3.1.1 Cuantización

La técnica de cuantización consiste, principalmente, en convertir los parámetros de la red de coma flotante (32 bits) a un formato de menor precisión, como, por ejemplo, números enteros (8 bits) [130]. De esta manera, se reduce la complejidad computacional de las operaciones, pero también la precisión aritmética. Con números en coma flotante de 32 bits es posible representar más de 4 mil millones de posibles valores, lo cual ofrece un amplio rango dinámico de valores para los pesos de la red. Sin embargo, con una precisión aritmética de enteros de 8 bits solo es posible representar 256 valores. El proceso de cuantización consiste en establecer una correspondencia entre ambas precisiones aritméticas, escalando el vector de coma flotante a enteros de 8 bits en el rango de $[-128, 127]$ (ver Figura 2.8). Esta estrategia reduce el tamaño del modelo, el uso de memoria y el coste de la inferencia a expensas de una pérdida de precisión [131]. Para minimizar la pérdida de precisión es posible utilizar un conjunto de datos representativo del problema y ejecutar varios ciclos de inferencia para calibrar cuál es la representación en 8 bits que mejor se ajusta a los pesos, activaciones y datos de entrada en función de la distribución de valores de los tensores de la red [132].

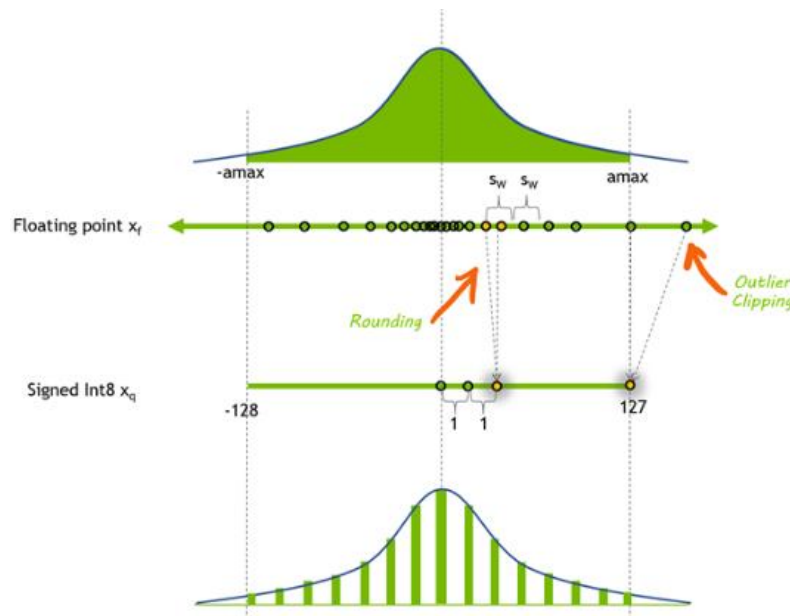


Figura 2.8. Ejemplo de cuantización de vector en coma flotante a un entero de 8 bits [133].

2.3.1.2 Poda de conexiones

La poda de conexiones o pesos en la red (*pruning*) es otra de las técnicas más populares en la optimización de modelos complejos [128]. Esta estrategia busca aumentar la dispersión de los valores de los pesos del modelo durante el proceso de entrenamiento. Se eliminan los parámetros (estableciendo su valor a 0) que tienen un impacto poco

significativo en el resultado de las predicciones del modelo, a costa de una leve disminución de la precisión [134]. De esta manera, se reduce el espacio que ocupa el modelo en disco. Además, la eliminación de conexiones entre neuronas abre la posibilidad de reducir la latencia de la inferencia, omitiendo el procesamiento de las operaciones donde los pesos de la capa sean igual a 0 (ver Figura 2.9). Sin embargo, actualmente, no hay herramientas de inferencia de modelos que saquen provecho a esta propiedad de los modelos podados, por lo que procesan la totalidad de operaciones en la red [134, 135].

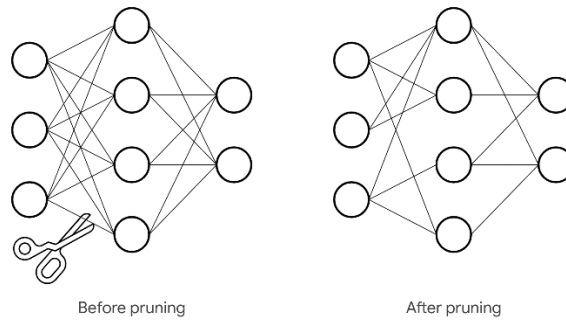


Figura 2.9. Ejemplo de podado (pruning) de conexiones en la red [136].

2.3.1.3 Destilación de conocimiento

La destilación de conocimiento es una de las técnicas más populares de optimización de modelos para permitir el despliegue de soluciones eficientes y precisas con rendimiento en tiempo real en dispositivos de bajas prestaciones. Por ejemplo, los autores en [137] se basan en la destilación de conocimiento para producir modelos precisos y ligeros para el posicionamiento visual que se ejecuta sobre vehículos aéreos no tripulados con limitados recursos. Otro ejemplo se describe en [138], donde se presenta un modelo eficiente de estimación de pose humana que opera en tiempo real en dispositivos empujados de bajo consumo. Finalmente, en [139] se desarrolla un módulo de estimación de profundidad que se ejecuta en teléfonos móviles apoyándose en la destilación de conocimiento.

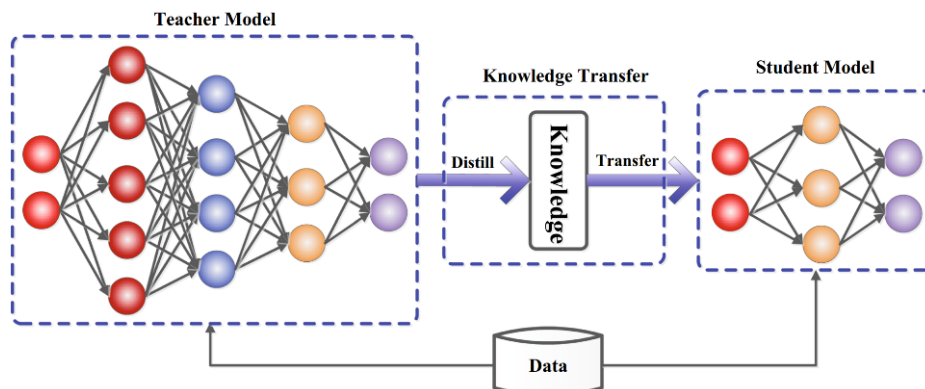


Figura 2.10. Destilación de conocimiento. Se transfiere el conocimiento de un modelo grande y complejo, denominado Maestro (Teacher) a un modelo más simple y eficiente Estudiante (Student) [123].

La destilación de conocimiento se basa en la transferencia de conocimiento desde un modelo complejo a uno más simple y eficiente (ver Figura 2.10) [129]. El proceso consiste en ajustar los pesos del modelo más simple, también llamado Estudiante (*Student*) para que el comportamiento y las predicciones sean lo más parecidas posible a las del modelo complejo o Maestro (*Teacher*) [140]. El procedimiento seguido en esta técnica se compone principalmente de 2 fases: 1) Se preentrena la arquitectura *Teacher* sobre el conjunto de datos deseado; 2) Se congelan los pesos del modelo *Teacher* y se optimizan los parámetros del modelo *Student* para que la distribución de probabilidad de las predicciones ante una misma entrada sean lo más parecidas posible. Esta metodología contribuye a mejorar la capacidad de reconocimiento del modelo más eficiente en comparación con el rendimiento que se obtendría al solo ajustarlo con los datos de entrenamiento. Esto es posible por medio del uso de *soft targets*. Los *soft-targets* se obtienen al definir un valor de temperatura (τ) en la función *softmax*. La función *softmax* (1) transforma la salida sin normalizar de la última capa de la red en una distribución de probabilidad con valores que suman 1.

$$q_i = \frac{e^{z_i/\tau}}{\sum_j e^{z_j/\tau}} \quad (1)$$

La función *softmax* estándar es equivalente a establecer $\tau = 1$. Para realizar el proceso de destilación se utilizan valores de temperatura más elevados, para así construir *targets* o vectores de predicción con una distribución suavizada (*soft target*). Los *soft-targets* proveen información acerca de las relaciones inter-clases extraídas por el modelo *Teacher*, lo cual es de gran utilidad para ayudar al modelo Estudiante a aprender a cómo generalizar [129].

Una de las ventajas de la destilación de conocimiento en comparación con las otras estrategias, es que permite optimizar la solución sin comprometer la capacidad de reconocimiento, de forma no dependiente a su arquitectura [140]. Es decir, una agrupación de grandes modelos con entradas multimodales puede destilar conocimiento a una red mucho más simple como MobileNet [141].

2.3.2 Herramientas de optimización de modelos

Paralelamente, se han introducido diversos marcos de trabajo y herramientas como Tensorflow Lite [135], OpenVINO [142] o TensorRT [143] para facilitar la ejecución de modelos complejos en dispositivos con recursos de procesamiento y memoria limitados, como aparatos móviles o dispositivos *SoC* e *IoT*. Tensorflow Lite es una biblioteca desarrollada por Google para la implementación de modelos en dispositivos de bajos recursos. Los modelos optimizados con Tensorflow Lite son compatibles con la aceleración por hardware y tienen una huella de memoria más baja y una inferencia más rápida en comparación con los modelos originales. Algunas de las técnicas utilizadas por Tensorflow

Lite para la optimización de los modelos son la cuantización [127] y la poda de pesos (*pruning*) [128].

Por otro lado, OpenVINO [142] introduce una herramienta para la optimización y despliegue de modelos entrenados en cualquiera de los marcos de trabajos existentes como Tensorflow o PyTorch. OpenVINO optimiza los modelos para obtener el mejor rendimiento posible en plataformas *hardware* de Intel, como procesadores, GPUs o FPGAs. Esta biblioteca optimiza los modelos de aprendizaje profundo mediante la fusión de grafos de computación, la reutilización de memoria, o la paralelización en las inferencias. En el proceso de optimización también se utilizan técnicas de compresión de las arquitecturas como la cuantización y la poda de pesos para reducir el tamaño del modelo y su uso de memoria para su despliegue en *hardware* de limitados recursos.

Por último, TensorRT [143], desarrollado por NVIDIA, es una de las bibliotecas más populares utilizada para el despliegue de modelos de manera eficiente. TensorRT es un entorno de ejecución e inferencia de alto rendimiento que ofrece baja latencia y una capacidad de procesamiento elevada para aplicaciones de inferencia de aprendizaje profundo. TensorRT permite optimizar las arquitecturas de aprendizaje automático reduciendo el uso de memoria y consumo energético de los modelos, limitando la precisión en coma flotante de las operaciones de la red, combinando operaciones y capas o seleccionando los mejores hiperparámetros para las capas de la red en función de las características de la GPU donde se desplegará el modelo. En concreto, el uso de TensorRT en dispositivos integrados de bajos recursos como la Nvidia Jetson Nano puede reducir el tiempo de inferencia hasta en un 40% [144].

3 Metodología

En este trabajo se propone el diseño y desarrollo de un sistema ciberfísico reconfigurable para la monitorización en tiempo real de actividades cotidianas de personas de avanzada edad o con necesidades de atención para promover la autonomía y mejora de la calidad de vida (ver Figura 3.1). El procedimiento seguido en esta tesis para el diseño e implementación del sistema se compone de dos tareas principales: a) desarrollo de módulo de reconocimiento de acciones; b) diseño de sistema ciberfísico reconfigurable con integración de módulo de monitorización y gestión de calidades y recursos.

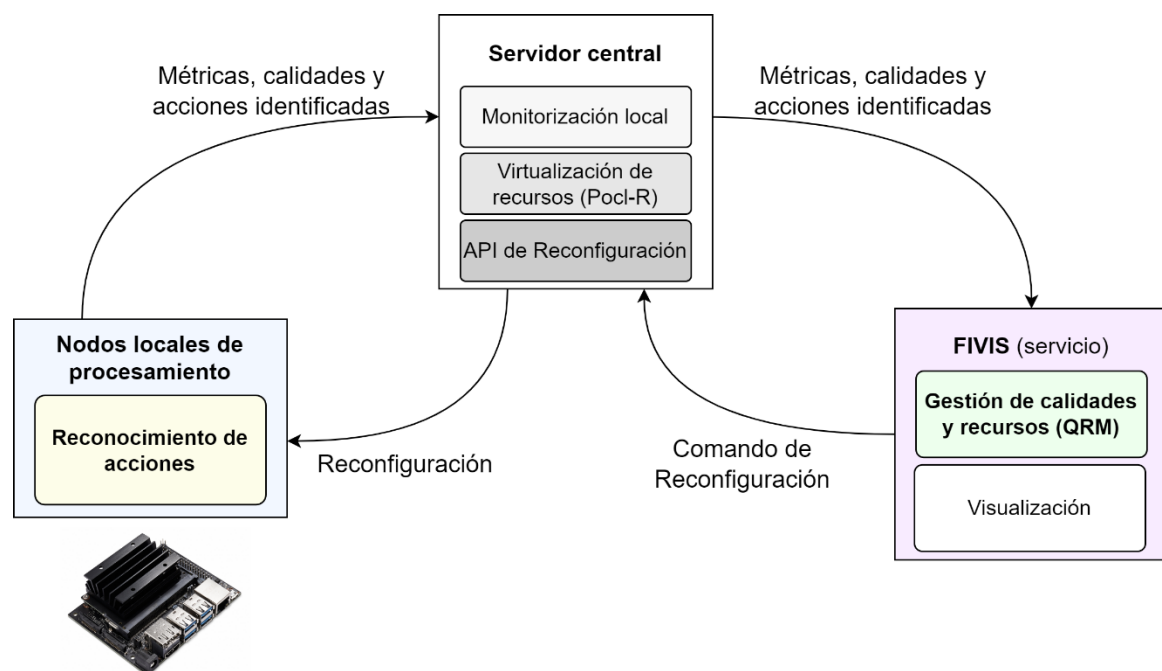


Figura 3.1. Visión general de relación entre principales componentes del sistema ciberfísico de monitorización de actividades.

El módulo de **reconocimiento de acciones** es uno de los componentes principales del sistema. Este componente es el encargado de, a partir del análisis de una o varias fuentes de datos, efectuar la identificación de las acciones realizadas por las personas monitorizadas. El desempeño de este componente tanto en términos de bondad de ajuste como de rendimiento depende principalmente de los siguientes factores: 1) **calidad de los datos**, disponer de un conjunto de datos de calidad y representativo del problema es uno de los elementos claves para que los modelos de aprendizaje automático tengan un buen desempeño en términos de evaluación; 2) selección y diseño de arquitectura de **aprendizaje automático**, ya que el modelo tiene un impacto directo sobre el compromiso entre rendimiento y bondad de ajuste que ofrece la solución; 3) **optimización** de modelos de aprendizaje profundo, buscando una mayor eficiencia de las arquitecturas sin comprometer significativamente la precisión de la solución, permitiendo la ejecución en tiempo real en nodos locales de limitados recursos.

Una vez desarrollado el módulo de reconocimiento de acciones, buscando maximizar la precisión de reconocimiento de situaciones críticas y la optimización de las arquitecturas, se integra el componente en un sistema ciberfísico. El diseño del sistema ciberfísico reconfigurable de arquitectura nodo-nube se apoya en la integración de una herramienta de monitorización y Gestión de Calidades y Recursos (QRM). Este componente tiene como función principal monitorizar y analizar la información generada por el sistema para tomar decisiones de reconfiguración inteligentes. El despliegue de reconfiguraciones inteligentes contribuye a hacer un uso optimizado de los escasos recursos disponibles, maximizando el tiempo de funcionamiento y capacidad de reconocimiento del sistema.

3.1 Arquitecturas hardware

El equipo utilizado para el desarrollo e implementación del trabajo presentado en esta tesis ha constado de computadores de altas prestaciones para el entrenamiento de modelos de aprendizaje profundo y nodos de procesamiento empotrados para el análisis de las actividades en la periferia de la nube.

3.1.1 Servidor central

Uno de los equipos utilizados para el desarrollo de esta tesis ha sido un computador con una GPU de altas prestaciones. Este equipo tiene las siguientes especificaciones: 6 núcleos de CPU (Intel i5-8400 CPU @2,8 Ghz), 32Gb de memoria principal (RAM), 2TB de almacenamiento en disco HDD y una GPU Nvidia RTX 2080Ti. La RTX 2080Ti es una unidad de procesamiento gráfico que dispone de 4352 núcleos CUDA y 11 GB de VRAM.

La cantidad de núcleos que posee el procesador de este ordenador ha sido de utilidad para ejecutar tareas utilizando multiprocesamiento y obtener resultados con una rapidez mayor que si se dispusiera de tan solo un núcleo. Por otra parte, un componente esencial para ser capaz de entrenar y realizar inferencia sobre los modelos de redes neuronales a una velocidad razonable ha sido la tarjeta gráfica. Esta tarjeta gráfica es la GPU de Nvidia con más núcleos CUDA de la serie 2000. También ha sido de vital importancia contar con 11GB de VRAM, ya que en varias ocasiones el entrenamiento de algunos modelos ha llegado a superar los 10GB de utilización. Por otra parte, la memoria secundaria disponible ha permitido recopilar varios conjuntos de datos de vídeos de acciones, así como realizar procesamiento sobre los mismos y almacenar el resultado para ser computados posteriormente por una red.

En el sistema, este servidor central se encarga de las tareas de control dentro del CPS. Recopilando las métricas y calidades del sistema a nivel de los nodos, actuando como pasarela con la herramienta QRM. A su vez, la alta disponibilidad de recursos de computación permite la ejecución de arquitecturas de aprendizaje profundo complejas de gran precisión de manera puntual con la finalidad de confirmar situaciones críticas.

3.1.2 Nodos de procesamiento local

El procesamiento distribuido en el sistema ciberfísico se realiza utilizando dispositivos SoC empotrados para el análisis de actividades eficiente cerca de la fuente de los datos. En concreto, se han integrado en el sistema dos tipos de nodos de la familia Jetson de Nvidia. La Jetson TX2 y la Jetson Nano. Como primera aproximación, se integran en el sistema nodos Jetson TX2, habilitando la monitorización eficiente de actividades. Posteriormente, se integran en el sistema nodos Jetson Nano, los cuales son dispositivos empotrados aún más eficientes computacionalmente y con recursos más limitados.

3.1.2.1 Nvidia Jetson TX2

La Nvidia Jetson TX2 es un dispositivo SoC introducido por Nvidia para el procesamiento eficiente de soluciones de aprendizaje automático. Este módulo ofrece un rendimiento razonable en tareas de aprendizaje profundo consumiendo una cantidad de recursos y energía relativamente bajas. Esto es posible gracias a que integra una GPU de 256 núcleos con arquitectura Pascal y 6 núcleos de CPU ARM de 64 bits. Además, cuenta con 8 GB de memoria RAM compartida entre CPU y GPU que da la posibilidad de ejecutar soluciones de IA requieren un elevado consumo de memoria.

Además de eso, permite cumplir con uno de los objetivos establecidos para esta tesis, el cual es desarrollar un sistema de monitorización de actividades garantizando la privacidad de los individuos. Esto es debido a que este dispositivo puede instalarse en cualquier hogar o residencia y efectuar el análisis del vídeo cerca de la fuente de los datos, sin ser necesario enviar información o el vídeo capturado fuera de la red local.



Figura 3.2 Fotografía Jetson TX2.

En la Figura 3.2 se muestra una fotografía del dispositivo en cuestión donde se pueden apreciar sus principales componentes, así como su tamaño.

3.1.2.2 Nvidia Jetson Nano

La Jetson Nano es el dispositivo empotrado para el procesamiento de soluciones de aprendizaje automático más eficiente presentado por Nvidia. Este módulo de alta eficiencia

energética es capaz de ejecutar sistemas de visión por computador o procesamiento de lenguaje natural consumiendo menos de 5 W (vatios). Este dispositivo integra una GPU de tan solo 128 núcleos CUDA de arquitectura Maxwell y una CPU de 4 núcleos ARM [145]. Además, cuenta con tan solo 4GB de memoria RAM compartida entre CPU y GPU, lo que supone una limitación importante al implementar modelos de elevada complejidad en el nodo.

A diferencia de la Jetson TX2, este dispositivo tiene una GPU con unas prestaciones inferiores y una arquitectura más antigua (Maxwell en lugar de Pascal). Como se puede observar en la Figura 3.3, la Jetson Nano es el módulo más compacto de los productos de la familia Jetson de Nvidia.



Figura 3.3. Fotografía de Jetson Nano.

Este dispositivo integrado permite la monitorización de personas a través del procesamiento local, garantizando así la privacidad de los individuos. Sin embargo, dada la alta complejidad de las arquitecturas de aprendizaje profundo del estado del arte para el reconocimiento de acciones, supone un gran reto que el sistema opere en tiempo real.

Este módulo ofrece por defecto 2 modos de operación predefinidos. En uno de ellos se limita la frecuencia máxima de funcionamiento de los componentes hardware del dispositivo (CPU, GPU y RAM), reduciendo así el consumo del dispositivo a cambio de limitar sus prestaciones. El otro modo de funcionamiento utiliza todos los componentes a la máxima frecuencia para obtener el máximo rendimiento.

3.1.3 Módulo de telemedicina

Para mejorar el reconocimiento de acciones críticas, se propone el uso de información multimodal de vídeo y pulso cardíaco. Para ello, se ha seleccionado el dispositivo pulsioxímetro inalámbrico desarrollado por la empresa *RGB Medical Devices* [112] para medir las señales vitales de los sujetos de una forma no invasiva. Este módulo de SpO₂ monitoriza la señal de pulso cardíaco, que provee información acerca de cambios en la presión de los vasos sanguíneos. Su frecuencia de muestreo es de 66,67 Hz (cada medida se extrae aproximadamente cada 15 milisegundos).

Basándose en esta señal fisiológica, el módulo es capaz de extraer dos medidas principales: 1) Saturación de oxígeno en sangre (0-100%); y 2) Pulso cardíaco (30-250 pulsaciones por minuto - ppm). Este módulo de telemedicina ofrece una alta resolución y precisión en sus mediciones. En cuanto a los valores de saturación de oxígeno en sangre, ofrece una resolución de un 1% con una exactitud de ± 2 dígitos en el rango de 70-100% SpO₂; para el ritmo cardíaco, la resolución es de 1 ppm con una exactitud de $\pm 3\%$. Adicionalmente, este dispositivo puede alertar al usuario cuando el sensor se ha desconectado o la señal es débil.

En este trabajo, se analiza la señal PPG (pulso cardíaco) para identificar acciones cotidianas. Si bien es cierto que otros dispositivos también son capaces de proveer esta señal de pulso cardíaco, este módulo SpO₂ es también capaz de monitorizar la saturación de oxígeno, permitiendo hacer una detección temprana de hipoxemia, que consiste en un valor anormalmente bajo de oxígeno en sangre ($< 93\%$) [113].

3.2 Módulo de reconocimiento de acciones

En la Figura 3.4 se muestra la metodología y tareas definidas para la elaboración del módulo de reconocimiento de acciones. El primer paso para el diseño del componente de reconocimiento de acciones es la construcción del **conjunto de datos**. Seguidamente, se realiza un estudio del estado de la técnica en clasificación de acciones en vídeo para seleccionar y diseñar las arquitecturas de aprendizaje profundo que mejor se ajustan para resolver el problema de identificación de actividades cotidianas en espacios interiores. Una vez definidos los conjuntos de datos y los modelos de aprendizaje automático que se integrarán en el trabajo, se procede con el entrenamiento de los modelos. En esta etapa se establece la metodología a seguir para el entrenamiento de las arquitecturas, aportando indicaciones acerca del preprocesamiento de los datos, de los hiperparámetros de ajuste, así como de las distintas fases en las que se efectúa la optimización de los pesos de la red.

Por otra parte, para impulsar el reconocimiento de situaciones **potencialmente peligrosas**, se propone la utilización de información **multimodal**, de vídeo y de estimación de movimiento, o bien, de vídeo y de señales vitales de los pacientes. Se pretende que a través de la monitorización la información multimodal mejore la precisión de la solución, particularmente para la identificación de situaciones críticas.

A continuación, con la finalidad de minimizar el impacto en términos de recursos de las arquitecturas propuestas, se define un procedimiento a través de la **destilación de conocimiento** para optimizar el uso de recursos de las soluciones a expensas de un leve impacto en la precisión.

Finalmente, se despliegan las arquitecturas en los nodos locales haciendo uso de herramientas como TensorRT [146] o Triton [147] para la implementación eficiente de los modelos en el nodo, acelerando el procesamiento mediante TensorRT y con un uso

optimizado de recursos de memoria mediante el servidor de inferencia Triton. Además, se introduce una estrategia para la **asignación eficiente de tareas a recursos** con la finalidad de orquestar el análisis de las acciones llevadas a cabo por los sujetos entre los nodos de procesamiento disponibles.

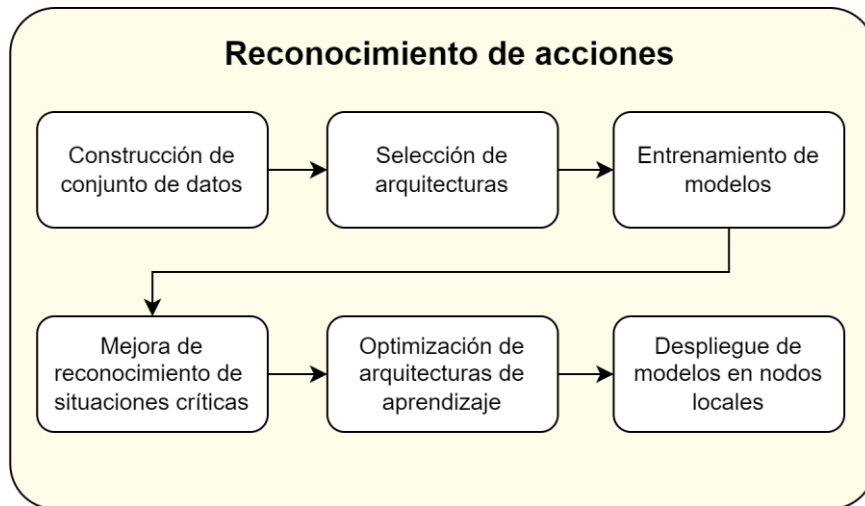


Figura 3.4. Metodología para desarrollo del módulo de reconocimiento de acciones.

3.2.1 Construcción de conjunto de datos

Tal y como se ha introducido en el epígrafe 2.2.2, existen varios conjuntos de datos disponibles para la clasificación de acciones en vídeo de carácter general. Sin embargo, de forma individual no contienen suficientes acciones representativas para la monitorización de actividades cotidianas.

Por esta razón, se construyen conjuntos de datos a medida, para la clasificación de acciones en el hogar a partir de vídeo. En concreto, se siguen dos aproximaciones diferenciadas. Por un lado, se construye un conjunto de datos seleccionando muestras de *datasets* públicos de actividades. Por otro lado, se procede con la creación de un conjunto de datos multimodal que incluye información de vídeo y pulso cardíaco, de forma que el análisis de datos de ambas fuentes permita mejorar la capacidad de reconocimiento de las soluciones. En ambos casos, se incluyen actividades cotidianas que son de utilidad para la monitorización de hábitos y la asistencia en el hogar.

En primer lugar, se seleccionan acciones críticas como las caídas. Las caídas suelen tener consecuencias graves en personas de avanzada edad, y actuar lo antes posible en estas situaciones es crucial para disminuir los efectos perjudiciales que puede suponer una demora en la atención.

Por otra parte, detectar si una persona está comiendo es también interesante para extraer de forma superficial sus patrones de alimentación.

A su vez, reconocer cuando se está limpiando el suelo puede ser importante, ya que en el caso de que la persona lo esté fregando y el piso esté mojado, pueden aumentar las

posibilidades de que se produzca una caída. Resulta también interesante conocer con qué frecuencia y cuánto tiempo dedican al día los sujetos a ver la televisión. De esta manera, en el caso de que sus rutinas sean muy sedentarias, se les podría animar a tener una vida más activa, si su salud lo permite. Todo esto con la finalidad de promover un buen estado de salud en cada individuo.

Por último, acciones como caminar, sentarse o levantarse pueden ser también relevantes para conocer el estilo de vida de la persona en su hogar y por qué localizaciones de su casa se suele desplazar.

3.2.1.1 CIAV Dataset

Para abordar el problema de escasez de datos representativos de actividades cotidianas se recopilan clips a partir de diversos conjuntos de datos, seleccionando 18 acciones consideradas relevantes para la monitorización en interiores y construir el *Combined Indoor Action Video (CIAV) dataset* (ver Tabla 3.1).

Las acciones y los ejemplos fueron recopilados de 6 conjuntos de datos de reconocimiento de acciones a partir de vídeo: *HMDB51* [99], *UCF-101* [98], *Fall Detection* [105], *Charades* [103], y *Kinetics* [94]. De esta forma, el nuevo conjunto incluye varias acciones de interés para la monitorización de hábitos. Para desempeñar la monitorización de actividades cotidianas en el hogar es crucial la construcción de este conjunto de datos, ya que actualmente no hay disponibles *datasets* que incluyan una cantidad de acciones razonable de utilidad para la monitorización de estilos de vida. Además, el combinar instancias de diferentes fuentes de datos permite la creación de un *dataset* heterogéneo que representa clases con diferentes actores y escenas, contribuyendo a una mejor capacidad de generalización de la solución (ver Figura 3.5).

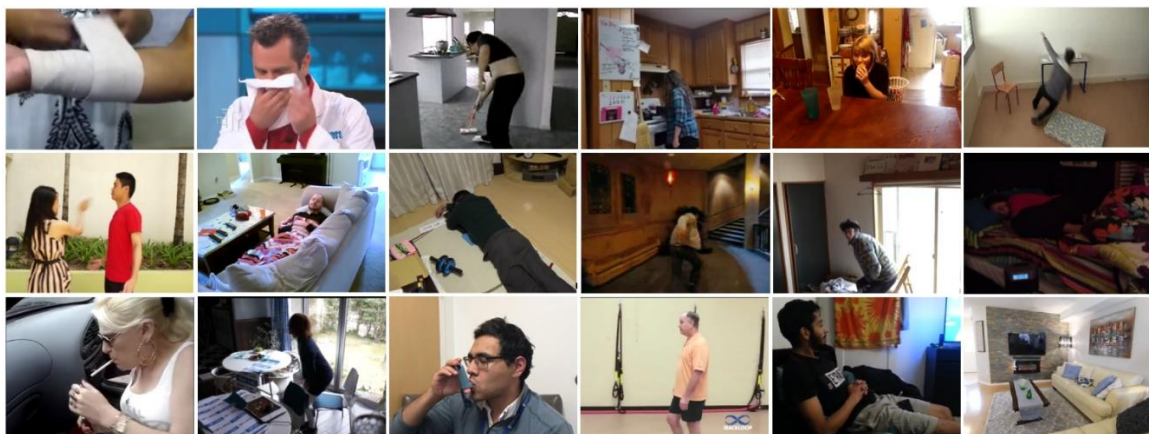


Figura 3.5. Muestras del CIAV dataset. Cada ejemplo está extraído de cada una de las clases mostradas en la Tabla 3.1, ordenadas de izquierda a derecha y de la parte superior a la inferior.

Dado que algunas acciones disponen de un número bajo de ejemplos, se etiquetan nuevas muestras de forma manual a partir de vídeos de YouTube. También se ha añadido

una clase denominada “no acción” a partir de las siguientes fuentes: 1) vídeos generados a partir del conjunto de datos *Indoor Scene Recognition* [148], que incluye espacios del interior del hogar sin personas; 2) vídeos de YouTube en el que se muestran tours virtuales por diversos hogares (alrededor de 950 clips). Los clips de vídeo que representan esta acción solo incluyen escenas en espacios interiores sin personas. Con la inclusión de esta clase se busca asistir al modelo a relacionar la presencia de los sujetos con las acciones, evitando así que el modelo tome las decisiones sesgadas solo en función de la escena o el entorno.

Tabla 3.1. Muestras por clase del CIAV dataset creado a medida

Acción	Entrenamiento	Validación	Test	Total
Vendarse (<i>bandaging</i>)	619	48	99	766
Sonarse o estornudar (<i>blowing nose or sneezing</i>)	1140	99	187	1426
Limpiar el suelo (<i>cleaning floor</i>)	1876	160	298	2334
Cocinar (<i>cooking</i>)	1965	186	332	2483
Comer (<i>eating</i>)	1183	165	215	1563
Caerse (<i>falling</i>)	808	72	121	1001
Golpear (<i>hitting</i>)	1014	74	120	1208
Tumbarse en la cama o sofá (<i>lying on bed or sofa</i>)	300	39	40	379
Estar tumbado en el suelo (<i>lying on the floor</i>)	1186	148	148	1482
Correr (<i>running</i>)	403	48	49	500
Sentarse (<i>sitting</i>)	857	103	104	1064
Dormir (<i>sleeping</i>)	574	47	96	717
Fumar (<i>smoking</i>)	895	54	81	1030
Ponerse de pie (<i>standing up</i>)	549	65	66	680
Usar inhalador (<i>using inhaler</i>)	533	46	96	675
Caminar (<i>walking</i>)	1212	147	148	1507
Ver la televisión (<i>watching tv</i>)	541	48	95	684
No acción (<i>no accion</i>)	1500	200	200	1900
Total	17155	1749	2495	21399

El conjunto de datos dispone de un total de 21.399 vídeos, con una duración media de 10 segundos cada uno, este *dataset* se divide en los siguientes subconjuntos, entrenamiento (80%), validación (8%) y test (12%). Dada la diferente naturaleza de los conjuntos de datos originales, el *dataset CIAV* está altamente desbalanceado, con algunas clases conteniendo significativamente más ejemplos que otros. Por ejemplo, se incluyen más de 1.800 muestras de la clase *comer* y menos de 600 instancias de *ver la televisión*.

Se ha compartido públicamente un repositorio¹ para construir el conjunto de datos descrito a partir de las muestras de los *datasets* públicos utilizados. De esta forma, se busca facilitar el acceso a datos relacionados con la identificación de actividades cotidianas en espacios interiores para futuros trabajos de investigación.

3.2.1.2 Conjunto de datos multimodal (IAPV)

Debido a que los sistemas basados en Aprendizaje Automático aprenden a partir del análisis de ejemplos, uno de los factores más importantes cuando se construyen este tipo de soluciones es la disponibilidad de conjuntos de datos que se encuadren en el campo de aplicación. La cantidad y calidad de los datos es crucial y tiene un impacto directo en la capacidad de reconocimiento del sistema [149]. Actualmente, existen conjuntos de datos para realizar reconocimiento de acciones a partir de vídeo [94, 150], y otros a partir de pulso cardíaco [116]. Sin embargo, no hay disponibles conjuntos de datos que combinen información de **ambas modalidades** de forma **sincronizada** para el reconocimiento de acciones.

Tabla 3.2. Número de clips del conjunto de datos IAPV

Acción	Entrenamiento	Validación	Test	Total
Sonarse o estornudar	38	10	15	63
Limpiar	55	17	34	106
Comer	57	18	32	107
Caerse	34	11	16	61
Estar tumbado en el suelo	53	12	36	101
Sentarse	66	22	29	117
Ponerse de pie	95	30	41	166
Caminar	158	34	90	282
Ver la televisión	60	19	43	122
No acción	57	14	32	103
Total	673	187	368	1228

¹ <https://github.com/DaniDeniz/CIAV-dataset>

Se pretende que la información multimodal ayude a mejorar la identificación de situaciones críticas y pueda servir de apoyo a los profesionales sanitarios en la evaluación de la relación entre las constantes vitales de los individuos y las actividades realizadas.

Para solventar este problema, se construye un conjunto de datos multimodal para reconocimiento de actividades cotidianas (*Indoor Action Pulses and Video - IAPV*), recopilando datos de vídeo y pulso cardíaco de forma sincronizada. Este conjunto de datos contiene personas realizando diversas acciones en espacios interiores del hogar como pueden ser: habitaciones, salas de estar o cocinas. El conjunto de datos fue grabado por 5 sujetos (3 hombres y 2 mujeres) en el hogar. Las acciones se grabaron en bloques continuos de unos 10 minutos, usando el módulo de telemedicina de *RGB Medical* (pulsioxímetro – SpO₂) para la medición del pulso cardíaco.

Los actores grabaron diversas acciones de interés para la monitorización de hábitos como puede ser *limpiar, comer, sentarse, caminar o ver la televisión*. También, se incluyeron acciones críticas pueden ser las caídas o estar tumbado en el suelo, ya que son de especial relevancia para identificar si un sujeto ha sufrido una situación peligrosa y requiere asistencia. En la Tabla 3.2 se detallan las acciones incluidas en el conjunto de datos y el número de clips grabados. Por otro lado, en la Figura 3.6 se ilustran 3 ejemplos de algunas de estas acciones grabadas por los actores.

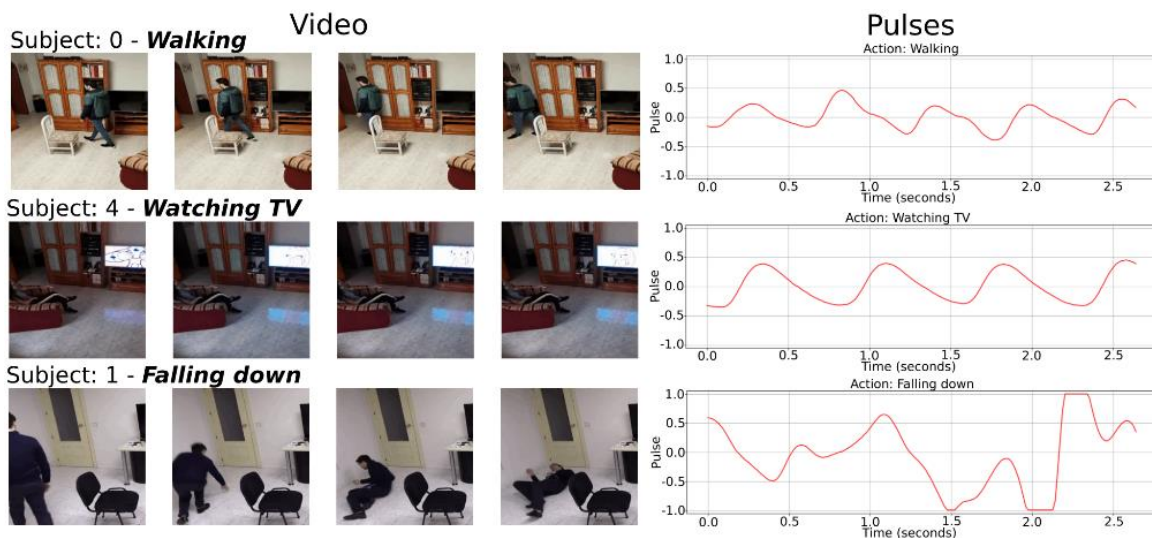


Figura 3.6. Ejemplos del conjunto de datos IAPV [151]. Se muestran a tres sujetos realizando las siguientes acciones: caminar, ver la televisión y sufrir una caída. Obsérvese como los pulsos están sincronizados con el vídeo y cómo varían de forma significativa dependiendo de cada acción.

Con respecto a la preparación de los datos, en primer lugar, los vídeos se fragmentan en clips individuales y estos últimos se etiquetan de forma manual. En segundo lugar, la información de pulso cardíaco se etiqueta de forma automática ya que están **sincronizados** con el vídeo. En la Tabla 3.2 se muestran las nueve etiquetas de acciones incluidas en el conjunto de datos. La clase *no acción* muestra escenas de habitaciones vacías sin personas

y contribuye a reducir el **sesgo** de los modelos por los escenarios donde se realizan las acciones. Esta clase no se tuvo en cuenta en la parte del conjunto de datos asociado a la información de pulso cardíaco (ya que ninguna persona participa en esta clase).

Las instancias multimodales se asignaron manualmente a cada subconjunto de la siguiente forma: 55% para entrenamiento, 15% en validación y 30% para test. Los datos se dividen en estos subconjuntos de forma cuidadosa para garantizar que todos los clips extraídos de un mismo vídeo se asignan al mismo subconjunto. Asegurando así la fidelidad del experimento.

Por otro lado, en la Tabla 3.3 se muestra el número de clips grabados por cada actor. El identificador del usuario (ID) se puede utilizar para efectuar el análisis individual del rendimiento del sistema para cada usuario. De esta manera, es posible medir el impacto en las prestaciones de las soluciones en función de los datos disponibles de cada sujeto.

Tabla 3.3. Número de clips grabados por cada sujeto

ID Sujeto	Género	Entrenamiento	Validación	Test	Total
0	Hombre	406	92	123	621
1	Hombre	32	0	85	117
2	Hombre	70	25	29	124
3	Mujer	70	0	63	133
4	Mujer	95	70	68	233

3.2.1.3 Indoor Action Dataset

El *Indoor Action Dataset* representa la modalidad de vídeo del conjunto de datos multimodal presentado en el anterior epígrafe 3.2.1.2 (IAPV). La distribución de los datos es la misma que la presentada en la sección anterior, solo que en este caso se trabaja tan solo con la información de vídeo.

Este conjunto de datos se ha **liberado** públicamente en GitHub² para la comunidad científica. De modo que el desarrollo de soluciones de reconocimiento de acciones para la monitorización de actividades cotidianas pueda ampliarse e impulsarse en otros trabajos.

3.2.1.4 Métricas de evaluación

Los conjuntos de datos con los que se trabaja en esta tesis están altamente desbalanceados, por lo que se seleccionan las métricas de evaluación más adecuadas para lidiar con esta situación. Las métricas utilizadas son la sensibilidad o *recall* ($\frac{VP}{VP+FP}$), el valor

² <https://github.com/DaniDeniz/IndoorActionDataset>

predictivo positivo o *precision* ($\frac{VP}{VP+FN}$) y el valor de F1-Score, que equivale a la media armónica entre los valores *precision* y *recall*. Además, debido a la naturaleza multi clase de este trabajo, se calculan las macro-métricas, que se obtienen calculando la media aritmética por clase de las métricas previamente mencionadas, haciendo así que todas las clases tengan la misma importancia en la evaluación. Por motivos de simplicidad, a partir de ahora nos referiremos a estas **macro**-métricas simplemente como *precision*, *recall* y F1-Score.

3.2.2 Selección de arquitecturas

Una vez definidos los conjuntos de datos representativos para la clasificación de acciones cotidianas en interiores se procede con la selección y diseño de las arquitecturas de aprendizaje profundo.

En primer lugar, se realiza un estudio del estado del arte para conocer el desempeño de las distintas arquitecturas en el reconocimiento de acciones. Principalmente, se tienen dos factores en cuenta a la hora de seleccionar las arquitecturas que mejor se ajustan a los requisitos de este trabajo: 1) precisión de las soluciones, se busca diseñar modelos con elevada capacidad de reconocimiento de acciones en vídeo, con prestaciones a nivel del estado de la técnica; 2) coste computacional de las arquitecturas, dado que el reconocimiento de acciones se llevará a cabo en nodos locales de procesamiento de limitados recursos, es crucial seleccionar aquellas arquitecturas que ofrezcan un mejor compromiso entre consumo de recursos y precisión. Debido a que soluciones del estado del arte de gran precisión como los Transformers no podrían empotrarse en los nodos de procesamiento y ofrecer un rendimiento en tiempo real, en este trabajo se optan por soluciones basadas en el uso de convoluciones 3D y redes convolucionales recurrentes.

Para el diseño de las arquitecturas se tienen presente calidades como la precisión, rendimiento (en fotogramas por segundo), consumo energético y tamaño o complejidad del modelo. Estando esto último determinado por el número de parámetros y número de operaciones en coma flotante del modelo.

Se selecciona la arquitectura *RGBI3D* como el esqueleto principal para las redes propuestas en este trabajo. Una de las ventajas de usar esta arquitectura es que se pueden utilizar los parámetros preentrenados sobre el conjunto de datos de gran tamaño Kinetics600 [152]. Esto habilita la posibilidad de entrenar los modelos de HAR en interiores en una cantidad de tiempo razonable en una sola GPU (RTX 2080 Ti).

En este trabajo, se proponen cuatro aproximaciones principales para el Reconocimiento de Acciones Humanas (dos de ellas se basan en el modelo *RGBI3D* mencionado anteriormente):

RGBI3D: Una de las arquitecturas seleccionadas es el modelo *RGBI3D*. La red *RGBI3D* es la arquitectura de red utilizada para procesar la información de vídeo en el modelo *TwoStream I3D*. Es una versión 3D del modelo *Inceptionv1* [122]. Este modelo se

compone de bloques *Inception* concatenados, aplicando convoluciones de distintos tamaños para reducir los requisitos de cómputo, y aumentar la profundidad y anchura de la arquitectura. En la Figura 3.7 se muestra la arquitectura del modelo, construido a partir de la reutilización de bloques *Inception* (en color amarillo). El modelo final contiene alrededor de 12 millones de parámetros.

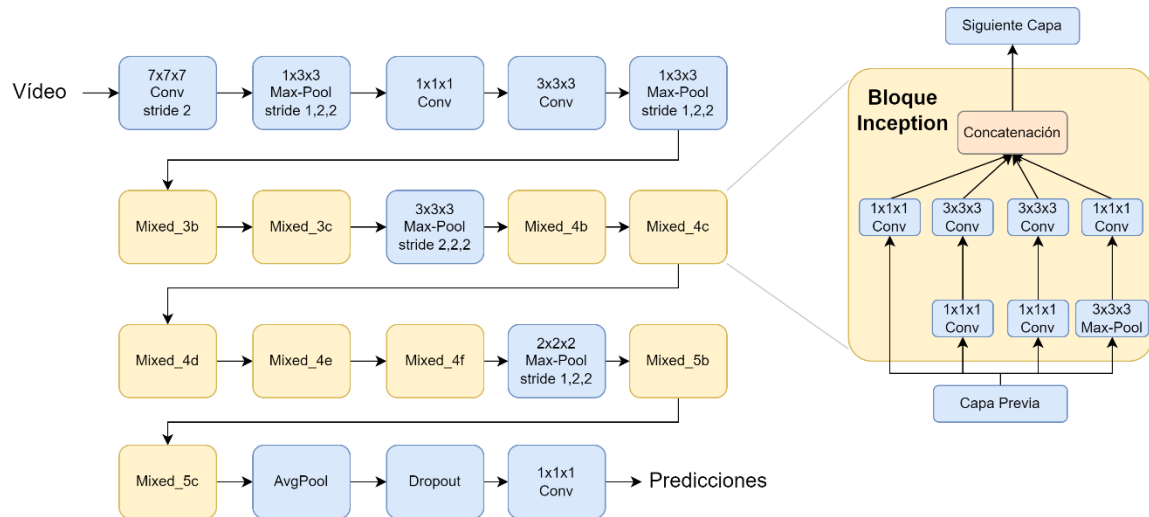


Figura 3.7. Arquitectura de modelo RGBI3D. Integra operaciones convolucionales 3D en bloques *Inception* (en color amarillo) para extraer información espacio-temporal de los vídeos.

RGBI3D reduced: Teniendo en cuenta que el reconocimiento de acciones se va a integrar en un dispositivo de limitados recursos, se propone un modelo más eficiente con menores requisitos de cómputo. Aunque, esto suponga una pérdida en términos de precisión. Este modelo se compone de dos partes diferenciadas: 1) el modelo *RGBI3D* hasta la capa *Mixed_4b*, usando los pesos preentrenados de esta red para extraer características de nivel general de los vídeos; 2) para el resto de las capas finales, se siguió un enfoque similar al introducido en [92]. Se incluyen capas convolucionales 3D factorizadas, simplificadas en capas convolucionales 2D más 1D (2+1)D. Tras las capas convolucionales (2+1)D, se integran unas capas finales similares a las del modelo original. El modelo resultante tiene solo 5,5 millones de parámetros (menos de la mitad que el original). La arquitectura *RGBI3D reduced* se muestra en la Figura 3.8.

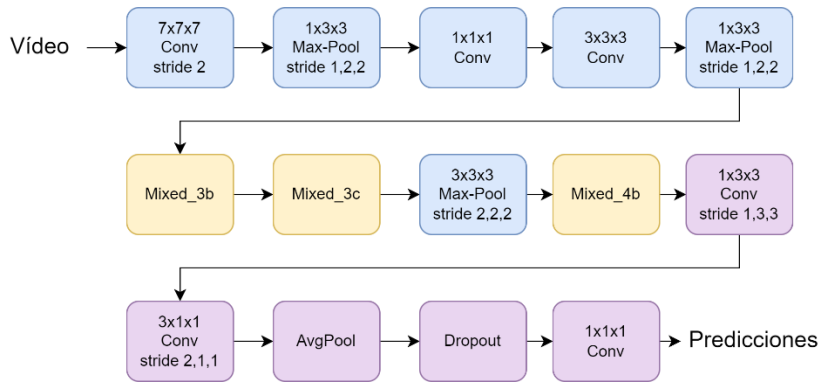


Figura 3.8. Arquitectura de modelo RGBI3D reduced. Las capas de color azul y amarillas forman parte del modelo original (con los pesos tras el entrenamiento sobre Kinetics [75]). En la red RGBI3D reduced, las capas de color lila sustituyen las últimas capas de la red original y son reentrenadas en el conjunto de datos creado a medida para la monitorización de actividades cotidianas.

RGB2D Mobilenet GRU: Esta arquitectura está inspirada en el trabajo de Donahue et al. [85]. Cada fotograma es procesado por la red *Mobilenet* [153] para extraer la información espacial. Estas características son posteriormente analizadas por una capa recurrente GRU. Esta capa se utiliza para aprender la dinámica de las acciones en términos temporales. Esta arquitectura tiene tan solo 4,38 millones de parámetros. Cabe mencionar que la arquitectura *Mobilenet* se introdujo originalmente para mejorar el rendimiento en tiempo real de las soluciones basadas en Aprendizaje Profundo que se ejecutan en dispositivos de limitados recursos.

RGB3D fully trained: Esta red es una versión simplificada de la arquitectura *RGBI3D reduced*, y además, sus parámetros no han sido preentrenados sobre grandes conjuntos de datos. El modelo final cuenta con 2,6 millones de parámetros.

Para el reconocimiento de acciones, partimos de las cuatro alternativas anteriormente presentadas como aproximación inicial. Además, el desarrollo del módulo de reconocimiento de acciones se ha realizado siguiendo un enfoque incremental, en el que iterativamente se han introducen nuevas arquitecturas, así como modificaciones a los modelos presentados para ofrecer un mejor balance entre bondad de ajuste y uso de recursos de las soluciones.

3.2.3 Entrenamiento de modelos

Una vez seleccionadas las arquitecturas se procede con el entrenamiento de las soluciones. En primer lugar, se define la resolución de los datos de entrada a la red, tanto en términos temporales como espaciales. A continuación, se presentan las técnicas de preprocesamiento utilizadas, así como la metodología seguida para la aumentación de datos.

Las tareas de preprocesamiento y aumentación de datos se ejecutan cada vez que se alimenta a la red con nuevos datos para optimizar los pesos en la fase de entrenamiento. El entrenamiento está compuesto de dos etapas diferenciadas, *Transfer Learning* (transferencia de aprendizaje) [154] y *Fine Tuning* (ajuste fino) [155] (ver Figura 3.9). Mediante la aplicación de estas estrategias se previene el sobreajuste de la red y se reduce el tiempo de entrenamiento sin requerir un gran conjunto de datos. Las técnicas de *Transfer Learning* y *Fine Tuning* permiten la reutilización de modelos de *DL* pre-entrenados sobre tareas similares, aprovechando el conocimiento extraído por estos modelos reentrenándolos solo durante unas pocas épocas.

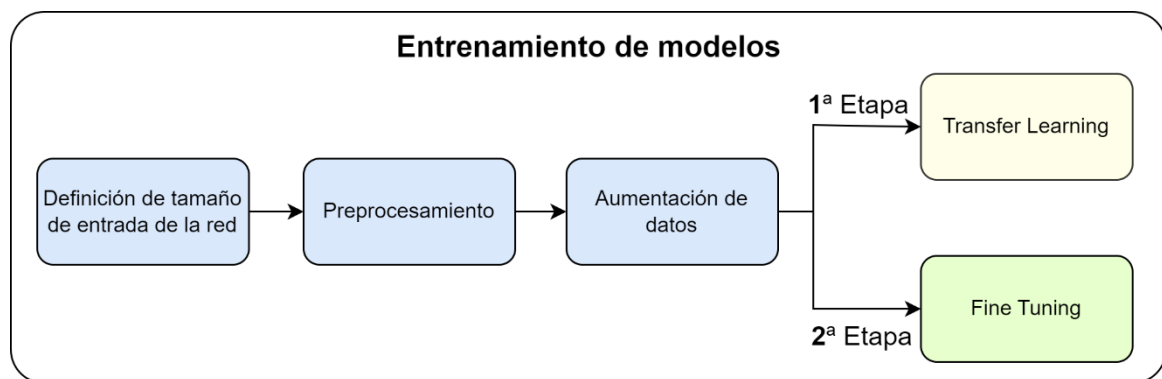


Figura 3.9. Procedimiento de entrenamiento en dos etapas. A las muestras que analiza la red en la fase de entrenamiento se le aplican transformaciones para facilitar el ajuste de los parámetros del modelo. La optimización de los parámetros se lleva a cabo en dos etapas; 1) mediante *Transfer Learning* se congelan los pesos de una arquitectura preentrenada en una tarea similar y tan solo se entrenan las capas densas finales; 2) Seguidamente, a través del *Fine Tuning* se adaptan los pesos de la red para especializarla en la tarea de clasificación de acciones cotidianas.

3.2.3.1 Procedimiento de entrenamiento

Dado que la arquitectura principal con la que se trabaja es la *RGBI3D*, para los datos de entrada se toma una configuración similar a la presentada en [75]. En concreto, la red se configura para analizar un flujo de 64 fotogramas con una resolución de 224x224 píxeles grabados a 25 fotogramas por segundo. Esto significa que la red tiene que ser alimentada con un clip de vídeo de unos 2560 milisegundos (~2,5 s) para dar una predicción. En términos generales, esta ventana de tiempo es suficiente para representar acciones cotidianas como *caminar*, *sentarse* o *sufrir una caída*.

En cuanto al preprocesamiento de los datos, la resolución de los vídeos se reduce hasta que su lado más corto tenga un tamaño de 256 píxeles. Posteriormente, se selecciona una ventana de 224x224 píxeles de 64 fotogramas de forma aleatoria de cada instancia de entrenamiento. El tamaño de esta ventana coincide con el tamaño de entrada del modelo. Además, se siguen otros procesos de aumentación adicionales como el volteo horizontal o

la rotación espacial. Finalmente, los píxeles se normalizan en el rango de $[-1, 1]$ para alimentar a la red con valores de pequeña magnitud que ayude a estabilizar el proceso de entrenamiento.

La implementación de técnicas de aumentación de datos permite alimentar a la red en cada época con ejemplos diferentes, de forma que pueda favorecer la generalización del modelo, evitando un sobreajuste de los parámetros sin tener que disponer de un conjunto de datos con elevado número de muestras.

En cuanto a los hiperparámetros, se escoge el uso del optimizador Adam con una tasa de aprendizaje inicial de 0,001. A su vez, dado que se trata de un problema multi clase, se trabaja con la función de pérdida *cross-entropy* (entropía cruzada). El objetivo de esta función es optimizar los pesos de la red para que la salida de la confianza de las predicciones sea lo más cercana posible a la etiqueta de cada muestra. Cuanto menor es el valor de esta función, mejor se ha ajustado a los datos el modelo.

Por ello, en lugar de definir un número de épocas a entrenar, se evalúa tras cada época el valor de la función de pérdida sobre el conjunto de validación, con la finalidad de escoger los pesos del modelo en la época que se ha minimizado la pérdida en un conjunto sobre el que no ha sido entrenado. Siguiendo esta aproximación se busca optimizar los parámetros de la arquitectura sin llegar al punto en el que se produzca un sobreajuste.

Por otra parte, para evitar que el modelo recale en mínimos locales de la función de pérdida, se monitoriza su valor sobre el conjunto de validación, y si tras un número definido de épocas (por ejemplo, cinco épocas) no se detecta mejora, se disminuye la tasa de aprendizaje utilizando un factor de 0,2. Este enfoque ayuda al optimizador a realizar un ajuste más fino de los parámetros del modelo, evitando que la arquitectura tienda al sobreaprendizaje (*overfitting*).

Tal y como se ha presentado en el epígrafe 3.2.1 en esta tesis se trabaja con conjuntos de datos desbalanceados. Este es un problema bien conocido en el campo del *Deep Learning*; un conjunto de entrenamiento desbalanceado conduce a los modelos a aprender a identificar de forma más precisa aquellas clases con mayor número de ejemplos. Consecuentemente, el modelo ve reducido su rendimiento de clasificación de las clases con pocas instancias. Este problema se aborda **ponderando** la contribución de cada clase a la función de pérdida dependiendo del número de ejemplos disponibles de cada acción en el conjunto de entrenamiento.

3.2.3.2 Transfer Learning

La técnica de **Transfer Learning** [154] permite sacar provecho al aprendizaje de una arquitectura para resolver una tarea, utilizándolo para acelerar el entrenamiento del modelo en una tarea de similares características. En concreto, esta estrategia consiste en congelar todos los pesos de la red preentrenada sobre un conjunto de datos de grandes dimensiones y tan solo entrenar las capas densas finales, evitando así perder el conocimiento obtenido

del *dataset* en el que se pre-entrenó el modelo, favoreciendo a su vez la capacidad de generalización de la solución.

Al congelar la mayor parte de las capas conservamos los filtros aprendidos por el modelo inicial para extraer información relevante para resolver la tarea, como puede ser la identificación de contornos, texturas o formas. Por medio de la transferencia de aprendizaje podemos conservar esos filtros y que el entrenamiento esté focalizado en la optimización de las últimas capas. De esta forma, el ajuste de los parámetros será más estable y convergerán al valor óptimo más rápido. Por lo tanto, a pesar de contar con un conjunto de datos pequeño es posible entrenar una solución evitando un posible sobreajuste y pudiendo obtener mejores métricas de bondad que las que se conseguirían al entrenar el modelo con los pesos aleatoriamente inicializamos.

En la sección de Resultados y Discusión se estudian los beneficios que aporta la implementación de *Transfer Learning* en la tarea de reconocimiento de acciones.

3.2.3.3 *Fine Tuning*

La estrategia de *Fine Tuning* [155] se utiliza para optimizar el modelo y especializarlo para el reconocimiento de actividades en interiores. En esta etapa, es donde el modelo ajusta los pesos de sus capas iniciales para mejorar la capacidad de reconocimiento de la solución en la tarea en cuestión.

La técnica de *Fine Tuning* permite, a partir de un modelo al que se le ha aplicado transferencia de aprendizaje, descongelar los pesos de todas las capas y entrenar el modelo de extremo a extremo. Por medio de esta metodología es posible especializar la arquitectura en el reconocimiento de acciones cotidianas, pero con una convergencia más rápida y estable que partiendo de un modelo al que no se le ha aplicado *Transfer Learning*.

Para estabilizar aún más el ajuste de los pesos y no perder parámetros relevantes para la identificación de características, es posible aplicar *Fine Tuning* de manera iterativa. De forma que, en un proceso de entrenamiento de varias etapas, se descongelen las capas desde la más profunda hasta la que recibe la entrada de los datos de forma progresiva.

3.2.4 Mejora de reconocimiento de situaciones críticas

Los modelos introducidos previamente ofrecen un buen desempeño en el reconocimiento de actividades cotidianas. Sin embargo, es importante que la solución ofrezca una alta precisión en la identificación de situaciones críticas como las caídas. Para evitar que los usuarios pierdan la confianza en el sistema, es crucial que el número de falsas alarmas se minimice a la vez que se detecten la mayor proporción posible de ejemplos positivos.

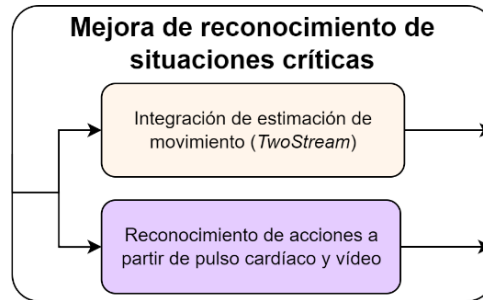


Figura 3.10. Aproximaciones propuestas para la mejora del reconocimiento de acciones críticas.

Para abordar este problema, se proponen dos aproximaciones alternativas (ver Figura 3.10). Una de ellas consiste en la integración del modelo *TwoStream I3D* en el sistema, requiriendo la estimación costosa de flujo óptico, así como el análisis de vídeo y de movimiento. Mediante este enfoque gracias a la información de movimiento se busca mejorar la identificación de situaciones críticas, aunque a un alto coste computacional, haciendo un elevado uso de recursos en los nodos de procesamiento local.

Por otro lado, el otro enfoque estudiado para la mejora de identificación de situaciones críticas es el uso de información multimodal, en concreto, datos de vídeo y de señales vitales como puede ser el pulso cardíaco. Extrayendo conocimiento de ambas fuentes de información se pretende mejorar la capacidad de reconocimiento de las soluciones.

3.2.4.1 Integración de estimación de movimiento (*TwoStream I3D*)

La red *TwoStream I3D* [75] es seleccionada para su integración en el módulo HAR. Esta red ofrece un gran rendimiento en términos de precisión sobre el conjunto de datos *Kinetics* [94]. Este modelo es computacionalmente muy complejo e incluye gran número de parámetros, principalmente debido a sus operaciones convolucionales 3D, las cuales son muy exigentes en términos de recursos. Uno de los flujos o *streams* de esta red es la arquitectura *RGBI3D*. El segundo *stream* (*Optical Flow I3D*) tiene una arquitectura equivalente a la red *RGBI3D*, pero se entrena para identificar acciones a partir de información de movimiento extraída por medio de la estimación de flujo óptico.

Es de especial interés en este trabajo la mejora de la precisión global del sistema, especialmente para el reconocimiento de acciones críticas como las caídas. Según [75], la precisión mejora significativamente al usar la red *TwoStream I3D*, en concreto, al añadir el *stream* de *Optical Flow I3D* para extraer conocimiento a partir del movimiento. El patrón que se da cuando una persona está sufriendo una caída es muy informativo y contribuye a mejorar la precisión. Sin embargo, la estimación de flujo óptico es una tarea intensiva computacionalmente que conlleva un gran coste de energía, recursos y tiempo de procesamiento. Debido a que el reconocimiento de acciones se realiza en los nodos locales, el alto coste de la arquitectura *TwoStream I3D* es un problema aún más preocupante dado los limitados recursos de los nodos.

Para lidiar con este problema se propone la integración de la herramienta *PoCL-R* [63] para transferir la ejecución (*offloading*) de tareas computacionalmente intensivas en equipos de altas prestaciones. Esto contribuye a la liberación de recursos, así como a una reducción del consumo energético en el nodo, ayudando a ampliar el tiempo de funcionamiento del sistema con nodos alimentados por baterías. El tiempo de procesamiento también se ve reducido gracias a la paralelización de los flujos de datos: la estimación de flujo óptico en el servidor central, y la inferencia de los *streams RGBI3D* y *Optical Flow I3D* en los nodos locales.

Virtualización de GPU remota a través de PoCL-R: La ejecución remota de tareas de forma transparente en una GPU remota se consigue mediante la virtualización de estos componentes para que aparezcan como si estuviesen conectados localmente al dispositivo empotrado, a través de *PoCL* y su controlador *PoCL-R*. Este controlador expone la GPU remota a la aplicación si fuera la propia GPU local del dispositivo. Las aplicaciones pueden forzarse a usar la GPU remota sin efectuar ninguna modificación en el código fuente.

3.2.4.2 Reconocimiento de acciones a partir de pulso cardíaco y vídeo

Por otra parte, se estudia la contribución del análisis de información multimodal de señales vitales y vídeo en la precisión de las arquitecturas. Para ello, se presenta una arquitectura de Aprendizaje Profundo optimizada que procesa información multimodal que monitoriza los hábitos de las personas.

El desarrollo de esta solución implica la consecución de los siguientes pasos: 1) se elabora un nuevo conjunto de datos de actividades cotidianas en espacios interiores, incluyendo datos de dos modalidades, vídeo y pulso cardíaco (*IAPV*, introducido en el epígrafe 3.2.1.2); 2) se desarrolla una nueva arquitectura *TwoStream* de Aprendizaje Profundo para el reconocimiento de actividades humanas, combinando información de vídeo y pulso cardíaco (ver Figura 3.11); 3) se optimizan los modelos HAR para su integración en dispositivos de bajo consumo ofreciendo rendimiento en tiempo real; 4) Finalmente, se describe un modelo optimizado para obtener una solución mejorada que se adapta a los usuarios con el objetivo de mejorar la precisión, lo cual es especialmente importante para la identificación de acciones críticas o de riesgo como las caídas.

En primer lugar, los dos *streams* de la red neuronal se construyen de forma independiente. En cada *stream*, una red neuronal profunda analiza o bien el vídeo o el pulso cardíaco. A continuación, un nuevo modelo combina ambas redes para crear una arquitectura de red Multimodal que ofrece una mayor capacidad de reconocimiento. Con esta mejora de prestaciones, se busca impulsar el reconocimiento de acciones críticas con el objetivo de reducir falsas alarmas. Posteriormente, se estudia un enfoque adaptativo al usuario para evaluar las ventajas de especializar el modelo de *DL* a diferentes sujetos en términos de capacidad de reconocimiento. La adaptación del modelo a cada sujeto es

recomendable si se tiene en cuenta que el sistema será de uso personal y que, además, permitiría la posibilidad de mejorarlo a lo largo del tiempo.

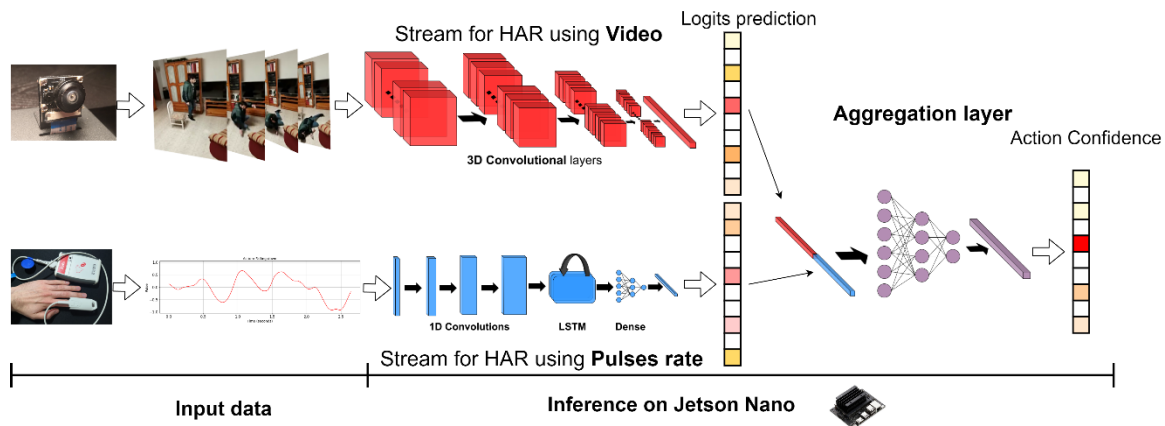


Figura 3.11. Arquitectura de Red Neuronal Profunda Multimodal para el reconocimiento de acciones humanas [151]. El modelo *TwoStream* propuesto combina vídeos (*stream superior*) y datos de pulso cardíaco usando un dispositivo de SpO_2 . La predicción de la acción se lleva a cabo en los dispositivos empotrados eficientes, *Jetson Nano*, alcanzando rendimiento en tiempo real.

El primer *stream* ejecuta la red *RGBI3D* para extraer patrones espacio-temporales del vídeo. Por otro lado, el segundo *stream* es una red *1DCNN + LSTM* que extrae los patrones temporales de la señal unidimensional de PPG. Este último *stream* es más eficiente computacionalmente con respecto al anterior. Sin embargo, el *stream* que analiza los fotogramas es más preciso, ya que la información contenida en el vídeo es más ilustrativa a la hora de identificar acciones. Además, las últimas capas de la red *TwoStream* combinan las características extraídas a partir de ambas modalidades de datos usando una capa de red creada a medida denominada *WeighPerClass*. Esta capa pondera la contribución que tiene cada uno de los *streams* en la predicción final de cada una de las actividades. El código de este desarrollo se puede encontrar de forma pública en GitHub³.

Red de análisis de vídeo: La arquitectura que analiza las acciones a través de vídeo se basa en la red *RGBI3D* [75]. Este modelo realiza el reconocimiento de acciones sobre vídeos grabados a 25 fotogramas por segundo (fps) con una resolución de 224x224 píxeles. Sin embargo, esta es una red muy exigente computacionalmente. Debido a que los nodos de procesamiento son dispositivos empotrados de limitados recursos, es necesario adaptar esta arquitectura. Para ello, se reduce la resolución espacial del vídeo de entrada, de esta forma, el número de operaciones en coma flotante que tiene que llevar a cabo la red se ve significativamente disminuida. En definitiva, el modelo se diseña para procesar una fuente de vídeo compuesta de 64 fotogramas a una resolución espacial de 112x112 píxeles.

³ <https://github.com/DaniDeniz/deep-multimodal-action-recognition>

Con respecto al entrenamiento de la red de vídeo, se sigue el procedimiento descrito en la sección 3.2.3. En la etapa de *Transfer Learning* se adaptan los pesos del modelo sobre el conjunto de datos *CIAV* (epígrafe 3.2.1.1). Este conjunto de datos de vídeo contiene un mayor número de muestras que el *dataset* construido para este estudio (*IAPV*) y permite hacer una adaptación suavizada de los parámetros del modelo para el reconocimiento de actividades cotidianas. Finalmente, en la etapa de *Fine Tuning* se reentrena el modelo sobre la modalidad de vídeo del conjunto *IAPV*.

Además, tras la fase de entrenamiento y con la perspectiva de crear un modelo más preciso que se adapte al comportamiento de los usuarios, se hace el *Fine Tuning* del modelo de nuevo de forma particular para cada usuario. Especializar el modelo de *DL* para cada sujeto contribuye a modelar de forma más precisa las características que son relevantes para el reconocimiento de acciones de cada uno. Este simple enfoque para un sistema adaptable a los usuarios permite obtener mejoras significativas en términos de la calidad de identificación de las acciones [34].

Red de análisis de pulso cardíaco: un pulsioxímetro mide el pulso cardíaco, una señal 1D cuya frecuencia indica el ritmo cardíaco de los pacientes. La amplitud y frecuencia de la señal del pulso cardíaco varía en función de la actividad que el usuario esté realizando [117].

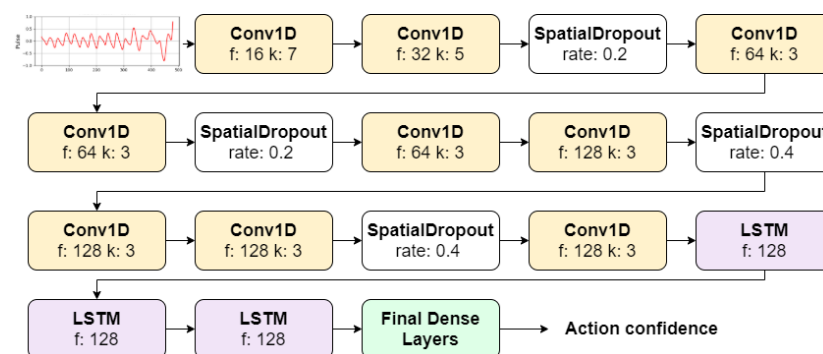


Figura 3.12. Modelo 1DCNN + LSTM para reconocimiento de acciones a partir de información de pulso cardíaco [151]. Utiliza capas convolucionales 1D para analizar las variaciones locales de los pulsos, capas LSTM para aprender patrones temporales, y capas de dropout para evitar el sobreajuste. Este modelo tiene un total de 650.000 parámetros, por lo que es aproximadamente 20 veces menos complejo que la arquitectura que procesa vídeo. *F*: número de filtros, *k*: tamaño del kernel, *rate*: proporción de conexiones podadas entre capas.

Se propone un nuevo modelo (ver Figura 3.12) para analizar el pulso cardíaco e identificar acciones a partir de ello. Este modelo está compuesto por dos tipos de capas de red principales: 1) capas de convoluciones 1D y; 2) capas de LSTM. Las capas convolucionales 1D se integran en el modelo para extraer variaciones locales en la señal. Por otro lado, la LSTM es una red neuronal recurrente que extrae la información temporal a una escala global a través del uso de la memoria para aprender secuencias de patrones en

la señal. Esta red neuronal también incluye capas *dropout*: elimina una proporción de conexiones entre capas al momento de entrenamiento para disminuir el sobreajuste de la solución.

Adicionalmente, este modelo se ha diseñado para poder procesar una señal con un tamaño variable. Por ello, la red se alimenta con un número de puntos de la señal diferente tanto en entrenamiento como en inferencia. De esta manera, la red puede aprender a partir de intervalos de tiempo variables, favoreciendo así la capacidad de generalización del modelo.

Con respecto al procedimiento de entrenamiento, se realiza como se describe a continuación:

- Primero, se aplican técnicas de aumentación de datos muestreando ventanas temporales de distinto tamaño en diferentes posiciones de la señal cardíaca en cada época. Esta aproximación contribuye a mejorar la precisión del modelo a la vez que se previene el sobreajuste. Esto es de gran utilidad sobre todo cuando la cantidad de datos disponibles es escasa para entrenar el modelo. Se asegura que al menos, los últimos 2,5 segundos de la señal muestreada corresponden con la acción que está siendo analizada. El resto de la muestra puede coincidir parcialmente con la anterior acción (teniendo en cuenta que el conjunto de datos se ha grabado con vídeos de forma continua).
- La red *1DCNN+LSTM* se entrena desde cero durante 100 épocas usando el optimizador Adam y un tamaño de *batch* de 4. Finalmente, nos quedamos con los pesos que el modelo tenía al final de aquella época en donde el error sobre el conjunto de validación era más bajo. De media, el tamaño de la señal de la que se alimenta la red tiene una duración de alrededor 6,6 segundos (aproximadamente 440 puntos de valores)

De forma similar al enfoque seguido con el *stream* que procesa el vídeo, este modelo es especializado a las características, comportamientos y escenarios asociados a cada usuario. Este procedimiento hace que el modelo final ofrezca una mayor exactitud (*accuracy*).

Arquitectura Multimodal de vídeo y pulsos: la red propuesta para la monitorización de actividades integra la información proveniente de los *streams* que procesan vídeo y pulso cardíaco. En particular, se plantean dos enfoques para combinar el resultado del procesamiento de ambos *streams*: 1) añadir una capa que agrega sumando elemento a elemento la salida de las predicciones no normalizadas de los *streams*; 2) Añadir una nueva capa denominada *WeighPerClass* que aprende a ponderar la contribución de cada *stream* a la predicción final de cada clase. La primera alternativa consiste en agregar las predicciones sin normalizar de ambos *stream* y pasarlos a través de una función de activación *softmax*. El segundo enfoque implementa el siguiente procedimiento: las predicciones de ambos *streams* se concatenan (*L*), y esto sirve como entrada a una capa de

red personalizada llamada *WeighPerClass*. Esta capa se construye usando una matriz de pesos W de tamaño ($NClases \times NStreams$), inicializada a 1. Para este estudio, los valores de $NStreams$ y $NClases$ son 2 y 10 respectivamente. Posteriormente, una función *softmax* combina la contribución de cada flujo para cada clase (ver (2)).

$$WeighPerClass(W, L)_{i,j} = \frac{e^{W_{i,j}}}{\sum_{k=1}^{nstreams} e^{W_{i,k}}} \cdot L_{i,j} \quad (2)$$

$\forall i \in \{1, \dots, nclases\} \text{ y } \forall j \in \{1, \dots, nstreams\}$

Tras la fase de ponderación de pesos, otra función de activación *softmax* procesa la salida de la anterior capa para obtener la predicción final de la arquitectura de *DL* Multimodal.

La capa *WeighPerClass* se entrena durante solo 5 épocas, comenzando con un valor de velocidad de aprendizaje o *learning rate* de 0,01, que es reducido por un factor de 2 al final de cada época. Hay que considerar que el *learning rate* (*lr*) tiene un gran impacto en el resultado final: ratios de aprendizaje altos puede llevar a la red a converger a soluciones subóptimas, a la vez que valores muy pequeños pueden hacer que apenas se produzcan modificaciones en los pesos originales de la red. Esto haría que el proceso de entrenamiento se alargara más de lo debido o se estanque.

Como resultado, el modelo *TwoStream* de vídeo y pulso mejora la capacidad de identificación de actividades, especialmente para acciones que pueden suponer situaciones de riesgo para personas de avanzada edad, como caerse o estar tumbado en el suelo. De esta forma, también es posible reducir falsos positivos, aprovechando los beneficios derivados del análisis simultáneo de información de vídeo y pulso cardíaco.

3.2.5 Optimización de arquitecturas de aprendizaje

Las decisiones en el momento del diseño y selección de los modelos que mejor se ajustan al sistema se rigen por un equilibrio o compromiso de calidades teniendo en cuenta:

- Los limitados recursos *hardware* de los nodos locales en términos de núcleos GPU y memoria.
- La necesidad de realizar la inferencia en tiempo real para acelerar la toma de decisiones.
- La complejidad de la arquitectura que determina el consumo y, por lo tanto, el tiempo total que podrían operar nodos locales alimentados por batería.
- La precisión de las soluciones, especialmente en la identificación de situaciones críticas.

Las arquitecturas propuestas, y especialmente el modelo *RGBI3D*, son soluciones de una elevada complejidad computacional. Esto dificulta la ejecución en tiempo real del reconocimiento de actividades cotidianas en nodos de procesamiento de bajos recursos. Por ello, la optimización de las arquitecturas juega un papel importante para lograr el despliegue de soluciones eficientes de monitorización de hábitos con rendimiento en tiempo real.

La complejidad de las soluciones depende principalmente del número de parámetros y número de operaciones en coma flotante (sumas, restas y multiplicaciones) necesarias para realizar una predicción. Para reducir las operaciones en coma flotante (FLOPs) de las arquitecturas se proponen dos enfoques diferenciados.

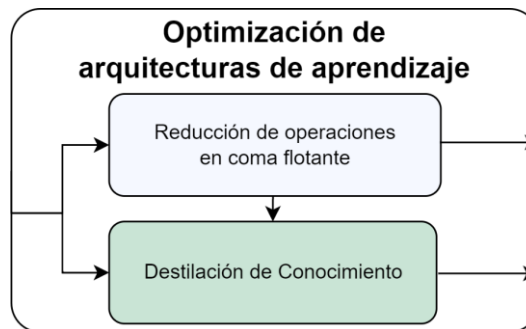


Figura 3.13. Enfoques introducidos para la optimización de arquitecturas. Cabe destacar que es posible combinar ambas metodologías de optimización para impulsar la capacidad de reconocimiento de las soluciones con un uso reducido de recursos.

Para abordar este problema se proponen dos enfoques (ver Figura 3.13). Por un lado, se estudia el impacto de la reducción del tamaño de entrada de las alternativas, midiendo los beneficios en términos de recursos gracias a la reducción del número de operaciones a costa de una pérdida en bondad de ajuste de la solución. Por otro lado, se propone la implementación de técnicas de compresión y aceleración de modelos. En concreto, se introduce la destilación de conocimiento para reducir la complejidad de las soluciones con un bajo impacto en términos de precisión. Esta técnica permite transferir conocimiento desde modelos complejos a arquitecturas más simples y eficientes, permitiendo que el desempeño de los modelos más sencillos sea mayor al que se obtendría al simplemente entrenar sobre un conjunto de datos.

3.2.5.1 Reducción de operaciones en coma flotante

El modelo original *RGBI3D* procesa vídeos de 64 fotogramas con una resolución de 224x224 píxeles, lo que supone una elevada cantidad de datos para ser procesado por los nodos de procesamiento empujados. Para reducir la complejidad de las soluciones, ofrecer rendimiento en tiempo real y minimizar el consumo de energía se propone adaptar las arquitecturas para procesar los datos de entrada a una menor resolución temporal y espacial a la del diseño original; en particular, la reducción espacial de los datos de entrada disminuye drásticamente los recursos necesarios de realizar una inferencia.

Esta reducción no solo disminuye el número de FLOPs del modelo, sino que también reduce la huella de memoria en GPU producida por el modelo. Esto permite aumentar el tamaño del *mini-batch* de entrenamiento, lo que puede contribuir a realizar una optimización de los parámetros más estable. A su vez, teniendo en cuenta que las soluciones se diseñan para ser implementadas en nodos de procesamiento local de bajos recursos, es muy importante minimizar el uso de memoria de estas arquitecturas. Si el uso de memoria es muy elevado, la inferencia no se puede acometer en los nodos, en cambio, si el uso de memoria es reducido, se podrían cargar varios modelos de forma concurrente en un mismo nodo de procesamiento.

En caso de modificar la resolución temporal y espacial de las soluciones es muy importante realizar el *Fine Tuning* del modelo de extremo a extremo. Los parámetros de las capas iniciales de la red están ajustados para extraer información de vídeo con una resolución concreta. Por esto, se requiere un reentrenamiento completo del modelo para así adaptar los pesos para extraer información relevante a partir de la nueva resolución.

3.2.5.2 Destilación de conocimiento

Una de las técnicas más populares para facilitar el despliegue de modelos precisos y eficientes en dispositivos de bajos recursos con rendimiento en tiempo real es la destilación de conocimiento. Brevemente, la destilación gestiona la transferencia del conocimiento adquirido por un modelo complejo y preciso a una arquitectura más simple y eficiente [123]. El resultado es una nueva arquitectura que reduce el tiempo de inferencia en el nodo local a la vez que ofrece una alta precisión sacando partido al conocimiento destilado del modelo complejo. Por ello, se propone la aplicación de la destilación de conocimiento para reducir la complejidad computacional de los modelos de aprendizaje profundo sin deteriorar la capacidad de reconocimiento [129].

En primer lugar, se ha diseñado una versión *TwoStream* de la red *S3DG*, que analiza datos de vídeo y de estimación de movimiento (flujo óptico) para el reconocimiento de acciones. Esta red *TwoStream* es equivalente a la presentada en el epígrafe 3.2.4.1 pero utilizando una iteración del modelo *RGBI3D* que contiene optimizaciones a nivel de arquitectura, incluyendo diseños que la hacen más eficiente, como pueden ser las convoluciones temporalmente separables o los mecanismos de activación de características espacio-temporales. Esta alternativa ofrece una gran capacidad de reconocimiento, pero tiene el inconveniente de que es computacionalmente muy exigente.

Por esta razón, se seleccionan arquitecturas de aprendizaje profundo más eficientes a las que destilar el conocimiento, en busca de ofrecer unos valores de exactitud razonables, pero a un menor coste de recursos.

A través de la destilación, es posible transferir el conocimiento extraído por grandes modelos o como el caso de este trabajo, por un conjunto de modelos entrenados individualmente (RGB y estimación de movimiento), a una arquitectura más sencilla y

eficiente [129]. Los modelos destilados son entrenados sobre un *dataset* definido, aplicando una distribución *soft target*, usando como referencia el modelo más complejo con una alta temperatura en su función *softmax*. La función *softmax* transforma la salida sin normalizar de la última capa de la red en una distribución de probabilidad con valores que suman 1.

La función *softmax* estándar es equivalente a establecer la temperatura $\tau = 1$ (ver capítulo 2.3.1.3. En el proceso de destilación, se utilizan valores de temperatura más elevados, para construir *targets* o vectores de predicción con una distribución suavizada (*soft target*). Los *soft targets* proveen información acerca de las relaciones inter-clases, lo cual es de gran utilidad para ayudar al modelo a aprender a cómo generalizar.

A través de la utilización de esta técnica se busca mejorar la capacidad de reconocimiento de los modelos sobre el *Indoor Action Dataset*. Se selecciona la arquitectura más potente y precisa (*TwoStream*) como *Teacher* o Maestra y los *Students* o Estudiantes son las arquitecturas eficientes de aprendizaje profundo con los datos de entrada redimensionados (ver epígrafe 3.2.5.1 Reducción de operaciones en coma flotante).

Para el proceso de destilación, se sigue este procedimiento: 1) Se aplican técnicas de aumentación sobre los datos de entrada como la rotación, recorte o redimensionamiento aleatorio; 2) las redes *Teacher* y *Student* se alimentan con la misma muestra, solo que en caso de la estudiante esta puede tener una resolución menor. La función de pérdida de la destilación ($\mathcal{L}_{KL}$) se calcula haciendo uso de la divergencia de *Kullback-Leibler* para estimar las diferencias entre las distribuciones de probabilidad extraídas por las redes *Teacher* y *Student*; 4) También se estima la pérdida *cross-entropy* ($\mathcal{L}_{CE}$) del modelo *Student* con relación a la etiqueta real de la muestra. La función de pérdida final ($\mathcal{L}_{global}$) se construye mediante la combinación de las funciones descritas en los pasos 3 y 4, estableciendo el valor $\lambda=0,1$ para ponderar la contribución de $\mathcal{L}_{KL}$ (90%) y $\mathcal{L}_{CE}$ (10%) a la pérdida global. En la Ecuación (3), τ se refiere a la temperatura de destilación, ψ es la función *softmax*, Z_s y Z_t son las predicciones de las redes estudiante y maestra respectivamente, y se refiere a la etiqueta real del ejemplo. De esta forma, la arquitectura *Student* se optimiza para hacer coincidir las predicciones con la red *Teacher*. 6) Se realiza el proceso de descenso del gradiente sobre la arquitectura *Student* para minimizar $\mathcal{L}_{global}$.

$$\mathcal{L}_{global} = (1 - \lambda)\mathcal{L}_{KL}(\psi(Z_s/\tau), \psi(Z_t/\tau)) + \lambda\mathcal{L}_{CE}(\psi(Z_s), y) \quad (3)$$

Cada modelo se destila utilizando distintos valores de temperatura τ entre 1 y 9. Valores de temperatura más elevados ayudan a extraer información acerca de las relaciones entre clases. Lo cual es útil para transferir al modelo destilado cómo la arquitectura *Teacher* tiende a generalizar.

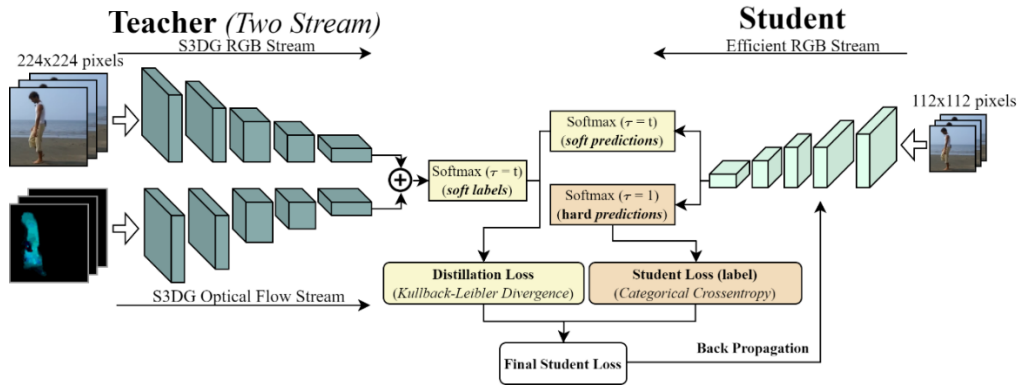


Figura 3.14. Ejemplo de destilación [156]. Se utiliza la alternativa más compleja y precisa (TwoStream) como Teacher. Después, se selecciona una red estudiante, que es una arquitectura de Deep Learning más eficiente y que recibe unos datos de entrada equivalentes, pero a una menor resolución espacial y temporal. Se entrena la red estudiante midiendo cómo de diferente es la distribución de sus predicciones en comparación con la arquitectura Maestra. Cabe mencionar que la estimación de movimiento sólo se requiere en las fases de entrenamiento y destilación para alimentar a la red Teacher. Los modelos estudiantes destilados sólo procesan vídeo en formato RGB. Valores de temperatura (τ) elevados producen una distribución entre clases más suavizada.

La Figura 3.14 muestra el enfoque seguido para realizar la destilación de conocimiento a partir de la arquitectura multimodal *TwoStream* que procesa vídeo RGB e información de movimiento, transfiriendo el conocimiento a una red *Student* más eficiente y que analiza solo vídeo RGB a menor resolución. Hacer un submuestreo de los datos de entrada de la red afecta negativamente a la precisión de la arquitectura, pero también reduce el número de operaciones en coma flotante (FLOPs) necesarias para realizar una inferencia. Esto es fundamental para construir modelos de aprendizaje profundo más eficientes, con tiempos de inferencia más bajos, pero sin deteriorar su capacidad de reconocimiento.

La optimización de los modelos mediante destilación de conocimiento permite el despliegue de arquitecturas más precisas y eficientes en el borde. Además, gracias a la destilación es posible construir modelos eficientes que permiten la monitorización de varios flujos de vídeo concurrentemente en un único nodo de procesamiento, a la vez que ofrece una precisión comparable a soluciones no destiladas más exigentes en términos de recursos.

3.2.6 Despliegue de modelos en el nodo local

Finalmente, la última etapa del diseño del módulo de reconocimiento de actividades es el despliegue de las arquitecturas en los nodos locales. Para hacer un uso eficiente de los escasos recursos disponibles se plantea la integración de herramientas como TensorRT [146] o Triton [147].

TensorRT es una de las herramientas más populares para el despliegue de modelos de manera eficiente. Esta herramienta permite la optimización de los modelos cuantificando (*quantization*) los pesos y funciones de la red a una precisión menor (por ejemplo, INT8 – enteros de 8 bits, o FP16 - números en coma flotante representados en 16 bits, en lugar de números en coma flotante representados con 32 bits). También permite ahorrar recursos por medio de la fusión de algunas capas u optimizar el uso de la memoria. Estas optimizaciones contribuyen a reducir el tiempo de inferencia, siendo de media un 10% más rápidas, a la vez que reduce el uso de memoria en la GPU hasta en un 25%.

La optimización de uso de memoria puede permitir el despliegue de varios modelos en el nodo de forma simultánea. Para hacer una gestión eficiente de la memoria disponible se integra la herramienta Triton. Triton es un servidor de inferencia de código abierto que permite hacer una gestión eficiente de los recursos de computación y de memoria disponibles, pudiendo ejecutar múltiples modelos concurrentemente en un único dispositivo. Esto es de gran utilidad para poder efectuar el redespliegue de arquitecturas de aprendizaje automático con una latencia muy baja. Esto facilitaría la adaptación del sistema en tiempo de ejecución, pudiendo desplegar modelos con distintas prestaciones en términos de compromiso entre precisión y consumo de recursos.

3.2.6.1 *Asignación eficiente de tareas a recursos*

En un entorno con recursos limitados, es crucial hacer un uso optimizado de los recursos de cómputo disponibles para ejecutar las tareas garantizando operación en tiempo real [157]. Seguir un enfoque basado en el procesamiento distribuido en el borde ayuda a hacer una gestión eficiente de los recursos de cómputo locales disponibles al monitorizar las actividades. La toma de decisiones sobre cómo asignar tareas a los nodos locales es crucial para garantizar un procesamiento en tiempo real.

Los modelos introducidos en esta tesis son intensivos en el uso de GPU, y su rendimiento es altamente dependiente de la capacidad computacional de este componente. Por esta razón, el uso medio de la GPU sirve como indicador de la carga computacional de los nodos locales [158, 159]. La carga computacional estimada para la ejecución en tiempo real de cada modelo se calcula por adelantado, antes del despliegue del sistema. Esto supone que cada arquitectura de aprendizaje automático se ejecuta de antemano en el nodo local, y se monitoriza la utilización media de la GPU durante un tiempo prefijado (30 segundos) mientras la plataforma procesa un flujo de vídeo. El valor de la utilización de GPU se almacena para ser consultado cuando el sistema esté en funcionamiento. El uso medio de GPU indica la carga computacional estimada de analizar un flujo de vídeo usando un modelo concreto. A partir de ahora, nos referiremos a esta calidad como **carga computacional**.

Para realizar la asignación eficiente de tareas a recursos se sigue un enfoque de balanceo de carga adaptativo basado en recursos [160]. El algoritmo de balanceo de carga distribuye el procesamiento de vídeo, priorizando la asignación de tareas a los nodos con

menor carga computacional para prevenir la sobrecarga de un nodo mientras se garantiza su funcionamiento en tiempo real. El esquema de balanceo de carga recaba la información de la carga computacional de los nodos en una Tabla de Información de Carga. La distribución del procesamiento tiene en consideración el número de flujos de vídeos a procesar, así como la utilización de recursos de las tareas en cada nodo de procesamiento. En concreto, se implanta una metodología de orquestación basada en recursos, de forma que el procesamiento se distribuye en función de los recursos disponibles en los nodos y de la carga computacional de las tareas (ver Figura 3.15). Este método de balanceo de carga dinámico monitoriza la asignación de tareas de forma periódica para distribuir de forma equitativa el procesamiento entre los nodos [161].

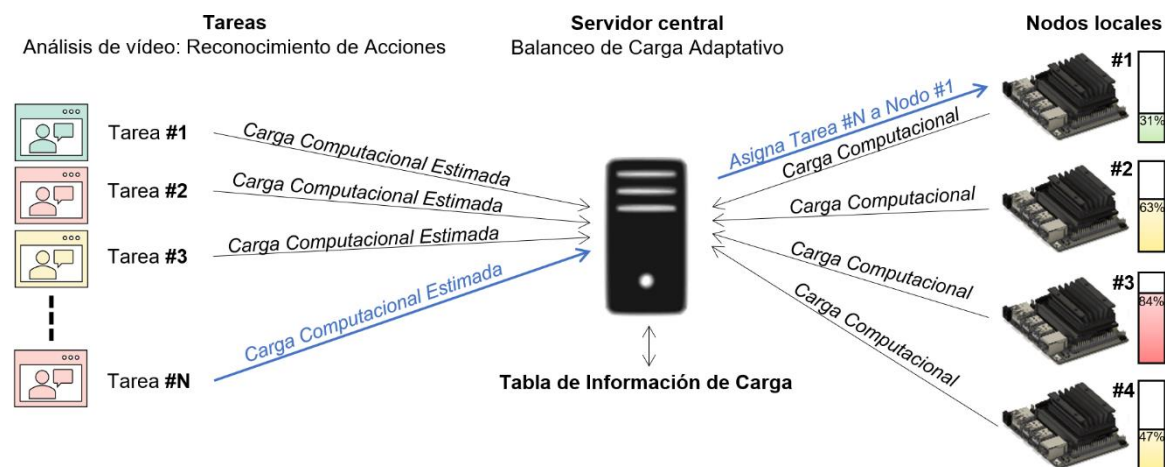


Figura 3.15. Diagrama de ejemplo de funcionamiento del balanceo de carga adaptativo basado en recursos. El servidor central obtiene la carga computacional de los nodos de procesamiento, almacenando esta información en la Tabla de Información de Carga. Las tareas se distribuyen entre los nodos del borde siguiendo un enfoque basado en recursos. Esto resulta en que el procesamiento de vídeo se asigna a los nodos con carga computacional más baja, para distribuir así de forma balanceada el procesamiento entre los nodos locales. En la figura, el ejemplo muestra que un nuevo sujeto va a ser monitorizado (líneas en azul, Tarea #N) y cómo el procesamiento se despliega en el edge_node_#1, el cual en dicho momento es el que está soportando una menor carga de trabajo.

El proceso seguido de balanceo de carga dinámico es el siguiente: el módulo consulta en la Tabla de Información de Carga cuáles son los nodos con mayor y menor carga de trabajo. Estos datos incluyen qué sujetos están siendo monitorizados por cada nodo y la carga computacional asociada a cada tarea. Posteriormente, el módulo evalúa las consecuencias de reasignar una de las tareas del nodo que soporta más carga de trabajo en el nodo más liberado. Finalmente, se efectúa la reasignación de la tarea en caso de que ello minimice la diferencia de nivel de carga de trabajo entre ambos nodos.

3.3 Sistema ciberfísico reconfigurable

A continuación, se presenta el diseño de un sistema ciberfísico distribuido adaptable automáticamente y reconfigurable usando una plataforma remota de monitorización en línea y una herramienta QRM, que es capaz de lanzar reconfiguraciones en los nodos locales basándose en el análisis de la información registrada.

El sistema ciberfísico (ver Figura 3.16) se construye principalmente sobre los siguientes componentes: a) módulo de reconocimiento de acciones; b) núcleo reconfigurable del CPS; c) herramienta de Gestión de Calidades y Recursos (QRM).

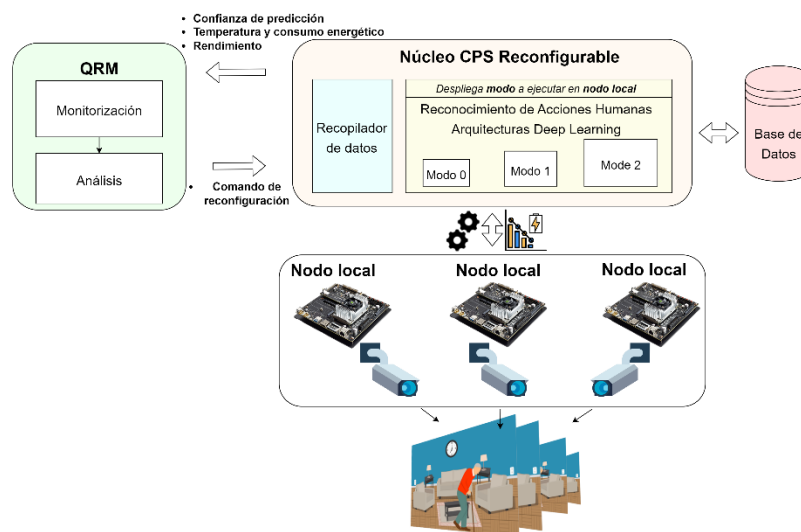


Figura 3.16. Visión general de la arquitectura del sistema ciberfísico reconfigurable.

El módulo de reconocimiento de acciones es el encargado de realizar el análisis de las actividades de las personas mayores o con necesidades especiales utilizando los recursos de los nodos de procesamiento **local** alimentados por **baterías**. Para ello, pone a disposición del sistema modelos con distinto compromiso entre precisión y consumo de recursos. Esto permite implementar reconfiguraciones en el sistema para que se adapte de forma automática a distintos escenarios, efectuando el redespliegue de diferentes modelos de aprendizaje profundo en los nodos locales.

El núcleo reconfigurable del CPS (CPS Core) es un módulo que se ejecuta en el servidor central dentro de la red local que tiene la función de orquestar las reconfiguraciones, de ser un primer agregador de los datos y una puerta de enlace entre el sistema local y la herramienta QRM. El CPS Core recopila datos de cada uno de los nodos locales acerca del funcionamiento del sistema y los envía a través de una conexión cifrada al módulo QRM. Estos datos incluyen: el resultado del análisis del vídeo, como puede ser la confianza de las acciones; calidades a nivel de aplicación, como el rendimiento en términos de fotogramas por segundo; y métricas a nivel de sistema como el consumo energético o la frecuencia de funcionamiento de los componentes *hardware*. Los datos son analizados de forma continua por una tarea de análisis en el módulo QRM, el cual envía

solicitudes de reconfiguraciones de alto nivel al componente CPS *Core*, que ejecuta acciones para la adaptación en tiempo de ejecución del sistema en los nodos locales.

Conceptualmente, la adaptación en tiempo de ejecución del sistema es una instancia del ciclo MAPE-k (ver Figura 3.17); un bucle de control de ámbito general para sistemas adaptativos [162], que está compuesto por varias fases que trabajan sobre un Conocimiento compartido (Monitorización, Análisis, Planificación y Ejecución). La adaptabilidad es generalmente alcanzada a través de la reconfiguración de los sistemas en la fase de Ejecución en respuesta a información recopilada en la fase de Monitorización. A pesar de que esta técnica es conceptualmente sencilla, en la práctica, construir un sistema adaptivo tiene implicaciones en la arquitectura de este, lo cual requiere una implementación específica a nivel de sistema de cada una de las fases.

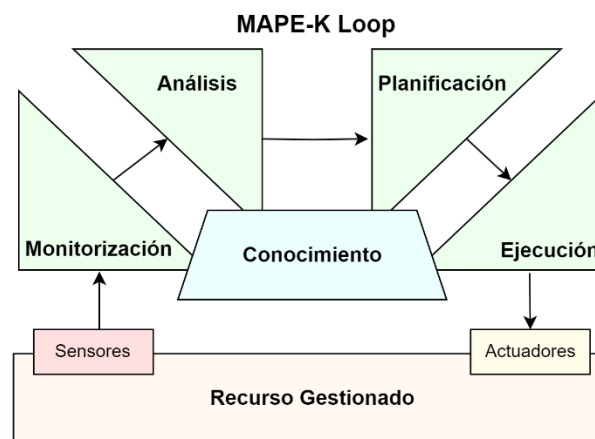


Figura 3.17. Ciclo MAPE-K. La metodología de Monitorización, Análisis, Planificación y Ejecución sobre una base de conocimiento compartida permite configurar la adaptación automática en tiempo real de un recurso gestionado.

La Gestión de Calidades y Recursos (*Quality and Resource Management – QRM*) tiene un papel crucial en el CPS, debido a que la configuración de un sistema presenta muchos grados de libertad, y encontrar una configuración óptima para una determinada tarea en distintas circunstancias se convierte en un problema de optimización multi-objetivo nada trivial [163]. Por lo tanto, la herramienta QRM provee soluciones en forma de configuraciones óptimas para distintas situaciones, lo que requiere un compromiso entre la utilización de ancho de banda, tiempo de respuesta, precisión y consumo energético [54]. La adaptación del sistema en tiempo de ejecución se efectúa por medio de políticas de reconfiguración [164, 165], definiendo modos de funcionamiento diseñados para alcanzar unos niveles de precisión, rendimiento o de utilización de recursos específicos [166].

Un elemento importante de los sistemas ciberfísicos adaptativos, y especialmente de aquellos en los que intervienen personas, es el hacer su funcionamiento observable para garantizar que operan (y toman decisiones) tal y como se espera. Durante su funcionamiento, estos sistemas generan gran cantidad de datos, que necesitan ser

procesados y presentados de forma adecuada para que puedan ser consultados por humanos. Para gestionar las reconfiguraciones y la visualización de los datos, el CPS integra FIVIS (*Fitoptivis Intelligent Visualization System*, que es una adaptación del sistema IVIS [167]). Esta herramienta es un *framework* (marco de trabajo) que integra elementos para el almacenamiento, análisis y visualización de los datos monitorizados por el sistema [168].

En definitiva, a través el diseño del CPS reconfigurable se persigue la optimización de tres calidades principales a nivel de sistema: **precisión**, **rendimiento** y **consumo de recursos** (ver Figura 3.18). Por un lado, se plantea el diseño de un sistema que ofrezca rendimiento en tiempo real usando nodos de procesamiento eficientes. A su vez, se busca que se maximice la precisión de la solución en el reconocimiento de actividades, especialmente en la identificación de acciones críticas para prestar rápida atención a personas en situaciones de emergencia, reduciendo las falsas alarmas. Por último, realizar un uso optimizado de los recursos disponibles es crucial para reducir el consumo energético de los nodos de procesamiento, permitiendo así aumentar el tiempo de operación de los nodos alimentados por baterías.

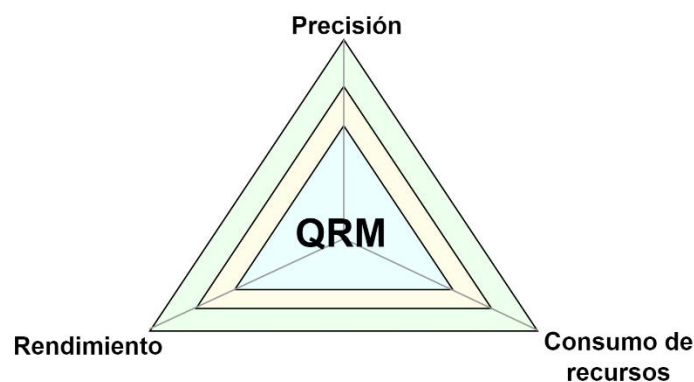


Figura 3.18. Optimización de calidades a nivel de sistema: precisión, rendimiento y consumo de recursos. Mediante la implementación de un módulo QRM para la adaptación en tiempo de ejecución del sistema se persigue la optimización de las calidades mostradas.

3.3.1 Herramienta de monitorización y Gestión de Calidades y Recursos

La herramienta de monitorización y Gestión de Calidades y Recursos (QRM) contribuye a hacer un uso optimizado de los escasos recursos disponibles, maximizando el tiempo de funcionamiento y capacidad de reconocimiento del sistema. La capacidad de adaptación del sistema a diversas situaciones está supeditada al diseño de este componente. Por un lado, la monitorización de las calidades del sistema permite mantener un registro del estado del sistema en términos de rendimiento, uso de recursos y de reconocimiento de actividades, generando alarmas para dar una rápida asistencia cuando los usuarios sufren situaciones potencialmente peligrosas.

Por otro lado, la gestión de recursos abre la posibilidad de tomar decisiones de **adaptación** del sistema en **tiempo de ejecución** a partir del análisis de la información monitorizada. La implementación de una política de reconfiguración permite efectuar

reajustes y modificar el comportamiento del sistema en tiempo de ejecución para hacer un uso optimizado de los recursos limitados disponibles, o bien mejorar la capacidad de reconocimiento de la solución para confirmar situaciones críticas y disminuir falsas alarmas.

El módulo de reconocimiento de acciones ofrece varios modelos de *DL* con diversas características; el nodo *hardware* puede también funcionar con distintos niveles de rendimiento, lo que resulta en un sistema con múltiples reconfiguraciones posibles. El objetivo de la herramienta QRM es adaptar en tiempo de ejecución la configuración del sistema para cumplir con objetivos de más alto nivel. Como puede ser reducir el consumo energético para extender el tiempo de funcionamiento del sistema sin comprometer el reconocimiento de actividades.

Como se ha mencionado en el epígrafe anterior, la reconfiguración del sistema en tiempo de ejecución se implementa como un bucle MAPE-K, con diferentes componentes responsables de diferentes tareas. El bucle se construye alrededor de un componente central, el conocimiento o *knowledge*, que es provisto por FIVIS, y que recopila la información necesaria para tomar decisiones de adaptación del sistema.

3.3.1.1 Monitorización

La monitorización es un elemento esencial en cualquier bucle de retroalimentación (*feedback loop*). En el sistema que se propone en este trabajo, los nodos locales envían la información a nivel de sistema al núcleo de reconfiguración (CPS Core), que agrega los datos provenientes de múltiples nodos y los envía a FIVIS. Además de hacer los datos disponibles para otros consumidores de información (y otros componentes del bucle de adaptación), FIVIS hace posible la visualización de los datos monitorizados para permitir a los usuarios observar y supervisar el funcionamiento del sistema.

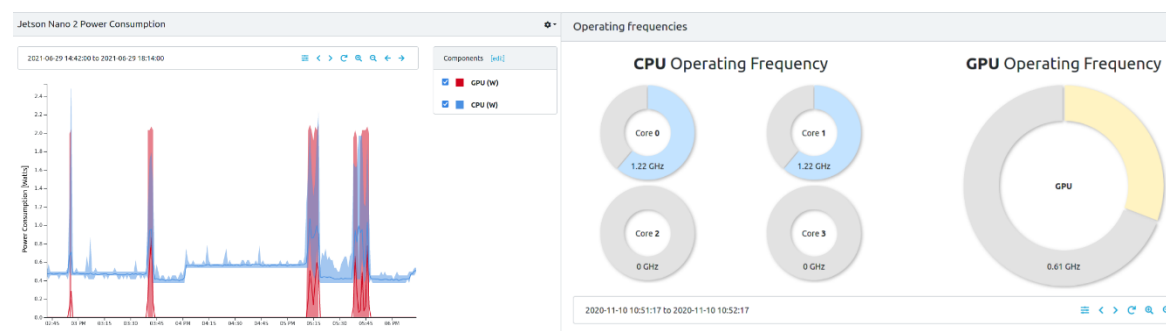


Figura 3.19. Consumo energético y frecuencia de funcionamiento de los componentes del sistema.

En este caso concreto el sistema recopila métricas a nivel de sistema y a nivel de aplicación. A nivel de sistema, se monitoriza la utilización, temperatura, y frecuencias de funcionamiento de los componentes *hardware* (CPU, GPU, DRAM) así como el consumo

energético. Esto se ilustra en la Figura 3.19, en la que se muestran algunas visualizaciones de métricas a nivel de sistema. A nivel de aplicación, se monitoriza tanto el rendimiento del reconocimiento de acciones, la confianza de las acciones identificadas por los modelos de *DL*, así como el modo de operación en el que está funcionando el sistema. Esto se destaca en la Figura 3.20, que muestra visualizaciones generadas a partir de métricas a nivel de aplicación.

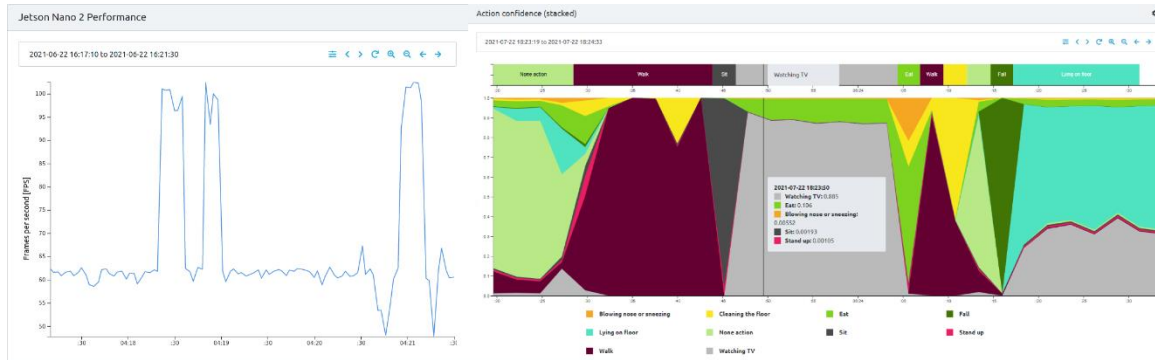


Figura 3.20. Rendimiento del reconocimiento de acciones en FPS y diagrama de líneas apiladas de la confianza de las acciones identificadas.

3.3.1.2 Análisis y Planificación

Las fases de análisis y planificación del bucle MAPE-k se llevan a cabo a través de una tarea de análisis definida y desplegada en FIVIS. Esta tarea procesa datos de dos de las calidades monitorizadas para definir el modo de operación más adecuado para cada nodo del sistema. Se monitoriza el consumo energético del sistema y la confianza de las predicciones de las acciones con la finalidad de maximizar el tiempo de funcionamiento de los nodos alimentados por baterías y minimizar las falsas alarmas al identificar situaciones potencialmente peligrosas (por ejemplo, caídas). FIVIS ejecuta esta tarea de análisis cada vez que la plataforma recibe nuevos datos de algunas de las calidades mencionadas. Debido a que esta tarea de análisis puede llegar a ser costosa computacionalmente, FIVIS permite la ejecución de estas tareas de forma distribuida en varias máquinas para balancear la carga de trabajo.

Las políticas de reconfiguración implementadas en la tarea de análisis son las que determinan cuál es el modo de funcionamiento del sistema más apropiado en cada momento en función de la monitorización de las siguientes calidades:

- *Consumo energético de las baterías:* La carga de batería disponible es una de las calidades monitorizadas cuando se ejecutan los modelos en los nodos locales. Cuando el nivel de batería restante se sitúa por debajo de un umbral predefinido, el sistema valorará el despliegue de una arquitectura de aprendizaje profunda que sea menos compleja computacionalmente (para reducir consumo energético), pero sin descuidar la precisión del sistema.

- *Confianza del reconocimiento de acciones:* Los modelos de *DL* son capaces de reconocer muchas acciones diferentes, sin embargo, para tomar decisiones de reconfiguración inteligentes, es necesario evaluar la duración y el riesgo que supondría que ocurriese cada acción. Con respecto al nivel de riesgo, en este trabajo se categorizan las acciones entre **críticas** y **no críticas**. Las acciones críticas se corresponden con situaciones de riesgo o que pongan en peligro la integridad de las personas como las *caídas* o *estar tumbado en el suelo*, y eventualmente cuando estas acciones se identifiquen, lanzar alarmas al sistema de emergencia. Con relación a la duración de las acciones, se distingue entre acciones de duración **breve** o **larga** [169]. Las acciones definidas como **largas** ocurren durante un periodo de tiempo extenso (generalmente de forma repetida), e incluye acciones del tipo: *comer* o *ver la televisión*. Por otra parte, ejemplos de acciones **breves** serían: *sentarse*, *ponerse de pie*, o *estornudar*. La información acerca de la confianza de las predicciones unido a esta categorización de las actividades, permiten al sistema seleccionar el modelo de *DL* más adecuado para cada situación. Por ejemplo, cuando se identifiquen acciones de **larga** duración que no sean críticas, el sistema tomará la decisión de usar un modelo de *DL* más simple y menos complejo. Esto se traduce en un menor consumo energético, conservando unos niveles de precisión razonables. Por otra parte, cuando el sistema reconozca una acción **crítica**, activa el uso del modelo de *DL* más preciso (y también más potente y demandante de recursos) para mejorar la precisión del reconocimiento de acciones y confirmar si una supuesta situación **crítica** realmente se dio, evitando así falsas alarmas.

La tarea de análisis mencionada ofrece dos salidas de datos. Una de ellas ofrece información acerca del estado del sistema (alarma / no alarma) a la consola de monitorización central, facilitando detalles específicos acerca del momento en el que se lanzó la alarma, así como cuál fue la situación que la provocó. Esto permite a un operario humano hacer una supervisión y verificar la alarma para activar una acción para hacer frente a esa emergencia. El otro tipo de salida de datos que ofrece esta tarea da información sobre cuál es el modo de funcionamiento de los nodos más aconsejable en cada momento, y facilita estos datos al núcleo reconfigurable (CPS Core).

A continuación, se muestra un extracto del código en *Python* integrado en FIVIS para automatizar la toma de decisiones de reconfiguración en la tarea de análisis. En el capítulo 4.2.1 se detallarán más aspectos sobre la implementación de la política de reconfiguración.

```

class ActionsMonitor:
    def __init__(self, dispatch: Callable = lambda *args: None, state=None,
                  minimum_conf_long_actions: float = 0.75,
                  critical_threshold: float = 0.5, check_critical_between_top_n: int = 2,
                  buffer_time_range: float = 10):
        """
        Monitors the recognized actions and triggers reconfigurations
        :param minimum_conf_long_actions: Minimum confidence to consider that a long temporally
        action is being identified with enough confidence.
        :param critical_threshold: Confidence threshold to be sure that a critical action occurred
        :param check_critical_between_top_n: Check a critical action is between the top N predictions
        :param buffer_time_range: Number of actions that we use to determine if an action is occurring
        continuously. (in seconds)
        """
        # dict with the relation of the active operating mode and the Deep Learning model name.
        self.operating_modes = {
            "motion_detector": 0, # This mode is not implemented yet
            "rgb_16_112": 1,
            "rgb_32_112": 2,
            "rgb_64_112": 3, # Changes between mode 2 and 3 depends on remaining battery.
            "two_stream_64_112": 4
        }
        self.minimum_conf_long_actions = minimum_conf_long_actions
        self.critical_threshold = critical_threshold
        self.buffer_time_range = buffer_time_range
        self.check_critical_between_top_n = check_critical_between_top_n
        self.last_actions_detected = {"name": [],
                                      "confidence": [],
                                      "time": []}
        # The actions that usually occur continuously for a long period of time.
        # We consider brief actions the ones that are not included in this list
        self.temporal_long_actions = ["bandaging", "cleanfloor", "cook", "eat", "lyingonbed/sofa",
                                      "sleeping", "smoke", "watchingtv", "lyingonfloor", "none"]
        # List of actions that we consider critical.
        self.critical_actions = ["fall"]
        self.last_alarm = ("", 0) # (action_name, time_reference_triggered)
        self.dispatch = dispatch

    def check_change_to_mode_1(self, video_ts: float, ordered_actions):
        if self.current_active_mode == 1 or self.current_active_mode == 4:
            return False
        top_action_name, top_action_confidence = ordered_actions[0]
        # Check if a long action is occurring with high confidence
        if self.is_long_action(top_action_name) and
            top_action_confidence >= self.minimum_conf_long_actions:
            # Unique list of names of the actions
            unique_actions = list(set(self.last_actions_detected["name"]))
            # Check if the same long action was occurring with high confidence for a period of time
            if len(unique_actions) == 1 and unique_actions[0] == top_action_name:
                confidence_higher_than_minimum = [conf >= self.minimum_conf_long_actions
                                                  for conf in
                                                  self.last_actions_detected["confidence"]]
                if sum(confidence_higher_than_minimum) == len(confidence_higher_than_minimum):
                    # Trigger reconfiguration command here
                    self.dispatch({"type": Event.MODE_CHANGE, "payload": {
                        "operating_mode": 1,
                        "video_reference": video_ts,
                        "ts_triggered": time.time(),
                        "reason": "{0} occurring continuously for more than {1} seconds "
                                "with a confidence higher than {0.2f}%".format(
                                    top_action_name, self.buffer_time_range)
                    }})
                    logger.info("Change operating mode to 1")
                    self.change_active_mode_to = 1
                    return True
        return False

```

3.3.1.3 Ejecución

La fase de ejecución del bucle MAPE-k (y el hacer efectiva la reconfiguración) se lleva a cabo por el componente *reconfigurable core* (CPS Core), que recibe una notificación de reconfiguración desde la tarea de análisis en forma de *callback* en un *endpoint REST*. Cuando el modo de funcionamiento del nodo difiere del que se indica, el núcleo reconfigurable del sistema modifica en tiempo de ejecución la configuración del nodo local para activar el modo de operación deseado.

El modo de operación que está activo en el sistema es otra de las calidades enviadas a FIVIS para su monitorización, de forma que permite a los desarrolladores y usuarios del sistema tener una perspectiva más amplia acerca del funcionamiento de este. Al que igual que con otras métricas, FIVIS también hace posible la visualización de esta información en un gráfico creado a medida, tal y como se muestra en la Figura 3.21. Las barras horizontales indican el intervalo de tiempo durante el cual estuvo activa una configuración concreta del sistema, mientras que las líneas verticales señalan en qué momento fue solicitada la reconfiguración por la tarea de análisis.

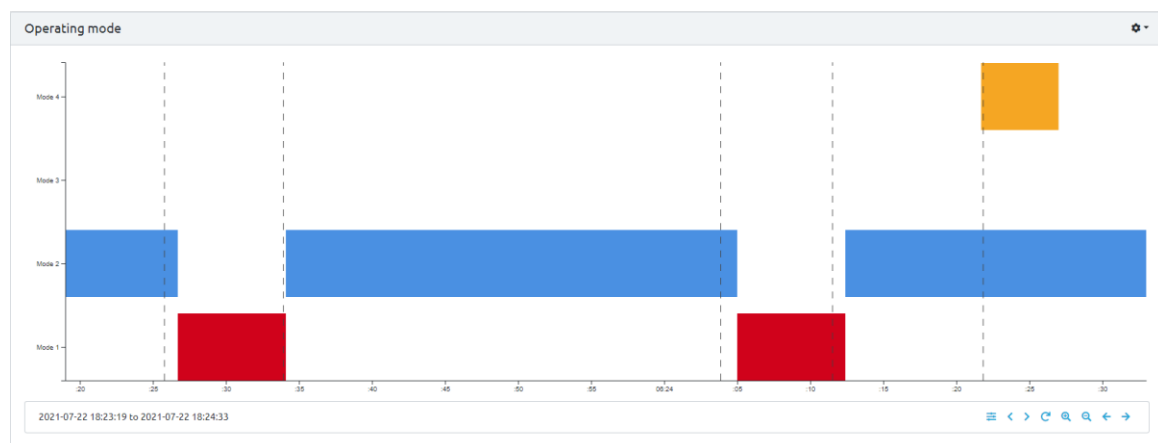


Figura 3.21. Gráfico que indica el modo de operación activo en el sistema y los eventos de reconfiguración.

4 Resultados y Discusión

En esta sección, en primer lugar, se analiza la precisión y consumo de recursos de los modelos propuestos. Posteriormente, se realiza un estudio de los beneficios de la utilización de información multimodal para la mejora de la precisión de las arquitecturas, especialmente en la identificación de situaciones críticas. Finalmente, se muestra los beneficios aportados por la implementación de técnicas de compresión y aceleración de modelos como la destilación de conocimiento.

En segundo lugar, se muestra la implementación de un CPS reconfigurable con arquitectura nodo-nube, donde se integra un módulo de reconocimiento de acciones con redes neuronales precisas y eficientes, y donde se implementa en el módulo QRM una política de reconfiguración que persigue dos objetivos: 1) extender el tiempo de funcionamiento del sistema alimentado por baterías al ser reducido su consumo energético; 2) optimizar la precisión global del sistema al identificar acciones **críticas**.

4.1 Módulo de reconocimiento de acciones

El módulo de reconocimiento de acciones que se presenta a continuación se construye siguiendo el procedimiento descrito en el capítulo 3.2 de la metodología. A continuación, se evalúa el rendimiento en términos de bondad de ajuste y consumo de recursos de las primeras arquitecturas diseñadas para el reconocimiento de acciones. Seguidamente, se evalúa el impacto de la reducción de operaciones en coma flotante de las arquitecturas en la precisión de la solución.

Por otro lado, se estudia la contribución de incluir información multimodal (estimación de movimiento y señales vitales) en la mejora de la precisión a nivel global de la solución, y particularmente para la identificación de acción críticas. Finalmente, se analizan los beneficios generados por la destilación de conocimiento para mejorar el compromiso entre precisión y consumo de recursos de los modelos de identificación de actividades cotidianas en espacios interiores.

4.1.1 Primeras arquitecturas seleccionadas

Se introducen arquitecturas de aprendizaje automático basadas en convoluciones 3D y redes convolucionales recurrentes. La aplicación de técnicas como Transfer Learning en la fase de entrenamiento permite que la arquitectura RGBI3D ofrezca un F1-Score superior al 78% en la clasificación de actividades cotidianas sobre el dataset CIAV. Se estudia cómo la integración de los modelos en dispositivos empotrados puede reducir la eficiencia energética de la solución hasta en 30 veces en comparación con el uso de GPUs de altas prestaciones.

En esta sección, se evalúa la bondad de ajuste de los modelos introducidos en el epígrafe 3.2.2, los cuales representan la base del desarrollo del módulo de reconocimiento

de acciones. En esta etapa se estudia el equilibrio entre precisión y consumo energético de las arquitecturas de aprendizaje profundo seleccionadas, evaluándolas sobre el *CIAV dataset* (ver sección 3.2.1.1), el conjunto de datos heterogéneo compuesto por muestras recopiladas de diversos *datasets* públicos de acciones de carácter general.

La Tabla 4.1 presenta la evaluación de los modelos presentados usando las métricas de exactitud, *precision*, *recall* y F1-Score (ver sección 3.2.1.4). Los resultados muestran que el modelo *RGBI3D original* es la alternativa que ofrece mejor bondad de ajuste, no solo en términos de exactitud (*accuracy*), sino también de F1-Score. Esto supone que este modelo es el que clasifica correctamente, en media, la mayoría de las clases. A pesar de ello, el valor de F1-Score es menor que el de exactitud (*accuracy*), algo que también sucede con el resto de los modelos. Esto es debido a que las arquitecturas clasifican correctamente aquellas clases que incluyen muchos ejemplos, pero tienden a fallar cuando las instancias de entrenamiento de algunas clases no son suficientes. La métrica macro F1-Score, a diferencia de la exactitud, tiene en cuenta este hecho y todas las clases contribuyen de forma equitativa a la métrica F1. Por ello, tiene siempre un menor valor.

Tabla 4.1. Métricas de evaluación de los modelos propuestos

Modelo de DL	Exactitud	Precision	Recall	F1-Score
<i>RGBI3D original</i>	84,24	77,83	79,30	78,14
<i>RGBI3D reduced</i>	81,00	76,65	75,81	74,99
<i>RGB2D Mobilenet GRU</i>	76,27	69,99	70,95	70,08
<i>RGB3D fully trained</i>	65,69	58,77	60,26	58,35

Por otro lado, la Figura 4.1 compara las curvas *precision-recall* (PR) para cada modelo. Las curvas PR muestran un compromiso entre las métricas de *precision* y *recall* para diferentes umbrales de confianza. Los valores más altos de área bajo la curva o *area under the curve* (AUC) denotan altos valores de *precision* y *recall*. Como podría esperarse, el modelo más complejo (*RGBI3D original*) alcanza el mejor valor de AUC. Sin embargo, el modelo *RGBI3D reduced*, que es una versión más simplificada de la solución anterior, ofrece tan solo 2 puntos menos de AUC. Al contrario, el rendimiento del modelo *RGB3D fully trained* es significativamente peor. Es por esto por lo que se descarta esta alternativa para su despliegue en los nodos. Este hecho apoya el potencial que ofrece la técnica de *Transfer Learning* cuando no se trabaja con suficientes ejemplos en el conjunto de datos. La red *RGB3D fully trained* se ha incluido en esta comparativa para ilustrar los beneficios que aportan los modelos de Aprendizaje Profundo pre-entrenados en grandes conjuntos de datos.

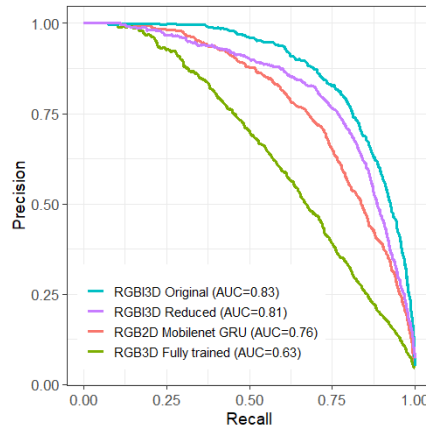


Figura 4.1. Curvas Macro Precision-Recall de los modelos propuestos [35]. El modelo RGBI3D original es la solución que ofrece una mayor área bajo la curva, obteniendo altos valores de precision y recall. Sin embargo, el modelo RGBI3D reduced obtiene valores muy cercanos con una arquitectura más simplificada. Por otro lado, el modelo RGB3D fully trained es el que peor rendimiento ofrece.

En la Figura 4.2 se ilustra el rendimiento del módulo de reconocimiento de acciones con algunos resultados cualitativos. En la columna de la derecha se indica la confianza en la predicción para estos ejemplos de test, en los que la red RGBI3D original ofrece los mejores resultados seguida de cerca de la red RGBI3D reduced. Se utiliza la técnica Grad-Cam para generar explicaciones visuales de cuáles son las regiones de la secuencia de fotogramas en las que la red RGBI3D original se centra para predecir la etiqueta. Esto se representa mediante mapas de calor superpuestos. Destacan regiones de atención del área alrededor de las manos para el caso del ejemplo de comer (*eating*) o las herramientas para el caso de limpiar (*cleaning floor*).

Label	Video	Confidence
Eating		RGBI3D Original: 85.88% RGBI3D Reduced: 72.23% RGB2D Mobilenet G.: 72.50% RGB3D Fully trained: 33.23%
Cleaning floor		RGBI3D Original: 96.57% RGBI3D Reduced: 80.52% RGB2D Mobilenet G.: 68.00% RGB3D Fully trained: 45.97%
Falling		RGBI3D Original: 97.39% RGBI3D Reduced: 91.92% RGB2D Mobilenet G.: 81.77% RGB3D Fully trained: 31.14%
Cooking		RGBI3D Original: 88.52% RGBI3D Reduced: 77.16% RGB2D Mobilenet G.: 72.45% RGB3D Fully trained: 53.58%

Figura 4.2. Ejemplos cualitativos de vídeo de test sobre CIAV dataset [35]. Para cada ejemplo, mostramos su etiqueta (label), una secuencia de fotogramas con un mapa de calor superpuesto, y la confianza de las predicciones de los modelos. Los mapas de calor se estiman usando la red RGBI3D original, la cual ofrece la mayor confianza en los ejemplos.

Estos mapas señalan las regiones de la secuencia de imágenes donde el modelo extrae las características para identificar las acciones. Por ejemplo, en los vídeos de cocinar (cooking) y comer (eating), las manos proporcionan la información más relevante para identificar la acción. Sin embargo, en el caso de la acción limpiar el suelo (cleaning floor), la atención se centra en el cepillo de barrer.

Dado que no se han dedicado esfuerzos para la optimización de estas soluciones, los modelos se despliegan sobre el dispositivo SoC Jetson TX2, que, a pesar de ser un dispositivo empujado altamente eficiente, no es la arquitectura *hardware* de más limitados recursos de la familia Jetson.

Con respecto a la eficiencia energética, se mide el consumo de energía medio en Vatios (*Watts* – W) para cada alternativa realizando 30 inferencias de forma continuada mientras se mide el consumo energético del dispositivo cada 100 ms. La Figura 4.3 muestra una comparativa de los modelos integrados en los nodos locales (izquierda: integrados en la Jetson TX2) con respecto a una GPU de altas prestaciones (derecha: Nvidia RTX 2080Ti). En primer lugar, se destaca las diferencias significativas en términos de consumo de energía, siendo el dispositivo empujado entre 20 y 30 veces más eficiente. Esta eficiencia se consigue gracias a la arquitectura *hardware* de la plataforma SoC, su frecuencia de funcionamiento y la disponibilidad limitada de recursos. En segundo lugar, los modelos de *DL* más complejos requieren de una mayor cantidad de recursos para su procesamiento, lo cual afecta al rendimiento de la solución. Por ejemplo, mientras las alternativas *RGBI3D original* y *RGBI3D reduced* procesan lotes de 64 imágenes, la red *RGB2D Mobilenet GRU* analiza en cada inferencia bloques de 5 fotogramas equidistantes que representan una ventana temporal equivalente al de las anteriores alternativas. Dejando de lado la complejidad de cada una de las arquitecturas, tan solo la diferencia de los datos de entrada que procesan supone una reducción significativa del número de operaciones que se tiene que acometer en cada inferencia.

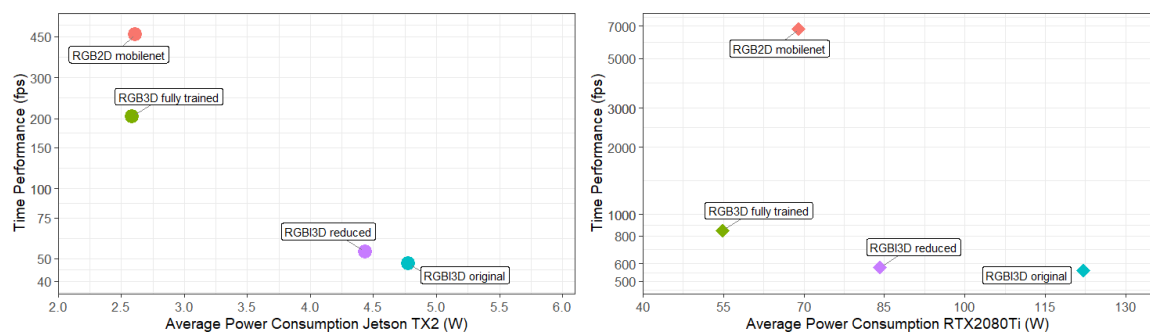


Figura 4.3. Izquierda y derecha: Rendimiento en fps vs. consumo energético en Jetson TX2 y RTX 2080Ti respectivamente. [35].

Este hecho se ve reflejado en el siguiente ejemplo: la red *RGBI3D original* es hasta 10 veces más lenta que la solución *RGB2D Mobilenet GRU*, y a su vez, consume el doble de energía, tanto en la implementación del nodo local como en la GPU de altas prestaciones.

Finalmente, existe una pequeña diferencia en el consumo de energía en el nodo Jetson TX2 entre los modelos *RGBI3D original* y *RGBI3D reduced*, con una diferencia entorno a un 15-20%. En el caso de la plataforma con la RTX, este consumo se reduce hasta un 30%. Por último, se destaca como el nodo local es significativamente más eficiente en términos de energía que la GPU de altas prestaciones, ofreciendo una reducción del consumo de hasta un 98%.

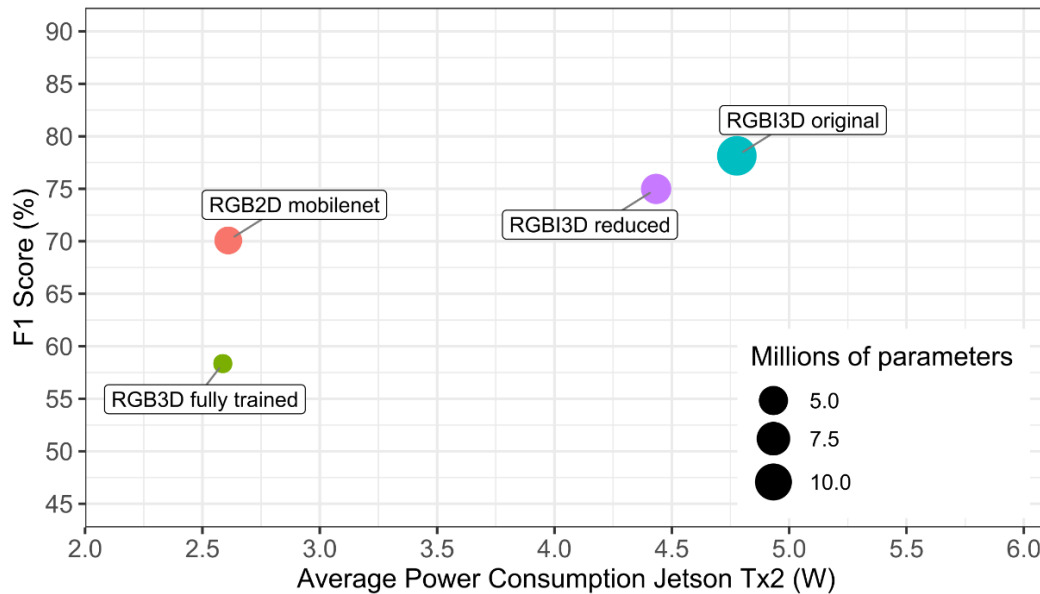


Figura 4.4. Macro F1-Score vs. Consumo energético en GPU de Jetson TX2 y tamaño de los modelos (en millones de parámetros) [35].

La Figura 4.4 muestra más variables de la comparativa de las soluciones, concretamente, el F1-Score y consumo de energía en los nodos locales, así como el tamaño de los modelos en millones de parámetros. En primer lugar, los modelos que alcanzan una mayor precisión (F1-Score) requieren una mayor cantidad de recursos para su procesamiento, lo que implica un mayor consumo de memoria y energía. En otras palabras, las soluciones con un mayor número de parámetros ofrecen un mejor rendimiento en la clasificación a expensas de un mayor consumo energético. En este trabajo se buscan soluciones que ofrezcan un buen compromiso entre precisión y consumo, para el despliegue de los modelos en los nodos locales.

4.1.2 Reducción de operaciones en coma flotante

Se propone la modificación de las arquitecturas para hacerlas computacionalmente más eficientes mediante la reducción de operaciones en coma flotante a través del ajuste de la resolución temporal y espacial de la capa de entrada de los modelos. Esta aproximación permite implementar arquitecturas que operan en tiempo real en el dispositivo de recursos más limitados de la familia Jetson (Jetson Nano). En concreto, se introducen modelos hasta un 75% más eficientes en términos de GFLOPs que ven reducida su precisión en menos de un 3% con respecto a la arquitectura original.

Como se ha observado en los resultados del anterior epígrafe, la cantidad de datos que analiza la red afecta directamente al número de operaciones en coma flotante que tiene que ejecutar la solución para producir una predicción. Por esta razón, como se ha introducido en la sección 3.2.5.1, se propone la modificación de las arquitecturas para reducir el coste computacional y el número de operaciones en coma flotante de los modelos. En concreto, para realizar el estudio del impacto de la reducción de los FLOPs de los modelos en términos de consumo energético y precisión, se escoge la arquitectura *RGBI3D original*, a la que a partir de ahora nos referiremos como *RGBI3D*. Esta se corresponde con la arquitectura que ha ofrecido una mejor bondad de ajuste en términos de exactitud y F1-Score para el reconocimiento de actividades cotidianas.

Debido a que se propone la reducción de la complejidad computacional de las arquitecturas, el rendimiento de estas se mide sobre nodos de procesamiento Jetson Nano. La Jetson Nano es el dispositivo SoC de Nvidia para la ejecución de aplicaciones de inteligencia artificial más eficiente y con menos recursos disponible actualmente en el mercado.

El modelo original *RGBI3D* procesa vídeos de 64 fotogramas con una resolución de 224x224 píxeles, lo que supone una elevada cantidad de datos para ser procesado por los nodos de procesamiento empotrados. Para reducir la complejidad de las soluciones, ofrecer un rendimiento en tiempo real y reducir el consumo de energía se presentan las siguientes 6 alternativas que se muestran en la Tabla 4.2. Con respecto a la resolución espacial, se estableció una disminución de la resolución de las imágenes de entrada a 112x112 píxeles. También se muestrea la resolución temporal mediante la selección de fotogramas equidistantes de un lote de 64 fotogramas, seleccionando 32 o 16 fotogramas (quedándonos con 1 de cada 2 o 1 de cada 4 del conjunto de 64 imágenes).

Todas las alternativas se entrenan siguiendo el mismo procedimiento, partiendo del modelo *RGBI3D* con los pesos pre-entrenados en [75]. En primer lugar, esta arquitectura se modifica y adapta para procesar una entrada de un tamaño definido. Seguidamente, los modelos se entrenan aplicando técnicas de *Transfer Learning*, lo que significa que se

congelan los pesos de la red y tan solo se entrenan las últimas capas. El objetivo de esta técnica es adaptar progresivamente la arquitectura de *DL* al *CIAB dataset*, evitando así perder el conocimiento obtenido del *dataset* en el que se pre-entrenó el modelo, con la finalidad de favorecer la generalización. Posteriormente, se aplica *Fine Tuning* para optimizar el modelo y especializarlo para el reconocimiento de actividades en interiores. En esta etapa, es donde el modelo ajusta los pesos de sus capas iniciales para adaptarse a las distintas configuraciones de entradas de datos. En resumen, la combinación de las técnicas de *Transfer Learning* y *Fine Tuning* mejora significativamente la capacidad de reconocimiento de las alternativas.

En la Tabla 4.2 se muestra el rendimiento y la utilización de recursos de las arquitecturas presentadas, valorando el macro F1-Score, y el tiempo de inferencia en el nodo. En cuanto a la métrica F1-Score, la red *RGBI3D 64 – 224x224* ofrece el mejor rendimiento, aunque como se ha mencionado previamente, esta alternativa viene ligada a un alto coste en términos de tiempo de procesamiento y recursos. Al compararse con la arquitectura *RGBI3D 16 – 112x112*, que es el modelo más veloz, solo se ve reducida la precisión en un 5%, a cambio de reducir el tiempo de inferencia en más de 13 veces con respecto al modelo más preciso.

Tabla 4.2. Rendimiento de las arquitecturas de Deep Learning

Arquitectura	#Frames	Resolución espacial	F1-Score	Inferencia (ms)
RGBI3D	64	224x224 (x3)	82,81	5551,58
RGBI3D	64	112x112 (x3)	80,50	1595,42
RGBI3D	32	224x224 (x3)	80,35	2878,06
RGBI3D	32	112x112 (x3)	79,60	829,78
RGBI3D	16	224x224 (x3)	78,35	1448,67
RGBI3D	16	112x112 (x3)	78,00	426,50

El procesamiento en tiempo real se garantiza para aquellas alternativas con un tiempo de inferencia inferior a 2560 ms (64 fotogramas a 25 fps), cumplido por la mayoría de las soluciones como se ha mostrado en la última columna de la Tabla 4.2. La Figura 4.5 muestra la velocidad inferencia a nivel de FPS de los modelos *DL* desarrollados. La red neuronal *RGBI3D 16 – 112x112* alcanza hasta los 150 fps, lo que es 6 veces más veloz que el umbral de tiempo real definido previamente. Como se puede observar, todas las soluciones que procesan una entrada con una resolución espacial de 112x112 y el modelo *RGBI3D 16 – 224x224* son capaces de cumplir con el requisito de operación en tiempo real en el nodo. Esto destaca la importancia de optimizar el número de operaciones de la red, y por lo tanto su complejidad, con vistas a diseñar arquitecturas de alto rendimiento.

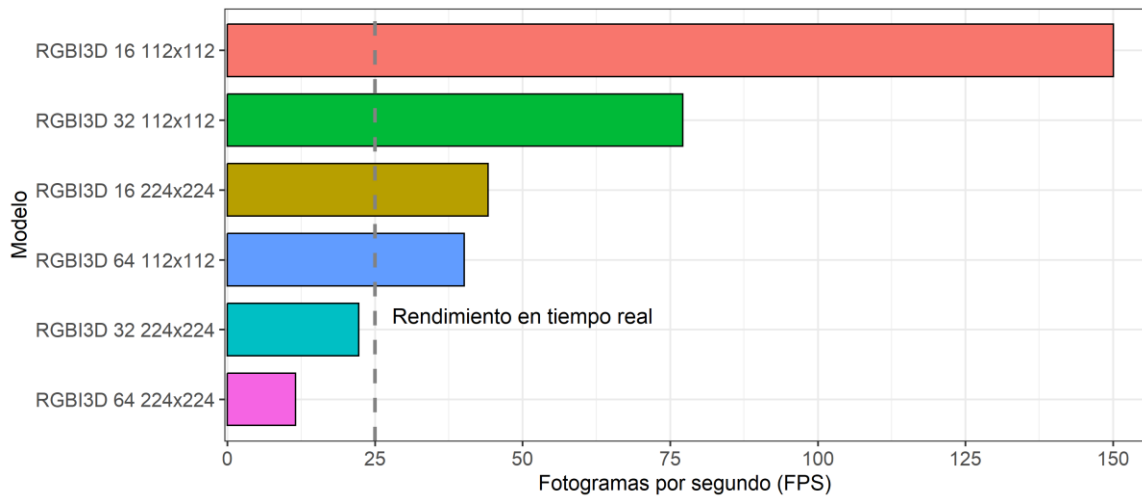


Figura 4.5. Rendimiento de los modelos propuestos en términos de FPS. La operación en tiempo real se garantiza cuando el tiempo de inferencia es menor o igual al tiempo necesario para recibir un nuevo lote de 64 fotogramas a 25 fps (línea discontinua).

Las operaciones en coma flotante (*floating point operations* o *FLOPs*) representan el número de multiplicaciones y sumas que se tiene que computar en cada inferencia por los modelos de *DL*, por lo tanto, se considera una medida de la complejidad computacional de las soluciones. Es por esto por lo que el número de operaciones en coma flotante tiene una relación directa con el rendimiento (tiempo de inferencia) y consumo energético de los modelos. En la Figura 4.6 se muestra una relación entre la precisión del sistema (F1-Score) y el número de Giga operaciones en coma flotante (GFLOPs, equivale a $10^6 \cdot \text{FLOPs}$) para cada uno de los modelos. El resultado más interesante que se obtiene de esta figura es que cuando se reduce la resolución espacial de los datos de entrada, se consigue una disminución significativa de los GFLOPs, sin tener un impacto negativo elevado en términos de precisión.

Por ejemplo, la red *RGBI3D 64 – 112x112* computa alrededor de un 75% menos de operaciones en comparación con el modelo *RGBI3D 64 – 224x224*. Al mismo tiempo, la métrica F1-Score se ve reducida en menos de un 3%. Por otra parte, se observa que la resolución temporal (número de fotogramas) tiene un gran impacto en la precisión del modelo: las alternativas *RGBI3D 64 – 112x112* y *RGBI3D 16 – 224x224* procesan un número similar de operaciones (GFLOPs), pero el primer modelo es un 2,7% más preciso.

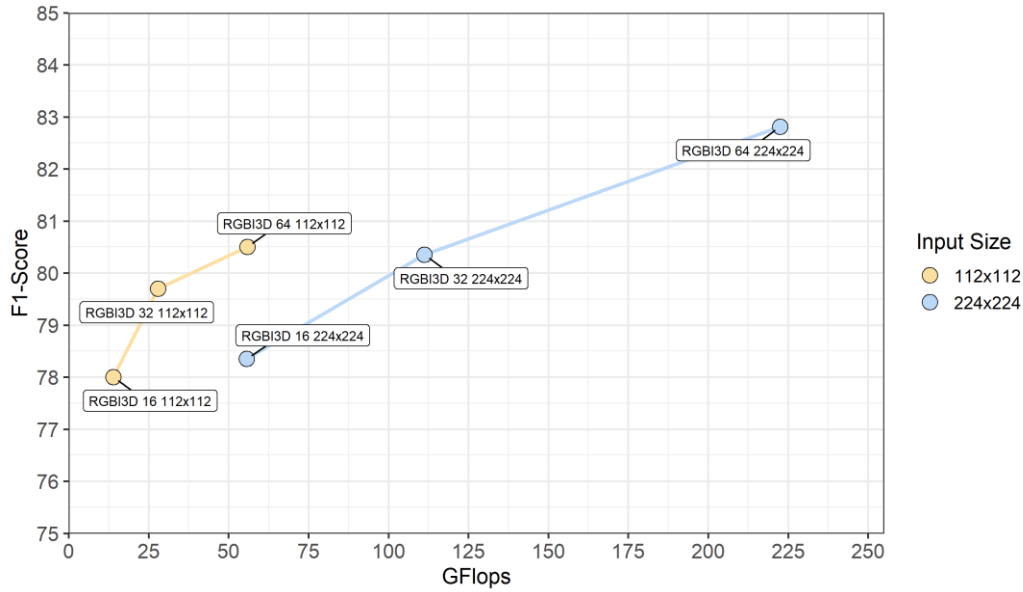


Figura 4.6. F1-Score vs. GFLOPs. Los GFLOPs representan el número de operaciones de sumas y multiplicaciones ejecutadas en la inferencia del modelo. Por lo tanto, el valor de GFLOPs es directamente proporcional al rendimiento, uso de recursos y consumo energético de cada una de las alternativas. Destaca por ejemplo como las soluciones con una entrada de menor resolución espacial (112x112) obtienen un valor de F1-Score razonable usando menos recursos computacionales (menos GFLOPs) que los modelos mayor resolución espacial.

4.1.3 Mejora de reconocimiento de situaciones críticas

A continuación, se presentan dos aproximaciones basadas en el uso de redes multimodales para impulsar la capacidad de reconocimiento de la solución en la identificación de situaciones críticas. Esto se propone para mejorar la tasa de verdaderos positivos y minimizar los falsos positivos, especialmente para el caso de situaciones potencialmente peligrosas. La mejora de precisión del sistema tiene un impacto directo en la confianza que los usuarios depositan en él, debido a que las personas se sienten más seguras al saber que el sistema cuenta con una mayor capacidad de discriminación entre los casos reales y falsos positivos, especialmente en situaciones que puedan representar un riesgo potencial para su salud. Por ejemplo, una tasa elevada de falsas alarmas puede redundar en la pérdida de confianza en el sistema por parte de los sujetos y los servicios de atención.

Por un lado, se propone el análisis de información de movimiento de forma explícita en la red, ya que la dinámica de acciones como las caídas pueden ayudar a discernir y a identificar correctamente la actividad monitorizada. Por otro lado, se busca mediante la medición de señales vitales de los pacientes, en particular, el pulso cardíaco, mejorar la capacidad de reconocimiento del sistema.

4.1.3.1 Integración de información de movimiento

La implementación del modelo TwoStream I3D permite mejorar la capacidad de reconocimiento general del sistema mediante el análisis de vídeo y de estimación de movimiento. En concreto, se mejora el F1-Score hasta en más de 5 puntos con respecto al modelo más preciso presentado en el capítulo 4.1.1. Esto es posible a costa de un mayor consumo de recursos, principalmente debido a las exigencias computacionales de la estimación de flujo óptico. Sin embargo, la integración de la herramienta PoCLR permite transferir la estimación de movimiento al servidor central, aliviando el consumo de energía en los nodos locales y acelerando la inferencia hasta en un 46%.

Partiendo de la base de lo presentado en el anterior epígrafe, se implementa el modelo *TwoStream I3D*. Este modelo está compuesto por dos flujos o *streams* de procesamiento. Uno de los *streams* es la red *RGBI3D*, y el segundo de ellos se denomina *Optical Flow I3D*. La red *Optical Flow I3D* tiene una arquitectura homóloga a *RGBI3D*, sin embargo, en lugar de procesar vídeo grabado con cámaras de espectro visible, esta se entrena para identificar acciones a partir de información de movimiento extraída por medio de la estimación de flujo óptico.

Para desarrollar la arquitectura *TwoStream I3D* se procede con el diseño y entrenamiento de la red *Optical Flow I3D*. Al igual que en caso de la red *RGBI3D* en la anterior sección, se presentan dos configuraciones alternativas variando la resolución espacial de los datos de estimación de movimiento para reducir la complejidad computacional de la solución (ver Tabla 4.3). Dada la importancia del factor temporal en el análisis del movimiento se ha descartado realizar un submuestreo de la dimensión temporal de los datos.

Tabla 4.3. Rendimiento de arquitecturas de Flujo Óptico

Arquitectura	#Frames	Resolución espacial	F1-Score	Inferencia (ms)
Optical Flow I3D	64	224x224 (x2)	78,21	5072
Optical Flow I3D	64	112x112 (x2)	79,17	1473

La Figura 4.7 destaca también como los modelos que procesan el flujo óptico (*Optical Flow I3D*) procesan en media un 8% menos de operaciones en comparación con la arquitectura *RGBI3D* con similar configuración. Los datos de vídeo se representan con 3 canales (RGB), y el flujo óptico en tan solo dos, representando las componentes horizontal (x) y vertical (y) del desplazamiento de los píxeles entre fotogramas consecutivos. Finalmente, se alcanza el mayor rendimiento de clasificación con la red *TwoStream I3D 64*

– 112×112 , ofreciendo un F1-Score un 1% mayor que la alternativa $RGBI3D\ 64 - 224 \times 224$ usando menos de la mitad de recursos computacionales.

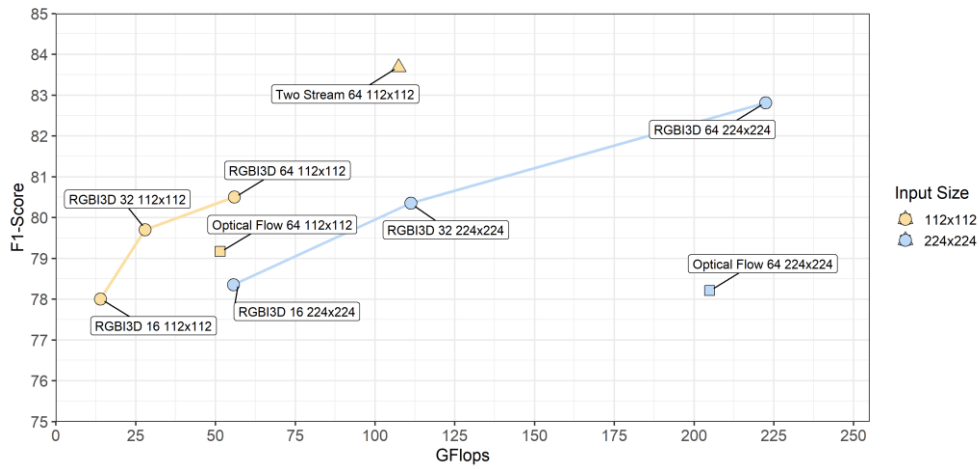


Figura 4.7. F1-Score vs. GFLOPs con red TwoStream [60]. Los GFLOPs representan el número de operaciones de sumas y multiplicaciones ejecutadas en la inferencia del modelo. Destaca como el modelo TwoStream 64 – 112×112 es hasta un 1% más precisa que la alternativa más exigente computacionalmente, requiriendo menos de la mitad de operaciones en coma flotante en la fase de inferencia.

La inferencia en la arquitectura *TwoStream I3D* se realiza combinando los resultados del procesamiento de los *streams RGBI3D 64 – 112×112* y *Optical Flow I3D 64 – 112×112* . Esta solución es la que ofrece una mejor precisión en la identificación de situaciones críticas. La estimación de flujo óptico se puede realizar en la CPU del nodo local, en su GPU, o bien en un servidor remoto a través de **PoCL-R**. La opción de usar la CPU local no es factible, ya que tomaría más de 30 segundos hacer la estimación de flujo óptico de un vídeo de 64 fotogramas con una resolución de 112×112 . La opción de usar la GPU local (CUDA) es una alternativa más viable ya que la estimación es más rápida, requiriendo de $2,645 \pm 0,016$ s. Sin embargo, utilizar *PoCL-R* para acceder a una GPU remota a través de una conexión 1000BASE-T Ethernet (1 Gbps) consigue acelerar la estimación de flujo óptico, tardando tan solo $0,986 \pm 0,09$ segundos. Además, el utilizar una GPU remota aporta un gran valor añadido al sistema, y es que permite la reducción del tiempo de inferencia para la arquitectura *TwoStream I3D* hasta en un 46%.

Tabla 4.4. TwoStream I3D, alternativas de estimación de flujo óptico

Arquitectura	Estimación Flujo Óptico	#Frames	Resolución	F1-Score	Inferencia (ms)
TwoStream I3D	CUDA (nodo)	64	112×112	83,63	5770
TwoStream I3D	RTX 2080Ti (PoCL-R)	64	112×112	83,63	3060

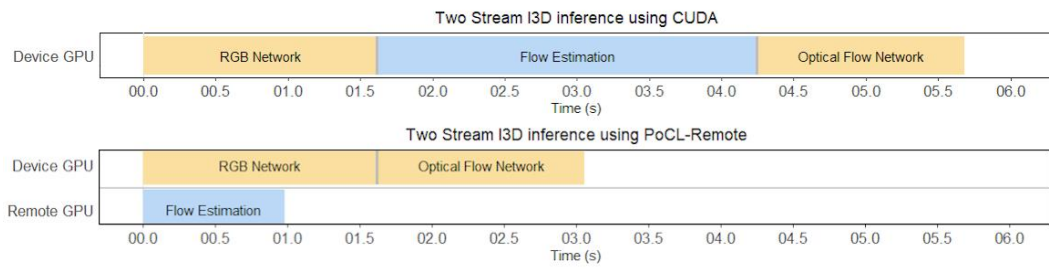


Figura 4.8. Comparativa de rendimiento entre usar GPU local en el nodo o un servidor remoto a través de PoCL-R para la inferencia de la red TwoStream I3D [60]. La opción que hace uso de la GPU remota consigue reducir en un 46% el tiempo de inferencia por medio del offloading de la estimación de flujo óptico. Además también consigue mitigar el consumo de energía en el nodo con respecto a la aproximación CUDA.

En la Figura 4.8 y la Tabla 4.4 se observa como la utilización de la GPU remota permite la paralelización del procesamiento de la inferencia de la red *RGBI3D* y la estimación del flujo óptico. De esta forma, justo cuando termina la inferencia del *stream* de vídeo en la GPU del nodo local, se comienza con la inferencia de la red *Optical Flow I3D* para el análisis del flujo óptico. El uso de la GPU remota por medio de *PoCL-R* consigue ahorrar casi 2,6 segundos de procesamiento y reducir el consumo de energía en los nodos locales alimentados por baterías.

Tras el análisis efectuado, se seleccionan cuatro de los modelos que mejor compromiso entre F1-Score vs. GFLOPs ofrecen: *RGBI3D 16 – 112x112*, *RGBI3D 32 – 112x112*, *RGBI3D 64 – 112x112* y *TwoStream I3D (PoCL-R) 64 – 112x112*. Esto facilita la implementación de soluciones energéticamente eficientes en los nodos de procesamiento locales. La Figura 4.9 muestra la precisión, rendimiento y consumo de energía para cada una de las arquitecturas HAR seleccionadas en los nodos Jetson Nano. A pesar de que la red *TwoStream I3D* es la alternativa más precisa, viene acompañada de un alto consumo de energía y de tiempos de inferencia más lentos. Esto provoca que no sea viable su ejecución en tiempo real de forma continuada en los nodos. Sin embargo, sí puede ser de gran utilidad su utilización puntualmente para confirmar si han ocurrido situaciones potencialmente peligrosas.

En general, los modelos que requieren de una cantidad menor de recursos para su inferencia en el nodo (GFLOPs), y menor consumo de energía, también son los que ofrecen un mejor rendimiento en términos de fps. No obstante, el consumo energético de las cuatro alternativas es relativamente bajo, sin exceder un consumo medio de tan solo 4.5 W en los nodos locales Jetson Nano. Este consumo se mide en los dispositivos SoC realizando 30 inferencias sobre una entrada de vídeo de 64 fotogramas.

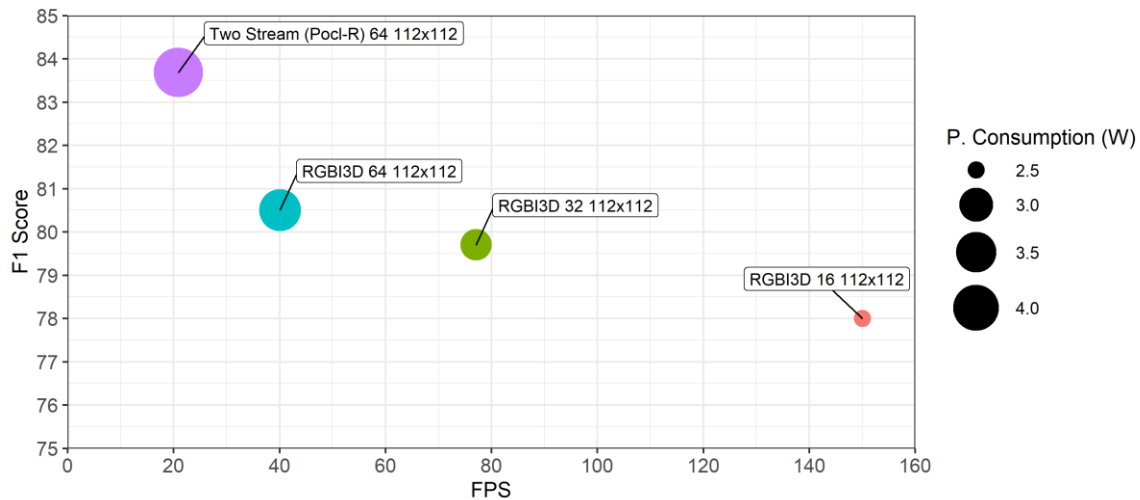


Figura 4.9. Comparativa de rendimiento de las arquitecturas de DL seleccionadas en términos de F1-Score, fps y consumo de energía [60]. Las soluciones que alcanzan un mayor valor de F1-Score tienen un peor rendimiento (fps) a un coste energético más alto. Por ejemplo, el modelo RGBI3D 16 – 112x112 ofrece el mejor rendimiento en términos de fps con un consumo bajo de energía, pero a expensas de perder 5,6 puntos de F1-Score en comparación con la red TwoStream I3D.

4.1.3.2 Reconocimiento de acciones a partir de pulso cardíaco y vídeo

El análisis de información multimodal de vídeo y pulso cardíaco recopilada en el conjunto de datos IAPV (ver epígrafe 3.2.1.2) permite mejorar hasta en un 2,4% el F1-Score de la solución con respecto a solo usar información de vídeo a un coste muy bajo de recursos. En concreto, se impulsa hasta en un 18% la bondad de ajuste de posibles acciones de emergencia, pudiendo reconocer a personas caerse o que están tumbadas en el suelo con un valor de F1-Score de 93,75% y 72% respectivamente. Asimismo, se propone una etapa donde el sistema es capaz de adaptarse a los usuarios y mejorar aún más el reconocimiento de estas situaciones críticas. Mejorando la capacidad de reconocimiento de media en un 21%.

Por otro lado, otro de los enfoques que se proponen para mejorar la capacidad de reconocimiento de las soluciones consiste en la integración de información de vídeo y señales vitales para la clasificación de acciones. En concreto, estas dos fuentes de datos se combinan para desarrollar una arquitectura multimodal para el reconocimiento de acciones. El modelo integra dos *streams* en los que se procesa vídeo y pulso cardíaco (*TwoStream*, ver epígrafe 3.2.4.2). El análisis del vídeo se lleva a cabo utilizando una versión optimizada de la red *RGBI3D* [75], que es una de las alternativas que ofrece una capacidad de reconocimiento cercana a la del estado del arte. El segundo *streams* procesa la información de pulso cardíaco medida a través de un módulo pulsioxímetro de RGB Medical [112].

Para llevar a cabo este estudio es necesario utilizar un conjunto de datos multimodal, como el IAPV (*Indoor Action Pulses and Video*), que ha sido construido desde cero con grabaciones en el hogar de cinco voluntarios realizando actividades cotidianas para el desarrollo de sistemas de monitorización de estilos de vida. En el epígrafe 3.2.1.2 de esta tesis se describe en más detalle el proceso seguido para la grabación de este conjunto de datos, así como su estructura y distribución de muestras o ejemplos por cada clase.

A continuación, se introduce y evalúa la capacidad de reconocimiento de la arquitectura Multimodal *TwoStream* propuesta para el análisis de vídeo y pulso cardíaco. Después, se presenta la evaluación del resultado de la etapa de adaptación del sistema, que busca especializar el sistema para que responda mejor a cada usuario en concreto. Finalmente, se introduce un estudio de ablación para entender la contribución independiente de cada uno de los *streams* a la precisión final de la solución.

Evaluación de la red Multimodal *TwoStream*: La red *TwoStream* procesa datos de dos fuentes de entrada (vídeo y pulso cardíaco). La evaluación se realiza usando una ventana deslizante con un tamaño de 64 fotogramas y obteniendo una media de las predicciones de las inferencias realizadas a lo largo del clip. Con respecto a la señal de pulso cardíaco, en la inferencia se procesan fragmentos de 6,6 segundos (440 muestras) que están sincronizados con el vídeo. Una vez el sistema se despliega, los datos multimodales se sincronizan estableciendo marcas de tiempo sobre las señales recibidas en el nodo local. Posteriormente, el modelo de *DL* procesa lotes de datos de vídeo y pulso cardíaco que representan el mismo instante en el tiempo.

Tabla 4.5. Evaluación de modelo *TwoStream* de vídeo y pulso cardíaco

Modelo	Exactitud	F1-Score
<i>TwoStream</i> (enfoque simple)	79,34	76,35
<i>TwoStream</i> (capa de ponderación)	80,70	78,27

Esta red combina la información extraída por los *streams* de vídeo y pulso cardíaco individuales siguiendo dos enfoques: 1) un enfoque simple que suma las predicciones de ambos *streams*; 2) una aproximación que usa una capa personalizada *WeighPerClass* para ponderar la contribución de cada *stream* a cada clase. La Tabla 4.5 muestra como la alternativa que usa la capa *WeighPerClass* (capa de ponderación) ofrece mejores resultados en términos de exactitud y F1-Score, alcanzando valores un 1,7% y 2,5% respectivamente más altos con respecto al enfoque más simple. Esta mejora de rendimiento usando la capa personalizada de ponderación es debida a que la red es capaz de aprender cómo debe contribuir de cada *stream* en la identificación de cada clase.

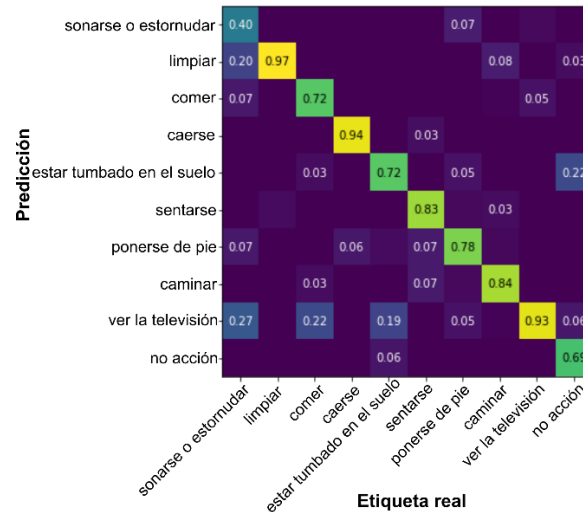


Figura 4.10. Matriz de confusión que evalúa el modelo TwoStream con la capa WeighPerClass sobre el conjunto de datos IAPV [151]. El modelo ofrece un F1-Score de 93,75% sobre la clase de *caerse*. El modelo también reconoce de forma exitosa otras acciones de interés para la monitorización de hábitos como: *caminar* o *ver la televisión*. La capacidad de reconocimiento se ve degradada ligeramente en la detección de acciones como *sonarse o estornudar*, pero esto ocurre debido a que el usuario hace esta acción mientras está realizando de forma concurrente otras actividades como *limpiar* o *ver la televisión*.

Por ejemplo, debido al uso de la capa de ponderación, para la acción de *ponerse de pie*, la red que analiza solo el vídeo tiene una contribución de un 71% para la predicción final de esta clase. Esto significa que los datos extraídos del vídeo son más informativos para la identificación de esta clase que las señales vitales. Sin embargo, el pulso cardíaco es algo más informativo para la identificación de *caídas* o de personas *tumbadas en el suelo*. Para estas acciones, la contribución del *stream* de vídeo en la confianza de la acción predicha se ve reducida a un 60% y la contribución de la red de pulsos aumenta hasta un 40%.

Finalmente, observe en la matriz de confusión (ver Figura 4.10) como el modelo *TwoStream* ofrece un valor de *recall* y *precision* de 93,75% para la clase *caerse*. Y alcanza un valor F1-Score de 72% para la identificación de la clase *estar tumbado en el suelo*. Este modelo también presenta gran capacidad de reconocimiento para acciones como *caminar* o *ver la televisión* entre otras. Por ejemplo, ofrece gran sensibilidad para la acción de *ver la televisión*, y una exactitud razonable al identificar personas *comiendo*, dos acciones de interés para definir estrategias que promuevan hábitos saludables. Sin embargo, presenta una leve confusión entre algunas de las clases. Hay que tener en cuenta también que la precisión (*accuracy*) top 2 del modelo de *DL* es 89,9%. Esto significa que, con una elevada probabilidad, la acción llevada a cabo será identificada como una de las dos acciones predichas con mayor confianza.

Evaluación de arquitectura adaptable al usuario: El modelo *TwoStream* presentado se ha entrenado usando la totalidad del conjunto de datos IAPV. Sin embargo,

se propone estudiar el efecto de especializar el modelo a cada actor. Tras una primera fase de adaptación al usuario, se espera que la tasa de reconocimiento general del sistema se vea mejorada significativamente, lo que llevaría a una solución con mayor grado de interés para los usuarios.

La especialización se aborda en dos pasos: primero, cada alternativa se entrena omitiendo uno de los actores en el conjunto de datos, después el modelo de *DL* se ajusta usando el conjunto de datos IAPV al completo (incluyendo datos de entrenamiento referentes al nuevo sujeto). A pesar de que en este segundo paso se podría haber usado solo los datos del nuevo sujeto para entrenar el modelo, no se sigue esta aproximación debido a la escasez de datos de entrenamiento. Para esto último sería crucial construir un conjunto de datos más amplio, con más actores y ejemplos.

Tabla 4.6. Evaluación de adaptación a los usuarios

Sujeto	Antes de especialización		Después de especialización	
	Exactitud	F1-Score	Exactitud	F1-Score
0	67,47	57,34	82,92	76,71
1	78,82	72,57	91,76	87,34
2	62,06	41,88	72,41	51,74
3	61,90	58,06	68,25	57,53
4	76,47	71,80	95,58	95,15

La Tabla 4.6 presenta los resultados del enfoque de adaptación a los usuarios; se evalúa la exactitud (*accuracy*) y el F1-Score solo sobre el conjunto de test de cada usuario seleccionado antes y después de la especialización con la arquitectura Multimodal *TwoStream*. La precisión tras la especialización se ve en media aumentada en un 18%, y el F1-Score en 21,85%. Esto muestra cómo adaptar el modelo a los usuarios contribuye notablemente a mejorar la capacidad de reconocimiento del sistema.

Veamos, por ejemplo, el caso del *Sujeto 0* y *Sujeto 4*. Ambos sujetos son los que disponen de un mayor número de muestras (ver Tabla 3.3), y, por consiguiente, son los que acaban obteniendo mejoras más altas en términos de F1-Score (alrededor del 33%). Claramente, el número de instancias grabadas por cada actor tiene un impacto en el resultado de este enfoque adaptable a los usuarios, pero incluso una pequeña cantidad de muestras (como los 32 clips del *Sujeto 1*), puede conducir a una mejora sustancial (20% en F1-Score). En cualquier caso, usar muestras de otros actores en la fase de entrenamiento fomenta la generalización del sistema, lo que se traduce en una mejora en la capacidad de reconocimiento.

Estudio de ablación: seguidamente, se estudia la contribución de cada *stream* en los resultados de la red Multimodal *TwoStream*.

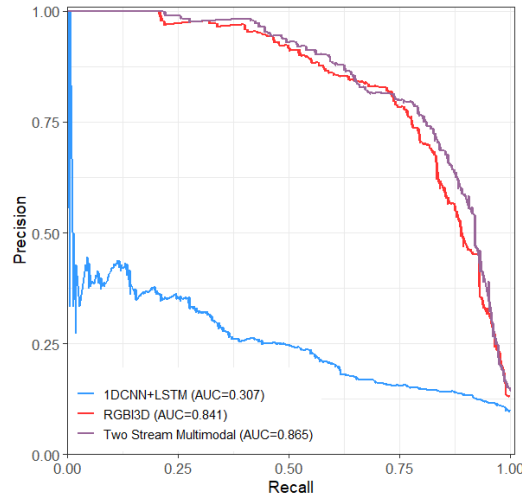


Figura 4.11. Curvas de macro Precision-Recall para la red *TwoStream* y cada uno de los *streams*. Aunque la red de pulso cardíaco (*1DCNN+LSTM*) ofrece valores muy bajos de precision-recall, esta red computacionalmente eficiente contribuye a mejorar en términos de AUC un 2,8% a la arquitectura *TwoStream* con respecto al stream que analiza el vídeo. Ayudando así a la mejora de la identificación de acciones críticas.

La Figura 4.11 muestra las curvas de *precision-recall* (PR) para la arquitectura *TwoStream* y cada uno de los *streams* de forma independiente. Las curvas PR muestran un balance de las métricas *precision* y *recall* para diferentes umbrales. El área bajo la curva (area under the curve – AUC) se está relacionada con los valores *precision-recall*: un valor mayor de AUC indican mejores resultados. Por ejemplo, la red *1DCNN+LSTM* ofrece unos valores muy bajos de AUC cuando se evalúa de forma individual. Sin embargo, contribuye en la mejora del AUC de la arquitectura *TwoStream* en un 2,8% con respecto a la red que analiza vídeo (*RGBI3D*). Debido a que los valores aquí mostrados corresponden a la media de todas las acciones, puede parecer que esta contribución no es muy relevante. Sin embargo, el procesamiento de datos multimodales contribuye a la obtención de notables mejoras en el reconocimiento de algunas acciones, tal y como se muestra en la Tabla 4.7 y se analizará en los siguientes párrafos.

Tabla 4.7. Evaluación por clase de cada una de las redes individuales

Acción	1DCNN+LSTM		RGBI3D	
	Precision	Recall	Precision	Recall
Estornudar o sonarse	0,00	0,00	58,33	46,67
Limpiar	41,67	29,41	64,70	97,05

Comer	13,63	9,37	85,18	71,87
Caerse	20,00	56,25	93,75	93,75
Estar tumbado en el suelo	33,33	41,67	66,67	55,55
Sentarse	16,67	10,34	92,00	79,31
Ponerse de pie	38,00	46,34	82,05	78,04
Caminar	50,68	41,11	94,73	80,00
Ver la televisión	47,45	65,11	64,51	93,02
No acción	-	-	76,67	71,87

La red *1DCNN+LSTM* analiza la señal de pulso cardíaco a través de convoluciones 1D y capas LSTM. La baja complejidad de esta arquitectura unido a la naturaleza unidimensional de la señal que procesa la red hace que este sea un modelo computacionalmente eficiente, ofreciendo tasas de inferencia muy altas. El análisis de la Tabla 4.7 muestra que usar solo el *stream* que se procesa con este modelo no es suficiente para el reconocimiento de acciones. Esto es debido a la ausencia de características en la señal de pulsos que aporten información relevante para distinguir entre algunas acciones como *comer* o *sentarse*. Sin embargo, el modelo es mucho más preciso para distinguir entre acciones como *caminar*, *caerse*, o *estar tumbado en el suelo*.

Con respecto a la red *RGBI3D*, ofrece una alta capacidad de reconocimiento a expensas de un alto coste computacional, principalmente debido al uso de operaciones convolucionales 3D. En concreto, tiene 20 veces más parámetros que la red que analiza los pulsos. No obstante, alcanza una alta precisión para la monitorización de hábitos. Al contrario de la red que analiza los pulsos, es capaz de reconocer de forma precisa acciones como: *comer*, *sentarse* o *ver la televisión*. De media, ofrece una exactitud (*accuracy*) de 78,26% sobre el conjunto de datos IAPV. A pesar de esto, la red presenta cierta confusión entre acciones como *estar tumbado en el suelo* y *no acción*. Esto es debido a que ambos casos la red no identifica ningún movimiento en la escena.

Como se ha mostrado, ambas redes son complementarias, y la combinación de estas impulsa el rendimiento del modelo multimodal, incluyendo la mejora de la identificación de situaciones críticas, lo cual es de especial interés en este trabajo. Por ejemplo, la métrica F1-Score para la clase *estar tumbado en el suelo* se ve mejorada en un 18%. Además, el F1-Score de otras actividades cotidianas como *caminar* o *limpiar* se ve mejorada en un 3% y 8,9% respectivamente (ver Figura 4.10). Por lo tanto, a pesar de su bajo rendimiento cuando se evalúa de forma aislada, la red *1DCNN+LSTM* contribuye a conseguir predicciones más precisas para la arquitectura Multimodal *TwoStream* a un bajo coste.

Finalmente, la Tabla 4.8 compara los tiempos de inferencia, consumos de energía y F1-Score de cada una de las redes individuales, así como del modelo Multimodal final. La

arquitectura Multimodal ofrece rendimiento en tiempo real con tiempos de inferencia por debajo de 1,7 segundos procesando lotes de 64 fotogramas (2560 ms). La red eficiente *1DCNN+LSTM* es capaz de analizar con un consumo energético muy reducido una señal de 6,6 segundos de duración en tan solo 16 ms. Como resultado, combinar información de vídeo y pulsos conduce a una solución con una capacidad de reconocimiento mejorada en términos de F1-Score de 2,4%, a un bajo coste computacional y energético adicional con respecto a la red de vídeo.

Tabla 4.8. Rendimiento arquitectura Multimodal en la Nvidia Jetson Nano

Modelo	Tiempo de inferencia (ms)	GPU (W)	Dispositivo (W)	F1-Score
RGBI3D	1604,15 $\pm$ 0,702	1,24 $\pm$ 0,87	3,68 $\pm$ 0,97	76,41
1DCNN+LSTM	16,60 $\pm$ 0,480	0,09 $\pm$ 0,02	2,42 $\pm$ 0,05	30,48
<i>TwoStream</i> Multimodal	1626,96 $\pm$ 2,812	1,25 $\pm$ 0,87	3,75 $\pm$ 0,97	78,27

Por último, para ilustrar el consumo energético, la arquitectura Multimodal podría ejecutarse de forma continuada durante más de 30 horas en una Jetson Nano conectada a una batería de 5V 25.000mAh.

4.1.4 Optimización de arquitecturas de aprendizaje profundo

Los modelos introducidos hasta ahora ofrecen una gran capacidad de reconocimiento de actividades cotidianas en el hogar. Sin embargo, las arquitecturas más precisas son también las más exigentes computacionalmente, con los efectos asociados que esto supone: tiempos de inferencia mayores o una utilización de gran cantidad de recursos de cómputo y memoria, lo cual también se traduce en un elevado consumo energético.

Nuestra propuesta consiste en introducir un sistema ciberfísico distribuido con arquitectura nodo-nube, donde los nodos de procesamiento local son los encargados de ejecutar tareas de análisis de aprendizaje automático de forma eficiente y con escasos recursos. Acercar los nodos de cómputo a la fuente de los datos contribuye a reducir la latencia de respuesta del análisis y a preservar la privacidad de los pacientes evitando compartir información sensible fuera de la red local. A pesar de esto, los nodos de procesamiento son dispositivos de muy limitados recursos, por ejemplo, la Nvidia Jetson Nano cuenta tan solo con 128 núcleos CUDA en su GPU, que es menos de un 1% del total de núcleos de los que dispone una GPU de altas prestaciones de nueva generación como la RTX 4090.

Ante este problema, la optimización de los modelos juega un papel crucial para lograr el despliegue de modelos precisos y eficientes en dispositivos de bajos recursos. Una de las

técnicas más populares para la aceleración y compresión de los modelos es la destilación de conocimiento. Este método permite implementar una nueva arquitectura que reduce el tiempo de inferencia en el nodo a la vez que ofrece una elevada precisión, sacando partido al conocimiento extraído por modelos más complejos.

4.1.4.1 Destilación de conocimiento

Para permitir el despliegue de arquitecturas precisas en dispositivos empotrados de bajos recursos, se optimizan los modelos a través de la técnica de destilación de conocimiento. Se estudia cómo es posible mejorar la precisión de arquitecturas eficientes destilando conocimiento a partir de redes neuronales multimodales (de vídeo y estimación de movimiento). En concreto, se mejora el F1-Score de uno de los modelos computacionalmente más eficientes diseñados en esta tesis en un 8,3%, logrando un F1-Score de 82,65% sobre la modalidad de vídeo del IAPV dataset con un rendimiento de más de 250 fps en los nodos de procesamiento locales Jetson Nano. Esta optimización permite el análisis en tiempo real de varios sujetos en un mismo nodo de procesamiento de forma concurrente.

La destilación de conocimiento consiste en la transferencia de conocimiento de uno o varios modelos complejos (*Teacher*) y precisos a una arquitectura más simple y eficiente (*Student*), de forma que esta segunda arquitectura alcance una precisión más elevada a la que se conseguiría entrenando solamente sobre un conjunto de datos.

Esta técnica difiere del *Transfer Learning* en varios aspectos. Por un lado, el *Transfer Learning* aprovecha los pesos preentrenados de una arquitectura para resolver una tarea y los utiliza para que el proceso de entrenamiento sobre otra tarea de similares características sea más rápido, estable y tienda menos al sobreajuste cuando se trabaja con conjuntos de datos de pequeñas dimensiones. Sin embargo, la arquitectura de la red es exactamente la misma, por lo que salvando pequeñas diferencias que puedan producirse, el rendimiento y uso de recursos será el mismo.

Por otro lado, para la destilación de conocimiento, el modelo complejo de partida sí debe estar entrenado para resolver la tarea deseada. En cambio, por medio de esta técnica es posible transferir el conocimiento adquirido sobre el problema a una nueva arquitectura más simple y eficiente, que no tiene por qué ser similar ni procesar la misma información que el modelo original.

En esta tesis, para estudiar los beneficios de la destilación de conocimiento en la clasificación de acciones se ha diseñado una versión *TwoStream* de la red *S3DG*, que analiza datos de vídeo y de estimación de movimiento para el reconocimiento de acciones. Esta alternativa ofrece una gran capacidad de reconocimiento, pero tiene el inconveniente de que es muy exigente computacionalmente. Por ello, se diseñan modelos más simples

Student a los que destilar el conocimiento, de forma que el análisis de vídeo y de estimación de movimiento contribuya a mejorar de manera implícita la bondad de ajuste de las alternativas más eficientes.

La red *S3DG* es una versión optimizada de la arquitectura *RGBI3D* que integra convoluciones temporalmente separables que contribuyen a reducir el número de parámetros y operaciones en coma flotante de la red. En concreto, se escogen y adaptan los modelos *RGBI3D* y *S3DG* basados en 3D-ConvNets para su despliegue en los nodos de procesamiento empujados.

El análisis de los beneficios generados por parte de la destilación se realiza sobre el conjunto de datos *Indoor Action Dataset* (ver epígrafe 3.2.1.3). Este conjunto es la modalidad de vídeo del *dataset Indoor Action Pulses and Video* (IAPV), construido durante el desarrollo de esta tesis. Este conjunto de datos contiene ejemplos muy representativos de personas realizando actividades cotidianas en el hogar grabadas con cámaras en una posición fija. De esta manera, el movimiento en la escena viene originado por los sujetos y su interacción con el entorno. Este conjunto de datos representativo de actividades cotidianas es limitado en término del número total de ejemplos (alrededor de 1300) en comparación con grandes *datasets*. El número medio de ejemplos por clases es $122 (\pm 63 \sigma)$ por lo que es un conjunto de datos desbalanceado.

Por esta razón, para este estudio también se utiliza un conjunto de datos con un gran número de ejemplos. En concreto, se utiliza una versión refinada del *Combined Indoor Action Video (CIAV) Dataset* [60] con más de 14000 clips. El *CIAV dataset* (ver epígrafe 3.2.1.1) es un conjunto revisado manualmente para evitar alimentar a la red con etiquetas erróneas y para homogeneizar la representación de cada una de las clases. Sin embargo, la calidad de los datos que representan estas actividades es aún deficiente (por ejemplo, algunas muestras sufren de movimiento de cámara o de un fuerte desenfoque de movimiento). A pesar de ello, el uso de un conjunto de datos grande favorece la generalización de los modelos, previniendo el sobreajuste.

A continuación, se comparan diversas arquitecturas basadas en convoluciones 3D y redes convolucionales recurrentes. En particular, se seleccionan las arquitecturas *RGBI3D* y *S3DG* como punto de partida de los modelos diseñados. Sin embargo, para disponer de un abanico de arquitecturas más completo, también se ha diseñado el modelo *Recurrent 2D* que integra una red convolucional 2D (MobileNetV2 [141]) entrenada de forma conjunta a una capa *Long-short Term Memory* (LSTM) [170].

Entrenamiento: se valoran diferentes configuraciones para las tres arquitecturas seleccionadas (*RGBI3D*, *S3DG*, y *Recurrent2D*) donde se proponen variaciones de resolución para las dimensiones espaciales y temporales. En concreto, se muestrean fotogramas equidistantes, escogiendo [8, 16, 32, 64] fotogramas equidistantes de una secuencia de 64 imágenes; a su vez, la entrada también se muestrea a diferentes resoluciones [112, 140, 168, 196, 224].

Con respecto al proceso de entrenamiento, en primer lugar, cada modelo se pre-entrena en el *dataset*, con gran número de ejemplos, *CIAB Dataset*. Seguidamente, se realiza un ajuste fino (*Fine Tuning*) de los modelos sobre el conjunto de datos *Indoor Action Dataset*. Seguir esta metodología contribuye a sacar provecho del conocimiento aportado por el conjunto de datos con un mayor número de ejemplos para favorecer la capacidad de generalización de los modelos, y posteriormente especializarlo para identificar acciones de interés en este trabajo, actividades cotidianas (*Indoor Action Dataset*).

Posteriormente, los modelos se destilan usando la red *TwoStream* como *Teacher* para mejorar la capacidad de reconocimiento sin afectar negativamente al rendimiento o uso de recursos de las alternativas. En concreto, se selecciona el modelo *TwoStream S3DG* como Maestra, la cual está diseñada para procesar un conjunto de 64 fotogramas con una resolución espacial de 224x224 píxeles. Esta arquitectura es el modelo más complejo (250 GFLOPs), y ofrece un F1-Score de 89,47, ejecutándose a menos de 7 fps en los nodos locales, un rendimiento muy alejado del tiempo real (al menos 25 fps).

Análisis: Posteriormente, se comparan los modelos en términos de rendimiento y precisión sobre el conjunto de test del *Indoor Action Dataset*. Por un lado, se escoge como métrica de precisión la macro F1-Score. Por otro lado, el rendimiento se evalúa en términos de fotogramas por segundo (fps) al ejecutar el modelo en los nodos del borde. Concretamente, un uso menor de recursos se traduce en inferencias más rápidas (mayores fps) y menor consumo de energía en comparación con alternativas computacionalmente más complejas que necesitan más tiempo para ejecutar una inferencia, consumiendo más energía.

La Tabla 4.9 muestra una selección de las arquitecturas destiladas y se señala el porcentaje de mejora de F1-Score conseguido tras la destilación. Los modelos se etiquetan de la siguiente forma *modelo_muestreoTemporal_resoluciónEspacial*: donde *modelo* indica la arquitectura de red, *muestreoTemporal* es la resolución temporal y *resoluciónEspacial* es la resolución espacial. Por ejemplo, el modelo *RGBI3D_16_112* está basado en la arquitectura *RGBI3D* que procesa 16 fotogramas (de cada 64) con una resolución de 112x112.

Tabla 4.9. Mejora aportada por la destilación en las arquitecturas de DL

Arquitectura	Inferencia (ms)	F1-Base	F1-Destilado
RGBI3D_8_224	401	80,85	84,88 (5,0% ↑)
RGBI3D_16_112	239	76,34	82,65 (8,3% ↑)
RGBI3D_16_224	751	86,16	86,52 (0,4% ↑)
RGBI3D_64_140	1275	85,23	88,06 (3,3% ↑)

RGBI3D_64_168	1804	84,65	87,56 (3,4% ↑)
RGBI3D_64_196	2408	87,67	90,55 (3,3% ↑)
S3DG_8_224	416	84,41	85,63 (1,4% ↑)
S3DG_16_112	256	78,12	79,90 (2,3% ↑)
S3DG_16_224	807	86,55	88,13 (1,8% ↑)
S3DG_64_140	1306	78,56	84,12 (7,1% ↑)
S3DG_64_168	1842	88,18	89,13 (1,1% ↑)
S3DG_64_196	2553	87,04	91,74 (5,4% ↑)
Recurrent2D_8_224	133	75,19	76,64 (1,9% ↑)
Recurrent2D_16_112	96	64,32	66,78 (3,8% ↑)
Recurrent2D_16_224	303	74,98	76,91 (2,6% ↑)
Recurrent2D_64_168	633	71,80	73,16 (2,1% ↑)

Antes del procedimiento de destilación, el valor F1-Score medio era $80,62 \pm 6,82$, y tras la destilación el F1-Score alcanza una mejora media de un 3,2%. Aunque pueda parecer una mejora marginal, esta varía dependiendo de la arquitectura de las soluciones, ofreciendo mejoras de hasta un 8,3% y 7,1% para los modelos *RGBI3D_16_112* y *S3DG_64_140* respectivamente. Una de las mayores ventajas que ofrece la destilación es que el tiempo de inferencia requerido para procesar un flujo de vídeo (2560 ms – 64 imágenes a 25 fps) no se ve perjudicado a pesar de la mejora de precisión.

Por motivos de visualización, la Figura 4.12 solo muestra los modelos más representativos, escogiendo aquellos que ofrecen un mejor equilibrio entre precisión y rendimiento. El eje vertical de la izquierda muestra el F1-Score alcanzado por los modelos (F1-Score antes de destilar en color naranja). La línea roja indica el tiempo de inferencia (en fps) de los modelos optimizados al ejecutarse en los nodos *edge* (ver eje derecho). Los modelos se muestran en el eje horizontal, usando columnas apiladas que muestran el F1-Score en naranja y la mejora obtenida tras la destilación en color azul. Destaca el compromiso de eficiencia y precisión de los diferentes modelos: en la parte derecha de la figura los modelos alcanzan valores de precisión más elevados en detrimento de tiempos de inferencia más altos; la parte izquierda muestra modelos con un rendimiento superior a 250 fps pero obteniendo alrededor de 10 puntos menos de F1-Score. En resumen, las soluciones con un mayor valor de F1-Score tienen un mayor impacto en término de uso de recursos y rendimiento. Hay que tener presente que el tamaño de entrada de los modelos (resolución espacial y temporal) tiene una consecuencia directa en la precisión y el rendimiento: resoluciones más bajas producen soluciones más eficientes, pero menos precisas.

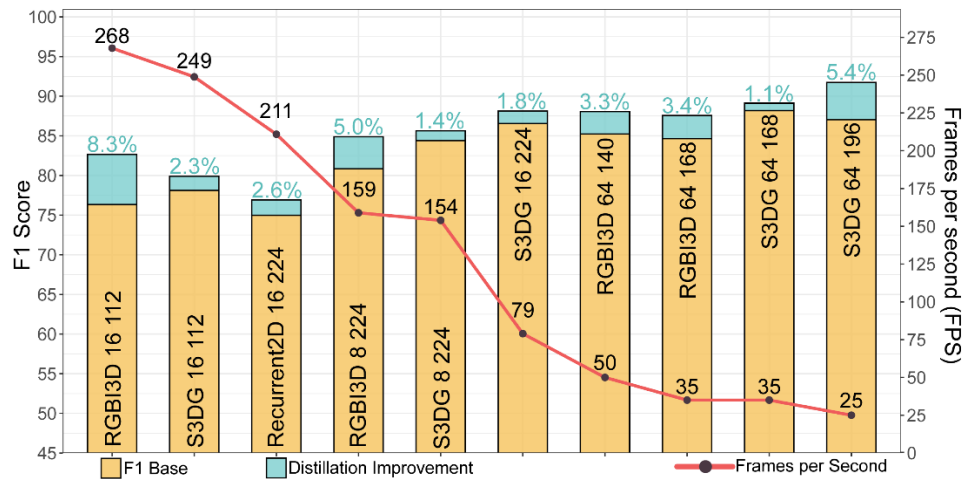


Figura 4.12. Evaluación del rendimiento de modelos para el reconocimiento de acciones [156]. El valor de F1-Score se muestra en el eje vertical izquierdo, en las columnas apiladas también se indica en azul la mejora obtenida por medio de la destilación. El eje vertical derecho muestra el rendimiento (fps) en los nodos locales; los modelos (en columnas) están ordenados de mayor a menor rendimiento de izquierda a derecha (línea roja). En concreto, las soluciones mostradas a mitad del eje horizontal son arquitecturas que ofrecen un buen balance entre precisión y rendimiento. Concretamente, modelos como Distilled S3DG_16_224 o Distilled RGBI3D_64_140 logran una mayor precisión en comparación con la arquitectura más intensiva en recursos no destilada (S3DG_64_196 – 25 fps), con un F1-Score superior al 88%, ofreciendo rendimiento en tiempo real a más de 79 y 50 fps respectivamente.

En cuanto a la mejora obtenida por medio de la destilación, indicada en color azul en las columnas, se muestra la mejora conseguida de F1-Score de los modelos al transferir conocimiento desde la red *Teacher TwoStream*. Por ejemplo, el F1-Score del modelo más eficiente mostrado en la Figura se ve mejorado a 82,65 puntos, un aumento de un 8% de precisión conservando el mismo rendimiento. Esto hace que la diferencia de precisión entre la alternativa destilada más eficiente (*Distilled RGBI3D_16_112*) y la arquitectura más precisa antes de la destilación (*S3DG_64_196*) se reduzca de 12 a tan solo 4 puntos de F1-Score. Además, la arquitectura destilada usa un 87% menos de recursos computacionales (14 vs. 102 GFLOPs), lo que se traduce en inferencias hasta 10 veces más rápidas. En otras palabras, antes de la destilación y comparando estos dos modelos, la red *RGBI3D_16_112* ofrece 5,45 puntos de F1-Score por GFLOP. Tras la destilación, este modelo eficiente ofrece un 8% más puntos de F1-Score por cada GFLOP, que es significativamente mayor a la alternativa *S3DG_64_196* que ofrece 0,85 puntos de F1-Score por GFLOP.

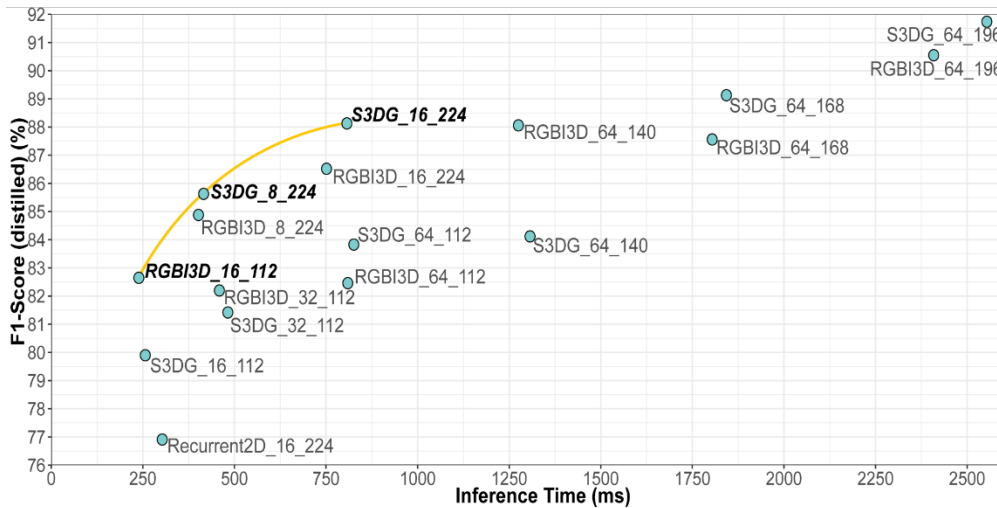


Figura 4.13. Compromiso de F1-Score vs. tiempo de inferencia de los modelos destilados [156]. El tiempo de inferencia (en milisegundos) indica el tiempo necesario para efectuar una inferencia sobre una muestra de vídeo de 2560 ms (64 fotogramas a 25 fps) en el nodo. En el gráfico, habría que buscar aquellos modelos con menores tiempos de inferencia y valores de F1-Score más altos. Por lo tanto, para el despliegue del sistema se seleccionan las alternativas situadas en la parte superior izquierda de la Figura (conectadas por una línea amarilla), ya que ofrecen el mejor balance entre F1-Score y uso de recursos.

Por otra parte, la Figura 4.13 muestra el compromiso entre F1-Score y tiempo de inferencia en el *edge* de los modelos destilados. El propósito de esta Figura es seleccionar los modelos más apropiados para desplegarlos en el nodo. Se seleccionan aquellas arquitecturas con menores tiempos de inferencias y valores de precisión más elevados. En el gráfico, estos modelos se resaltan en la parte superior izquierda del gráfico (conectados por una línea amarilla): *Distilled RGBI3D_16_112*, *Distilled S3DG_8_224*, y *Distilled S3DG_16_224*. La ejecución de soluciones eficientes poco intensivas en recursos permitiría analizar varios flujos de vídeo en el mismo nodo, optimizando así el uso de los recursos de cómputo del sistema distribuido. En cambio, los modelos más precisos no permitirían monitorizar a más de una persona simultáneamente en un único nodo.

En resumen, a través de la destilación de conocimiento se optimizan las arquitecturas de aprendizaje automático, abriendo la posibilidad de analizar varias cámaras de forma concurrente en un mismo nodo de procesamiento empujado. Esto es posible gracias a la reducción de recursos manteniendo una precisión muy cercana a la ofrecida por modelos más complejos que consumen hasta 10 veces más recursos.

Finalmente, en la Figura 4.14 se muestra de forma cualitativa el análisis de los modelos seleccionados sobre muestras de la modalidad de vídeo del *IAPV dataset*. Destaca que el modelo más eficiente (*Distilled RGBI3D_64_112*) es capaz de identificar correctamente las acciones, aunque no lo haga con la máxima confianza. Por otro lado, la arquitectura más precisa que consume una mayor cantidad de recursos es capaz de clasificar con un 99% de confianza todas las situaciones. Esto es de especial relevancia para la

identificación de emergencias, contribuyendo a la reducción de falsas alarmas y a la confirmación de acciones críticas.

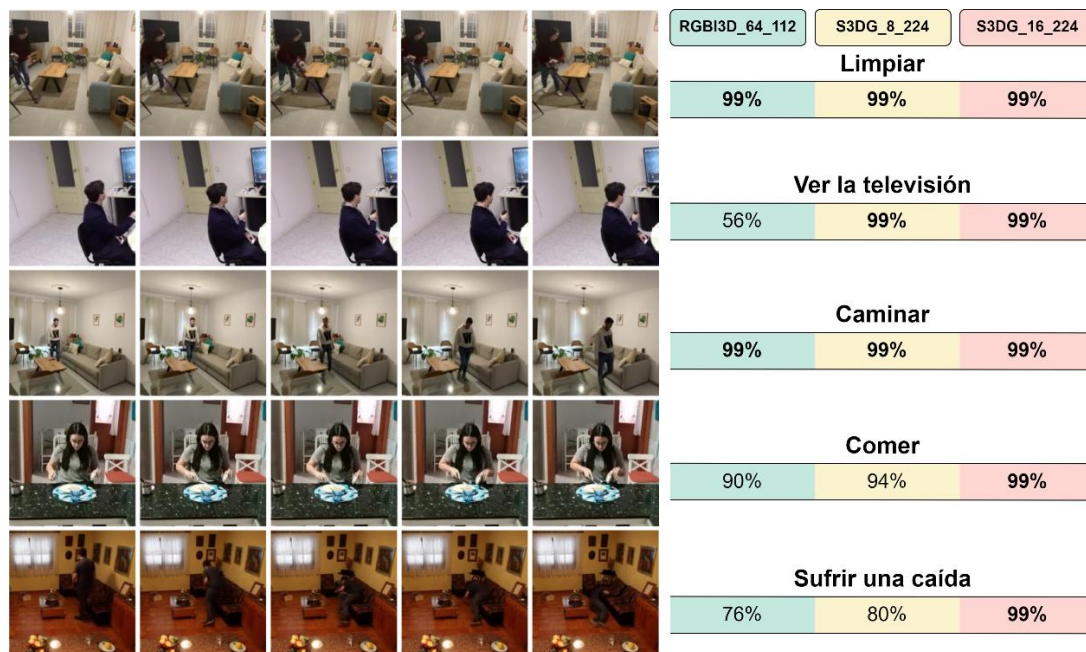


Figura 4.14. Confianza de las predicciones de los modelos destilados seleccionados sobre muestras de test del conjunto de datos.

4.2 Sistema ciberfísico reconfigurable

Una vez diseñado y desarrollado el módulo de reconocimiento de acciones que formará parte del sistema ciberfísico, se presentan el resto de los componentes integrados en el CPS con la finalidad de evaluar los beneficios aportados por la implementación de estos módulos en la aplicación.

En particular, se introduce un CPS eficiente y reconfigurable para la monitorización de actividades en dispositivos empotrados, integrando la monitorización remota y una herramienta QRM, así como una plataforma de virtualización de recursos remotos (*PoCL-R*) (ver Figura 4.15).

El procesamiento de vídeo se basa en arquitecturas de *Deep Learning* del estado del arte [75] que son optimizadas para el despliegue en los nodos locales: dispositivos Nvidia Jetson. Estos dispositivos empotrados ejecutan de forma automática el reconocimiento de acciones y son capaces de reconfigurarse automáticamente. Las soluciones que se proponen en este trabajo para estos nodos en el borde de la nube alimentados por baterías ofrecen un rendimiento en tiempo real a un coste energético muy bajo.

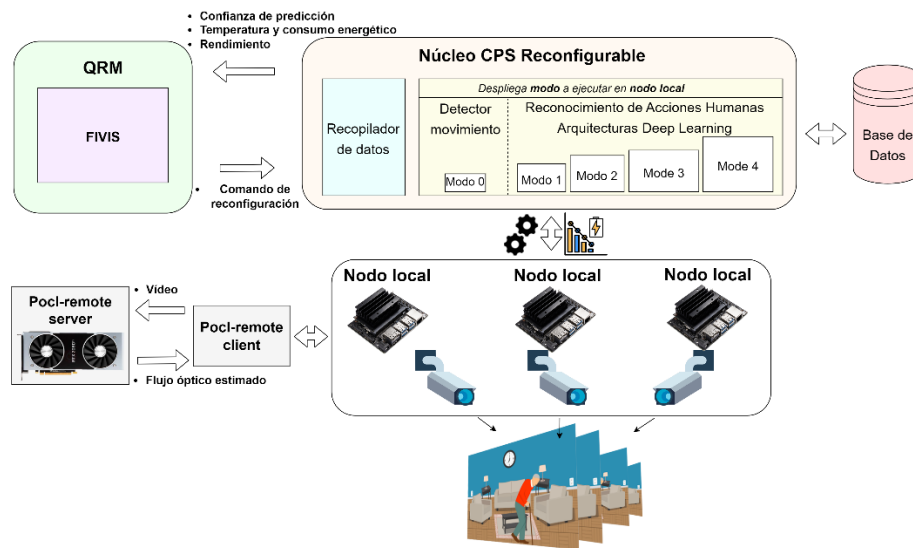


Figura 4.15. Arquitectura de sistema ciberfísico reconfigurable. Los nodos locales realizan la monitorización de acciones en interior a partir de vídeos formados por lotes de 64 fotogramas capturados a 25 fps (fotogramas por segundo). El núcleo reconfigurable (reconfigurable core) del sistema recibe las etiquetas de las acciones identificadas e información acerca de diversas calidades de los dispositivos locales. Las calidades del sistema se monitorizan de forma remota a través de la herramienta QRM, FIVIS. FIVIS analiza los datos y toma decisiones acerca de reconfiguraciones en tiempo de ejecución para optimizar la precisión del reconocimiento de acciones y el tiempo de funcionamiento del sistema (teniendo en cuenta que los nodos locales de procesamiento están alimentados por baterías). Finalmente, PoCL-R permite el offloading (ejecución remota) de la estimación de flujo óptico en un servidor de altas prestaciones para la mejora de la precisión global del sistema en la identificación de situaciones críticas a la par que se evita un consumo energético excesivo en los nodos locales.

Con respecto al módulo de reconocimiento de acciones, por motivos de simplicidad se integran los modelos presentados en el capítulo 4.1.2, ya que lo que se busca es evaluar las ventajas que aporta la integración de un módulo de monitorización en línea y de Gestión de Calidades y Recursos (QRM) en el sistema para optimizar la precisión, rendimiento y el consumo de recursos de la solución final.

Finalmente, se presenta una política de reconfiguración que persigue dos objetivos: 1) extender el tiempo de funcionamiento del sistema alimentado por baterías al ser reducido su consumo energético; 2) optimizar la precisión global del sistema al identificar acciones **críticas**. Por último, se evalúan los beneficios aportados por la adopción de esta política de reconfiguración ejemplificando varios escenarios diferentes.

Con respecto a los nodos locales de procesamiento, se implementa el Reconocimiento de Acciones Humanas en el sistema mediante su despliegue en los dispositivos empujados. Estos dispositivos SoC energéticamente eficientes disponen de 4 núcleos CPU y una GPU con 128 núcleos CUDA, compartiendo 4GB de memoria principal. Es un dispositivo

potente, pero es limitado en términos de recursos para aplicaciones de aprendizaje automático en comparación a arquitecturas *hardware* de altas prestaciones utilizadas para procesar soluciones del estado del arte. Este es precisamente el motivo por el que la reconfiguración tiene un papel muy importante en este sistema, ya que permite hacer un uso optimizado de los recursos al desplegar el sistema en los nodos locales de bajo costes y energéticamente eficientes. Una de las ventajas que presenta la plataforma *hardware* Jetson Nano, es que es un dispositivo totalmente configurable, lo que supone que tanto la frecuencia de operación de la CPU y GPU pueden ser modificadas. Con vistas a minimizar en la medida de lo posible el consumo energético, se configuran los nodos para: 1) usar tan solo 2 núcleos CPU a 1,2Ghz; 2) se reduce la frecuencia de operación de la GPU a 614Mhz.

4.2.1 Política de reconfiguración del sistema

La introducción de una política de reconfiguración permite la adaptación en tiempo de ejecución del sistema. La optimización multi-objetivo busca maximizar el tiempo de funcionamiento de los nodos de procesamiento locales (con baterías) y minimizar las falsas alarmas asociadas a situaciones críticas mediante la mejora de la precisión del reconocimiento de acciones. Se proponen cuatro modos de operación posibles para el despliegue de diversos modelos en los nodos que ofrecen un distinto compromiso entre precisión y consumo de recursos. La toma de decisiones se realiza en base al análisis del tipo de acción detectada (breve o larga duración, crítica o no) y del consumo energético de la solución.

Comúnmente, se suele optimizar el diseño los modelos de *DL* para lograr un rendimiento al nivel del estado del arte. Sin embargo, cuando se despliegan en el mundo real, puede haber otros requisitos u objetivos que también deban cumplirse para garantizar un buen funcionamiento del sistema. Para lidiar con la optimización multi-objetivo, los sistemas del mundo real deben ser provistos de mecanismos para la adaptación en tiempo de ejecución. Estos mecanismos se controlan por medio de políticas de reconfiguración, cuyo diseño viene determinado a partir de objetivos de optimización específicos.

En este trabajo, la optimización multi-objetivo busca maximizar el tiempo de funcionamiento de los nodos de procesamiento locales (con baterías) a la vez que se minimizan las falsas alarmas asociadas a situaciones críticas mediante la mejora de la precisión del reconocimiento de acciones. Esta política de reconfiguración se implementa como una tarea de análisis ejecutada en FIVIS, que monitoriza y analiza las calidades a nivel de sistema y aplicación y lanza comandos de reconfiguración para asegurar el cumplimiento de los objetivos propuestos.

Tabla 4.10. Modos de operación de los nodos locales

Modo ID	Nombre	F1-Score	Inferencia (ms)	Consumo (W)
0	Detector de movimiento	-	-	1,76 ± 0,03
1	RGBI3D 16 – 112x112	78,00	426,50	2,50 ± 0,74
2	RGBI3D 32 – 112x112	79,70	829,78	2,86 ± 0,96
3	RGBI3D 64 – 112x112	80,50	1595,42	3,61 ± 0,98
4	TwoStream I3D (PoCL-R) 64 - 112x112	83,68	3060,55	4,44 ± 0,55

En la Tabla 4.10 se resumen los cinco modos de funcionamiento en los que puede operar el sistema usando las arquitecturas de aprendizaje profundo introducidas en el capítulo 4.1.2. Los modos de operación presentados están ordenados con números del 0 al 4 dependiendo de su F1-Score y consumo de energía (desde más bajo a más alto). El *Modo 0* es un simple *detector de movimiento*, este modo de funcionamiento evita estar ejecutando ninguna arquitectura de *DL*, y por lo tanto, es la alternativa más eficiente. El *detector de movimiento* sólo analiza las diferencias presentes entre dos fotogramas separados por un segundo de distancia; si no se detecta diferencia entre las imágenes, se considera que no ha habido cambios en la escena. Los modos del 1 al 4 ejecutan las redes neuronales seleccionadas: el *modo 1* ejecuta el modelo más simple (*RGBI3D 16 – 112x112*), siendo la alternativa que menos recursos consume; el *modo 2* activa el uso de la red *RGBI3D 32 – 112x112*, este modo de funcionamiento ofrece un valor de F1-Score razonable a un coste bajo de energía; el *modo 3* por su parte implementa el modelo *RGBI3D* que alcanza la mayor precisión garantizando un procesamiento en tiempo real (*RGBI3D 64 – 112x112*), a un coste energético medio; Por último, el *modo 4* hace uso del modelo *TwoStream I3D 64 – 112x112* (ver capítulo 4.1.3.1) para confirmar si una acción crítica ha ocurrido realmente, reduciendo así las falsas alarmas de emergencia.

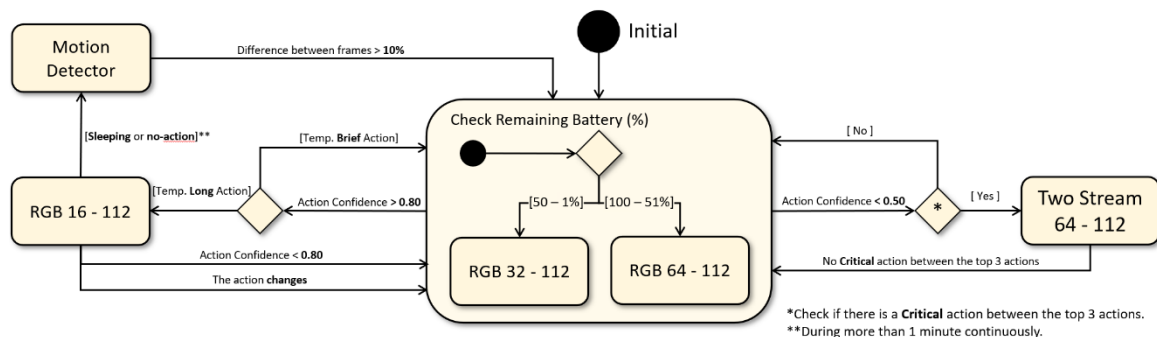


Figura 4.16. Diagrama de transición de estados de la política de reconfiguración [60]. Se muestra el diseño de la política de reconfiguración para la adaptación del sistema en tiempo de ejecución lanzando comandos de reconfiguración. Izquierda) los modos 0 y 1 se activan cuando se identifican acciones de **larga** duración; centro) los modos de operación 2 y 3 se

despliegan cuando los sujetos están realizando acciones **breves**, la decisión de cuál de los dos modos se activa depende del nivel de batería restante en el nodo local; derecha) el modo de operación 4 se utiliza para confirmar si una acción crítica realmente ha ocurrido.

La política de reconfiguración se describe en un diagrama de transición de estados en la Figura 4.16 que representa cómo el sistema se adapta de forma automática en tiempo de ejecución, lanzando comandos de reconfiguración.

- En el diagrama de transición de estados, cuando los sujetos están realizando acciones **breves** se utiliza el *modo 2* o *modo 3*: si la carga de batería restante es superior al 50%, se activa el *modo 3*, ofreciendo una gran capacidad de reconocimiento; De lo contrario, se habilita el *modo 2* una solución más eficiente en términos de energía y que contribuye a la extensión del tiempo de funcionamiento de las baterías a expensas de una reducción limitada de la precisión.
- Cuando el sistema identifica acciones de **larga** duración con una alta confianza, se despliega el *modo 1* en los nodos *edge*. Se monitoriza al sujeto que está realizando esta acción de **larga** duración de forma continua en este modo de operación, utilizando una pequeña cantidad de recursos computacionales. Un cambio de acción o una reducción de la confianza de la identificación de esta provoca que el sistema cambie de nuevo al *modo 2* o *modo 3*.
- Cuando la tarea de análisis detecta que el sujeto se encuentra durmiendo o no hay nadie en la escena durante unos pocos minutos, se despliega el *modo 0*. Este modo de funcionamiento consigue ahorrar al sistema una cantidad significativa de recursos y energía. Cuando se detecte movimiento en la escena, se lanza una reconfiguración para cambiar al *modo 2* o *modo 3*.
- El *modo 4* se despliega en los nodos de procesamiento cuando se reconoce una acción **crítica**. Para confirmar que una situación potencialmente peligrosa pudo haber ocurrido, el sistema reanaliza el vídeo activando el *modo 4*, que usa el modelo *TwoStream I3D*. Posteriormente, si el sistema detecta un falso positivo (confirma que esta situación no ocurrió), se lanza una reconfiguración para volver al modo de funcionamiento 2 o 3.

4.2.2 Evaluación del sistema en escenarios reales

La integración de la reconfiguración aporta gran flexibilidad al sistema, se valida sobre escenarios prácticos cómo la implementación de la política de reconfiguración introducida permite extender hasta en un 63% el tiempo de operación del sistema con nodos alimentados por baterías a expensas de una limitada pérdida de precisión. Además, la reconfirmación de posibles situaciones críticas permite mejorar la capacidad de reconocimiento de la solución al identificar emergencias, reduciendo así la generación de falsas alarmas. Por otro lado, la asignación eficiente de tareas a recursos y el despliegue de modelos destilados puede llegar a reducir hasta en un 75% los recursos de cómputo necesarios (nodos de procesamiento) para ejecutar la solución.

Para ilustrar los beneficios del sistema propuesto se definen dos escenarios: 1) la rutina de actividades diarias de una persona mayor en su hogar según los datos de [171]; 2) un accidente que provoca una alarma. En este experimento, por motivos de simplicidad, se asume que solo se despliega un único nodo de procesamiento. Para estimar el tiempo de funcionamiento total y el rendimiento del sistema se valora el uso de unas baterías de 5V 25000 mAh para los nodos locales. Con una capacidad de 125 Wh, el sistema puede operar durante 25 horas consumiendo en media 5W.

Finalmente, se incluye un tercer escenario donde se muestra un caso de uso en el que se despliega el sistema *edge-cloud* en una residencia de ancianos y se evalúan los beneficios de hacer un uso optimizado de los recursos disponibles mediante la asignación eficiente de tareas a recursos.

4.2.2.1 Escenario 1: Un día de una persona de avanzada edad

Este escenario muestra la monitorización de una persona de avanzada edad en su día a día. La Figura 4.17 muestra la media de horas al día que dedica una persona mayor a diferentes actividades según la Agencia de Estadísticas Laborales y el Ministerio de Trabajo de Estados Unidos [171].

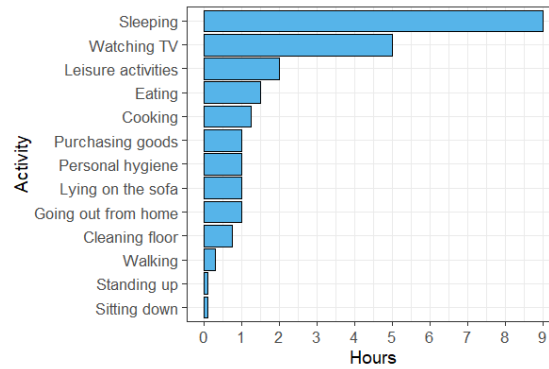


Figura 4.17. Media de horas al día dedicadas a distintas actividades por una persona de avanzada edad (mayor de 65) [171].

Teniendo en consideración las actividades seleccionadas para nuestro sistema, la Figura 4.18 muestra una comparativa del tiempo teórico de funcionamiento del sistema dependiendo de si se implementa o no la política de reconfiguración descrita en el epígrafe 4.2.1. Las tres primeras columnas muestran casos donde no se ha integrado la política de reconfiguración, en las que el sistema ejecuta un único modelo de *DL* respectivamente: *RGBI3D 64 – 112x112 (modo 3)*, *RGBI3D 32 – 112x112 (modo 2)*, *RGBI3D 16 – 112x112 (modo 1)*.

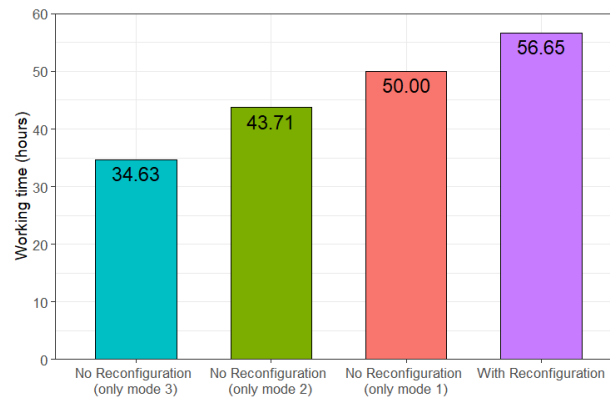


Figura 4.18. Tiempos de funcionamiento estimados de los nodos locales alimentados por baterías de 25V 25000mAh [60]. Considerando la duración y actividades descritas en la Figura 4.17. De izquierda a derecha, las columnas de la 1 a la 3 muestran el tiempo de funcionamiento para los modos de operación del 3 al 1 respectivamente, la cuarta columna muestra la extensión del tiempo de funcionamiento del sistema cuando se implementa la reconfiguración en tiempo de ejecución.

La Figura 4.18 muestra como el tiempo de operación del sistema puede extenderse hasta en un 63% cuando se llevan a cabo reconfiguraciones, llegando a poder ejecutarse durante 22 horas adicionales en comparación con sólo desplegar el *modo 3*. También, se observa como el tiempo de funcionamiento se amplía en un 13% con el respecto al *modo 1*, el cual utiliza el modelo de Aprendizaje Profundo más eficiente que se ha seleccionado.

Por medio de la integración de las reconfiguraciones, es posible activar el *modo* de operación más apropiado en función de la situación. La adaptación del sistema contribuye a garantizar un buen rendimiento de clasificación y a optimizar el consumo energético del mismo. Como se ha podido comprobar, la gestión automática del uso de recursos de los nodos de procesamiento permite grandes ahorros de consumo de energía. Esto es posible por medio de la implementación de la política de reconfiguración y el uso de métodos o modelos menos intensivos en recursos cuando la habitación se encuentra sin sujetos o se están identificando actividades de **larga** duración (ver Figura 4.19). Esto también garantiza el funcionamiento continuo e ininterrumpido. Hay que considerar que a medida que aumentan el número de nodos de procesamiento utilizados en el sistema, mayor será el ahorro energético.

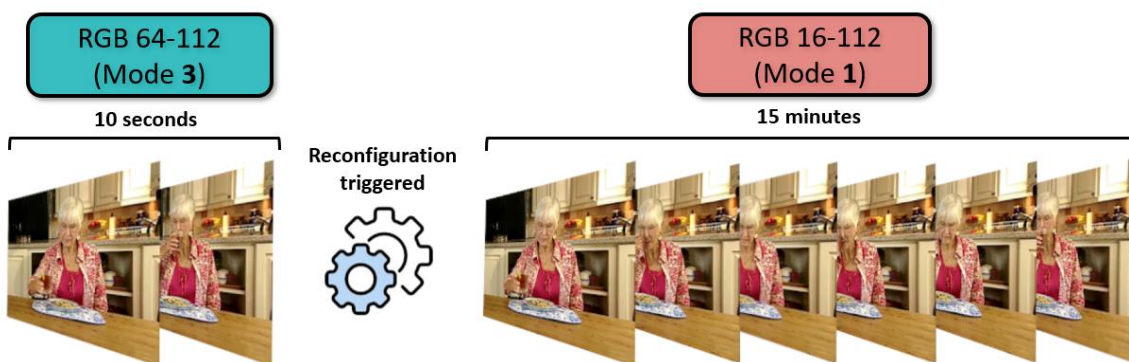


Figura 4.19. Escenario 1: Reducción del consumo energético por medio de la reconfiguración [60]. Una persona está comiendo, y el sistema identifica con una alta confianza que se trata de una acción de larga duración. FIVIS lanza una reconfiguración para pasar a usar un modelo de DL más eficiente para seguir monitorizando la actividad de larga duración. El modelo RGBI3D 16 – 112x112 continúa identificando la acción correctamente pero con una confianza menor. Con esto se puede conseguir hasta un 30% de ahorro de energía durante la monitorización de esta acción en comparación con usar exclusivamente el modelo RGBI3D 64 – 112x112 sin reconfiguración.

El sistema registra continuamente las actividades identificadas, así como otras calidades. Esta información, mostrada en los paneles de visualización de la Figura 4.21 y Figura 4.21 busca asistir a los profesionales de la salud en la monitorización de los hábitos de las personas. Este panel muestra el tiempo medio dedicado por los sujetos a cada actividad y notifica con una alarma cuando una situación crítica se identifica.



Figura 4.20. Panel de visualizaciones del sistema de monitorización de hábitos con información agregada de tres personas.



Figura 4.21. Panel de visualizaciones del sistema de monitorización de hábitos de vida. Panel que muestra sólo la información asociada al sujeto con ID 2. Este cuadro de mando está destinado a profesionales de la salud para la supervisión de los hábitos y el estilo de vida que están siguiendo las personas monitorizadas. Entre otros, se muestra el tiempo medio

caso, destaca como la estimación de flujo óptico extrae información del movimiento que se produce en las caídas.

4.2.2.3 Escenario 3: Despliegue del CPS en residencia de mayores

A continuación, se describe un caso de uso concreto donde se despliega un sistema ciberfísico reconfigurable en el que se implementan las arquitecturas de redes neuronales destiladas presentadas en el capítulo 4.1.4.

Se introduce un CPS *edge-cloud* eficiente para la monitorización de actividades en centros de cuidados de mayores o personas con necesidades especiales, que implementan el análisis en nodos locales de procesamiento distribuidos. Estos nodos identifican acciones a partir de flujos de vídeos utilizando redes neuronales destiladas (una visión a nivel general del sistema se muestra en la Figura 4.23). La finalidad del diseño de este sistema es mitigar los costes del cuidado de la salud de estas instituciones (por ejemplo, residencias de mayores) a la vez que mejorar la calidad de vida de sus residentes. En la solución propuesta, los residentes son monitorizados en sus habitaciones usando nodos locales de visión. Al mismo tiempo, la orquestación y monitorización es llevada cabo por un módulo QRM que se ejecuta en el servidor central.

En primer lugar, la herramienta QRM monitoriza calidades a nivel de nodo y de sistema como el consumo de energía, temperatura o rendimiento, así como la confianza global del sistema en las predicciones, al igual que en la arquitectura de CPS propuesta en el anterior epígrafe. Al monitorizar la confianza de las acciones, la herramienta QRM es capaz de generar alarmas para ofrecer una rápida asistencia cuando se detecta una situación crítica. En segundo lugar, como nuevo añadido al CPS, el servidor central gestiona el uso óptimo de los limitados recursos de cómputo disponibles, distribuyendo el procesamiento entre los nodos siguiendo un enfoque de Balanceo de Carga Adaptativo [160] (ver epígrafe 3.2.6.1), que debe garantizar el funcionamiento en tiempo real del sistema.

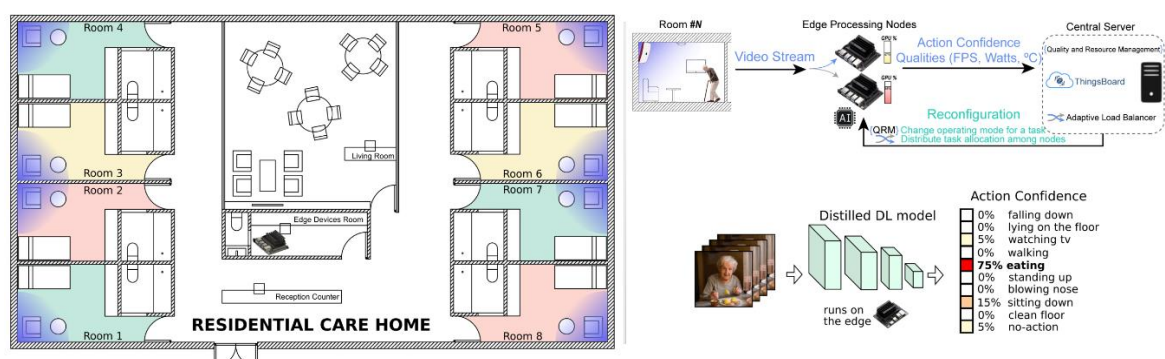


Figura 4.23. Visión general del sistema. Sistema ciberfísico para la monitorización eficiente de actividades en espacios interiores [156]. La solución es una plataforma borde-nube con 1) cámaras distribuidas en las habitaciones de los residentes; nodos locales que ejecutan soluciones de IA para el reconocimiento de acciones en vídeo, a través de modelos

destilados. Los nodos locales son dispositivos System-On-Chip (SoC) con GPUs integradas diseñados para la ejecución de soluciones de aprendizaje automático. El servidor central recopila las predicciones de las acciones de los nodos borde y diversas calidades del sistemas a través de la plataforma de QRM. En la Figura, el color de fondo de las habitaciones señalan el riesgo potencial de caída de los residentes (verde, amarillo y rojo) en función de sus patologías.

Uno de los objetivos de este sistema es fomentar un estilo de vida activo y saludable: la monitorización de forma continua de actividades cotidianas servirá como base a un futuro sistema que facilita información a los profesionales sanitarios y da orientaciones o sugerencias para actividades personalizadas.

Como se mostró en la Figura 4.23, la intención es desplegar el sistema en una residencia de personas de avanzada edad. Todos los nodos de procesamiento y el servidor central se encuentran localizados dentro de las instalaciones de la residencia, evitando así compartir información sensible fuera de la red local. Para el despliegue final, los nodos de procesamiento se desacoplan de las cámaras, de forma que el módulo encargado de la gestión de recursos garantice un uso eficiente de los recursos disponibles, analizando múltiples cámaras de forma simultánea desde un único nodo de procesamiento. Cabe también destacar que en la Figura 4.23, cada habitación tiene un color de fondo diferente (verde, amarillo o rojo). A partir de una evaluación del estado físico y cognitivo del residente realizado por un profesional sanitario, el color de la habitación indica el nivel de riesgo de sufrir una caída del residente: bajo, medio o elevado respectivamente. Esto indica la probabilidad de sufrir una caída por parte de un residente, en el sistema propuesto, esto determina el tipo de análisis que se llevará a cabo con cada usuario, de forma que se haga un uso optimizado de los recursos disponibles a la par que se garantice el mejor cuidado posible a cada persona. Por ejemplo, es sabido que los pacientes de Alzheimer pueden deambular por la noche, y por lo tanto, su nivel de riesgo debe definirse como elevado (color rojo). El sistema tendrá esto en cuenta y monitorizará al residente con el modelo de reconocimiento más preciso, para identificar las caídas robustamente y prevenir cualquier otra consecuencia dañina derivada de la caída.

Procesamiento en el borde distribuido: Brevemente, el análisis de los hábitos de los residentes a través del procesamiento local distribuido ofrece varias ventajas como: procesamiento eficiente, ejecución en tiempo real, uso de ancho de banda y reducción de costes en comparación con plataformas de altas prestaciones en la nube.

El sistema propuesto implementa los modelos destilados mostrados en la Tabla 4.11 (ver capítulo 4.1.4.1), donde se describen los modelos para los pacientes de bajo, medio y elevado riesgo de caída. En esta tabla se incluye información del valor de F1-Score medio, la carga computacional (uso de medio de GPU), y el F1-Score de detección de caídas de cada una de las alternativas. Destaca como se asigna a los pacientes con mayor riesgo el modelo con mayor precisión de identificación de caídas.

Tabla 4.11. Carga computacional en los nodos Jetson Nano

Modelo de DL	Nivel de Riesgo	F1-Score Caídas	F1-Score Medio	Carga Computacional
RGBI3D_16_112	Bajo	90,32	82,65	10%
S3DG_8_224	Medio	93,33	85,63	17%
S3DG_16_224	Elevado	96,77	88,13	32%
Detector de Movimiento	-	-	-	3%

En primer lugar, la optimización de los modelos que se acaban de presentar unido al procesamiento distribuido adaptativo permite el análisis de múltiples habitaciones utilizando un único nodo local. En segundo lugar, cada habitación es analizada utilizando el modelo *DL* asignado al riesgo del residente o con el módulo de detección de movimiento. Se utilizan las arquitecturas más precisas para analizar los flujos de vídeos de residentes con mayor nivel de riesgo en un intento de maximizar la detección de acciones críticas. De la misma manera, los modelos computacionalmente más eficientes se asignan a residentes con un nivel de riesgo medio o bajo.

El módulo de balanceo de carga adaptativo (**asignación eficiente de tareas a recursos**) orquestado por la herramienta QRM en el servidor central contribuye a que el sistema opere en tiempo real sin sobrecargar ningún nodo de procesamiento. A su vez, esto ayuda a minimizar el número de nodos locales necesarios para el procesamiento distribuido.

Por otra parte, el esquema de balanceo de carga adaptativo tiene como objetivo distribuir las tareas de cómputo entre los diferentes nodos de procesamiento para así evitar la sobrecarga de estos. De esta manera, se consigue reducir el uso de recursos en el nodo y su consumo energético a la vez que se garantiza funcionamiento en tiempo real. Por ejemplo, en el escenario de la Figura 4.23 con ocho habitaciones (tres residentes de **bajo** riesgo, dos de **medio**, y tres de **elevado** riesgo), dos nodos locales son suficientes para garantizar el buen funcionamiento del sistema (ver Figura 4.24). Concretamente, al *edge_node#1* se le asigna el análisis de tres residentes de riesgo bajo, uno de riesgo medio y uno de alto riesgo (lo que supone $10+10+10+17+32 = 79\%$ de carga computacional). A su vez, el *edge_node#2* monitoriza los flujos de vídeo provenientes de las habitaciones de un residente de medio riesgo y dos de riesgo elevado ($17+32+32 = 81\%$). Si el sistema tuviese que monitorizar a un nuevo residente, su análisis se le asignaría al *edge_node#1* ya que es el nodo con más recursos disponibles o con la menor carga de trabajo.

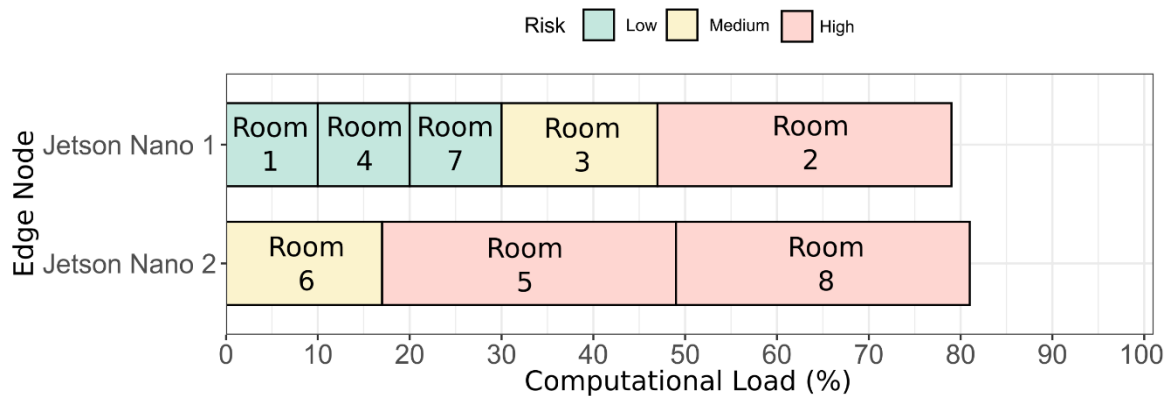


Figura 4.24. Carga computacional de los nodos locales al analizar los residentes de la residencia mostrada en la Figura 4.23. El `edge_node#1` analiza a los residentes de las habitaciones 1, 2, 3, 4 y 7. Y el `edge_node#2` procesa las cámaras de las habitaciones 5, 6 y 8. El módulo de balanceo de carga distribuye las tareas entre los nodos del borde dinámicamente. Si se tiene que monitorizar a un nuevo residente, se asignará de forma automática al nodo con menor carga de trabajo. Cabe destacar que este módulo también monitoriza y revisa la asignación de tareas de forma periódica para distribuir el procesamiento entre los nodos como se ha descrito en el epígrafe 3.2.6.1.

Siguiendo con este mismo ejemplo, se evalúan las consecuencias de la integración de los modelos destilados y de la **asignación eficiente de tareas a recursos** en el rendimiento del sistema, mostrando una comparativa en la que se proponen cuatro escenarios diferentes.

- 1) No se despliegan modelos destilados. Las habitaciones se asignan de antemano a los nodos, usando un nodo por tarea o habitación.
- 2) Se despliegan los modelos destilados. Además, todos los residentes se monitorizan usando el mismo modelo. En este caso, se utiliza el modelo asociado a las personas de alto riesgo para monitorizar a todos los residentes. Las tareas se distribuyen mediante una estrategia de *Round-Robin*.
- 3) Se integra el módulo QRM con reconfiguración en tiempo de ejecución, pero se utiliza una estrategia simple de distribución de tareas como *Round-Robin*.
- 4) La herramienta QRM y el módulo de balanceo de carga basado en recursos para la **asignación eficiente de tareas** a recursos se integra de forma completa.

Tabla 4.12. Análisis de integración de asignación eficiente de tareas a recursos con diferentes escenarios

Escenario	Balanceo de carga	F1-Score	F1-Score	# Nodos locales	Tiempo real [†]
		Caídas	Medio		
1	-	96,77	88,18	8	1,38
2	Round-Robin	96,77	88,13	3	1,04
3	Round-Robin	96,77	85,45	2	0,88
4	Resource-based	96,77	85,45	2	1,20

[†]Ratio de rendimiento en tiempo real (inversa de la carga computacional) del peor caso posible del nodo que soporta una mayor carga computacional. Valor mayor a 1 cumple con rendimiento en tiempo real.

La Tabla 4.12 resume los resultados de los escenarios definidos previamente. En el *Escenario#1*, no se despliegan modelos destilados. Por lo que monitorizan las habitaciones usando el modelo *S3DG_64_168* (ver capítulo 4.1.4.1), que ofrece una capacidad de reconocimiento similar al modelo destilado asignado a las personas con riesgo de caída más elevado (ver Tabla 4.11). Sin embargo, ejecutar este modelo sin destilar supone un 72% de carga computacional en los nodos. Esto hace que sean necesarios al menos 8 nodos para analizar todos los residentes de forma concurrente. Sin embargo, se garantiza un funcionamiento en tiempo real porque cada nodo solo monitoriza a un único residente.

En el *Escenario#2* se implementan modelos destilados. Por esto, a pesar de ofrecer una precisión similar al caso del anterior escenario, el análisis de cada habitación se reduce a una carga computacional de un 32% en el nodo (modelo destilado asignado a personas de elevado riesgo). Esto hace que tan solo 3 nodos puedan ser suficientes para analizar las 8 habitaciones de forma concurrente. Las tareas se asignan mediante un simple método de *Round-Robin*. Esto resulta en que al menos dos de los tres nodos estén soportando una carga computacional de un 96%.

En el *Escenario#3*, se asigna un modelo diferente a cada residente en función de su riesgo de caída. Esto supone una reducción de los requisitos computacionales, haciendo que ahora tan solo dos nodos puedan llevar a cabo el análisis. Además, si se detecta una caída, esta puede ser reanalizada con una arquitectura más compleja y precisa (ver política de reconfiguración descrita en el capítulo 4.2.1). De esta manera, se consigue mejorar la capacidad de identificación de situaciones críticas. A su vez, la carga de trabajo de los nodos locales puede reducirse aún más gracias al modo de operación de detección de movimiento. Por ejemplo, si un nodo monitoriza dos residentes, uno de bajo y otro de elevado riesgo, y el residente de bajo riesgo sale de su habitación, el consumo energético del dispositivo se reduce hasta en un 8,5% durante el tiempo en el que el paciente se

encuentre fuera de su habitación. Reduciendo también la carga de trabajo del nodo. En este escenario, se aplica un método basado en *Round-Robin* para el balanceo de carga. Este esquema de balanceo de carga no tiene en consideración los requisitos de cómputo de cada tarea. Por lo que puede resultar en un sistema con nodos sobrecargados que no ofrecen un rendimiento en tiempo real. En concreto, puede darse el caso en el que se asignen tres residentes de alto riesgo y un residente de riesgo medio al mismo nodo de procesamiento, lo que resultaría en un nodo con mayor carga computacional de la que puede soportar para operar en tiempo real.

Finalmente, en el *Escenario#4* se integra el módulo de balanceo de carga basado en recursos (**asignación eficiente de tareas a recursos**). Este método valora la complejidad computacional de cada tarea y los recursos disponibles en cada nodo. De esta forma, las tareas se asignan para distribuir de forma equitativa la carga computacional entre los nodos sin sobrecargar ninguno de ellos. Con ello, es posible que el sistema opere en tiempo real con tan solo 2 nodos de procesamiento locales.

Este sistema distribuido es altamente escalable. Si es necesario monitorizar nuevas habitaciones, estas pueden ser fácilmente integradas en el sistema sin la necesidad de adquirir nuevos nodos de procesamiento, hasta que los nodos alcancen su capacidad máxima de cómputo.

5 Conclusiones

En esta tesis se ha presentado el diseño y desarrollo de un **sistema ciberfísico distribuido y reconfigurable** para la monitorización de actividades por medio de la implementación de arquitecturas de aprendizaje automático. Esta solución tiene multitud de aplicaciones, ayudando a la monitorización de personas con alguna enfermedad o patología, asistiendo a profesionales del campo de la salud para el fomento de hábitos de vida saludable, y en general, para apoyar a las personas con necesidades de atención a vivir de forma autónoma y segura, ofreciendo una rápida respuesta ante situaciones de emergencia cuando sufren un accidente.

Este estudio permite dar respuesta a la hipótesis planteada, validando el uso de sistemas ciberfísicos reconfigurables basados en modelos de aprendizaje automático para ofrecer una solución precisa y eficiente para la monitorización de actividades cotidianas en tiempo real. A continuación, se describen en detalle las principales contribuciones de esta tesis y cómo se relacionan con los objetivos específicos definidos.

La implementación de este CPS reconfigurable se ha realizado en dos etapas. En primer lugar, se diseña el **módulo de reconocimiento de acciones**, introduciendo nuevas arquitecturas de aprendizaje automático con gran capacidad de reconocimiento para la clasificación de actividades de forma eficiente y en tiempo real. En segundo lugar, se diseña la arquitectura del sistema **ciberfísico reconfigurable** para la monitorización de hábitos, de forma que se adapta automáticamente, por ejemplo, en función de la precisión en el reconocimiento de actividades, permitiendo así su mejora o, de los recursos disponibles

El primer paso para el diseño del módulo de reconocimiento de acciones fue la recopilación y **creación** de nuevos conjuntos de datos a medida, más específicos para la clasificación de actividades cotidianas. Para su recolección, se tuvo en cuenta que incluyeran una amplia diversidad de acciones de la vida diaria que son comunes en entornos domésticos, y se garantizara la representación de situaciones relevantes para brindar asistencia efectiva en el hogar. Con esta contribución se busca impulsar la investigación de sistemas para la monitorización de actividades de la vida diaria basado en HAR. Entre los conjuntos de datos introducidos en esta tesis se destaca el *dataset IAPV (Indoor Action Pulses and Video)*. Este conjunto de datos, con más de mil muestras, contiene grabaciones de personas en sus hogares realizando actividades cotidianas como caminar, comer, limpiar o ver la televisión, así como situaciones de emergencia como sufrir una caída o estar tumbado en el suelo. En este conjunto de datos se recopila información de **vídeo** y de **señales de signos vitales** de manera sincronizada. Tal y como se ha estudiado en el capítulo 4.1.3.2, el análisis de esta información multimodal de vídeo y pulso cardíaco permite mejorar la capacidad de reconocimiento del sistema hasta en un **2,4%**, mejorando la identificación de acciones críticas. Esta contribución se alinea con el objetivo de **recopilar y crear un conjunto de datos multimodal que represente actividades de la vida diaria relevantes para el marco de aplicación de asistencia en el hogar**.

Por otro lado, se introducen varios modelos de aprendizaje automático para el análisis de actividades cotidianas. Para ello, se diseñan diversas arquitecturas basadas en aprendizaje profundo utilizando operaciones convolucionales 3D o redes convolucionales recurrentes para extraer características espaciotemporales que permitan clasificar acciones a partir de un clip de vídeo. Tras el estudio pormenorizado de las arquitecturas del estado del arte y su comparación, se selecciona el modelo **RGBI3D**, una red basada en convoluciones 3D que ofrece una precisión en términos de F1-Score superior al 78% en la identificación de actividades de la vida diaria sobre el conjunto de datos *CIAV* (ver capítulo 4.1.1). El desarrollo del módulo de reconocimiento de acciones sigue un enfoque incremental: partiendo de la base de la arquitectura seleccionada, se introducen de manera iterativa nuevas modificaciones y ajustes a nivel de modelo con el objetivo de ofrecer un mejor equilibrio entre la bondad de ajuste y el uso de recurso de las alternativas propuestas. Para mejorar la precisión de la solución, especialmente para la clasificación de situaciones críticas como las caídas, se introduce la arquitectura **TwoStream I3D** (capítulo 4.1.3.1), que por medio del análisis de **vídeo** y de estimación de **movimiento** es capaz de mejorar en más de **5 puntos** el F1-Score de la solución a costa de un aumento significativo del uso de recursos. Sin embargo, se minimiza el impacto en el uso de recursos y consumo energético en el nodo local transfiriendo el procesamiento de la estimación de movimiento al servidor central, usando la herramienta de virtualización de recursos *PoCL-R*. Este desarrollo responde al objetivo específico de **diseñar e implementar modelos de aprendizaje automático para el reconocimiento de actividades de la vida diaria**.

El análisis de las actividades se realiza en nodos locales empotrados de recursos limitados como las Nvidia Jetson. El procesamiento en la periferia de la nube aporta diversos beneficios al sistema entre los que se destaca: una menor latencia de respuesta, la preservación de la privacidad de los usuarios o la reducción de uso de ancho de banda al procesar la información cerca de la fuente de los datos, evitando el envío de información sensible fuera de la red local. Sin embargo, la ejecución en **tiempo real** de los modelos de aprendizaje profundo complejos en estos dispositivos **SoC** energéticamente eficientes supone uno de los retos de este trabajo. Por esto, es crucial **optimizar** las arquitecturas de *Aprendizaje Profundo* para integrarlas en los nodos de procesamiento. Se demuestra cómo es posible mejorar la precisión de arquitecturas eficientes **destilando conocimiento** a partir de redes neuronales más complejas, garantizando su funcionamiento en tiempo real, llegando a ofrecer una precisión comparable a modelos que consumen hasta 10 veces más recursos. En concreto, destilar conocimiento de un conjunto de redes neuronales multimodales de vídeo y estimación de movimiento **mejora** el F1-Score de uno de los modelos computacionalmente más eficientes (*Distilled RGBI3D_16_112*) en un **8,3%** (alcanzando un 82,65% de F1-Score sobre la modalidad de vídeo del *IAPV dataset*), con un rendimiento de más de **250 fps**. Esta optimización permite el análisis en **tiempo real** de varios residentes en un mismo nodo de procesamiento de forma concurrente. Esta contribución responde al objetivo específico de **optimizar y desplegar modelos de aprendizaje automático en dispositivos empotrados con rendimiento en tiempo real**.

A continuación, se presenta la implementación del CPS distribuido y reconfigurable con arquitectura nodo-nube. El sistema presentado está construido a partir de la integración de varios componentes: un servidor central de altas prestaciones para las tareas de alta carga computacional; nodos locales de procesamiento que realizan el reconocimiento de acciones en tiempo real con un bajo consumo de energía sobre dispositivos empotrados energéticamente eficientes; una plataforma de virtualización de recursos hardware, *PoCLR*, que virtualiza recursos, permitiendo que el procesamiento de tareas de alta carga computacional se lleven a cabo en el servidor remoto de altas prestaciones para aliviar el uso de recursos en el nodo; y la herramienta de Gestión de Calidades y Recursos (QRM), FIVIS, dedicada al análisis de los datos generados por el sistema para la toma de decisiones, así como a la visualización de los mismos.

A través del componente QRM, se monitorizan de forma continua los datos operacionales y rendimiento del sistema, así como la confianza de las predicciones o el consumo energético de los nodos de procesamiento. Este módulo puede generar comandos de **reconfiguración en tiempo de ejecución** para adaptar el funcionamiento del sistema completo automáticamente con la finalidad de **optimizar** el consumo energético o la precisión de este.

La integración de la reconfiguración aporta gran **flexibilidad** al sistema, con la adaptación de forma automática de los modelos desplegados en los nodos locales alimentados por **baterías**. Con la política de reconfiguración presentada, el **tiempo de funcionamiento** del sistema puede ampliarse hasta en un 63% con una pérdida poco significativa de la precisión general. Además, se mejora el reconocimiento de **acciones críticas**, lo cual es un aspecto crucial en los sistemas de monitorización del ámbito del *Smart-Health*. Esta mejora permite reducir la generación de **falsas alarmas**, evitando que los usuarios pierdan la confianza en el sistema. Esta contribución se relaciona con el objetivo específico de **diseñar la política de reconfiguración multi-objetivo del sistema completo, con el objetivo de optimizar distintas calidades como precisión o consumo**.

Por último, el sistema distribuido nodos-nube está formado por la integración de los componentes mencionados anteriormente, con el foco en diseñar una solución con gran capacidad de reconocimiento y eficiente en el uso de recursos. Adicionalmente, la **asignación eficiente de tareas a recursos** unida a la destilación de conocimiento puede contribuir a **reducir** hasta en un 75% los recursos de cómputo necesarios para desplegar el sistema para el caso de uso de la residencia de ancianos mostrado en el capítulo 4.2.2.3, y consecuentemente también reduciendo el coste *hardware* (el sistema inicial mostrado en el escenario práctico correspondiente pasa de requerir 8 nodos para su eficaz funcionamiento en la versión inicial, a tan sólo necesitar 2 tras la optimización de recursos). La implementación de este sistema, unido a las contribuciones presentadas anteriormente permiten validar el objetivo específico de **diseñar, implementar y validar la arquitectura de sistema ciberfísico usando nodos locales y un servidor central para procesamiento**

distribuido, que permita su reconfiguración en tiempo real en función del análisis de diferentes calidades como la precisión del sistema o el consumo total del mismo.

Finalmente, señalamos algunas potenciales direcciones para futuros trabajos. En primer lugar, se plantea continuar el trabajo con la construcción de un sistema que ofrezca guías para actividades e informes a profesionales de la salud basándose en la monitorización continua de las actividades en interiores para promover estilos de vida saludables. También se propone la integración de este sistema en hospitales, residencias y hogares para dar asistencia y promover hábitos de vida saludables. Otra de las líneas de trabajo futuro es la detección de actividades con múltiples personas en la escena. Esta situación representa nuevos retos, por ejemplo: modelos computacionalmente más complejos que no ofrecerían rendimiento en tiempo real en nodos de procesamiento eficientes, el problema de la oclusión entre personas, o el de la perspectiva de la cámara que añade ambigüedad a la clasificación de actividades de los sujetos en la escena.

Publicaciones

A continuación, se presentan los artículos científicos publicados o que se encuentran en estado de revisión que han sido elaborados durante el proceso de desarrollo de la tesis doctoral. En concreto, varios de los trabajos están estrechamente relacionados con las contribuciones presentadas en esta tesis.

El artículo publicado en la conferencia *ETFA* (2020) muestra una primera aproximación para el diseño de un sistema ciberfísico reconfigurable simple en cuanto a la toma de decisiones y análisis de calidades, pero que explora diversas arquitecturas del estado del arte para el reconocimiento de acciones a partir de vídeo.

Por otro lado, el artículo publicado en *Journal of Signal Processing Systems* (2023) presenta el conjunto de datos multimodal de vídeo y pulso cardíaco creado a medida (IAPV) y estudia los beneficios del análisis de datos de diversa naturaleza en la mejora de la precisión global del sistema, especialmente para la identificación de situaciones críticas.

El artículo en revisión enviado a *IEEE Transactions on Mobile Computing* (2023) se centra en la optimización de las arquitecturas de aprendizaje profundo mediante la destilación de conocimiento para mejorar la precisión de las soluciones de reconocimiento de actividades cotidianas más eficientes. Esta contribución se valida proponiendo el despliegue de un sistema ciberfísico reconfigurable en una residencia de mayores, estudiando como la destilación de los modelos y la asignación eficiente de tareas a recursos permite el diseño de una solución eficiente y precisa que opera en tiempo real en nodos de procesamiento empotrados de bajos recursos.

Finalmente, el trabajo en revisión en la revista *Engineering Applications of Artificial Intelligence* (2023) presenta el diseño integral de un sistema ciberfísico reconfigurable con arquitectura nodo-nube. Se introduce una política de reconfiguración para la adaptación automática del sistema en tiempo de ejecución a partir del análisis de diversas calidades monitorizadas. Esto unido con la integración de una plataforma de virtualización de recursos permite el diseño de un CPS eficiente en cuanto al uso de recursos y preciso en la identificación de situaciones de emergencia como las caídas.

Artículos de revista

- **Deniz, D.**, Jimenez-Perera, G., Nolasco, R., Corral, J., & Barranco, F. (2023). Deep Multimodal Habit Tracking System: A User-adaptive Approach for Low-power Embedded Systems. *Journal of Signal Processing Systems*, 1-14.

Journal of Signal Processing Systems for Signal Image and Video Technology			
Métricas de revista en 2021			
Factor de impacto	Categoría	Cuartil JIF	JIF Rank
1.813	Engineering, Electrical & Electronic	Q3	196/276

JIF= Journal Impact Factor

- Isern, J., Barranco, F., **Deniz, D.**, Lesonen, J., Hannuksela, J., & Carrillo, R. R. (2020). Reconfigurable cyber-physical system for critical infrastructure protection in smart cities via smart video-surveillance. *Pattern Recognition Letters*, 140, 303-309.

Pattern Recognition Letters			
Métricas de revista en 2020			
Factor de impacto	Categoría	Cuartil JIF	JIF Rank
3.756	Computer Science, Artificial Intelligence	Q2	46/139

JIF= Journal Impact Factor

Artículos de revista en revisión

- **Deniz, D.**, Isern, J., Solanti, J., Jääskeläinen, P., Hnětynka, P., Bulej, L., Ros, E., & Barranco, F. (2023). Efficient reconfigurable CPS for monitoring the elderly at home via Deep Learning. *Engineering Applications of Artificial Intelligence*. Under review.

Engineering Applications of Artificial Intelligence			
Métricas de revista en 2021			
Factor de impacto	Categoría	Cuartil JIF	JIF Rank
7.802	Automation & Control Systems	Q1	8/65
	Computer Science, Artificial Intelligence	Q1	27/145
	Engineering, Electrical & Electronic	Q1	30/276

JIF= Journal Impact Factor

- **Deniz, D.**, Ros, E., Ortigosa, E.M., & Barranco, F. (2023). Optimized edge-cloud system for activity monitoring using knowledge distillation. *IEEE Transactions on Mobile Computing*. Under **major** revision.

IEEE Transactions on Mobile Computing			
Métricas de revista en 2021			
Factor de impacto	Categoría	Cuartil JIF	JIF Rank

6.075	Computer Science, Information Systems	Q1	30/164
	Telecommunications	Q1	17/93

JIF= Journal Impact Factor

- **Deniz, D.**, Fermüller, C., Ros, E., Rodriguez-Alvarez, M., & Barranco, F. (2023). Event-based Vision for Early Prediction of Manipulation Actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Under review.

IEEE Transactions on Pattern Analysis and Machine Intelligence			
Métricas de revista en 2021			
Factor de impacto	Categoría	Cuartil JIF	JIF Rank
24.314	Computer Science, Artificial Intelligence	Q1	2/145
	Engineering, Electrical & Electronic	Q1	2/276

Artículos de conferencia

- **Deniz, D.**, Barranco, F., Isern, J., & Ros, E. (2020). Reconfigurable cyber-physical system for lifestyle video-monitoring via deep learning. *In 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (Vol. 1, pp. 1705-1712). IEEE.
- **Deniz, D.**, Ros, E., Fermüller, C., & Barranco, F. (2023). When Do Neuromorphic Sensors Outperform cameras? Learning from Dynamic Features. *In 2023 57th Annual Conference on Information Sciences and Systems (CISS)* (pp. 1-6). IEEE.

Bibliografía

- [1] Lee EA (2008) Cyber physical systems: Design challenges. In: 2008 11th IEEE Int. Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC). pp 363–369
- [2] Ding D, Han Q-L, Wang Z, & Ge X (2019) A survey on model-based distributed control and filtering for industrial cyber-physical systems. *IEEE Trans Industr Inform* 15:2483–2499
- [3] Al-Ali AR, Gupta R, & Nabulsi A al (2018) Cyber physical systems role in manufacturing technologies. In: *AIP Conference Proceedings*. p 50007
- [4] Li K, Ni W, Emami Y, Shen Y, Severino R, Pereira D, & Tovar E (2019) Design and Implementation of Secret Key Agreement for Platoon-based Vehicular Cyber-physical Systems. *ACM Transactions on Cyber-Physical Systems* 4:1–20
- [5] Pal R, & Prasanna V (2016) The STREAM Mechanism for CPS SecurityThe Case of the Smart Grid. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 36:537–550
- [6] Chaâri R, Ellouze F, Koubâa A, Qureshi B, Pereira N, Youssef H, & Tovar E (2016) Cyber-physical systems clouds: A survey. *Computer Networks* 108:260–278
- [7] Zhang Y, Qiu M, Tsai C-W, Hassan MM, & Alamri A (2015) Health-CPS: Healthcare cyber-physical system assisted by cloud and big data. *IEEE Syst J* 11:88–95
- [8] Al-Ars Z, Basten T, de Beer A, Geilen M, Goswami D, Jääskeläinen P, Kadlec J, de Alejandro MM, Palumbo F, Peeren G, & others (2019) The FitOptiVis ECSEL project: highly efficient distributed embedded image/video processing in cyber-physical systems. In: *16th ACM Int. Conference on Computing Frontiers*. pp 333–338
- [9] Calderita L v, Vega A, Barroso-Ram\`virez S, Bustos P, & Núñez P (2020) Designing a cyber-physical system for ambient assisted living: a use-case analysis for social robot navigation in caregiving centers. *Sensors* 20:4005
- [10] Abdellatif AA, Mohamed A, Chiasserini CF, Tlili M, & Erbad A (2019) Edge computing for smart health: Context-aware approaches, opportunities, and challenges. *IEEE Netw* 33:196–203
- [11] Rashidi P, & Mihailidis A (2012) A survey on ambient-assisted living tools for older adults. *IEEE J Biomed Health Inform* 17:579–590
- [12] Dhiman C, & Vishwakarma DK (2019) A review of state-of-the-art techniques for abnormal human activity recognition. *Eng Appl Artif Intell* 77:21–45

-
- [13] Liu B, Zhang Y, Zhang G, & Zheng P (2019) Edge-cloud orchestration driven industrial smart product-service systems solution design based on CPS and IIoT. *Advanced Engineering Informatics* 42:100984
 - [14] Lara OD, & Labrador MA (2012) A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials* 15:1192–1209
 - [15] Huang J, Lin S, Wang N, Dai G, Xie Y, & Zhou J (2019) TSE-CNN: A two-stage end-To-end CNN for human activity recognition. *IEEE J Biomed Health Inform*
 - [16] Yao G, Lei T, & Zhong J (2019) A review of Convolutional-Neural-Network-based action recognition. *Pattern Recognit Lett* 118:14–22
 - [17] Li G, Ma X, Wang X, Yue H, Li J, Liu L, Feng X, & Xue J (2022) Optimizing deep neural networks on intelligent edge accelerators via flexible-rate filter pruning. *Journal of Systems Architecture* 124:102431
 - [18] Nations U, Social Affairs D of E, & Dynamics P (2019) 2019 Revision of World Population Prospects. In: *World Population Prospects 2019*. p 46
 - [19] Davis K, Hu J, Feijs L, & Owusu E (2015) Social hue: A subtle awareness system for connecting the elderly and their caregivers. In: *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. pp 178–183
 - [20] Doupe M, Brunkert T, Wagg A, Ginsburg L, Norton P, Berta W, Knopp-Sihota J, & Estabrooks C (2022) SCOPE: safer care for older persons (in residential) environments—a pilot study to enhance care aide-led quality improvement in nursing homes. *Pilot Feasibility Stud* 8:1–13
 - [21] Organisation for Economic Co-operation, & Development (2020) Long-term care resources and utilisation: long-term care recipients 2020
 - [22] Murillo JM (2019) Complex Event Processing for Health Monitoring. In: *Gerontechnology: First International Workshop, IWOG 2018, Cáceres, Spain, and Évora, Portugal, 14 and 17 December, 2018, Revised Selected Papers*. p 3
 - [23] Vinciguerra S, & Vinciguerra M (2019) Smart devices and healthy aging. *Nutr Healthy Aging* 5:13–19
 - [24] Buyl R, Beogo I, Fobelets M, Deletroz C, van Landuyt P, Dequanter S, Gorus E, Bourbonnais A, Giguère A, Lechasseur K, & others (2020) e-Health interventions for healthy aging: a systematic review. *Syst Rev* 9:1–15
 - [25] Bergen G, Stevens MR, & Burns ER (2016) Falls and fall injuries among adults aged 65 years—United States, 2014. *Morbidity and Mortality Weekly Report* 65:993–998

- [26] Yacchirema D, de Puga JS, Palau C, & Esteve M (2019) Fall detection system for elderly people using IoT and ensemble machine learning algorithm. *Pers Ubiquitous Comput* 23:801–817
- [27] Ganesan B, Gowda T, Al-Jumaily A, Fong KNK, Meena SK, & Tong RKY (2019) Ambient assisted living technologies for older adults with cognitive and physical impairments: a review. *Eur Rev Med Pharmacol Sci*
- [28] Syed AS, Sierra-Sosa D, Kumar A, & Elmaghraby A (2022) A deep convolutional neural network-XGB for direction and severity aware fall detection and activity recognition. *Sensors* 22:2547
- [29] Li J, Han X, Zhang X, & Wang S (2019) Spatiotemporal evolution of global population ageing from 1960 to 2017. *BMC Public Health* 19:1–15
- [30] Taramasco C, Rodenas T, Martinez F, Fuentes P, Munoz R, Olivares R, de Albuquerque VHC, & Demongeot J (2018) A novel monitoring system for fall detection in older people. *Ieee Access* 6:43563–43574
- [31] Abdellatif AA, Emam A, Chiasserini C-F, Mohamed A, Jaoua A, & Ward R (2019) Edge-based compression and classification for smart healthcare systems: Concept, implementation and evaluation. *Expert Syst Appl* 117:1–14
- [32] Tian S, Yang W, le Grange JM, Wang P, Huang W, & Ye Z (2019) Smart healthcare: making medical care more intelligent. *Global Health Journal* 3:62–65
- [33] Shah T, Yavari A, Mitra K, Saguna S, Jayaraman PP, Rabhi F, & Ranjan R (2016) Remote health care cyber-physical system: quality of service (QoS) challenges and opportunities. *IET Cyber-Physical Systems: Theory & Applications* 1:40–48
- [34] Frias-Martinez E, Magoulas G, Chen S, & Macredie R (2005) Modeling human behavior in user-adaptive systems: Recent advances using soft computing techniques. *Expert Syst Appl* 29:320–329
- [35] Deniz D, Barranco F, Isern J, & Ros E (2020) Reconfigurable cyber-physical system for lifestyle video-monitoring via deep learning. In: 2020 25th IEEE Int. Conference on Emerging Technologies and Factory Automation (ETFA). pp 1705–1712
- [36] Baheti R, & Gill H (2011) Cyber-physical systems. The impact of control technology 12:161–166
- [37] Shin Y-J, Cho E, Kim H, & Bae D-H (2022) Hands-On Field Operational Test Dataset of a Multi-Controller CPS: A Modeled Case Study on Autonomous Driving. In: 2022 17th Annual System of Systems Engineering Conference (SOSE). pp 205–212

- [38] Alsulami AA, & Zein-Sabatto S (2020) Detection and Defense from False Data Injection Attacks In Aviation Cyber-Physical Systems Using Artificial Immune Systems. In: 2020 International Conference on Computational Science and Computational Intelligence (CSCI). pp 69–75
- [39] Isern J, Barranco F, Deniz D, Lesonen J, Hannuksela J, & Carrillo RR (2020) Reconfigurable cyber-physical system for critical infrastructure protection in smart cities via smart video-surveillance. *Pattern Recognit Lett* 140:303–309
- [40] Krishna PV, Gurumoorthy S, Obaidat MS, Monisha K, & Rajasekhara Babu M (2019) A novel framework for healthcare monitoring system through cyber-physical system. *Internet of things and personalized healthcare systems* 21–36
- [41] Sharma SK, & Wang X (2017) Live data analytics with collaborative edge and cloud processing in wireless IoT networks. *IEEE Access* 5:4621–4635
- [42] Maheshwari S, Raychaudhuri D, Seskar I, & Bronzino F (2018) Scalability and performance evaluation of edge cloud systems for latency constrained applications. In: 2018 IEEE/ACM Symposium on Edge Computing (SEC). pp 286–299
- [43] Wang X, Xue H, Liu X, & Pei Q (2019) A privacy-preserving edge computation-based face verification system for user authentication. *IEEE Access* 7:14186–14197
- [44] Feng W, Qin Y, Zhao S, & Feng D (2018) AAoT: Lightweight attestation and authentication of low-resource things in IoT and CPS. *Computer Networks* 134:167–182
- [45] Barranco F, Diaz J, Pino B, & Ros E (2014) Real-time visual saliency architecture for FPGA with top-down attention modulation. *IEEE Trans Industr Inform* 10:1726–1735
- [46] Bharad P, Lee EK, & Pompili D (2014) Towards a reconfigurable cyber physical system. In: IEEE 11th Int. Conference on Mobile Ad Hoc and Sensor Systems. pp 531–532
- [47] Boschi F, Zanetti C, Tavola G, & Taisch M (2016) Functional requirements for reconfigurable and flexible cyber-physical system. In: Conf. of the IEEE Ind. Electronics Society. pp 5717–5722
- [48] Nogueira R, Reis J, Pinto R, & Gonçalves G (2019) Self-Adaptive Cobots in Cyber-Physical Production Systems. In: 24th IEEE Int. Conference on Emerging Technologies and Factory Automation (ETFA). pp 521–528
- [49] Gaur S, Almeida L, Tovar E, & Reddy R (2019) CAP: Context-Aware Programming for Cyber Physical Systems. In: 24th IEEE Int. Conference on Emerging Technologies and Factory Automation (ETFA). pp 1009–1016

- [50] Brusaferrì A, Ballarino A, Cavadini FA, Manzocchi D, & Mazzolini M (2014) CPS-based hierarchical and self-similar automation architecture for the control and verification of reconfigurable manufacturing systems. In: 2014 IEEE Emerging Technology and Factory Automation (ETFA). pp 1–8
- [51] Bonci A, Carbonari A, Cucchiarelli A, Messi L, Pirani M, & Vaccarini M (2019) A cyber-physical system approach for building efficiency monitoring. *Autom Constr* 102:68–85
- [52] Al-Ars Z, Basten T, de Beer A, Geilen M, Goswami D, Jääskeläinen P, Kadlec J, de Alejandro MM, Palumbo F, Peeren G, & others (2019) The FitOptiVis ECSEL project: highly efficient distributed embedded image/video processing in cyber-physical systems. In: 16th ACM Int. Conference on Computing Frontiers. pp 333–338
- [53] Barrère M, Hankin C, Barboni A, Zizzo G, Boem F, Maffei S, & Parisini T (2018) CPS-MT: A Real-Time Cyber-Physical System Monitoring Tool for Security Research. In: 2018 IEEE 24th Int. Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA). pp 240–241
- [54] Hendriks M, Geilen M, Goossens K, de Jong R, & Basten T (2020) Interface Modeling for Quality and Resource Management. *arXiv:200208181*
- [55] Zhou P, Zuo D, Hou KM, Zhang Z, Dong J, Li J, & Zhou H (2019) A Comprehensive Technological Survey on the Dependable Self-Management CPS: From Self-Adaptive Architecture to Self-Management Strategies. *Sensors* 19:. <https://doi.org/10.3390/s19051033>
- [56] Grafana (2021) Grafana
- [57] Kibana (2021) Kibana
- [58] Chronograf (2021) Chronograf
- [59] Peddoju SK, & Upadhyay H (2020) Evaluation of IoT Data Visualization Tools and Techniques. In: *Data visualization: Trends and challenges toward multidisciplinary perception*. Springer, pp 115–139
- [60] Deniz D, Isern J, Solanti J, Jääskeläinen P, Hnětynka P, Bulej L, Ros E, & Barranco F (2023) Efficient reconfigurable CPS for monitoring the elderly at home via Deep Learning. *Eng Appl Artif Intell* In press:
- [61] Gormley C, & Tong Z (2015) *Elasticsearch: the definitive guide*. O'Reilly
- [62] Bulej L, Bureš T, Filandr A, Hnětynka P, Hnětynková I, Pacovský J, Sandor G, & Gerostathopoulos I (2021) Managing latency in edge–cloud environment. *Journal of Systems and Software* 172:. <https://doi.org/10.1016/j.jss.2020.110872>

- [63] Solanti J, Babej M, Ikkala J, & Jääskeläinen P (2021) POCL-R: A Scalable Low Latency Distributed OpenCL Runtime. In: International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation
- [64] Jääskeläinen P, de La Lama CS, Schnetter E, Raiskila K, Takala J, & Berg H (2015) pocl: A Performance-Portable OpenCL Implementation. *Int J Parallel Program* 43:752–785. <https://doi.org/10.1007/s10766-014-0320-y>
- [65] Azimi I, Rahmani AM, Liljeberg P, & Tenhunen H (2017) Internet of things for remote elderly monitoring: a study from user-centered perspective. *J Ambient Intell Humaniz Comput* 8:273–289
- [66] Pomante L, Palumbo F, Rinaldi C, Valente G, Sau C, Fanni T, van der Linden F, Basten T, Geilen M, Peeren G, & others (2020) Design and management of image processing pipelines within CPS: 2 years of experience from the FitOptiVis ECSEL Project. In: 2020 23rd Euromicro Conference on Digital System Design (DSD). pp 378–385
- [67] Zdravevski E, Lameski P, Trajkovik V, Kulakov A, Chorbev I, Goleva R, Pombo N, & Garcia N (2017) Improving activity recognition accuracy in ambient-assisted living systems by automated feature engineering. *Ieee Access* 5:5262–5280
- [68] Torti E, Fontanella A, Musci M, Blago N, Pau D, Leporati F, & Piastra M (2019) Embedding recurrent neural networks in wearable systems for real-time fall detection. *Microprocess Microsyst* 71:102895
- [69] Rivero-Espinosa J, Iglesias-Pérez A, Gutiérrez-Duenas JA, & Rafael-Palou X (2013) SAAPHO: an AAL architecture to provide accessible and usable active aging services for the elderly. *ACM SIGACCESS Accessibility and Computing* 17–24
- [70] Karvonen H, Matilainen A, & Niemelä V (2019) Remote activity monitoring using indirect sensing approach in assisted living scenario. In: 2019 13th International Symposium on Medical Information and Communication Technology (ISMICT). pp 1–6
- [71] Er PV, & Tan KK (2018) Non-intrusive fall detection monitoring for the elderly based on fuzzy logic. *Measurement* 124:91–102
- [72] Chen Y, & Xue Y (2015) A deep learning approach to human activity recognition based on single accelerometer. In: IEEE Int. Conference on Systems, Man, and Cybernetics. pp 1488–1492
- [73] Zeng M, Nguyen LT, Yu B, Mengshoel OJ, Zhu J, Wu P, & Zhang J (2014) Convolutional neural networks for human activity recognition using mobile sensors. In: 6th Int. Conference on Mobile Computing, Applications and Services. pp 197–205

- [74] Choutas V, Weinzaepfel P, Revaud J, & Schmid C (2018) Potion: Pose motion representation for action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition. pp 7024–7033
- [75] Carreira J, & Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: IEEE Conference on Computer Vision and Pattern Recognition. pp 6299–6308
- [76] Liu G, Tian G, Li J, Zhu X, & Wang Z (2018) Human action recognition using a distributed rgb-depth camera network. *IEEE Sens J* 18:7570–7576
- [77] Kong Y, & Fu Y (2022) Human action recognition and prediction: A survey. *Int Journal of Computer Vision* 1–36
- [78] Hassan MM, Uddin MZ, Mohamed A, & Almogren A (2018) A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems* 81:307–313
- [79] Zhang H-B, Zhang Y-X, Zhong B, Lei Q, Yang L, Du J-X, & Chen D-S (2019) A comprehensive survey of vision-based human action recognition methods. *Sensors* 19:1005
- [80] Nascimento B, Oliveira T, & Tam C (2018) Wearable technology: What explains continuance intention in smartwatches? *Journal of Retailing and Consumer Services* 43:157–169
- [81] Wang J, Chen Y, Hao S, Peng X, & Hu L (2019) Deep learning for sensor-based activity recognition: A survey. *Pattern Recognit Lett* 119:3–11
- [82] Edel M, & Köppe E (2016) Binarized-blstm-rnn based human activity recognition. In: 2016 International conference on indoor positioning and indoor navigation (IPIN). pp 1–7
- [83] Gers FA, Schmidhuber J, & Cummins F (1999) Learning to forget: Continual prediction with LSTM
- [84] Yang J, Nguyen MN, San PP, Li X, & Krishnaswamy S (2015) Deep convolutional neural networks on multichannel time series for human activity recognition. In: *Ijcai*. pp 3995–4001
- [85] Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, & Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: IEEE conference on computer vision and pattern recognition. pp 2625–2634

-
- [86] Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, & Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:14061078
 - [87] Tran D, Bourdev L, Fergus R, Torresani L, & Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: IEEE Int. Conference on Computer Vision. pp 4489–4497
 - [88] Hara K, Kataoka H, & Satoh Y (2017) Learning spatio-temporal features with 3d residual networks for action recognition. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp 3154–3160
 - [89] He K, Zhang X, Ren S, & Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770–778
 - [90] Diba A, Fayyaz M, Sharma V, Karami AH, Arzani MM, Yousefzadeh R, & Van Gool L (2017) Temporal 3d convnets: New architecture and transfer learning for video classification. arXiv preprint arXiv:171108200
 - [91] Huang G, Liu Z, van der Maaten L, & Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4700–4708
 - [92] Tran D, Wang H, Torresani L, Ray J, LeCun Y, & Paluri M (2018) A closer look at spatiotemporal convolutions for action recognition. In: Conf. Computer Vision and Pattern Recognition. pp 6450–6459
 - [93] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, & Rabinovich A (2015) Going deeper with convolutions. In: IEEE conference on computer vision and pattern recognition. pp 1–9
 - [94] Kay W, Carreira J, Simonyan K, Zhang B, Hillier C, Vijayanarasimhan S, Viola F, Green T, Back T, Natsev P, & others (2017) The kinetics human action video dataset. arXiv:170506950
 - [95] Xie S, Sun C, Huang J, Tu Z, & Murphy K (2018) Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European conference on computer vision (ECCV). pp 305–321
 - [96] Zach C, Pock T, & Bischof H (2007) A duality based approach for realtime TV-L 1 optical flow. In: Joint pattern recognition symposium. pp 214–223
 - [97] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, & Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30:

-
- [98] Soomro K, Zamir AR, & Shah M (2012) UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv:12120402
 - [99] Kuehne H, Jhuang H, Garrote E, Poggio T, & Serre T (2011) HMDB: a large video database for human motion recognition. In: 2011 Int. Conference on Computer Vision. pp 2556–2563
 - [100] Wang L, Huang B, Zhao Z, Tong Z, He Y, Wang Y, Wang Y, & Qiao Y (2023) VideoMAE V2: Scaling Video Masked Autoencoders with Dual Masking. arXiv preprint arXiv:230316727
 - [101] Liu Z, Ning J, Cao Y, Wei Y, Zhang Z, Lin S, & Hu H (2022) Video swin transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 3202–3211
 - [102] Bertasius G, Wang H, & Torresani L (2021) Is space-time attention all you need for video understanding? In: ICML. p 4
 - [103] Sigurdsson GA, Varol G, Wang X, Farhadi A, Laptev I, & Gupta A (2016) Hollywood in homes: Crowdsourcing data collection for activity understanding. In: European Conference on Computer Vision. pp 510–526
 - [104] Challa H, Niu N, & Johnson R (2020) Faulty requirements made valuable: On the role of data quality in deep learning. In: 2020 IEEE Seventh Int. Workshop on Artificial Intelligence for Requirements Engineering (AIRE). pp 61–69
 - [105] Charfi I, Miteran J, Dubois J, Atri M, & Tourki R (2013) Optimized spatio-temporal descriptors for real-time fall detection: comparison of support vector machine and Adaboost-based classification. *J Electron Imaging* 22:41106
 - [106] Yoshikawa Y, Lin J, & Takeuchi A (2018) Stair actions: A video dataset of everyday home actions. arXiv:180404326
 - [107] Sun Z, Liu J, Ke Q, & Rahmani H (2020) Human Action Recognition from Various Data Modalities: A Review. arXiv preprint arXiv:201211866
 - [108] Wei H, Jafari R, & Kehtarnavaz N (2019) Fusion of Video and Inertial Sensing for Deep Learning–Based Human Action Recognition. *Sensors* 19:3680
 - [109] Owens A, & Efros AA (2018) Audio-visual scene analysis with self-supervised multisensory features. In: Proceedings of the European Conference on Computer Vision (ECCV). pp 631–648
 - [110] Raj C, Jain C, & Arif W (2017) HEMAN: Health monitoring and nous: An IoT based e-health care system for remote telemedicine. In: 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). pp 2115–2119

- [111] Angelucci A, Kuller D, & Aliverti A (2020) A home telemedicine system for continuous respiratory monitoring. *IEEE J Biomed Health Inform*
- [112] RGB Medical (2021) Telemedicine modules
- [113] Adochiei F, Rotariu C, Ciobotariu R, & Costin H (2011) A wireless low-power pulse oximetry system for patient telemonitoring. In: 2011 7th International Symposium on Advanced Topics in Electrical Engineering (ATEE). pp 1–4
- [114] Joseph G, Joseph A, Titus G, Thomas RM, & Jose D (2014) Photoplethysmogram (PPG) signal analysis and wavelet de-noising. In: 2014 annual international conference on emerging research areas: Magnetics, machines and drives (AICERA/iCMMD). pp 1–5
- [115] Jung S-J, Lee Y-D, Seo Y-S, & Chung W-Y (2008) Design of a low-power consumption wearable reflectance pulse oximetry for ubiquitous healthcare system. In: 2008 International Conference on Control, Automation and Systems. pp 526–529
- [116] Reiss A, Indlekofer I, Schmidt P, & van Laerhoven K (2019) Deep ppg: Large-scale heart rate estimation with convolutional neural networks. *Sensors* 19:3079
- [117] Biswas D, Everson L, Liu M, Panwar M, Verhoef B-E, Patki S, Kim CH, Acharyya A, van Hoof C, Konijnenburg M, & others (2019) CorNET: Deep learning framework for PPG-based heart rate estimation and biometric identification in ambulant environment. *IEEE Trans Biomed Circuits Syst* 13:282–291
- [118] Jindal V, Birjandtalab J, Pouyan MB, & Nourani M (2016) An adaptive deep learning approach for PPG-based identification. In: 2016 38th Annual international conference of the IEEE engineering in medicine and biology society (EMBC). pp 6401–6404
- [119] Lee MS, Lee YK, Pae DS, Lim MT, Kim DW, & Kang TK (2019) Fast emotion recognition based on single pulse PPG signal with convolutional neural network. *Applied Sciences* 9:3355
- [120] Brophy E, Muehlhausen W, Smeaton AF, & Ward TE (2020) CNNs for Heart Rate Estimation and Human Activity Recognition in Wrist Worn Sensing Applications. In: 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). pp 1–6
- [121] Boukhechba M, Cai L, Wu C, & Barnes LE (2019) ActiPPG: using deep neural networks for activity recognition from wrist-worn photoplethysmography (PPG) sensors. *Smart Health* 14:100082

- [122] Ioffe S, & Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Int. conference on machine learning. pp 448–456
- [123] Gou J, Yu B, Maybank SJ, & Tao D (2021) Knowledge distillation: A survey. *Int Journal of Computer Vision* 129:1789–1819
- [124] Tehrani AS, Cao H, Afsardoost S, Eriksson T, Isaksson M, & Fager C (2010) A Comparative analysis of the complexity/accuracy tradeoff in power amplifier behavioral models. *IEEE Trans Microw Theory Tech* 58:1510–1520. <https://doi.org/10.1109/TMTT.2010.2047920>
- [125] Peebles W, & Xie S (2022) Scalable Diffusion Models with Transformers
- [126] Markidis S, Chien SW Der, Laure E, Peng IB, & Vetter JS (2018) NVIDIA tensor core programmability, performance & precision. *Proceedings - 2018 IEEE 32nd International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2018* 522–531. <https://doi.org/10.1109/IPDPSW.2018.00091>
- [127] Courbariaux M, Bengio Y, & David J-P (2015) Binaryconnect: Training deep neural networks with binary weights during propagations. *Adv Neural Inf Process Syst* 28:
- [128] Liu Z, Sun M, Zhou T, Huang G, & Darrell T (2018) Rethinking the Value of Network Pruning. *7th International Conference on Learning Representations, ICLR 2019*
- [129] Hinton G, Vinyals O, Dean J, & others (2015) Distilling the knowledge in a neural network. *arXiv preprint arXiv:150302531* 2:
- [130] Wu J, Leng C, Wang Y, Hu Q, & Cheng J (2016) Quantized Convolutional Neural Networks for Mobile Devices. 4820–4828
- [131] Kwasniewska A, Szankin M, Ozga M, Wolfe J, Das A, Zajac A, Ruminski J, & Rad P (2019) Deep learning optimization for edge devices: Analysis of training quantization parameters. In: *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*. pp 96–101
- [132] Liu Z, Wang Y, Han K, Zhang W, Ma S, & Gao W (2021) Post-training quantization for vision transformer. *Adv Neural Inf Process Syst* 34:28092–28103
- [133] NVIDIA Achieving FP32 Accuracy for INT8 Inference Using Quantization Aware Training with NVIDIA TensorRT | NVIDIA Technical Blog. <https://developer.nvidia.com/blog/achieving-fp32-accuracy-for-int8-inference-using-quantization-aware-training-with-tensorrt/>. Accessed 2 Apr 2023
- [134] Blalock D, Gonzalez Ortiz JJ, Frankle J, & Gutttag J (2020) What is the state of neural network pruning? *Proceedings of machine learning and systems* 2:129–146

-
- [135] Google TensorFlow Lite | AA para dispositivos móviles y perimetrales. <https://www.tensorflow.org/lite?hl=es-419>. Accessed 31 Mar 2023
- [136] TensorFlow Model Optimization Toolkit — Pruning API — The TensorFlow Blog. <https://blog.tensorflow.org/2019/05/tf-model-optimization-toolkit-pruning-API.html>. Accessed 3 Apr 2023
- [137] Luo H, Chen T, Li X, Li S, Zhang C, Zhao G, & Liu X (2022) KeepEdge: A Knowledge Distillation Empowered Edge Intelligence Framework for Visual Assisted Positioning in UAV Delivery. *IEEE Trans Mob Comput.* <https://doi.org/10.1109/TMC.2022.3157957>
- [138] Yamazaki M, & Mori E (2021) Rethinking Deconvolution for 2D Human Pose Estimation Light yet Accurate Model for Real-time Edge Computing. In: 2021 16th IEEE Int. Conference on Automatic Face and Gesture Recognition (FG 2021). pp 1–5
- [139] Wang Y, Li X, Shi M, Xian K, & Cao Z (2021) Knowledge distillation for fast and accurate monocular depth estimation on mobile devices. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* pp 2457–2465
- [140] Beyer L, Zhai X, Royer A, Markeeva L, Anil R, & Kolesnikov A (2021) Knowledge distillation: A good teacher is patient and consistent. *arXiv preprint arXiv:210605237*
- [141] Sandler M, Howard A, Zhu M, Zhmoginov A, & Chen L-C (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* pp 4510–4520
- [142] Intel Corporation OpenVINO. <https://docs.openvino.ai/latest/home.html>. Accessed 31 Mar 2023
- [143] NVIDIA TensorRT SDK | NVIDIA Developer. <https://developer.nvidia.com/tensorrt>. Accessed 31 Mar 2023
- [144] Arabi S, Haghighat A, & Sharma A (2020) A deep-learning-based computer vision solution for construction vehicle detection. *Computer-Aided Civil and Infrastructure Engineering* 35:753–767. <https://doi.org/10.1111/MICE.12530>
- [145] NVIDIA (2020) Jetson Nano Developer Kit
- [146] NVIDIA (2020) TensorRT
- [147] NVidia (2022) NVidia Triton Inference Server
- [148] Quattoni A, & Torralba A (2009) Recognizing indoor scenes. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp 413–420

- [149] Anaby-Tavor A, Carmeli B, Goldbraich E, Kantor A, Kour G, Shlomov S, Tepper N, & Zwerdling N (2020) Do not have enough data? Deep learning to the rescue! In: Proceedings of the AAAI Conference on Artificial Intelligence. pp 7383–7390
- [150] Chaquet JM, Carmona EJ, & Fernández-Caballero A (2013) A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding* 117:633–659
- [151] Deniz D, Jimenez-Perera G, Nolasco R, Corral J, & Barranco F (2023) Deep Multimodal Habit Tracking System: A User-adaptive Approach for Low-power Embedded Systems. *J Signal Process Syst* 1–14
- [152] Carreira J, Noland E, Banki-Horvath A, Hillier C, & Zisserman A (2018) A short note about kinetics-600. [arXiv:1808.01340](https://arxiv.org/abs/1808.01340)
- [153] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, & Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. [arXiv:1704.04861](https://arxiv.org/abs/1704.04861)
- [154] Thrun S, & Pratt L (1998) Learning to learn: Introduction and overview. In: *Learning to learn*. Springer, pp 3–17
- [155] Tajbakhsh N, Shin JY, Gurudu SR, Hurst RT, Kendall CB, Gotway MB, & Liang J (2016) Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Trans Med Imaging* 35:1299–1312
- [156] Deniz D, Ros E, Ortigosa EM, & Barranco F (2023) Optimized edge-cloud System for Activity Monitoring using Knowledge Distillation. *IEEE Trans Mob Comput* In press:
- [157] Anwar A, & Raychowdhury A (2020) Autonomous navigation via deep reinforcement learning for resource constraint edge nodes using transfer learning. *IEEE Access* 8:26549–26560
- [158] Lahsen-Cherif I, Liu H, & Lamy-Bergot C (2022) Real-Time Drone Anti-Collision Avoidance Systems: an Edge Artificial Intelligence Application. In: *2022 IEEE Radar Conf. (RadarConf22)*. pp 1–6
- [159] Azizpour M, da Roza F, & Bajcinca N (2020) End-to-End Autonomous Driving Controller Using Semantic Segmentation and Variational Autoencoder. In: *2020 7th Int. Conf. on Control, Decision and Information Technologies (CoDIT)*. pp 1075–1080
- [160] Park G, Gu B, Heo J, Yi S, Han J, Park J, Min H, Piao X, Cho Y, Park CW, & others (2006) Adaptive load balancing mechanism for server cluster. In: *Int. Conference on Computational Science and Its Applications*. pp 549–557

-
- [161] Khan S, Nazir B, Khan IA, Shamshirband S, & Chronopoulos AT (2017) Load balancing in grid computing: Taxonomy, trends and opportunities. *Journal of Network and Computer Applications* 88:99–111
- [162] Kephart JO, & Chess DM (2003) The vision of autonomic computing. *Computer (Long Beach Calif)* 36:41–50. <https://doi.org/10.1109/MC.2003.1160055>
- [163] Guo Z, Zhang Y, Zhao X, & Song X (2020) CPS-based self-adaptive collaborative control for smart production-logistics systems. *IEEE Trans Cybern*
- [164] Jeong Y-S, Kim H-W, & Jang HJ (2013) Adaptive resource management scheme for monitoring of CPS. *J Supercomput* 66:57–69
- [165] Xia T, Xi L, Pan E, & Ni J (2017) Reconfiguration-oriented opportunistic maintenance policy for reconfigurable manufacturing systems. *Reliab Eng Syst Saf* 166:87–98
- [166] Eramo V, & Lavacca FG (2019) Proposal and investigation of a reconfiguration cost aware policy for resource allocation in multi-provider NFV infrastructures interconnected by elastic optical networks. *Journal of Lightwave Technology* 37:4098–4114
- [167] Sau C, Rinaldi C, Pomante L, Palumbo F, Valente G, Fanni T, Martinez M, van der Linden F, Basten T, Geilen M, & others (2021) Design and management of image processing pipelines within CPS: Acquired experience towards the end of the FitOptiVis ECSEL Project. *Microprocess Microsyst* 87:104350
- [168] Bulej L, Bureš T, Hnětynka P, Čamra V, Siegl P, & Töpfer M (2020) IVIS: Highly customizable framework for visualization and processing of IoT data. In: *Proceedings of EUROMICRO SEAA 2020, Portorož, Slovenia*. IEEE, p 4
- [169] Sigurdsson GA, Russakovsky O, & Gupta A (2017) What actions are needed for understanding human actions in videos? In: *Proceedings of the IEEE international conference on computer vision*. pp 2137–2146
- [170] Hochreiter S, & Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9:1735–1780
- [171] Bureau of Labor Statistics, & U.S. Department of Labor (2019) Average hours per day spent in selected activities by age