



**Universitat de les
Illes Balears**

*Modelado de Datos para la Detección de Caídas en Sistemas
Basados en Visión.*

Trabajo final de máster entregado a la Universidad de las Islas Baleares de acuerdo con los requisitos del **Máster Universitario en Análisis de Datos Masivos en Economía y Empresa (MADM)**

Autor

Elián González Hernández

Tutor

Gabriel Moyá Alcover

18/09/2020

Modelado de Datos para la Detección de Caídas en Sistemas Basados en Visión

Elián González Hernández
Tutor: Gabriel Moyá Alcover

Trabajo de fin de Máster Universitario en Análisis de Datos Masivos en Economía y Empresa (MADM)

Universidad de les Illes Balears
07122 Palma de Mallorca
elian880530@gmail.com

Resumen

Este trabajo propone la elaboración de un sistema capaz de detectar caídas, el cual será utilizado en las residencias de la empresa Fundación Amadip Esmment para monitorizar personas que, por diferentes motivos (discapacidad física, sensorial o intelectual), requieren de un apoyo constante.

La finalidad de este sistema es la de otorgar una cierta autonomía a las personas que cuidan a los residentes, pero también proporcionar esta autonomía al residente. Esto permite que dicho interno se sienta con menor dependencia y, además, con una mayor seguridad ya que si cae, el sistema lanzará una señal de aviso al sanitario responsable del área; todo esto con una monitorización no intrusiva para el paciente.

Para ello, se analizaron diferentes sistemas de detección que existen actualmente, mostrando sus ventajas e inconvenientes, para así identificar qué características necesita el sistema.

En este documento se describe el proceso de obtención de datos y el empleo de algoritmos para procesar las imágenes obtenidas mediante cámaras Intel de profundidad que se utilizarán para monitorizar a los residentes las 24 horas del día.

Una vez que se capturan estas imágenes se aplican algoritmos para realizar tres acciones fundamentales en el sistema: pre-procesamiento y segmentación de imágenes, clasificación de estas en cuatro estados y, por último, el seguimiento de personas y detección de caídas. Se realizaron simulaciones de caída en un entorno controlado hasta conseguir un patrón de detección que se utilice como base para disminuir al máximo el número de falsos positivos.

El sistema se ha entrenado con las caídas más frecuentes como son las frontales, traseras y laterales, lográndose un acierto superior al 90%.

Además de llegar a aumentar el acierto del sistema, como trabajo futuro se pretende refinar el algoritmo de detección de caídas. Este sistema podría contribuir en el desarrollo de nuevas investigaciones sobre la detección de caídas, ya sea por la obtención de nuevos datos o por su publicación.

Palabras clave: Caídas, Patrón, Pre-procesamiento, Segmentación, Imágenes, Distinción, Objetos, Personas, Seguimiento, Detección.

Abstract

This work proposes the development of a fall detection system that will be used in the residences of the Amadip Esmment Foundation to monitor people who, for different reasons (physical, sensorial or intellectual disability), require constant support.

The purpose of this system is to grant a certain autonomy of people who cares for the residents, but also to provide this autonomy to the resident. This allows to the resident to feel less dependent and, moreover, with greater security since if he falls, the system will send a warning signal to the monitor responsible for the area; all this with a non-intrusive monitoring for the patient.

For this, different detection systems that currently exist were analyzed, detecting their advantages and disadvantages, in order to identify what characteristics needs the system.

It describes the data collection process and the use of algorithms to process the images obtained by Intel depth cameras that will be used to monitor residents 24 hours a day.

Once these images are captured, algorithms are applied to perform three fundamental actions in the system: pre-processing and segmentation of images, classifying them into four states and, finally, tracking peo-

ple and detecting falls. Drop simulations were performed in a controlled environment to achieve a detection pattern that is used as a basis to minimize the number of false positives.

The system has been trained with the most frequent falls such as front, rear and side falls, achieving a hit of more than 90%.

In addition to increasing the success of the system, as future work it is intended to refine the fall detection algorithm. This system could contribute in the field of automatic fall detection, both by publishing it and by gathering information.

Keywords: Falls, Pattern, Pre-Processing, Segmentation, Images, Distinction, Objects, People, Follow-up, Detection.

1. Introducción

Las caídas son uno de los riesgos más altos a los que se enfrentan las personas que actualmente viven en residencias u hogares de atención, pero cuando padecen alguna discapacidad física o intelectual y además se tiene una avanzada edad, el riesgo aumenta exponencialmente, como se comenta en el artículo "Caídas en personas con discapacidad intelectual" de la Revista de Enfermería ROL [1]. Además, según este mismo artículo, las caídas son consideradas como la segunda causa de fallecimiento en el mundo por heridas no intencionadas. También añade que las personas con discapacidad intelectual suelen enfermar más, y presentan condiciones que, junto con factores ambientales, profesionales y organizacionales, pueden hacerlas más propensas a las caídas. El mismo artículo también indica que el 84% de las personas con discapacidad intelectual suelen caerse y sufrir lesiones por ello.

Según la Organización Mundial de la Salud (OMS), una caída se puede definir como "*un acontecimiento involuntario que hace perder el equilibrio y dar con el cuerpo en la tierra u otra superficie firme que lo detenga*" [2].

Para la salud pública, las caídas son consideradas como un gran problema a nivel mundial. Se calcula que anualmente se producen 646.00 caídas mortales, según los datos de la OMS. Esto lo convierte en una de las primeras causas a nivel mundial de fallecimientos por heridas no intencionales, sólo superadas por las lesiones producidas por el tráfico. Además, cifras superiores al 80% de las defunciones se relacionan con las caídas que se producen en países en vías de desarrollo o subdesarrollados [2].

De la misma forma, la mayor tasa de mortalidad por edad causadas por caídas se encuentra en mayores de 60 años para todas las regiones del mundo. Esto hace que sea una patología típica en la población anciana que, tal y como se ha comentado, tiene consecuencias muy graves pero es muy común que sea pasada por alto por los especialistas médicos. Hay diferentes razones

por las cuales no se estudia este tipo de sucesos: el paciente muchas veces no dice que se ha caído y no es frecuente que se pregunte sobre las caídas en el historial clínico debido a que no suelen producirse heridas directas por la caída; se suele atribuir la caída como un evento normal en el envejecimiento. Estas caídas son las principales causas de lesiones, incapacidades y en menor medida de muertes en este sector de la población, por esta razón, es considerado un factor de alto riesgo en los ancianos [2].

Según un estudio publicado en la Revista Española de Geriatria y Gerontología [3], sobre una porción de adultos que viven en España, mayores de 70 años, a los cuales durante un año se les realizó un seguimiento. El 28,4% de los participantes presentaron una o más caídas anuales. El 9,9% de ellos sufrieron varias caídas y de estos, un tercio fueron de carácter accidental. El 9,3% de la muestra que sufrieron caídas, tuvieron fracturas, y además el 3,1% padeció fractura de cadera. El 55,4% tuvo que recibir atención médica (en la sala de urgencias el 29% y tuvieron que ser ingresados un 7,3%). Se identificaron varios factores de riesgo, por ejemplo, la disminución del equilibrio o la fuerza y la avanzada edad cuando son mayores de 79 años.

Estas causas justifican la necesidad de desarrollar soluciones para detectar de manera automática o prevenir las caídas, para asegurar una respuesta eficaz cuando esto suceda. Ahora bien, debido a que los dispositivos no son fiables, la instalación no suele ser sencilla o debido a que los individuos no asimilan estos sistemas; producto del rechazo en cuanto a tecnología o por su coste; hoy en día no se suelen utilizar estos sistemas comerciales.

Debido a estos motivos, este proyecto tiene como motivación estudiar las soluciones que existen hoy en día, diseñar y desarrollar un software que se basa en la observación utilizando cámaras de profundidad Intel, capaces de realizar la captura de imágenes de una forma computacionalmente efectiva y no invasiva, para que detecten caídas en entornos similares a una residencia y que lancen un aviso a los monitores encargados de la vigilancia.

Esta investigación se llevará a cabo mediante el estudio de un conjunto de imágenes que servirán para entrenar un algoritmo que detecta las caídas con una precisión superior al 90% de los casos.

Para abordar estos objetivos, el trabajo se estructura en 8 secciones diferentes, contando como la primera sección esta introducción. En un primer lugar, en la sección 2, se observa el Estado del Arte, donde se presentan los diferentes tipos de detectores que existen actualmente y se realiza un análisis, comparando las diferentes propiedades, para así escoger cuál es el que mejor se ajusta a este estudio. Posteriormente, en la sección 3, se presenta cómo se han obtenido los vídeos y se han desfragmentado en imágenes para estudiar dichas caídas, y qué tratamiento se les ha aplicado, tales

cómo: en qué clases se han clasificado, qué limpieza de imágenes se ha realizado, etc. Una vez se tienen las imágenes de las secuencias listas, en la sección 4 se presenta un análisis de estas imágenes, estudiando la correlación que hay entre las diferentes clases en las que se clasifican.

En la sección 5, se pasa a presentar los modelos de predicción que se van a utilizar para realizar la clasificación de las imágenes y los diferentes parámetros de evaluación que se utilizarán en el trabajo para decidir qué modelo es el mejor en nuestro estudio. En la sección 6 primeramente se comentan los parámetros obtenidos y que se utilizarán en los modelos, el estudio de los resultados de la clasificación mediante las Matrices de confusión y los índices de evaluación presentados antes y el resultado de aplicar el modelo de Análisis de Componentes Principales, y como este no resulta satisfactorio en este trabajo. Una vez se tiene el mejor modelo seleccionado, en la sección 7 se aplica dicho modelo a varias secuencias de vídeo de caídas, y se estudia el seguimiento de la clasificación, mostrando si dicho estudio es útil para clasificar secuencias de caídas. Finalmente, en la sección 8 se presentan las conclusiones de este trabajo, explicando un breve resumen, mejoras que se podrían realizar y el posible trabajo a futuro de este estudio.

2. Estado del arte

Los sistemas que existen hoy en día para detectar las caídas abordan este problema de diferentes formas, tómese como ejemplo los dispositivos portátiles que se encargan de supervisar los movimientos del individuo, detectando variaciones de movimiento y posición o los que se basan en supervisar el entorno, situando sensores capaces de monitorizar todo lo que ocurre alrededor del usuario.

Sin embargo, estos sistemas de detección de caídas no detectan todos los casos y no permiten descartar falsos positivos. Por consiguiente, en los siguientes apartados se realizará una investigación respecto a las características de los diferentes sistemas que existen hoy en día, considerando sus efectividades y desventajas.

Por último, este artículo se enfocará en los sistemas que utilizan cámaras de profundidad con visión infrarroja, como se describe en la propuesta de solución, investigando múltiples algoritmos que se proponen para la detección de las caídas.

2.1. Clases de detectores de caídas

En la Revista Internacional de Matemáticas Puras y Aplicadas [4] se propone una clasificación detallada para los tipos de detectores de caídas que existen en la actualidad dividiéndolos en 5 grupos según los distintos enfoques utilizados para la creación de los sistemas y de ellos se derivan las técnicas utilizadas para su implementación según se muestra en la figura 1.

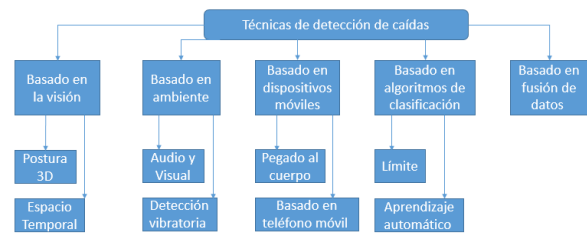


Figura 1: Clasificación para los sistemas detectores de caídas [4].

2.2. Análisis de los diferentes sistemas de detección de caídas

En esta sección se van a presentar una serie de diferentes sistemas de detección de caídas tomados en cuenta en esta investigación, haciendo énfasis en las principales características de estos y presentando sus ventajas o inconvenientes en el desarrollo de la investigación. Esto permitirá poder seleccionar cuál de estos sistemas de detección se ajusta mejor en este estudio.

2.2.1. Angel4

Esta clase de sensor consiste en un aparato de pequeñas dimensiones que va colocado al cinturón del paciente y que, mediante un acelerómetro y un algoritmo, detecta la caída y envía una alarma en un proceso totalmente automático. Contiene una aplicación móvil que envía la alarma a aquellos contactos que estén marcados en la lista del paciente [5].

➤ Ventajas:

- Tiene dimensiones pequeñas.
- Es controlado mediante una aplicación.
- Permite que se puedan enviar mensajes de difusión a diferentes contactos.
- Porcentaje alto de acierto.

➤ Inconvenientes:

- Su precio de venta es elevado si se compara con otros dispositivos de este tipo.
- No posee mucha autonomía de energía ya que utiliza pilas.

2.2.2. Speedy

Se trata de un reloj que dispone de dos acelerómetros, que se encargan de calcular la aceleración en las tres direcciones de los ejes, y posteriormente ejecutan un algoritmo de múltiples etapas para identificar el proceso de caída. Para ello, se obtienen dos resultados fundamentales, la velocidad y la normal, durante todo el recorrido de la caída. Si estos superan unos umbrales predefinidos, se considera que ha ocurrido una caída. Si tras 40 segundos de esa caída, no se registra actividad (movimiento), el dispositivo activará una alarma sonora y una alerta al centro de ayuda [6].

- Ventajas:
 - Todo el procedimiento está implementado en un reloj por lo que ofrece mucha comodidad al usuario.
 - El sistema permite descartar las falsas alarmas presionando un botón.
 - El sistema es capaz de mandar una alerta si el sonido de la alarma no ha sido desactivado.
- Inconvenientes:
 - Detecta con precisión únicamente las caídas hacia delante.

2.2.3. A Depth-Based Fall Detection System Using a Kinect Sensor

Este software se basa en la utilización de las cámaras Kinect de Microsoft para capturar los fotogramas. De igual manera realizar el reconocimiento de estancia y figuras utilizando rayos infrarrojos pero con la diferencia de que las cámaras se ubican en el techo para obtener un mejor ángulo de visión y realizar de forma más óptima el análisis de los fotogramas de profundidad [7]. A continuación se describen algunas ventajas e inconvenientes de este sistema de forma resumida:

- Ventajas:
 - Menos susceptibilidad a las variaciones en la intensidad de la luz.
 - La instalación del Kinect en la parte superior proporciona una vista completa de la escena y permite disminuir el número de dispositivos instalados por habitación, lo que reduciría los costes del proyecto.
 - La información 3D es generada por una sola cámara, mientras que un sistema estereoscópico es necesario el dominio RGB para lograr el mismo objetivo.
 - No es posible reconocer los detalles faciales de las personas capturadas por la cámara de profundidad. Esta característica ayuda a mantener la identidad confidencial.
- Inconvenientes:
 - No emite ninguna señal de alarma.
 - Tienen un elevado costo de venta.
 - La posición del Kinect en la parte superior de la habitación no le permite activar herramientas de seguimiento de esqueletos, que solo funcionan en modo vista frontal.

2.2.4. Otros sistemas analizados

En la presente investigación se estudiaron diferentes sistemas de detección de caídas, además de los descritos anteriormente, un ejemplo de ello es el sistema [9]

que se basa en una consola de juego que realiza el seguimiento de las imágenes desde una vista frontal. Utilizando este mismo ángulo de visión podemos mencionar la aplicación [10] que trabaja con cámaras Kinect.

Otro ejemplo interesante es el sistema [11] que utiliza sensores de movimientos y cámaras para realizar la monitorización de los pacientes.

Si se desea profundizar en el estudio de otros sistemas, se puede acceder a la Revista Internacional de Matemáticas Puras y Aplicadas [4] donde se realiza una comparación de diferentes aplicaciones que se utilizan actualmente y se describen los distintos enfoques y tecnologías utilizados en cada uno.

También se puede acceder a los resultados publicados en la encuesta realizada en el artículo Nuevos Avances y Desafíos de los Sistemas de Detección de Caídas [8], donde se describen en mayor número, los sistemas detectores de caídas existentes hasta el momento junto a sus diferentes ventajas y limitaciones en su utilización.

Por tal motivo en este apartado solamente se reflejan las características de los sistemas que más aportan a la investigación, considerando las ventajas e inconvenientes de cada uno.

De todos estos sistemas, el que más se asemeja al caso de estudio es el que utiliza los sensores Kinect ya que los dos poseen las mismas 4 ventajas que se describen en el epígrafe 2.2.3, pero con la mejora que la cámara Intel RealSense D435 es de un tamaño mucho menor y más compacto lo que posibilita una fácil manipulación y montaje en la superficie superior de las habitaciones.

Es importante destacar que la principal desventaja que presenta el sistema basado en sensores Kinect es que no posee un sistema de alerta o notificación. Sin embargo, esta es una característica que podemos ver entre las ventajas que presentan los otros sistemas Angel4 y Speedy.

Por todo lo anteriormente mencionado se propone para el desarrollo de esta investigación la creación de un nuevo sistema que posea las características mencionadas en el sistema Kinect pero adaptado a la cámara de profundidad Intel y a su vez dotarlo con un sistema capaz de alertar sobre posibles caídas.

3. Proceso de obtención y tratamiento de los datos

El proceso de obtención de los datos es uno de los puntos críticos en la fase de inicio de este proyecto, pues, de la información obtenida, dependerá en gran medida la capacidad del modelo para obtener mejores resultados.

Este proceso se realizará en un ambiente controlado donde se simularán diferentes tipos de caídas y trayectorias utilizando la cámara de profundidad Intel modelo D435 para grabar las secuencias de videos que

posteriormente se separarán en diferentes fotogramas formando así las imágenes.

3.1. Obtención de datos

Se escogió el modelo de cámara de profundidad Intel D435 a petición de la Fundación Amadip Esmert debido al bajo coste que supondrá una posible inversión para desplegar el proyecto en sus residencias.

Además, al ser una tecnología más reciente y en proceso de actualización constante desde su salida al mercado, posee una documentación bastante extensa y con múltiples ejemplos de código abierto en el mismo lenguaje que se utilizará para el desarrollo de esta investigación; donde se demuestra cómo utilizar la cámara en diferentes situaciones. En este caso resultó de gran utilidad debido a que explica detalladamente como grabar las diferentes secuencias de video y posteriormente separar los fotogramas para construir la base de datos.

Otra ventaja de esta cámara es la interfaz que tiene para realizar el proceso de grabación debido que es sencilla, intuitiva y fácil de utilizar. Además posibilita el trabajo con muchos más datos que los sensores Kinect, por ejemplo permite realizar 3 tipos de capturas de video dependiendo del tamaño de las imágenes y la rapidez con que se graban los fotogramas cumpliendo con la condición de que a mayor tamaño de imagen menos cantidad de fotogramas se obtienen en las secuencias de videos.

Por tal motivo, se pretende utilizar los videos que contienen imágenes con el menor tamaño permitido (480 x 270) ya que es capaz de abarcar toda el área de la habitación de las residencias. Además, con estas dimensiones la cámara es capaz de tomar 60 fotogramas por segundo, que es lo máximo permitido, y que resulta de gran utilidad cuando se graban las secuencias de caídas, debido a que, si se intentan grabar imágenes de mayor tamaño pero con menor cantidad de fotogramas por segundo, obtenemos mucha pérdida de información en la cadena de imágenes que se obtendrán del video.



Figura 2: Cámara de profundidad Intel modelo D435.

La característica más importante de los sistemas Kinect y que se puede reproducir fácilmente en las cámaras Intel es la capacidad para tomar imágenes de profundidad utilizando una lente secundaria al que se utiliza para las imágenes RGB, lo que posibilita que sea un sistema no invasivo. Las imágenes resultantes se construyen utilizando diferentes colores que representan la distancia a la cual se encuentran los objetos de la

lente de la cámara lo que permitirá posteriormente que los modelos puedan encontrar diferencias claves entre las imágenes.

Otro de los requisitos planteados es que se debe utilizar solamente una cámara por cada habitación para monitorizar permanentemente el área de dormir, donde se producen con mayor frecuencia las caídas. Las cámaras deberán estar ubicadas en la parte superior de la habitación para prevenir posibles interferencias por parte de los residentes que pueden padecer episodios de agresividad.

Por este motivo las simulaciones de caídas se grabarán en un entorno controlado tratando de recrear las mismas condiciones del ambiente real en que estarán desplegadas las cámaras.

La figura 3 muestra un plano real de las habitaciones de las residencias de la fundación con las medidas que se deben emplear para realizar las simulaciones de caídas.

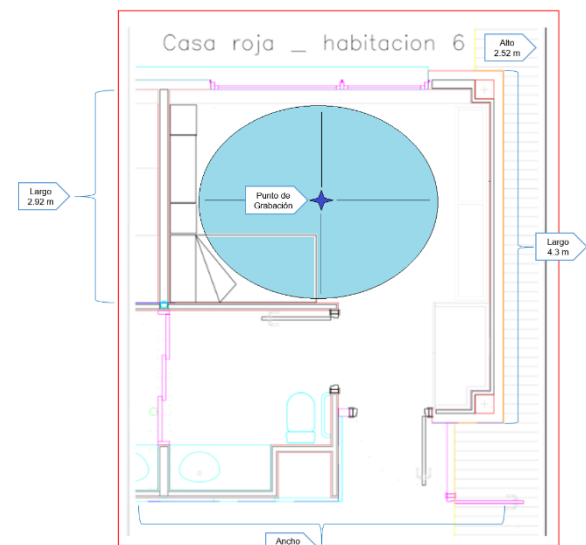


Figura 3: Plano de la habitación 6 de la residencia Casas de Colores, la cámara irá instalada en la parte superior de la imagen, donde se encuentra el área principal de la habitación.

3.2. Tratamiento de los datos

A partir de los diferentes vídeos obtenidos en las simulaciones de caídas se procede a desfragmentar estas secuencias de vídeo en imágenes y de esta manera conformar un único conjunto de datos con el cual se trabajará.

Para ello, se utilizará un software que posibilitará que se puedan obtener las grabaciones de vídeo en visión de profundidad, cumpliendo de esta manera con uno de los requisitos fundamentales de la Fundación Amadip. Además, este software permite que dichas grabaciones de profundidad se pasen a una escala de grises, que posteriormente facilitará al modelo la clasificación ya que se tiene una única escala definida y no diferentes escalas según el color.

Una vez se tienen los vídeos almacenados, se utilizan 60 imágenes por cada segundo de grabación para realizar la detección de una caída, es necesario clasificar las diferentes situaciones que se tienen en las imágenes desfragmentadas del vídeo. Para ello, se definen las siguientes clases posibles:

- Vacío: imagen del vídeo sin ninguna persona.
- Parado: imagen del vídeo con la detección de una persona de pie.
- Agachado: imagen del vídeo con la detección de una persona en proceso de caída o agachada.
- Fin de la Caída: imagen del vídeo con la detección de una persona caída en el suelo.

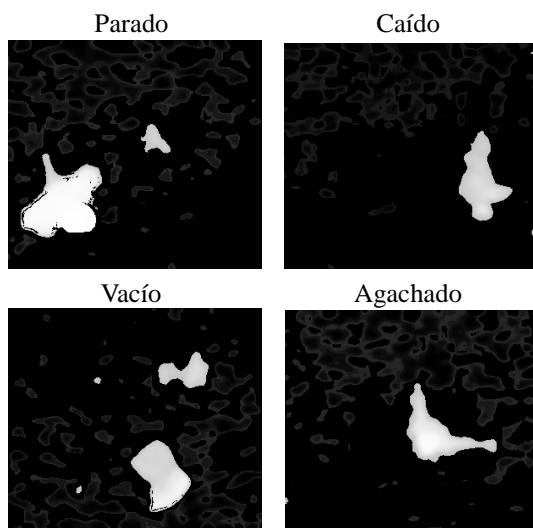


Figura 4: Imágenes de las diferentes clases previamente tratadas para reducir el ruido de fondo. La imagen Vacío muestra un caso en el que hay objetos de la habitación que introducen ruido.

3.3. Conjunto de datos.

En las clases en las que haya detectada una persona, hay que tener en cuenta que hay diferentes posibilidades dada la situación donde se encuentre la persona, o el tipo de caída que pueda producirse. Teniendo esto en cuenta, se obtuvieron 5000 imágenes de cada una de las clases para que, a la hora de realizar el modelo, estuvieran balanceadas. Cabe comentar que, lógicamente, había más imágenes de conjunto vacío y parado que de las otras dos clases, pero que tal y como se ha comentado antes referente al balanceamiento, se han escogido sólo 5000 imágenes de estas clases de manera aleatoria de la muestra original.

Además, para que el aprendizaje del modelo sea más eficiente, se aplicó una disminución de la resolución de la imagen que implica una reducción del ruido dado

por los contornos de la imagen, que dificultarían la diferenciación entre las clases. De forma tal que, se aplica una reducción de la resolución de las imágenes a 300 x 270 píxeles.

3.4. Eliminación del fondo.

Finalmente, se realiza un proceso de eliminación del ruido de fondo para permitir una detección más clara de las cuatro clases. Para ello, se han obtenido 40 imágenes del conjunto vacío, que se usarán para entrenar el modelo. Se ha calculado la media μ y la desviación estándar σ de todos los píxeles de estas 40 imágenes. Una vez se tiene esto, a cada una de las imágenes del conjunto de datos original, se le ha aplicado la siguiente fórmula, considerando x_{ij}^k como el valor original del píxel en la posición ij de la imagen k y y_{ij}^k el nuevo valor que se le asignará según dicha ecuación:

$$y_{ij}^k = \begin{cases} x_{ij}^k & \text{si } x_{ij}^k < \mu - \sigma \\ 0 & \text{si } \mu - \sigma < x_{ij}^k < \mu + \sigma, \\ x_{ij}^k & \text{si } \mu + \sigma < x_{ij}^k \end{cases}$$

Donde esta fórmula se aplica a todos los píxeles de la imagen k , es decir, se aplica a todos los ij . Esta ecuación funciona como un filtro, donde si x_{ij}^k es un valor típico (si este está en el rango $(\mu - \sigma, \mu + \sigma)$), pertenece a “ruido de fondo” y, por tanto, puede eliminarse su valor; por otro lado, si este no es un valor típico, su valor se mantiene, ya que es un objeto relevante.

Si se aplica este proceso, se obtiene el aspecto de las imágenes de la Figura 4, donde se puede ver que hay bastante área negra, haciendo referencia a píxeles de valor 0. Una vez realizado este proceso tenemos los datos listos para continuar con el análisis de los datos.

4. Análisis de datos

Un punto muy importante a la hora de realizar el modelo de clasificación, que detectará las cuatro clases comentadas en la sección anterior, es tener un conocimiento de la similitud que hay entre las diferentes imágenes de una misma clase y, a su vez, si hay mucha diferencia entre las diferentes clases.

Para ello, se va a hacer uso de la librería *Imgcompare* de Python que proporciona un coeficiente que indica mediante un porcentaje el grado de diferencia entre dos imágenes. Así pues, si este coeficiente es pequeño cuando se comparan imágenes de una misma clase, indicará que no habrá ruido dentro de la misma clase, facilitando así su reconocimiento. Por otro lado, si este coeficiente es grande cuando se comparan imágenes de diferentes clases, indicará que para el modelo será sencillo diferenciarlas y clasificarlas en clases diferentes. Ahora bien, si este coeficiente es pequeño, al

modelo le costará diferenciarlas y, debido a este ruido, no realizará una buena clasificación de clases.

Teniendo en cuenta esto, hay que comentar que este coeficiente se obtiene para cada pareja de imágenes. Por lo tanto, para obtener un coeficiente por cada clase y cada pareja de clases diferentes, se ha realizado la media de estos coeficientes para cada uno de los casos; obteniendo así una matriz de diferencias de las cuatro clases a estudiar. En la Tabla 1, se muestra la matriz comentada, junto a su varianza de cada una de las medias en la Tabla 2.

	Vacío	Parado	Agachado	Fin de la Caída
Vacío	5.97	8.36	8.52	8.36
Parado	8.36	12.53	11.51	11.53
Agachado	8.52	11.52	8.72	9.01
Fin de la Caída	8.36	11.53	9.01	7.61

Tabla 1: Matriz de diferencias entre las clases

	Vacío	Parado	Agachado	Fin de la Caída
Vacío	0.57	0.50	0.14	0.50
Parado	0.50	6.01	2.05	1.72
Agachado	0.14	2.05	1.08	0.69
Fin de la Caída	0.50	1.72	0.69	0.52

Tabla 2: Matriz de varianza entre las clases

En primer lugar, sorprende los valores tan bajos que se obtienen del coeficiente, ya que este es 0 si no hay ninguna diferencia, y 100 si la diferencia es total. Ninguno de los valores supera el 13% de diferencia según este coeficiente, lo que, a priori, indica que al modelo de clasificación le costará diferenciar las clases.

Ahora bien, estos valores tan bajos para los coeficientes se deben a que todas las imágenes están tomadas en el mismo plano de la habitación y, además, se ha realizado un proceso de limpieza del fondo de las imágenes, por lo que existen muchísimos píxeles iguales. Por lo tanto, la diferencia disminuye considerablemente, reflejándose el estado de la persona en una variación que no supera el 13% de diferencia.

Si se fija en los valores de esta matriz por clase, cabe esperar que los coeficientes sean menores en la diagonal que en el resto de la matriz. Ahora bien, tal y como se puede observar, esto sólo pasa en la clase Vacío y Fin de la Caída. En el caso de la clase Agachado, es menor la diferencia entre Agachado y Vacío que consigo misma. Prácticamente los coeficientes son iguales, y hay que tener en cuenta que, dentro del rango de la varianza, este podría ser menor que la diferencia entre Agachado y Vacío. Por otro lado, en el caso de la clase Parado, su diferencia consigo misma es mayor que con el resto de las clases. Esto se debe a que es una clase que tiene mucho ruido, tal y como se puede ver en la varianza consigo misma, debido a que una persona puede estar de pie en sitios diferentes de la habitación.

En referencia al coeficiente entre las diferentes clases, cabe comentar que la clase vacío presenta unos coeficientes de diferencia menores con las otras clases que el resto de las parejas posibles (sin contar los de la diagonal). Esto se debe a que, al no haber ninguna persona, la imagen sólo muestra la habitación y algunos objetos que puedan situarse en esta pero que, por lo general, forma parte del mismo “fondo” que se ve en el resto de las imágenes.

Finalmente, es interesante ver cómo el coeficiente de diferencia entre Fin de la Caída y Agachado es menor que Parado con estas dos. Esto se debe a que hay una mayor similitud en el estado de agacharse y estar caído en el suelo, que con una persona de pie, ya que la intensidad de blanco sobre negro captada por la cámara es mucho mayor en la clase Parado que en las otras dos. Una vez finalizado este proceso tenemos los datos listos para seguir con el proceso de selección de los modelos de predicción.

5. Modelos de predicción

Tras definir el conjunto de datos y sus clases, y un posterior estudio estadístico de la relación entre estas, se han de decidir qué modelos estadísticos de predicción se van a utilizar para realizar la clasificación o, mejor dicho, la detección.

A continuación, se muestra una lista de los 6 modelos utilizados, cuyas especificaciones respecto a los parámetros de estos se encontrarán en secciones posteriores.

- Regresión Logística: determina la probabilidad de una variable a pertenecer a una clase u otra mediante un ajuste lineal [12].
- K-Próximos Vecinos (K-NN, K-Nearest Neighbor en inglés): clasifica una variable en una clase determinada en función de las clases a las que pertenecen las k variables más cercanas [13].
- Clasificador Boosting con Gradiente (GBC, Gradient Boosting Classifier en inglés): se basa en generar diferentes Árboles de Decisión ajustándolos a diferentes copias creadas por el conjunto de datos original, con el objetivo final de combinarlos y crear un modelo de predicción final [14].
- Bosques Aleatorios (RF, Random Forest en inglés): genera Árboles de Decisión donde en cada una de las ramas del árbol no se tienen en cuenta las clases de todas las variables del conjunto, sino que se realiza una selección aleatoria de un número determinado de predictores que serán los candidatos que tomarán la decisión de la siguiente rama [15].
- Máquinas de Soporte Vectorial (SVM, Support Vector Machines en inglés): trata de buscar un hiperplano (un plano de más de dos dimensiones) que separe las diferentes clases de un conjunto de datos

de manera óptima, es decir, que haya la mayor distancia entre todos los puntos de los datos y este hiperplano [16].

- Perceptrón Multicapa: trata de simular las decisiones que se tomarían en una red de neuronas, teniendo diferentes “capas” de perceptrones concatenadas [17].

Debido a la longitud del trabajo, la explicación teórica detallada de cada modelo no se representa. Para obtener más información respecto a cada uno de los modelos véase la referencia al final de cada punto.

5.1. Medidas de evaluación

Antes de proseguir a la siguiente sección, es necesario definir unas medidas de evaluación para decidir cuál es la que mejor resultados proporciona. Estas medidas consisten en unos coeficientes que describen el acierto, u otras características, tales como: el Accuracy, la Precisión, el Recall y el F1-Score. A continuación, teniendo en cuenta las definiciones de la Tabla 3, se realiza una breve descripción de que aporta cada uno de estos índices junto a su fórmula:

- Accuracy: Indica el acierto de clases clasificadas correctamente en la predicción del modelo respecto al total [18].

$$Accuracy = \frac{VP + VN}{VP + FP + VN + FN}$$

- Precisión: Indica la cantidad de elementos clasificados como una determinada clase respecto al total de clasificados como aquella clase [18].

$$Precisión = \frac{VP}{VP + FP}$$

- Recall: Indica el grado de acierto de clasificación de una determinada clase [18].

$$Recall = \frac{VP}{VP + FN}$$

- F1-Score: Combina las métricas de la Precisión y el Recall mediante una media armónica [18].

$$F1 - Score = 2 \cdot \frac{Precisión \cdot Recall}{Precisión + Recall}$$

		Clase Predicha	
		Negativos	Positivos
Clase Real	Negativos	VN	FP
	Positivos	FN	VP

Tabla 3: Matriz de Confusión general de una determinada clase. El VP quiere decir Verdaderos Positivos, el VN Verdaderos Negativos, el FP Falsos Positivos y el FN Falsos Negativos.

6. Selección del modelo de clasificación

Una vez presentados los modelos que se van a considerar para realizar la clasificación en el apartado anterior, hay que realizar el ajuste al conjunto de datos.

Para ello, se crea un subconjunto de validación, que no formará parte del conjunto de datos con el que se ajustarán los modelos comentados en el apartado anterior, con el cuál se obtendrán los coeficientes necesarios para tomar la decisión de cuál es el modelo que mejor se ajusta.

Una vez separado este subconjunto de validación respecto el resto, teniendo un total de 24 mil imágenes, los modelos se ajustan al conjunto de entrenamiento aplicando la técnica de la Validación Cruzada [18], para evitar un sobreajuste al conjunto de entrenamiento.

Para ello, dado un modelo y sus diferentes combinaciones de parámetros, esta técnica comienza separando el conjunto de entrenamiento en 5 grupos diferentes del mismo tamaño. Una vez obtenido el resultado, se ajusta el modelo, con una combinación de parámetros determinados, a los 4 conjuntos de datos, usando el restante para la evaluación del modelo.

Este proceso se repite cuando los 5 grupos han formado parte de la evaluación, obteniendo así 5 índices de aciertos diferentes con los que hacer una media final. Todo este ajuste se repite para las diferentes combinaciones de parámetros que tenga el modelo determinado, eligiendo al final como mejor combinación el que mejor media de aciertos proporcione.

Cabe remarcar que esta combinación de parámetros se realiza como una búsqueda exhaustiva sobre unos posibles valores dados para cada uno de los parámetros, siendo una técnica conocida como *gridsearch*.

Debido a que se ha ajustado el modelo no sólo a un conjunto de datos, sino a diferentes bloques, y se ha evaluado en diferentes situaciones, permite que no haya el sobreajuste comentado previamente.

Finalmente, una vez se obtiene cuáles son los parámetros óptimos para cada modelo, se ajustan sobre el total del conjunto de entrenamiento y se evalúan sobre el conjunto de validación.

6.1. Parámetros obtenidos

Mediante el conjunto de ajuste del modelo, es decir, el conjunto de entrenamiento, se escogen cuáles son las especificaciones de los parámetros que hacen que dicho modelo se ajuste mejor al conjunto de datos. El código usado ajusta estos modelos con las combinaciones posibles de los diferentes parámetros que se tienen y escoge, mediante la Validación Cruzada y la búsqueda de parámetros, cuáles son los óptimos en términos del acierto en la clasificación general.

Considerando esto, se ha obtenido que en el caso del modelo de K-NN se han de usar 3 próximos vecinos.

En el caso del GBC, la combinación óptima ha sido de 50 estimadores, una profundidad máxima de los árboles de 10 y con un factor de *shrinkage* de 0.75. Para los RF, el conjunto que ha dado el ajuste óptimo ha sido de 10 estimadores, sin profundidad máxima, con un mínimo de 2 muestras para poder realizar una decisión y sin una limitación máxima de hojas por nodo. En referencia a los SVM, el parámetro C ha sido 1.0 junto a una función Kernel que, en este caso, ha sido uno lineal. Finalmente, en el caso del Perceptrón Multicapa, se ha escogido que la configuración de capas ocultas sea de 6 capas de 25, 50, 100, 100, 50, 25 perceptrones cada una, haciendo uso de 1000 iteraciones en el ajuste.

6.2. Análisis de la Matriz de Confusión y los índices

Esta evaluación que se realiza con el conjunto de prueba permite construir una matriz de confusión para comparar los diferentes resultados obtenidos para cada modelo. Las matrices de confusión para cada uno de los modelos se pueden observar en la Figura 5 donde el eje horizontal representa la clase predicha y el eje vertical la clase verdadera.

En esta Figura 5, se pueden observar las diferentes clasificaciones realizadas con cada uno de los modelos

de predicción. Si el modelo clasificase bien todas las imágenes, se tendría una diagonal blanca en la imagen, con un 1000 dentro de este. Ahora bien, como se puede ver en la Figura 5, esto no es así.

En primer lugar, se puede observar, que las 1000 imágenes del conjunto vacío están bien clasificadas en casi todos los modelos. Esto se debe a que es la imagen más fácil de identificar, ya que no hay una “figura humana” que pueda ser difícil de identificar en una determinada posición.

Resulta curioso la predicción de la clase Agachado en los modelos como la Regresión Logística, K-NN, SVM o el Perceptrón Multicapa, donde se puede observar que una gran parte de las imágenes que contienen a una persona agachada, se han clasificado como la clase Parado (en el caso del Perceptrón Multicapa llega a 342 clases Agachado clasificados como Parado). Esto se debe a que este modelo tiene la dificultad de diferenciar la clase Parado con la clase Agachado ya que, por “proximidad”, son imágenes bastante parecidas; como se puede observar en la matriz de correlación (véase Tabla 1); es mayor la diferencia entre la propia clase Parado que entre las clases Parado y Agachado). Además, pasa lo mismo con la clasificación de la clase Caída Fin en la Matriz del Perceptrón Multicapa, donde 213 imágenes de esta clase las clasifica como Agachado.

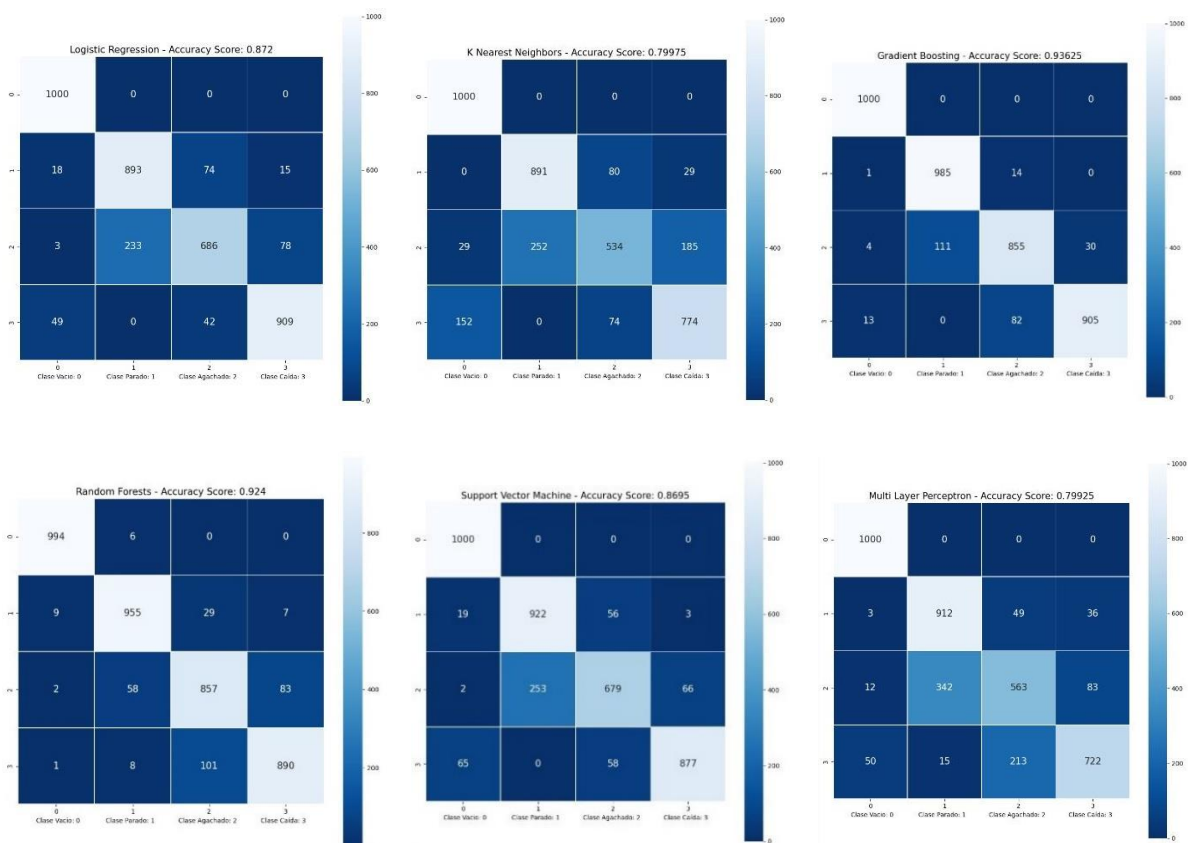


Figura 5: Matriz de confusión de cada modelo, para los cuales hay 1000 imágenes en cada una de las clases.

Mediante estas matrices de confusión, se puede analizar cada una de las predicciones para obtener conclusiones y detectar curiosidades de cada uno de los modelos. Ahora bien, para decidir cuál es el que mejor resultados proporciona, en vez de observar cada una de estas matrices, se puede hacer uso de las medidas de evaluación presentadas en la sección 5.1.

Teniendo en cuenta estos coeficientes, que se construyen mediante las matrices de confusión de cada uno de los modelos (véase Figura 5), se obtienen los valores mostrados en la Tabla 4.

Antes de comparar entre modelos, se considera necesario realizar una explicación sobre qué indican exactamente cada uno de estos coeficientes y cómo se utilizan a la hora de seleccionar el modelo final de clasificación. Por ejemplo, como se puede observar en el modelo del MLP, para la clase Parado, dicho modelo proporciona un Recall de un 91%. Esto quiere decir que el modelo casi siempre acertará cuándo la persona detectada en el vídeo está agachada.

Por otro lado, la Precisión es del 72%, indicando que un 28% de las veces en las que el modelo indica que hay una persona parada, no lo estará realmente (de hecho, de la matriz de confusión, se ve que estará agachada). Si lo que se pretende es detectar siempre cuándo una persona está parada, independientemente de que alguna vez avise de que está alguien parado y realmente no lo está, este resultado es perfecto, ya que casi siempre que haya alguien parado, lo indicará.

	Accuracy	Clase	Precisión	Recall	F1-Score
Regresión Logística	87.2%	Vacío	0.93	1.00	0.97
		Parado	0.79	0.89	0.84
		Agachado	0.86	0.69	0.76
		Caído	0.91	0.91	0.91
		Promedio	0.87	0.87	0.87
K-NN	79.9%	Vacío	0.85	1.00	0.92
		Parado	0.78	0.89	0.83
		Agachado	0.78	0.53	0.63
		Caído	0.78	0.77	0.78
		Promedio	0.80	0.80	0.79
GBC	93.6%	Vacío	0.98	1.00	0.99
		Parado	0.90	0.98	0.94
		Agachado	0.90	0.85	0.88
		Caído	0.97	0.91	0.94
		Promedio	0.94	0.94	0.94
RF	92.4%	Vacío	0.99	0.99	0.99
		Parado	0.93	0.95	0.94
		Agachado	0.87	0.86	0.86
		Caído	0.91	0.89	0.90
		Promedio	0.93	0.92	0.92
SVM	86.9%	Vacío	0.92	1.00	0.96
		Parado	0.78	0.92	0.85
		Agachado	0.86	0.68	0.76
		Caído	0.93	0.88	0.90
		Promedio	0.87	0.87	0.87
Perceptrón Multicapa	79.9%	Vacío	0.94	1.00	0.97
		Parado	0.72	0.91	0.80
		Agachado	0.68	0.56	0.62
		Caído	0.86	0.72	0.78
		Promedio	0.80	0.80	0.79

Tabla 4: Reportes de clasificación

En cambio, el Recall de la clase Caído es del 72%, esto quiere decir que solamente avisará un 72% de las

veces en que una persona se haya caído. Por lo tanto, no es muy útil si lo que importa es la salud de la persona ya que, cuando alguien se caiga de verdad, puede que el modelo no te indique que se haya caído. Por otra parte, la Precisión es del 86%, esto quiere decir que casi nunca dirá que una persona se ha caído en el resto de los estados en las que pueda estar, por lo que no creará “falsas alarmas”, sólo un 14% de los casos. Debido a esta diferencia entre Recall y Precisión, es por lo que es importante el F1-score, ya que es una especie de “ponderación” entre ambos coeficientes. Así pues, dependiendo de cuál sea el objetivo del modelo, y cuáles de las cuatro clases son las más relevantes, y en las que crear “falsas alarmas” no sea un problema, si lo que está en juego es la salud de las personas habrá que observar diferentes coeficientes.

Teniendo esto en cuenta, se sabe que, aparte de que el Accuracy sea alto, es importante, por la salud de la persona que vaya a estar en la habitación, detectar bien las clases Agachado y Caído, independientemente de que haya “falsas alarmas” en estas clases. Esto se debe a que es preferible que el modelo indique que una persona se ha caído e ir y que realmente no se haya caído, que a que la persona se haya caído y el modelo no te indique de dicho estado. Por esto, junto al Accuracy general del modelo, se deberá tener en cuenta que el Recall de las clases Agachado y Caído ha de ser alto, que, junto a una precisión que tampoco sea muy baja (ya que si no se tendrán demasiadas “falsas alarmas”), vendrá reflejado en el F1-Score.

Estudiando la Tabla 4, se puede ver que los modelos que proporcionan un mayor Accuracy general son el GBC y los RF. Ahora bien, fijándonos en los F1-Score de las cuatro clases del GBC, no sólo en la del estado Caído, o directamente en la media de las cuatro clases, se puede ver que son los más altos respecto al resto del modelo. Así pues, tanto por Accuracy general, como por F1-Score, en el que ya viene implícito un buen Recall en la clase Caído, queda claro que el mejor modelo de clasificación para este estudio es el GBC.

6.3. Análisis de Componentes Principales

Tras estos resultados obtenidos, resulta interesante aplicar una técnica que permite reducir la cantidad de variables de entrada, llamada Análisis de Componentes Principales (Principal Component Analysis en inglés, PCA) [20]. Esta técnica define nuevas variables mediante la combinación lineal de las variables de entrada. En este caso, las variables de entrada son los píxeles de cada una de las imágenes, permitiendo pasar de $300 \times 270 = 81000$ píxeles en cada una de las imágenes, a un conjunto más reducido de variables de entrada. Esto reducirá la varianza en los datos, mejorando, posiblemente, la predicción.

La Figura 6 del anexo representa la varianza explicada acumulada a la hora de escoger las componentes principales. Si se escogiesen todas las componentes

principales, se tendría el 100% de la varianza explicada, pero no se conseguiría nada, ya que escogeríamos la misma cantidad de variables de entrada que en el modelo sin PCA, lo que no serían las mismas ya que serían combinaciones lineales de estas. Así pues, mediante la Figura 6 del anexo de la varianza explicada acumulada y una lista de las variables, se puede ver que con 162 componentes principales, se explica un 85% de la varianza. Por lo tanto, con este proceso hemos conseguido reducir considerablemente la cantidad de variables de entrada al modelo, perdiendo sólo un 15% de la varianza explicada.

Prosiguiendo con estos nuevos componentes principales, con las 162 escogidas, se vuelven a aplicar los ajustes de los modelos al nuevo conjunto de entrenamiento, utilizando la Validación Cruzada para determinar la combinación de los parámetros óptimos en cada uno de los modelos, y se evalúan sobre el conjunto de prueba, obteniendo los coeficientes de evaluación que se muestran en la Tabla 5 del anexo.

Después de un estudio de los coeficientes obtenidos en la Tabla 5 del anexo se puede observar cómo, a excepción de los coeficientes de evaluación obtenidos para la clase Vacío en el GBC, todos los valores son menores que en los modelos de clasificación obtenidos sin la aplicación de PCA. Esto seguramente se deba a que la varianza que se está dejando, a la hora de escoger menos componentes principales, sí que es relevante a la hora de predecir la clase de otras imágenes. Por otro lado, cabe comentar que utilizando PCA también se obtiene que el mejor modelo de clasificación para este estudio es el GBC, aplicando la misma evaluación que en el apartado 6.2.

Este resultado de coeficientes de evaluación sugiere que aplicar PCA en este caso no es una técnica útil, ya que, a parte de obtener peores resultados en la clasificación, el tiempo que invierte el código para realizar las combinaciones lineales es mayor, añadiendo, además, que el previo estudio de cuántos componentes escoger también es un tiempo que invertir.

7. Aplicación del modelo en caídas reales

Una vez seleccionado el mejor modelo de clasificación como el obtenido mediante los GBC, se ha de hacer uso para realizar la detección de caídas a tiempo real, que es la finalidad de este trabajo.

Para ello, lo que se realiza es una desfragmentación de un vídeo, generando 60 imágenes por cada segundo de grabación; de la misma forma en las que se obtuvieron las imágenes para el ajuste del modelo de clasificación. Cuando se tienen estas imágenes, se aplica el modelo de predicción a cada una de ellas, y va clasificando la imagen que detecta en una de las cuatro clases comentadas en este trabajo.

En esta sección, se van a estudiar las 10 secuencias de caídas de pacientes, Figuras (7-16) del anexo, aplicando el modelo de clasificación obtenido mediante GBC en el apartado anterior, y representando en gráficas en qué clase se encuentra según la predicción.

Tal y como se puede visualizar en la Figura 7 del anexo, el proceso de estos vídeos consta de una habitación en la cual siempre se inicia con un individuo en el estado Parado, donde el modelo de predicción detecta correctamente en la mayoría de los casos, dicho estado con un recall promedio del 94% como se muestra en la Tabla 6 del anexo.

Posteriormente, la curva azul indica el cambio de estado a Agachado. Sin embargo, el modelo para dicha clase varía entre la clase Parado y Agachado, pero la clasifica más como Agachado que Parado. En este punto el modelo no clasifica tan bien ya que se puede observar en las imágenes, que el recall promedio para este estado es del 39% según la Tabla 6 del anexo.

En la tercera etapa, se puede ver cómo el individuo pasa a estar a clase Caído, pero sin embargo, el modelo no detecta que la persona está en ese estado durante todo el periodo que indica la línea azul. Ahora bien, la mayoría de veces se sitúa en el estado Caído.

Estos errores en la clasificación se deben a que el recall de la clase caído no es del 100%, por lo que puede clasificarlas en otra de las clases que, en su mayoría, como se podía ver en la matriz de confusión de la Figura 5, será como la clase Agachado. Este problema se evidencia en todas las Figuras (7-16) del anexo, donde se puede observar como el resultado promedio del recall en esta clase es del 29% de aciertos según se muestra en la Tabla 6 del anexo, es la clase que peor clasifica el modelo.

Tras la caída, el residente pasa al estado Agachado y previamente al Parado, donde el modelo entrenado presenta resultados similares que los explicados anteriormente.

Finalmente, se puede ver que la clase Vacío al final se clasifican algunas imágenes, como clase Parado, debido, seguramente, al ruido que generan los objetos de la habitación. El valor promedio del recall en esta clase, es del 87% de aciertos según se puede ver en la Tabla 6 del anexo.

Un tema interesante que tratar en estas figuras de seguimiento, es que se generan muy pocas “falsas alarmas” de caídas, es decir, son pocas las imágenes que se clasifican como Caída en cualquiera de los otros estados. Sin embargo, sí que hay estados Caído que no se han clasificado como tal. Esto podría suponer un problema en el caso en que el sistema no avisase al sanitario. Resulta curioso este resultado ya que, a la hora de evaluar el modelo con el conjunto de prueba, se obtuvo un recall del 91% según se ve en la Tabla 6 del anexo, mientras que en los vídeos el recall de esta clase es por lo general mucho más bajo.

8. Conclusiones

Este estudio ha mostrado una metodología para crear un detector de caídas que será utilizado en una residencia de personas con diferentes discapacidades físicas o mentales, que envíe una señal al monitor de guardia en caso de que el residente se caiga. Esta ha consistido en captar imágenes de vídeos grabados mediante una cámara de profundidad Intel D435, las cuales se han clasificado en 4 clases diferentes y generado un modelo de clasificación óptimo utilizando GBC, con un Accuracy del 93.6% y un F1-Score medio del 94%. Tras generar este modelo, mediante un código Python, se ha presentado un resumen de los resultados obtenidos del seguimiento a las 10 secuencias reales de caída estudiadas, donde se ha podido observar su utilidad.

En un caso real, como trabajo a futuro a desarrollar, el seguimiento se realizaría a tiempo real, donde la imagen captada por la cámara se desfragmentaría y se pasaría sobre el modelo para que detecte qué es lo que se observa en la imagen, determinando en cuál de las cuatro posibles clases se encuentra el individuo. Cuando el modelo detecte una caída, se enviaría una señal mediante la aplicación Telegram a los contactos designados previamente.

Por un lado, hay que destacar que en este trabajo se han presentado 6 modelos de clasificación y elegido el más óptimo de estos 6. Sin embargo, la cantidad de modelos de clasificación que se conocen es mucho mayor. Es decir, puede haber otros modelos de clasificación que arrojen mejores resultados que el GBC visto en este estudio.

Otra mejora que se podría aplicar a estudios futuros es la aplicación de modelos de detección de objetos dentro de las imágenes, los cuales clasificarían cada objeto que se encuentre dentro del fotograma.

Finalmente, hay que comentar que este modelo obtenido se iría actualizando periódicamente, todas las imágenes grabadas se añadirían al modelo para volver a reajustarlo, permitiendo así una mejora en el grado de detección.

Referencias

- [1] **BERNAL-CELESTINO RJ, R.-G. J.-R.-S.-H. (2014)**. Caídas en personas con discapacidad intelectual. *Revista Rol de Enfermería*, 240-246.
- [2] **OMS. (12 de 11 de 2018)**. Obtenido de Organización Mundial de la Salud: <http://www.who.int/es/news-room/fact-sheets/detail/falls>
- [3] **Rodríguez Molinero, A., Narvaiza, L., Gálvez Barrón, C., & de la Cruz, J. (Noviembre - Diciembre de 2015)**. Caídas en la población anciana española: incidencia, consecuencias y factores de riesgo. *Revista Española de Geriatria y Gerontología*, 50, 274-280.
- [4] **Birku, Y., & Agrawal, H. (2018)**. Survey on fall detection systems. *International Journal of Pure and Applied Mathematics*, 118, 2537-2543.
- [5] **Sense4Care. (12 de 11 de 2018)**. Obtenido de <https://www.sense4care.com/product/angel4-fall-detection/>
- [6] **M. R. y. S. W. Thomas Degen, H. J. (2003)**. Speedy: A fall detector in a wrist watch. *Wearable Computers, Seventh IEEE International Symposium*.
- [7] **Makris, G. M. (2012)**. Fall detection system using kinect's infrared sensors. *Journal of Real-Time Image Processing*, (págs. 635-646).
- [8] **Xu, T., Zhou, Y., & Zhu, J. (2018)**. New Advances and Challenges of Fall Detection Systems: A Survey. *Appl. Sci.* 418.
- [9] **Christian Marzahl, P. P. (s.f.)**. Unobtrusive Fall Detection Using 3D Images of a Gaming Console: Concept and First Results. *Stralsund, Germany: University of Applied Sciences*.
- [10] **Samuele Gasparrini, E. C. (2014)**. A Depth-Based Fall Detection System Using a Kinect Sensor. *Sensors* (págs. 2756-2775).
- [11] **Jin Wang, Z. Z. (2014)**. An enhanced fall detection system for elderly person monitoring using consumer home networks. *IEEE Transactions on Consumer Electronics*, 23-29.
- [12] **W. Hosmer, D., & Lemeshow, S. (1989)**. *Applied Logistic Regression*. New York: John Wiley & Sons, INC.
- [13] **JAMES M. KELLER, M. R. (1985)**. A Fuzzy K-Nearest Neighbor Algorithm. *IEEE Transactions On Systems, Man, And Cybernetics*, Vol. SMC-15, 580-585.
- [14] **Natekin, A., & Knoll, A. (2013)**. Gradient Boosting Machines, A Tutorial. *Frontiers in neurorobotics*, 21.
- [15] **Williams G. (2011)** Random Forests. In: *Data Mining with Rattle and R. Use R*. Springer, New York, NY.
- [16] **Deng, N. & Tian, Y. & Zhang, C.. (2012)**. Support vector machines: Optimization based theory, algorithms, and extensions. 10.1201/b14297.
- [17] **BISHOP, C. M. (1995)**. *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.
- [18] **Scikit-Learn Machine Learning in Python. (13 de 09 de 2019)**. Obtenido de <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics>
- [19] **Dunteman, G. H. (1989)**. *Principal Components Analysis*. California: Sage University Paper.

Repositorio

<https://github.com/elian880530/TFM-Eliau>

Anexo

	Accuracy	Clase	Precisión	Recall	F1-Score
Regresión Logística	61%	Vacío	0.61	1.00	0.76
		Parado	0.55	0.29	0.38
		Agachado	0.58	0.49	0.53
		Caído	0.68	0.66	0.67
		Promedio	0.60	0.61	0.58
K-NN	60.7%	Vacío	0.64	1.00	0.78
		Parado	0.66	0.42	0.52
		Agachado	0.53	0.45	0.48
		Caído	0.60	0.56	0.58
		Promedio	0.60	0.60	0.59
GBC	78.5%	Vacío	1.00	1.00	1.00
		Parado	0.69	0.87	0.77
		Agachado	0.63	0.55	0.58
		Caído	0.83	0.72	0.77
		Promedio	0.78	0.78	0.77
RF	68.5%	Vacío	0.75	1.00	0.86
		Parado	0.68	0.84	0.75
		Agachado	0.53	0.42	0.47
		Caído	0.75	0.48	0.58
		Promedio	0.67	0.68	0.66
SVM	60.2%	Vacío	0.58	1.00	0.74
		Parado	0.56	0.31	0.40
		Agachado	0.63	0.45	0.53
		Caído	0.64	0.65	0.64
		Promedio	0.60	0.60	0.57
Perceptrón Multicapa	68.5%	Vacío	0.73	1.00	0.84
		Parado	0.69	0.56	0.62
		Agachado	0.57	0.46	0.51
		Caído	0.71	0.72	0.71
		Promedio	0.67	0.68	0.67

Tabla 5: Reportes de clasificación utilizando PCA.

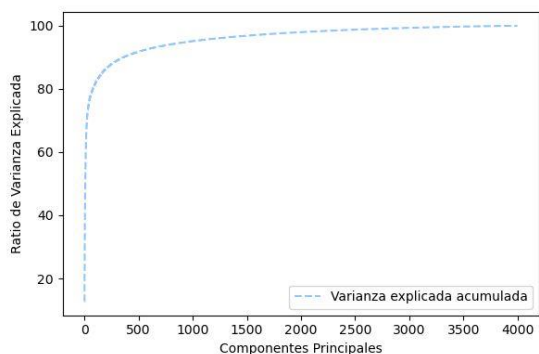


Figura 6: Representación gráfica de la varianza explicada.

	Vacío	Parado	Agachado	Caída
1	81%	98%	57%	50%
2	100%	95%	46%	26%
3	-	99%	45%	8%
4	66%	96%	25%	38%
5	-	85%	45%	16%
6	-	96%	37%	27%
7	91%	100%	36%	27%
8	100%	77%	22%	0.6%
9	100%	98%	6%	4.5%
10	77%	98%	75%	96%
Media	87%	94%	39%	29%

Tabla 6: Promedio del Recall en cada uno de los videos.

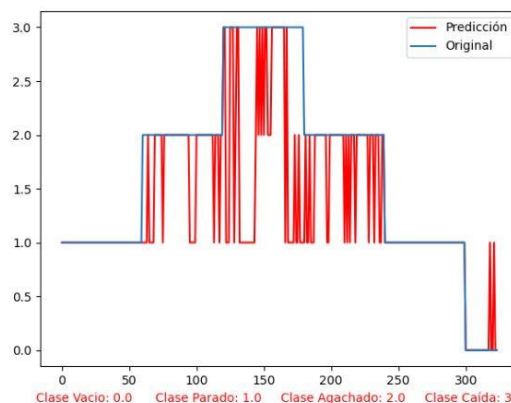


Figura 7: Seguimiento del video 7 donde se representan las 4 clases. Las cuatro clases vienen representadas en el eje vertical como 0 la clase Vacío, 1 la clase Parado, 2 la clase Agachado y 3 la clase Caída. Por otro lado, el eje horizontal representa la escala temporal, donde cada elemento es una imagen y 60 imágenes conforman 1 segundo.

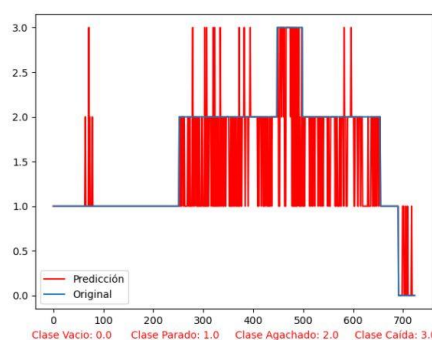


Figura 8: Seguimiento del video 1 representando las 4 clases.

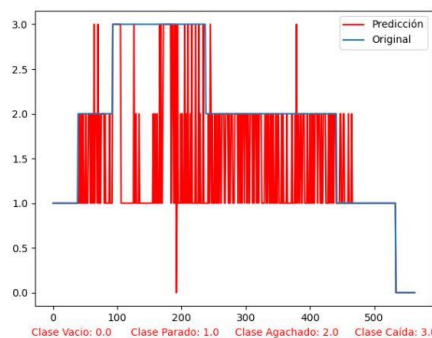


Figura 9: Seguimiento del video 2 representando las 4 clases.

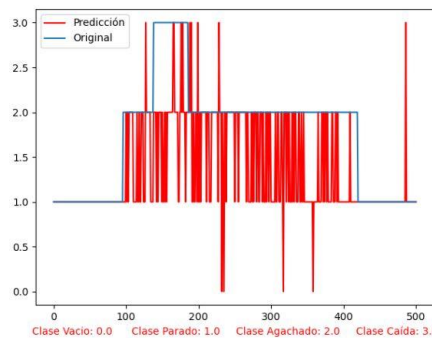


Figura 10: Seguimiento del video 3 representando las 4 clases.

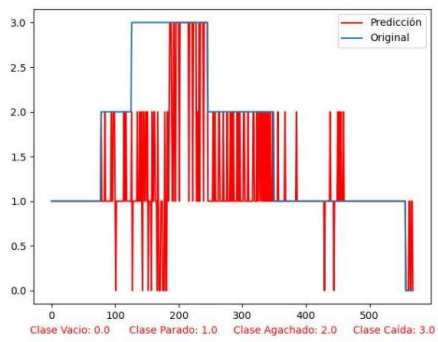


Figura 11: Seguimiento del video 4 representando las 4 clases.

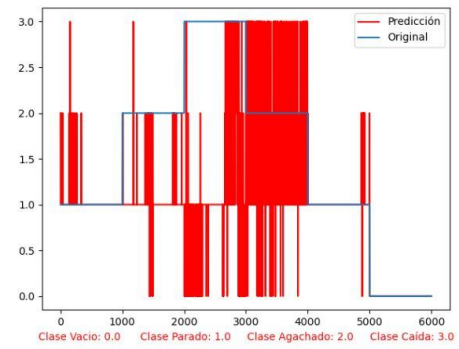


Figura 15: Seguimiento del video 9 representando las 4 clases.

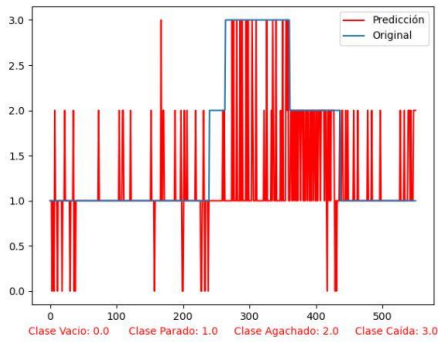


Figura 12: Seguimiento del video 5 representando las 4 clases.

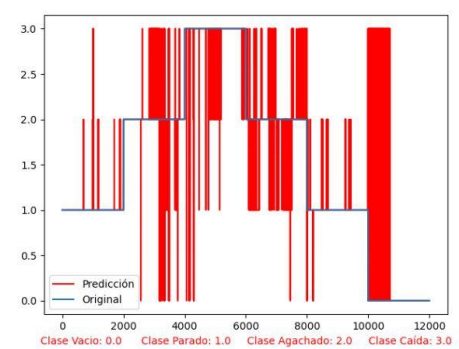


Figura 16: Seguimiento del video 10 representando las 4 clases.

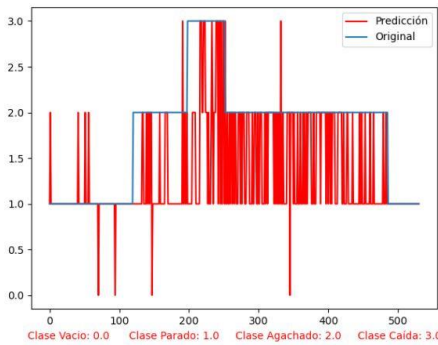


Figura 13: Seguimiento del video 6 representando las 4 clases.



Figura 17: Matriz de Confusión del video 1.

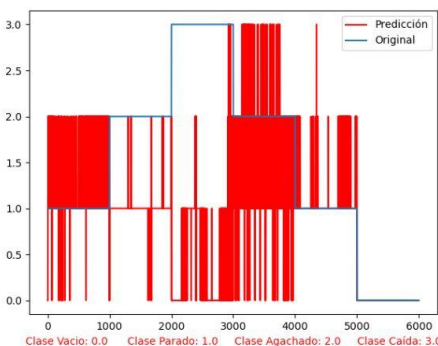


Figura 14: Seguimiento del video representando las 4 clases.

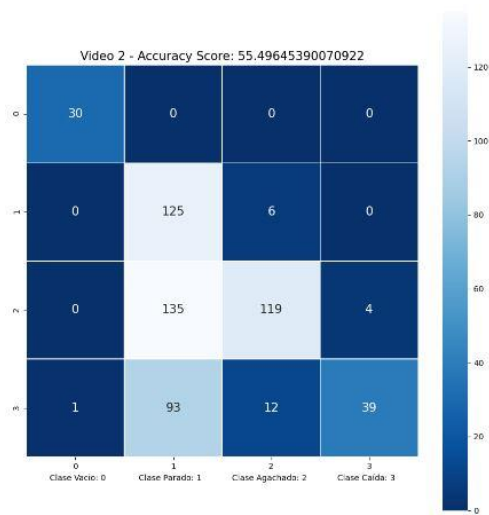


Figura 18: Matriz de Confusión del video 2.

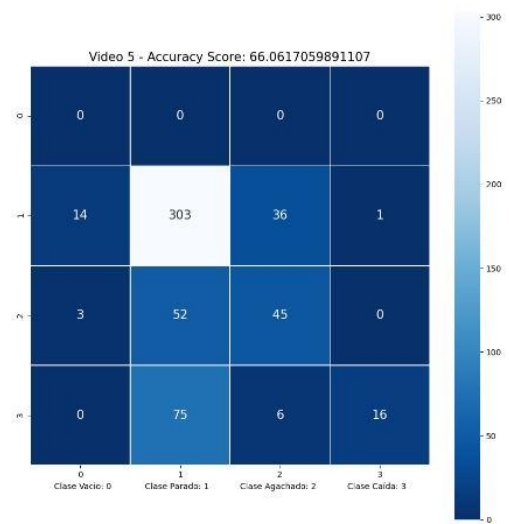


Figura 21: Matriz de Confusión del video 5.

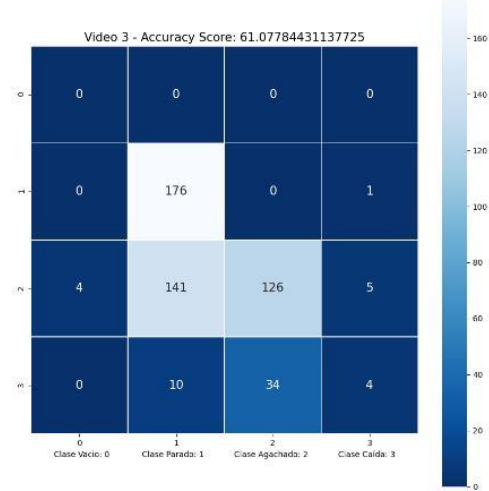


Figura 19: Matriz de Confusión del video 3.

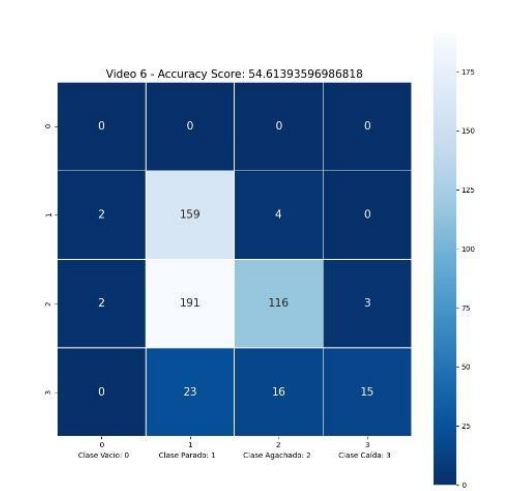


Figura 22: Matriz de Confusión del video 6.

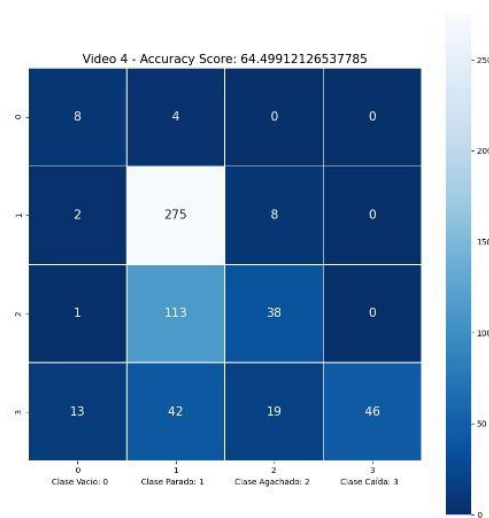


Figura 20: Matriz de Confusión del video 4.

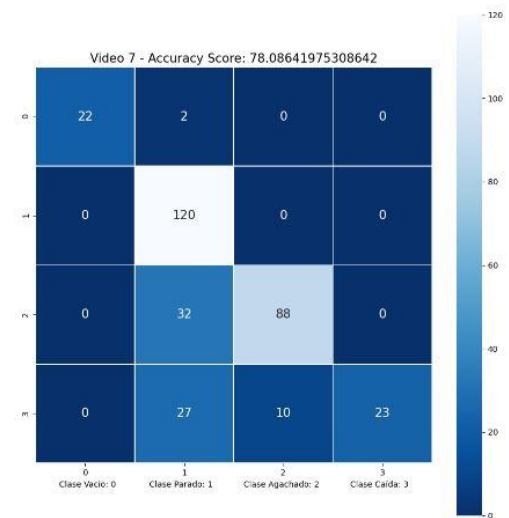


Figura 23: Matriz de Confusión del video 7.

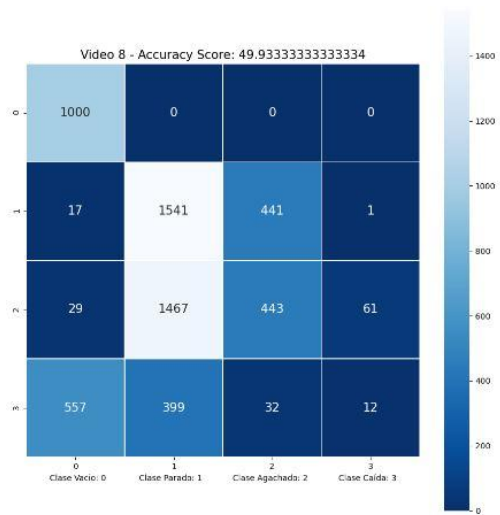


Figura 24: Matriz de Confusión del video 8.

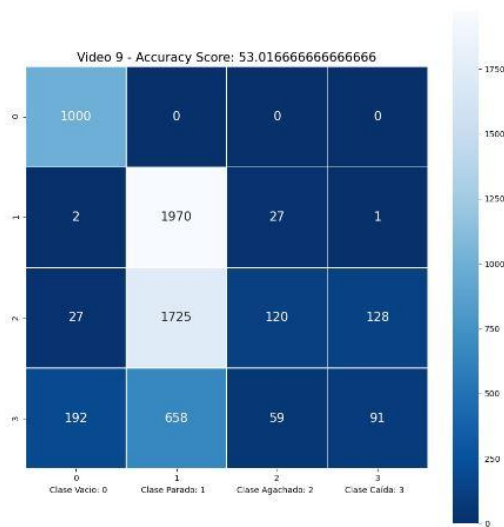


Figura 25: Matriz de Confusión del video 9.

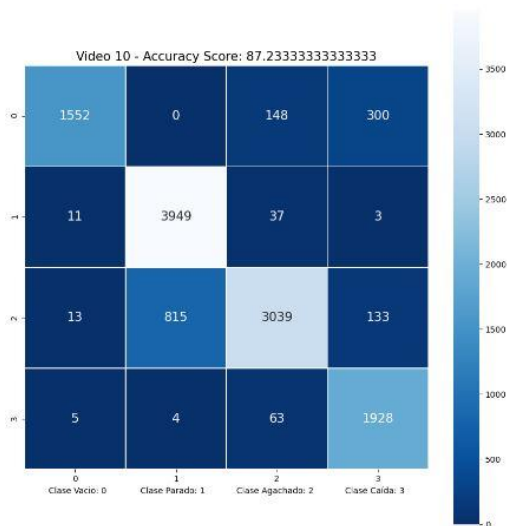


Figura 26: Matriz de Confusión del video 10.