Universidad Zaragoza
1542

Trabajo Fin de Grado

Monitorización de pacientes en estudios del sueño con cámaras de eventos
Patient monitoring in sleep studies with event cameras

Autora

Nerea Gallego Sánchez

Director

Eduardo Montijano Muñoz

Escuela de Ingeniería y Arquitectura

2023

Resumen

Los estudios del sueño son esenciales para el diagnóstico y tratamiento de los trastornos del sueño. Existen diferentes técnicas de monitorización del sueño capaces de medir diversos parámetros como la actividad muscular, la actividad cerebral y la respiración. Sin embargo, estas técnicas no proporcionan información sobre los movimientos del paciente mientras duerme, lo que puede limitar su capacidad para detectar ciertos trastornos como el síndrome de piernas inquietas o el sonambulismo. Con el objetivo de identificar patologías asociadas al movimiento durante el sueño, este TFG plantea un sistema de reconocimiento de acciones utilizando cámaras de eventos. Estos sensores son una alternativa menos invasiva para el paciente y tienen unas características ideales para capturar información precisa del movimiento en condiciones de baja iluminación.

El TFG propone un algoritmo de clasificación de las acciones del usuario durante el sueño a partir de los eventos que captura la cámara. El algoritmo es capaz de seleccionar los eventos relevantes para el problema de reconocimiento, agrupándolos en regiones de interés que describen la evolución espacial y temporal de cada acción. De estas regiones se extraen características representativas que son utilizadas por diferentes clasificadores para predecir las acciones realizadas.

Por otra parte, con el objetivo de poder evaluar el sistema desarrollado en condiciones realistas, se ha diseñado y grabado un dataset compuesto de varias secuencias en escenarios de sueño realistas en el marco de un proyecto de colaboración con la empresa Bitbrain. Este dataset es el primero de sus características, grabado con cámara de eventos en condiciones de escasa iluminación y reproduciendo movimientos durante el sueño.

En el TFG también se han realizado varios experimentos utilizando el dataset, obteniendo los mejores hiper-parámetros de los diferentes clasificadores considerados, comparándolos unos con otros y realizado una validación cruzada para analizar su capacidad de generalización en diferentes configuraciones. Los resultados obtenidos demuestran el potencial tanto del sistema de clasificación como del dataset en futuras aplicaciones de monitorización del sueño.

El carácter novedoso e investigador del trabajo realizado ha dado lugar a su inclusión como parte de un artículo de investigación enviado para su publicación a una conferencia internacional de prestigio.

Abstract

Sleep studies are essential for the diagnosis and treatment of sleep disorders. There are different sleep monitoring techniques capable of measuring various parameters such as muscle activity, brain activity and breathing. However, these techniques do not provide information on the patient's movements during sleep, which may limit their ability to detect certain disorders such as restless legs syndrome or sleepwalking. With the aim of identifying pathologies associated with movement during sleep, this TFG proposes an action recognition system using event cameras. These sensors are a less invasive alternative for the patient and have ideal characteristics for capturing accurate movement information in low light conditions.

The TFG proposes an algorithm for classifying the user's actions during sleep based on the events captured by the camera. The algorithm is able to select the events relevant to the recognition problem, grouping them into regions of interest that describe the spatial and temporal evolution of each action. Representative features are extracted from these regions and used by different classifiers to predict the actions performed.

Moreover, in order to be able to evaluate the developed system in realistic conditions, a dataset composed of several sequences in realistic sleep scenarios has been designed and recorded in the framework of a collaborative project with the company Bitbrain. This dataset is the first of its kind, recorded with an event camera in low-light conditions and reproducing movements during sleep.

In the TFG, several experiments have also been carried out using the dataset, obtaining the best hyper-parameters of the different classifiers considered, comparing them with each other and carrying out a cross-validation to analyse their generalisation capacity in different configurations. The results obtained demonstrate the potential of both the classification system and the dataset in future sleep monitoring applications.

The novel and investigative nature of the work has led to its inclusion as part of a research paper submitted for publication at a prestigious international conference.

Agradecimientos

Como presentación a este Trabajo Fin de Grado (TFG) quiero agradecer su apoyo y colaboración a todas aquellas personas que lo han hecho posible.

En primer lugar, me gustaría agradecer especialmente a mi tutor Eduardo por su dedicación, orientación y apoyo durante todo el desarrollo de este trabajo. Sus comentarios y sugerencias han sido fundamentales para dar forma a mis ideas y mejorar la calidad del trabajo. Su experiencia y disponibilidad han sido invaluables y estoy sinceramente agradecida por su ayuda.

En segundo lugar, agradecer al Instituto Universitario de Investigación en Ingeniería de Aragón (I3A) la oportunidad que me ha brindado de realizar el TFG con una de sus becas de investigación, en especial al área de Robótica, Percepción y Tiempo Real, que me ha permitido hacer uso de sus instalaciones durante los meses de trabajo. También agradecer a la empresa Bitbrain por cedernos sus laboratorios para la realización de grabaciones.

Me gustaría expresar mi más profundo agradecimiento a Carlos y Alberto que han contribuido de manera activa y desinteresada a mi investigación y desarrollo en este Trabajo de Fin de Grado. Sus valiosas aportaciones, asesoramiento y apoyo fueron fundamentales para el éxito de este proyecto.

Además quiero expresar mi agradecimiento a todos los participantes o sujetos de estudio que generosamente se ofrecieron a participar en este proyecto. Sin su colaboración y disposición, este trabajo no habría sido posible. Agradezco su tiempo y esfuerzo en contribuir a la investigación.

Por último, agradecer a mi familia, por su apoyo incondicional. A mis padres, quienes me han brindado su amor, comprensión y sacrificio a lo largo de mi vida y de esta importante etapa académica. Gracias por creer en mí y por ser mi fuente de motivación. Su constante apoyo financiero y emocional ha sido invaluable, y no podría haber llegado hasta aquí sin vosotros.

Índice general

Resumen **I**

Abstract **II**

Agradecimientos **III**

Índice general **IV**

Índice de figuras **VI**

Índice de tablas **VIII**

Índice de símbolos **1**

1. Introducción **2**

1.1. Objetivos y alcance 3

1.2. Organización de la memoria 5

2. Estado del arte **7**

2.1. Cámaras de eventos 7

2.1.1. Funcionamiento de las cámaras de eventos 7

2.1.2. Representación del conjunto de eventos 9

2.2. Reconocimiento de acciones 9

2.2.1. Reconocimiento de acciones durante el sueño 10

3. Metodología **12**

3.1. Filtrado de datos 12

3.2. Agrupación de eventos 14

3.3. Extracción de características 17

3.4. Clasificador de acciones 17

3.4.1. K Nearest Neighbors 20

3.4.2. Support Vector Classification 21

4. Evaluación en un entorno real **22**

4.1. Adquisición de datos 22

4.1.1. Participantes en el experimento 24

4.1.2. Etiquetado de datos 24

5. Resultados **26**

5.1. Diseño de experimentos 26

5.2. Métricas 27

5.3. Resultados 27

5.3.1. Ajuste de hiper-parámetros 27

5.3.2. Mejor clasificador 29

5.3.3. Validación cruzada por configuraciones 31

6. Conclusiones **32**

6.1. Líneas de trabajo futuro 33

Bibliografía **34**

Índice de figuras

1.1. Tecnologías de detección que se utilizan para diagnosticar y tratar trastornos del sueño. 3

1.2. Escenario de grabación 4

1.3. Diagrama de Gantt con los esfuerzos dedicados. 6

2.1. Magnitud de la señal con los umbrales de producción de eventos [1] 8

2.2. Representación de eventos 9

2.3. Ejemplos de la imagen original, el flujo óptico y el campo de flujo en cuatro canales calculado por [2] 10

3.1. Diagrama del algoritmo 12

3.2. Flujo del algoritmo con ejemplos 13

3.3. Ejemplo de eventos 14

3.4. Condición de pertenencia de un evento a un *cluster* 15

3.5. Representación de eventos. 16

3.6. Homogeneización de muestras 18

3.7. Etiquetas de las acciones a reconocer 19

3.8. Ejemplo de KNN en Sklearn 20

3.9. Ejemplo de SVM en Sklearn 21

4.1. Configuraciones de los sujetos en las grabaciones 23

5.1. Hiper-parámetros de *coarse* 28

5.2. Hiper-parámetros de *fine-grained* 29

5.3. Matriz de confusión 30

Índice de tablas

3.1. Etiqueta de las acciones a reconocer 19

4.1. Detalles de los sujetos 24

5.1. Separación de datos en tres conjuntos: *Train*, *Validation* y *Test*. 26

5.2. Resumen de hiper-parámetros 29

5.3. Medidas de los clasificadores 30

5.4. Tabla configuraciones 31

Índice de símbolos

ω	peso del clúster
c_x	coordenada x del centro de un clúster
c_y	coordenada y del centro de un clúster
e	evento
p	polaridad de un evento
r_x	dimensión del clúster en el eje x
r_y	dimensión del clúster en el eje y
r_{Kx}	espacio de búsqueda de eventos de un clúster en el eje x
r_{Ky}	espacio de búsqueda de eventos en un clúster en el eje y
t	timestamp de un evento
T^{last}	marca de tiempo del último evento en una coordenada x, y
$T_{NNb(x,y)}$	marca temporal del vecino más cercano al píxel x,y
T_{NNb}	umbral para el filtro *Nearest Neighbor Filter*
T_{ref}	umbral de tiempo mínimo que se impone entre dos eventos consecutivos que ocurren en el mísmo píxel
x	coordenada x de un evento
y	coordenada y de un evento

Capítulo 1

Introducción

El sueño es una función biológica muy importante para el ser humano y directamente relacionado con la salud y el bienestar. Aunque a menudo no se considera importante, una noche de sueño reparadora puede marcar la diferencia en el estado de ánimo, la energía y la capacidad de concentración. Durante los ciclos del sueño, el cuerpo y la mente pasan por varias fases del sueño en las que se incluye el sueño ligero, el sueño profundo y el movimiento ocular rápido (REM). Cada una de las fases tiene su función específica en la restauración y mantenimiento del cuerpo y la mente. Además, el sueño también está muy relacionado con el rendimiento y las funciones cognitivas durante el día.

Las alteraciones del sueño se han hecho prevalentes en la sociedad y los costes estimados en términos de días de enfermedad, tratamiento y otros impactos en la sociedad son considerables. Sin embargo, el interés de la investigación ha sido muy modesto durante la mayor parte del siglo pasado. Este interés principalmente se limitó a los profesionales de psiquiatría y psicología. Ya en el siglo XXI, los resultados de las investigaciones han empezado a describir el sueño como el factor clave de la restitución fisiológica, con implicaciones médicas de gran alcance [3]

Los algoritmos de reconocimiento de patrones pueden detectar patrones anormales de sueño, como el insomnio o la apnea del sueño, y proporcionan al médico una información valiosa para el diagnóstico y tratamiento. El reconocimiento de la actividad del sueño se ha estudiado con diferentes entradas sensoriales. La mayor parte de la literatura relacionada con el tema se basa en sensores portátiles como los polisomnogramas, que representan el estándar para medir la calidad del sueño [4], [5]. Sin embargo, el proceso de recopilación de las señales fisiológicas de un polisomnograma, como un electroencefalograma (EEG), requiere que los sujetos sean monitorizados por una unidad totalmente equipada y supervisada constantemente por personal técnico. Estos estudios se han realizado mediante dispositivos invasivos como electrodos en la cabeza, Figura 1.1, sensores de movimiento y de frecuencia cardíaca entre otros. Estos dispositivos realizan mediciones precisas pero resultan incómodos para el paciente, afectando al sueño y, por tanto, a la calidad de la

información recopilada. Otro de los grandes retos de los estudios preliminares sobre el análisis de imágenes para la monitorización del sueño son las condiciones de baja luminosidad. La alternativa más cercana a las cámaras externas es el uso de sensores de presión en la cama, que proporcionan una imagen del cuerpo del sujeto [6].

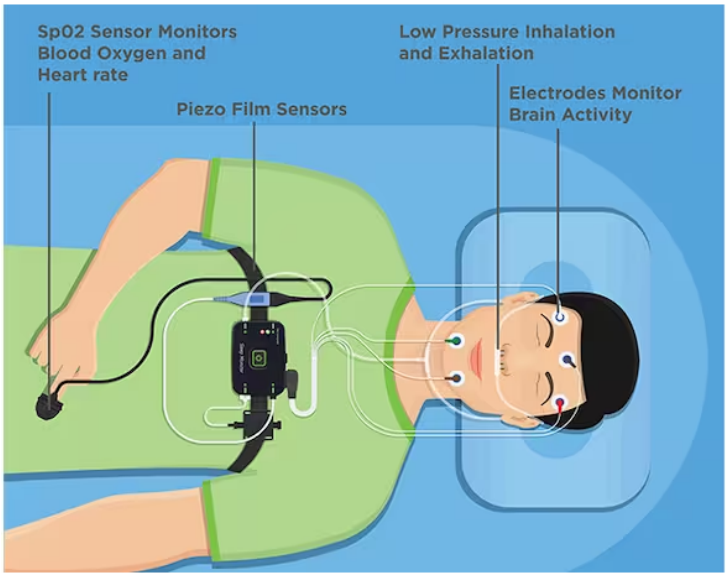


Figura 1.1: Tecnologías de detección que se utilizan para diagnosticar y tratar trastornos del sueño.

Motivado por los problemas expuestos con los dispositivos utilizados en los estudios del sueño y debido a las condiciones de baja iluminación que caracterizan a los estudios del sueño, en este TFG se propone utilizar cámaras de eventos como dispositivo alternativo para la monitorización en este tipo de estudios. Estos sensores suponen una alternativa menos invasiva para el paciente. Además, las cámaras de eventos disponen de alto rango dinámico y alta sensibilidad de captura, lo que las convierte en sensores ideales para los estudios del sueño (debido a las condiciones de baja iluminación). También, hay que tener en consideración que las cámaras de eventos son elementos más portátiles que algunos dispositivos (40 mm x 60 mm x 25 mm) convencionales lo que las hace una atractiva opción para estudiar el sueño en entornos clínicos. De esta manera, en el TFG se presenta un algoritmo de clasificación de las acciones que hace el usuario durante el sueño a partir de los datos proporcionados para una cámara de eventos, lo que puede ser muy útil para el futuro diagnostico de trastornos del sueño.

1.1. Objetivos y alcance

El objetivo principal del TFG es monitorizar pacientes en estudios del sueño con cámaras de eventos. Para cumplir este objetivo, por un lado en el TFG se ha propuesto un sistema innovador de reconocimiento de acciones a partir de datos proporcionados por

Figura 1.2: Se muestra la habitación utilizada para grabar el conjunto de datos que imita una habitación con una cama doble. Colocamos tres cámaras diferentes (Eventos, Profundidad e Infrarrojos) en el techo, justo encima de la cama.

estas cámaras. Por otro lado, se ha construido un dataset compuesto de varias secuencias en escenarios de sueño realistas, participando en las tareas de captura y etiquetado de la información en colaboración con la empresa Bitbrain (Figura 1.2). El carácter novedoso e investigador de la propuesta se ha incluido como parte de un artículo de investigación enviado para su publicación a una conferencia internacional de prestigio.

Al no haber trabajado previamente con cámaras de eventos, la primera tarea abordada en el TFG ha consistido en un estudio bibliográfico y la realización de un curso acerca de las cámaras de eventos impartido por la Universidad Técnica de Berlin Berlin[1], disponible online[2]. Así, se ha aprendido acerca del funcionamiento asíncrono de las cámaras con los cambios de la intensidad de la luz y el formato especial de salida que posee el sensor.

A continuación, se ha desarrollado un algoritmo con una arquitectura en varias etapas. Se parte de un conjunto de datos en crudo, muy ruidosos a causa de las condiciones de baja iluminación. Por lo tanto, la primera etapa del algoritmo construido es realizar un filtrado de los datos, para obtener únicamente los eventos de interés y eliminar el ruido. El siguiente paso es la agrupación de eventos, la extracción de características y la clasificación de acciones. Esto se debe a la innovación que supone aplicar las cámaras de eventos en un nuevo entorno como son los estudios del sueño. Para ello, se ha planteado una nueva

[1] `https://www.tu.berlin/`
[2] `https://sites.google.com/view/guillermogallego/teaching/event-based-robot-vision`

técnica de agrupación de los eventos de manera que mediante un *blob* se extrae una región de interés. Mediante estas regiones de interés se obtienen vectores de características que serán usadas como entrada en el clasificador de acciones. Las acciones de interés son realizar giros en la cama o tocarse la cabeza ya que estas acciones pueden inferir con otros sensores como electrodos en la cabeza y ayudan a los asistentes de laboratorio a intervenir en el experimento si así se requiere. Por último, con la extracción de características implementada se realiza una clasificación basada en acciones que se realizan durante el sueño con ayuda de la librería de Python Sklearn[3].

Con respecto al segundo objetivo, en colaboración con la empresa Bitbrain[4], se han realizado grabaciones de personas en características similares a un escenario de sueño real para la generación de un *dataset* realista. Para ello, se han utilizado dos sensores con los que se obtienen distintas imágenes. Se ha participado en la configuración del entorno de trabajo, mostrado en la Figura 1.2, en el proceso de grabación, indicando a los usuarios las acciones que debían realizar en cada instante, y en el procesado y etiquetado de la información grabada, de tal forma que pueda utilizarse en el algoritmo de clasificación. De esta forma, gracias a los datos obtenidos, se ha podido evaluar el algoritmo en un escenario real, analizando las capacidades del mismo y valorando la calidad de los datos obtenidos.

Entre las herramientas utilizadas en el TFG se encuentran las siguientes:

- Cámara de eventos DVXplorer camera 640 x 480 resolution
- Cámara de infrarrojos ELP HD Digital Camera
- Lenguaje de programación Python para el desarrollo de algoritmos integrado en un entorno de miniconda y configurado en la herramienta de desarrollo Spyder
- Repositorio de código en Github para mantener un historial de versiones
- Proyecto de Latex en Overleaf para la redacción del presente informe

1.2. Organización de la memoria

La organización de la memoria es la siguiente:

- En el capítulo 1 se pone en contexto al lector. También se exponen los objetivos y se plantea el problema.
- En el capítulo 2 se detalla el contexto del problema y el estado del arte del mismo.

[3] `https://scikit-learn.org/stable/`
[4] `https://www.bitbrain.com/es`

- En el capítulo 3 se expone la metodología empleada en el problema principal.
- En el capítulo 4 se narra la aplicación del problema en un entorno real.
- En el capítulo 5 se exponen los resultados obtenidos en el estudio
- En el capítulo 6 se detallan las conclusiones del estudio y se sugieren futuras líneas de trabajo relacionadas con este estudio.
- Finalmente, se muestra la bibliografía más importante.

En 1.3 se incluye el diagrama de Gantt con los esfuerzos dedicados a cada tarea.

Nombre de la tarea	Diciembre	Enero	Febrero	Marzo	Abril	Mayo	Junio	TOTAL
Curso	17,36	2,73						20,09
Bibliografía			15,31	1,05	3,17			19,53
Implementación		2,42	47,92	47,10	15,12	10,78	1,18	124,52
Dataset			2,33	3,50	4,15	5,14		15,12
Reuniones			1,00	3,33	2,50	4,64	3,33	14,80
Documentación			1,75	2,13		26,96	25,20	56,04
Analizar datos			3,42	21,25	15,64	6,74		47,05
Presentación							15,00	15,00
TOTAL	17,36	5,15	71,73	78,36	40,58	54,26	29,71	312,15

Figura 1.3: Diagrama de Gantt con los esfuerzos dedicados.

Capítulo 2

Estado del arte

2.1. Cámaras de eventos

2.1.1. Funcionamiento de las cámaras de eventos

En el *survey* [7] se explican en detalle las características y distinciones de las cámaras de eventos con respecto a una cámara convencional. A continuación se resumen los detalles de mayor relevancia en el contexto de este TFG.

Las cámaras de eventos son sensores asíncronos, lo que supone un cambio de paradigma en la forma de adquirir información visual en comparación con las cámaras convencionales. Esto se debe a que muestrean la luz según la dinámica de la escena. En lugar de capturar fotogramas a una velocidad fija, miden de forma asíncrona los cambios de intensidad de la luz en cada píxel y emiten una secuencia de eventos que codifican el tiempo, la ubicación y el signo de los cambios de intensidad. Los eventos se producen cuando la luminancia de un píxel o grupo de píxeles cambia por encima de un umbral, lo que permite una respuesta muy rápida a los cambios que surgen en la escena.

Cada píxel memoriza la intensidad logarítmica cada vez que envía un evento, y vigila continuamente si la magnitud de este valor memorizado es suficiente (Figura 2.1). Cuando el cambio supera un umbral, la cámara envía un evento, que se transmite desde el chip con la ubicación x, y, el tiempo t y la polaridad p de 1 bit del cambio (es decir, aumento ("ON") o disminución ("OFF") de la luminosidad). Por lo tanto, la salida de una cámara de eventos es una secuencia de "eventos" individuales, $e_k = (x_k, y_k, t_k, p_k)$, con una tasa de datos variable.

Comparadas con otras cámaras tradicionales, las cámaras de eventos resultan atractivas por motivos como: alta resolución temporal (del orden de μs), muy alto rango dinámico (140 dB vs. 60 dB), bajo consumo de energía y alto ancho de banda de píxeles (del orden

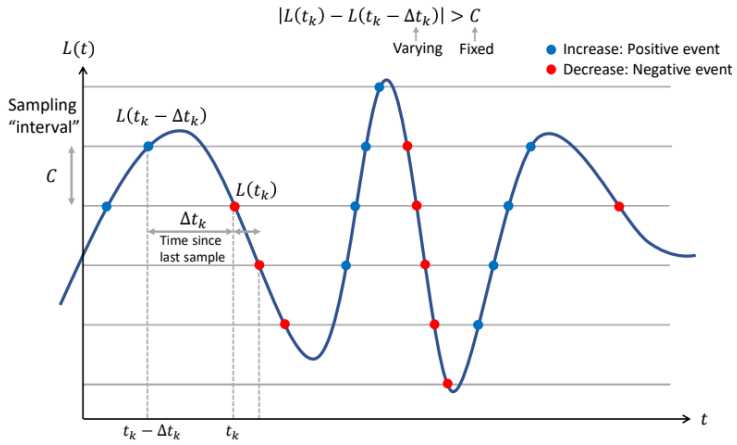


Figura 2.1: Magnitud de la señal con los umbrales de producción de eventos [1]

de kHz). Por lo tanto, las cámaras de eventos tienen un gran potencial para aplicaciones robóticas y en escenarios difíciles para las cámaras estándar, como la alta velocidad y el alto rango dinámico. Sin embargo, los nuevos métodos requieren procesar la salida no convencional de estos sensores para desbloquear su potencial.

Una de las cuestiones clave del cambio de paradigma que plantean las cámaras de eventos es cómo extraer información de los datos de eventos para cumplir una tarea determinada. Se trata de una tarea compleja, ya que la respuesta depende de la aplicación e impulsa el diseño algorítmico en función del contexto.

El aspecto temporal, especialmente la latencia, desempeñan un papel crucial en la forma que se procesan los eventos. Según el número de eventos que se procesen simultáneamente, pueden distinguirse dos categorías de algoritmos: (i) métodos que operan por evento, en los que el estado del sistema puede cambiar con la llegada de un solo evento, y (ii) métodos que operan sobre grupos o paquetes de eventos que introducen cierta latencia. Sin tener en cuenta las consideraciones de latencia, los métodos basados en grupos (es decir, ventanas temporales) de eventos pueden seguir proporcionando una actualización del estado a la llegada de cada suceso si la ventana se desliza por cada evento.

La detección y el seguimiento de características en el plano es fundamental para muchas tareas de visión, como la odometría visual, la segmentación de objetos y la comprensión de escenas. Las cámaras de eventos permiten realizar un seguimiento asíncrono adaptadas a la dinámica de la escena, con baja latencia, alto rango dinámico y bajo consumo. Para ello, los métodos desarrollados deben tener en cuenta las características espacio temporales y fotométricas únicas de la señal visual. Uno de los desafíos para resolver estos problemas con eventos es superar la variación de la escena causada por la dependencia del movimiento. El seguimiento requiere el establecimiento de correspondencias entre eventos (o características construidas a partir de eventos) en diferentes momentos. El segundo reto principal consiste en lidiar con el ruido del sensor y el posible desorden de eventos causado por el movimiento de la cámara.

2.1.2. Representación del conjunto de eventos

Los eventos que se obtienen a menudo se transforman (Figura 2.2) en representaciones alternativas que facilitan la extracción significativa de información ("características") para resolver una tarea determinada.

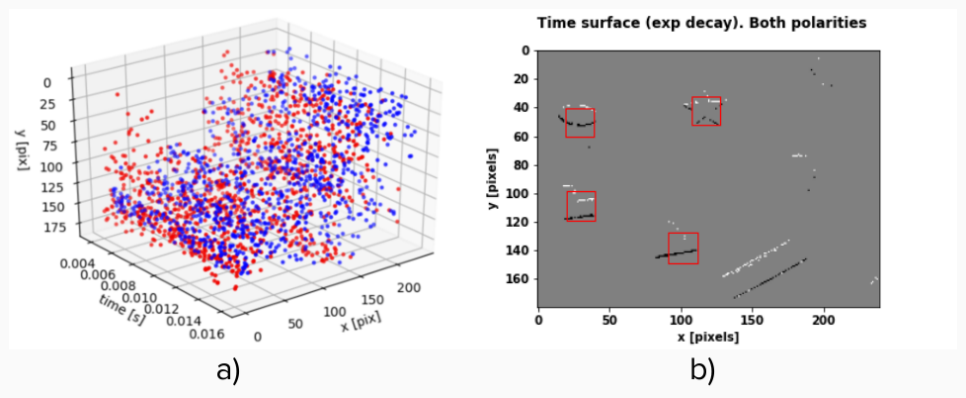


Figura 2.2: Representación de eventos. (a) eventos en el espacio-tiempo coloreados de acuerdo a su polaridad (positivos en azul, negativos en rojo). (b) frame de eventos

Los eventos individuales, e_k, son utilizados por métodos de procesamiento evento a evento, como los filtros probabilísticos y las redes neuronales con picos [7]. En el enfoque propuesto, un suceso se convierte en un punto en el espacio 3D y los flujos de sucesos forman nubes de eventos de eventos espacio-temporales en 3D [8].

En el sistema convencional de clasificación basado en cámaras de eventos, los flujos de eventos de dividen en múltiples segmentos para la extracción de características (*frames*). La segmentación temporal divide los eventos por intervalos de tiempo fijos, mientras que la segmentación suave obtiene segmentos predefinidos [8].

Por otro lado, en [9] se propone un algoritmo que se inspira en el enfoque de desplazamiento medio que implementa una agrupación continua de eventos asíncronos y el seguimiento de los *clusters* para la extracción de características. El algoritmo procesa cada evento a medida que se recibe sin almacenamiento en *buffer* de datos. Cada nuevo evento puede asignarse a un *cluster* en función de un criterio de distancia y se utiliza para actualizar el peso del *cluster* y posición central para el seguimiento.

2.2. Reconocimiento de acciones

La detección de objetivos humanos y el reconocimiento de comportamientos son los problemas básicos más comunes en las tareas de seguridad. Debido a la iniciativa subjetiva y a las características no rígidas de los seres humanos, se trata de un tema de investigación

complejo en el campo de la visión por computador. La detección humana detecta y localiza un cuerpo humano a partir de un vídeo, y el reconocimiento del comportamiento identifica y analiza el comportamiento temporal.

Figura 2.3: Ejemplos de la imagen original, el flujo óptico y el campo de flujo en cuatro canales calculado por [2]

Un diagrama de flujo típico de reconocimiento de acciones suele contener dos componentes principales a representación de la acción y la clasificación de la acción. El componente de representación de la acción básicamente convierte un vídeo de acción en un vector de características o una serie de vectores y el componente de clasificación de acciones infiere una etiqueta de acción a partir del vector. Recientemente, las redes neuronales profundas fusionan estos dos componentes en un marco unificado entrenable de extremo a extremo, que mejora aún más la clasificación de clasificación en general [10].

2.2.1. Reconocimiento de acciones durante el sueño

El reconocimiento de acciones en el sueño permite a los investigadores estudiar y comprender mejor los procesos cognitivos y neurológicos asociados con el sueño. Estudiar las acciones y movimientos que ocurren durante el sueño puede ayudar a desentrañar los misterios de los sueños y su función en el procesamiento de la memoria, la consolidación del aprendizaje y otros aspectos de la cognición humana. También puede ser útil en el diagnóstico de trastornos del sueño, como el sonambulismo y el trastorno del comportamiento del sueño REM. Estos trastornos se caracterizan por acciones anormales durante el sueño, como caminar, hablar o realizar movimientos violentos.

En [3] realiza una breve descripción sobre las características del sueño y una ligera introducción a las fases del sueño. El sueño se define a partir de la impresión combinada del electroencefalograma (EEG), el electrooculograma (EOG) y el electromiograma (EMG). El polisomnograma resultante identifica las etapas del sueño.

Hasta ahora, el reconocimiento de acciones en estudios del sueño tenía como objetivo reconocer las distintas fases de sueño. Con esta nueva aproximación de las cámaras de eventos a estudios del sueño se pretende realizar un seguimiento de las acciones que realiza el paciente a lo largo de la noche. Este reconocimiento de acciones permite a investigadores

y médicos detectar trastornos del sueño como apnea del sueño o síndrome de las piernas inquietas.

El fenómeno de que el sueño preserva la memoria del olvido está reconocido desde hace tiempo. Inicialmente, el efecto se explicaba por el efecto protector del sueño sobre los recuerdos recién aún lábiles, impidiendo que sean sobrescritos por nueva nueva información (interferencia retroactiva). Las explicaciones más recientes que consideran el mecanismo neuronal subyacente y, en particular, la repetición neuronal de los recuerdos durante el sueño, hacen hincapié en un papel activo del sueño en la consolidación de la memoria [11].

Mediante el uso de un algoritmo de reconocimiento de la conducta de dormir, se puede detectar la condición de dormir del personal de servicio, y se pueden tomar las medidas preventivas correspondientes para reducir la ocurrencia de accidentes. La adopción de un algoritmo de reconocimiento del comportamiento durante el sueño puede mostrar digitalmente el estado de trabajo del personal de guardia, mejorar la eficacia de la supervisión y evitar que los comportamientos peligrosos causados por los factores subjetivos del personal de guardia no se detecten y detengan a tiempo, lo que tendría graves consecuencias [12].

Capítulo 3

Metodología

Para resolver el problema, se ha propuesto e implementado un algoritmo con una arquitectura en varias capas. En la Figura 3.1 se muestra el diagrama de flujo del algoritmo. Primero se realiza un filtrado de datos, obteniendo así únicamente los eventos de interés. A continuación, se realiza una agrupación eventos a lo largo del tiempo, obteniendo así una región de interés. Con las agrupaciones de eventos se obtienen unos vectores de características que son usadas para realizar la clasificación de acciones mediante diferentes algoritmos de aprendizaje automático. A continuación se describe con más profundidad cada etapa del algoritmo. En 3.2 se muestran ejemplos del flujo del algoritmo construido.

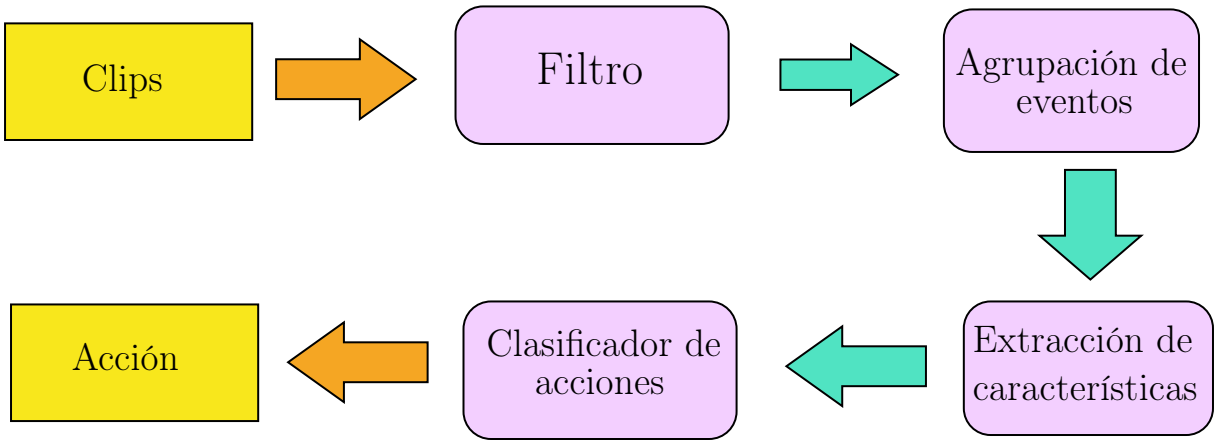


Figura 3.1: Diagrama del algoritmo

3.1. Filtrado de datos

Las cámaras de eventos producen ruido causado por la apertura del objetivo, ajustes de parámetros de la propia cámara o por agentes externos como puede ser la luz artificial.

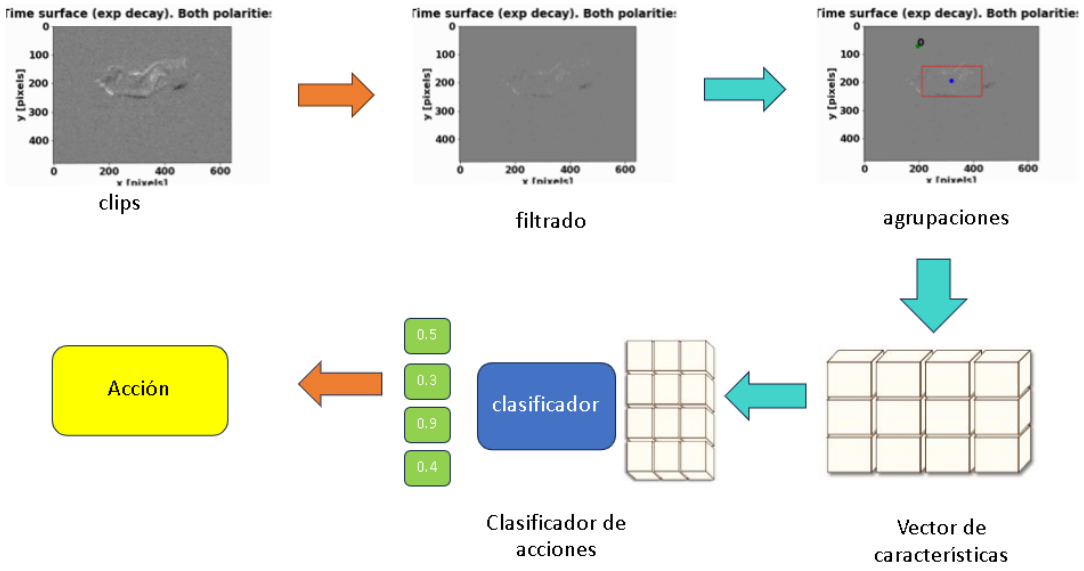


Figura 3.2: Flujo del algoritmo con ejemplos

Los estudios del sueño están caracterizados por unas condiciones de muy baja iluminación o con ausencia de ella. Por ello, en muchas ocasiones se modifican parámetros de la cámara como puede ser la apertura del objetivo y el aumento de la sensibilidad de la misma para que el sensor detecte suficientes eventos durante la grabación.

Grabar con una alta sensibilidad de la cámara produce un mayor número de eventos causados por el movimiento del sujeto, pero también produce más eventos causados por el ruido. En la Imagen 3.3 se muestra un ejemplo de los datos "en crudo". Se puede observar como ocurre ruido de sal y pimienta causado por diversos eventos que ocurren a lo largo de toda la imagen. También se muestra y compara con los datos procesados.

Una de las causas del ruido, es la sobre representación de eventos en espacios de tiempo muy pequeños. Esto produce ráfagas de alta frecuencia de eventos en determinados píxeles. Para evitar estas ráfagas se ha diseñado un filtro temporal que acota la diferencia mínima de tiempo entre dos eventos en cada píxel. Esto ayuda a eliminar los eventos de alta frecuencia producidos por el sensor.

El filtro impone un tiempo mínimo entre dos eventos consecutivos, T_{ref}, que aparecen en el mismo píxel. Sea el evento actual recibido por la cámara $e = (x, y, t, p)$ y T^{last} la marca de tiempo del último evento no filtrado en el píxel (x, y), independientemente de la polaridad. El filtro elimina todos los eventos en (x, y) que cumplan

$$t - T^{last} \leq T_{ref}. \tag{3.1}$$

En segundo lugar, el filtro también se encarga de eliminar los eventos que no ocurren en el área de la cama. Para ello, se obtienen las coordenadas de las esquinas de la cama y para cada evento $e = (x, y, t, p)$ se comprueba que su posición esté dentro del área de la

cama.

La última capa del filtro es un *Nearest Neighbor Filter*. Cuando un evento ocurre en el instante t y $e = (x, y, t, p)$ el filtro considera la marca temporal del vecino cercano más reciente $T_{NNb(x,y)}$ excluyendo al píxel actual y que esté en una distancia de 1 píxel considerando 8-vecindad. El evento pasa el filtro si la diferencia de tiempo entre t y $T_{NNb(x,y)}$ es menos que un determinado umbral T_{NNb}

$$t - T_{NNb(x,y)} < T_{NNb}. \tag{3.2}$$

Si la diferencia de tiempo es mayor que el umbral, entonces es más probable que se trate de un evento aislado causado por el ruido y es eliminado por este filtro.

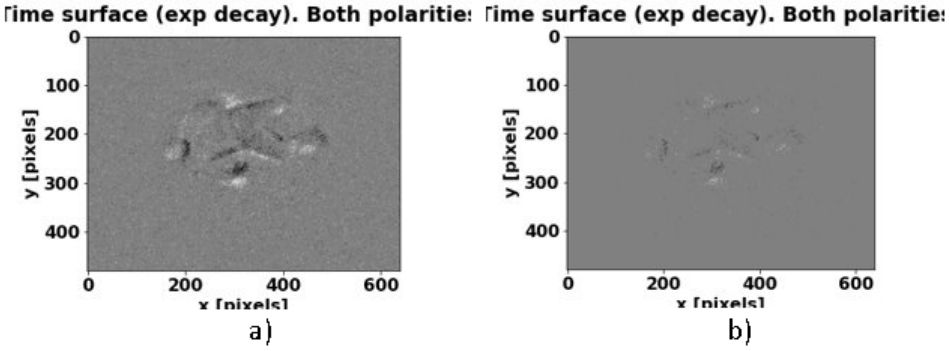


Figura 3.3: a) Datos grabados en "crudo". b) Datos filtrados

3.2. Agrupación de eventos

Una vez que se han filtrado los datos, el siguiente paso consiste en agrupar eventos extrayendo regiones de interés en la escena. Con este proceso se obtiene una representación más compacta de la información que se puede utilizar en las siguientes fases del algoritmo para realizar clasificación de acciones.

Esta componente del algoritmo se inspira en el enfoque *mean shift* e implementa agrupaciones continuas de los eventos y seguimiento de las agrupaciones. El método procesa cada evento que se recibe sin almacenarlos, lo que es especialmente importante en sistemas de bajo coste y pocos recursos en memoria. Además permite el procesamiento de los eventos en tiempo real.

Para ello, se define el concepto de *cluster*, caracterizado por su centro $(c_x(t),\ c_y(t))$, su tamaño $(r_x(t),\ r_y(t))$ y su peso $\omega(t)$. Estas características poseen una componente temporal ya que se modifican a lo largo del tiempo y se hace un seguimiento de ellas.

El procedimiento de generación y seguimiento consiste en asignar cada nuevo evento a un *cluster* en base a un criterio de distancia. Cuando se recibe un nuevo evento, se

busca entre los *clusters* existentes si el nuevo evento $e = (x, y, t, p)$ se encuentra a una distancia menor de R_K del centro de alguno de los *clusters* existentes. Se asume que los *clusters* tienen forma rectangular, por lo que la condición de pertenencia a alguna de las agrupaciones tiene que cumplir las siguientes condiciones (Figura 3.4)

$$|c_x(t) - x| < r_{Kx} \text{ y } |c_y(t) - y| < r_{Ky}. \tag{3.3}$$

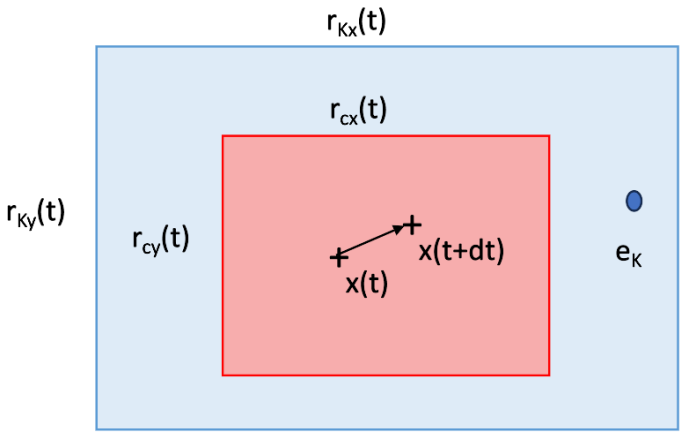


Figura 3.4: Condición de pertenencia de un evento a un *cluster*

En el caso de que ambas condiciones se cumplan para varios *clusters*, se elige el de mayor peso. Si no se encuentra un *cluster* al que pertenece dicho evento, se ignora el evento.

Se define r_{Kx} y r_{Ky} la distancia de búsqueda en el espacio para un determinado *cluster* como un múltiplo de su tamaño $r_x(t)$ y $r_y(t)$

$$r_{Kx}(t) = \text{mín}(R_{\text{máx}}, r_x(t)R_{multiple}) \tag{3.4}$$

$$r_{Ky}(t) = \text{mín}(R_{\text{máx}}, r_y(t)R_{multiple}), \tag{3.5}$$

dónde $R_{multiple}$ y $R_{\text{máx}}$ son parámetros del algoritmo.

Cuando un evento se asocia con un determinado *cluster*, se actualizan sus propiedades. Las nuevas coordenadas del centro del *cluster* $c_x(t + dt)$ y $c_y(t + dt)$ se calculan como

$$c_x(t + dt) = \alpha c_x(t) + (1 - \alpha)x \tag{3.6}$$

y

$$c_y(t + dt) = \alpha c_y(t) + (1 - \alpha)y, \tag{3.7}$$

dónde $0 < \alpha < 1$ es un parámetro del algoritmo y dt es la diferencia de tiempo desde el tiempo actual hasta el último evento que fue asignado al *cluster*.

También se actualiza su tamaño,

$$r_x(t+dt) = \text{máx}(R_{\text{mín}}, \alpha r_x(t) + (1-\alpha)(c_x(t) - x)) \tag{3.8}$$

$$r_y(t+dt) = \text{máx}(R_{\text{mín}}, \alpha r_y(t) + (1-\alpha)(c_y(t) - y)). \tag{3.9}$$

En este caso R_{min} es un parámetro del algoritmo y la condición de máximo asegura que el tamaño del *cluster* se mantenga en ciertos límites. El *cluster* inicialmente es cuadrado y tiene tamaño R_{min} en ambas direcciones.

Para el peso de los *clusters* $\omega(t)$ se utiliza la cantidad de eventos que hay en él. El peso de todos los *clusters* existentes en un determinado instante se actualiza cada vez que aparece un nuevo evento de la siguiente manera:

$$\omega(t) = \alpha\omega(t) + (1-\alpha)d \tag{3.10}$$

d tiene valor 1 si el evento se ha añadido a dicho *cluster* y d tendrá valor 0 para los *clusters* dónde no se ha añadido el evento. De esta manera se puede disminuir el peso de los *clusters* a los que se asignan pocos eventos (y que probablemente han sido creados a causa del ruido que se produce en los datos).

Para visualizar los *clusters* también se ha implementado una función de visualización mediante segmentación temporal fija.

Una vez se han acumulado los eventos correspondientes al segmento de tiempo, se muestra el último evento ocurrido en cada píxel, con su polaridad correspondiente. Además, en la visualización se muestra una caja que corresponde con la agrupación de eventos realizada por el algoritmo (Figura 3.5).

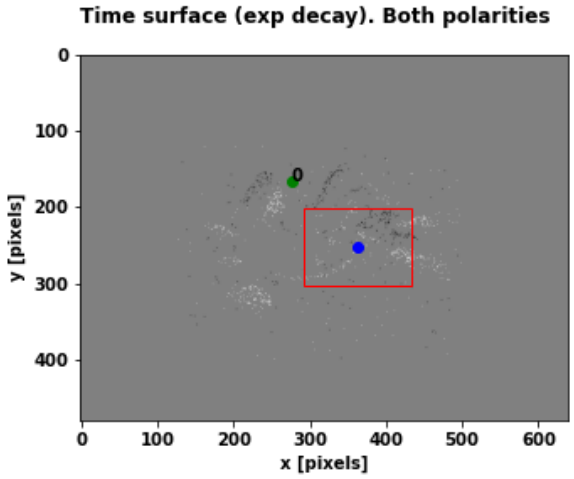


Figura 3.5: Representación de eventos.

3.3. Extracción de características

El objetivo final del presente proyecto es realizar reconocimiento de acciones que ocurren durante el sueño. Para ello, se han considerado como características las trayectorias y propiedades de los *clusters* que se generan cuándo ocurre una acción.

La trayectoria de los *clusters* se obtiene guardando en una estructura de datos de tipo diccionario su centro, tamaño y peso. Cada cierto tiempo, se realizan comprobaciones para eliminar los *clusters* con peso pequeño y se almacena la trayectoria del resto. El algoritmo fuerza que siempre haya al menos un *cluster* activo durante todo el proceso de reconocimiento de la acción.

El siguiente paso consiste en transformar la secuencia de valores de los parámetros del *cluster* en un vector de características de tamaño fijo, de tal forma que se puedan utilizar en el algoritmo de clasificación. Para esta etapa se plantean dos posibles soluciones, ajustar los datos de un clúster al tamaño deseado o utilizar un método de ventana deslizante de tamaño fijo sobre el *cluster*.

En el primer método se homogeneiza la cantidad de muestras a tamaño fijo N. Para realizar esta homogeneización hay que tener en cuenta dos posibles escenarios: en la trayectoria hay más de N muestras o en la trayectoria hay menos de N muestras. En el primer caso se realiza un muestreo de manera que se toman las muestras lo más equidistantes posibles (Figura 3.6). En el segundo caso, si hay menos de N muestras, se realiza una interpolación entre valores hasta lograr las N muestras necesarias.

El segundo método de homogeneización se basa en el uso de una ventana deslizante. Este tipo de homogeneización toma todas las muestras en ventanas de datos. Cada ventana se compone de un 20 % de datos que ya se incluyeron en la ventana anterior y un 80 % de datos nuevos. De esta manera se pueden obtener trayectorias completas de clips más largos sin perder información. Esta técnica es interesante para grabaciones que no se encuentran segmentadas por acciones, pudiendo así reconocer distintas acciones en una misma grabación, aunque asume una duración fija para todas las acciones que se pretende reconocer.

Por último se realiza una normalización de los datos, dividiendo por la altura y la anchura de la cama para que todos tengan valores entre 0 y 1.

3.4. Clasificador de acciones

El objetivo del proyecto es realizar un clasificador de acciones. Para ello, se han tenido en cuenta dos métodos tradicionales de clasificación con múltiples clases.

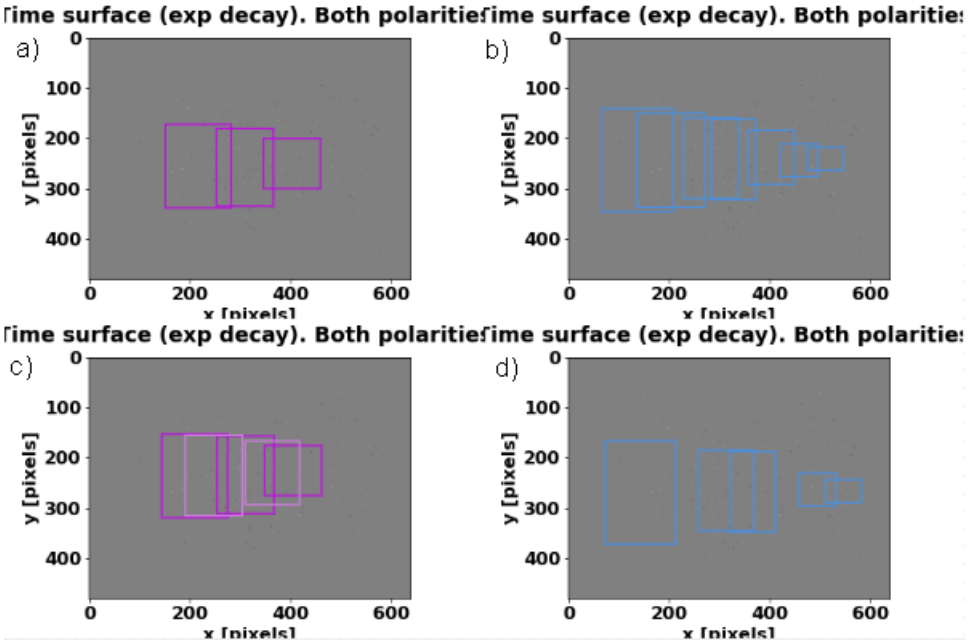


Figura 3.6: a) Trayectoria de clusters en la que hay menos de N ($N = 5$) muestras. b) Trayectoria de clusters en la que hay más de N ($N = 5$) muestras. c) Interpolación entre los clusters para obtener el número adecuado de muestras. d) Se ignoran algunos datos para obtener el número adecuado de muestras.

Se ha decidido usar métodos de clasificación tradicional debido a la cantidad de datos de la que se dispone. El *dataset* construido dispone de 14 sujetos en 3 configuraciones distintas. Este volumen de datos se considera adecuado para técnicas clásicas, pero insuficiente para poder realizar un entrenamiento de redes neuronales profundas que no produzca sobreajuste.

En cuanto a las acciones que se quieren clasificar, son acciones que ocurren típicamente durante el sueño y que proporcionan a los médicos e investigadores una información crucial sobre trastornos del sueño en determinados pacientes. Se distinguen acciones en dos niveles, por un lado, se encuentran las etiquetas *coarse* compuesto por 6 acciones a reconocer. Además, como parte del artículo de investigación enviado y la construcción del *dataset* se ha construido un nivel de etiquetas más detallado, referenciado como *fine-grained* compuesto por 10 acciones.
Las acciones a reconocer son:

- *HeadMove*: ajustes o reposicionamiento de la cabeza
- *Hands2Face*: tocar o colocar las manos en la cara o la cabeza
- *RollLeft* o *Roll*: rodar hacia el lado izquierdo de la cama
- *RollRight* o *Roll*: rodar hacia el lado derecho de la cama

- *LegsShake*: sacudir o mover las piernas
- *ArmsShake*: agitar o mover los brazos
- *LieLeft* o *Quiet*: estar tumbado sobre el lado izquierdo
- *LieRight* o *Quiet*: estar tumbado sobre el lado derecho
- *LieUp* o *Quiet*: estar tumbado boca arriba
- *LieDown* o *Quiet*: estar tumbado boca abajo

En la tabla 3.1 y en la figura 3.7 se encuentran las diferentes acciones que se van a reconocer y las equivalencias entre las etiquetas de los dos niveles de acciones.

Tabla 3.1: Etiqueta de las acciones a reconocer

Fine-grained	Coarse
HeadMove	*HeadMove*
Hands2Face	*Hands2Face*
RollLeft	*Roll*
RollRight	*Roll*
LegsShake	*LegsShake*
ArmsShake	*ArmsShake*
LieLeft	*Quiet*
LieRight	*Quiet*
LieUp	*Quiet*
LieDown	*Quiet*

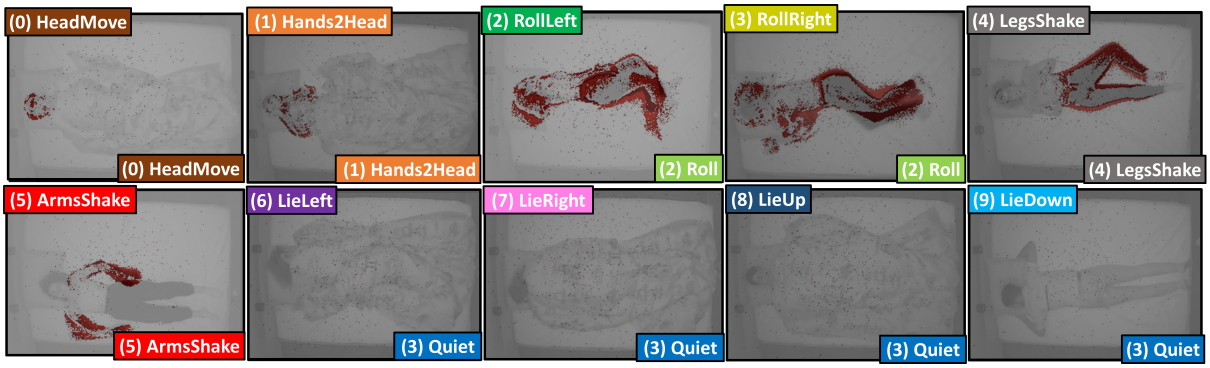


Figura 3.7: La matriz muestra un trazado por cada etiqueta *fine-grained* (no usadas en este trabajo). Su nombre aparece en la esquina superior izquierda y el de la etiqueta *coarse* correspondiente en la esquina inferior derecha. Cada gráfico corresponde a un fotograma de sucesos que se muestra encima del fotograma de infrarrojos correspondiente.

3.4.1. K Nearest Neighbors

La primera de las técnicas consideradas para reconocer las acciones es un clasificador *K Nearest Neigbors*.

K Nearest Neighbors es un algoritmo de aprendizaje supervisado que se utiliza en problemas de clasificación y regresión. La idea principal detrás de *KNN* es que los puntos de datos que están próximos entre sí son más propensos a compartir una etiqueta de clase o un valor objetivo similar.

En el método *KNN*, se define un valor positivo entero k, que representa el número de puntos de datos vecinos que se considerarán para determinar la etiqueta de clase o el valor objetivo de un punto de datos de interés. Una vez que se define k, se busca en el conjunto de datos el conjunto de k puntos de datos más cercanos al punto de datos de interés, utilizando una métrica de distancia. A continuación, se utiliza la etiqueta de la clase más común entre los k vecinos más cercanos para determinar la etiqueta de clase del punto de datos de interés.

En este caso se ha utilizado *KNN* con un parámetro de peso equivalente a la inversa de la distancia. De esta manera, los vecinos más cercanos tendrán una mayor influencia en un dato determinado a la hora de asignarle una clase.

Se ha utilizado la implementación de *Sklearn*[1] de *K Nearest Neighbors*, Figura 3.8.

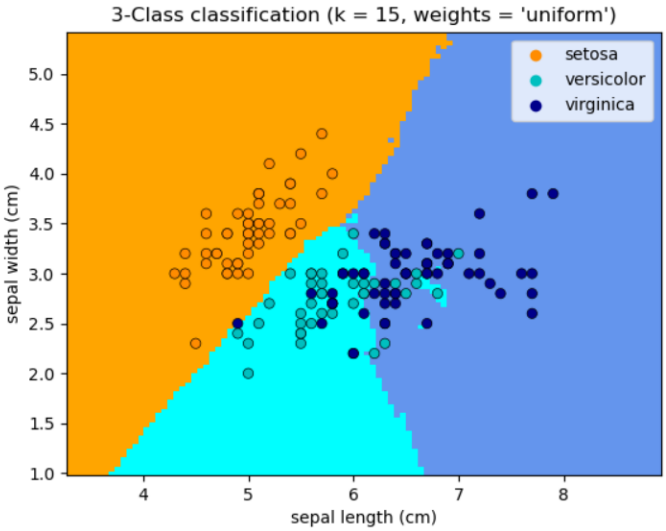


Figura 3.8: Ejemplo de KNN en Sklearn [2]

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html`

[2] `https://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html#sphx-glr-auto-examples-neighbors-plot-classification-py`

3.4.2. Support Vector Classification

La otra técnica estudiadad en el TFG es un clasificador *Support Vector Machine* (SVM).

SVM es un algoritmo de aprendizaje supervisado utilizado para realizar tareas de clasificación y regresión. Está diseñado para asignar valores de clase a nuevos puntos de datos basados en un conjunto de entrenamiento de puntos de datos que tienen valores de clase etiquetados.

El algoritmo SVM trabaja encontrando el conjunto de hiper-planos que mejor divide los puntos de datos en las distintas clases en el espacio de características. Un hiper-plano es una frontera de decisión lineal que se utiliza para separar los puntos de datos en diferentes clases. El objetivo es maximizar la distancia entre los puntos de datos de cada clase y la frontera de decisión o hiper-plano para mejorar la precisión de la clasificación.

En este caso, se ha utilizado SVC, un clasificador basado en SVM. En cuanto a parámetros, se ha considerado un kernel polinómico, ya que ajusta mejor las características de los datos, y un sistema de pesos balanceado, ya que utiliza los valores de la salida para ajustar automáticamente las ponderaciones de forma inversamente proporcional a las frecuencias de clase en los datos de entrada.

Se ha utilizado la implementación de *Sklearn* [3] de *Support Vector Classification*, Figura 3.9.

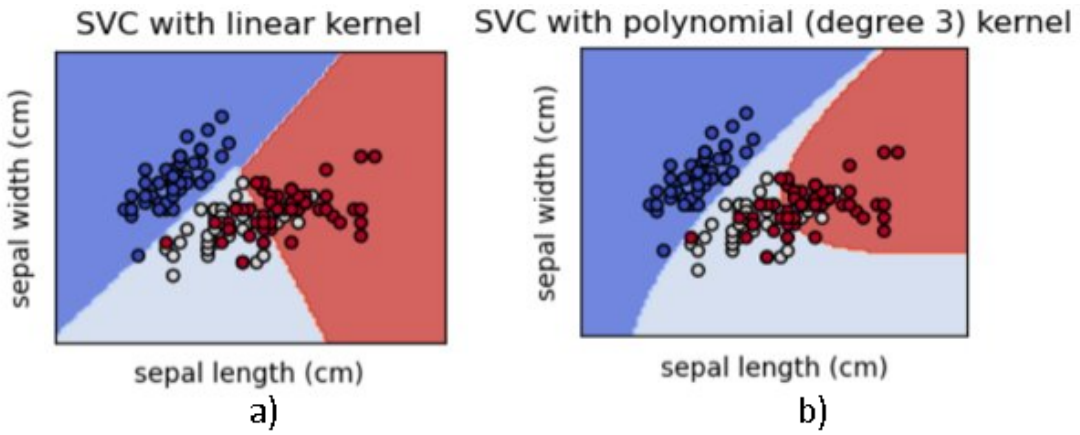


Figura 3.9: Ejemplo de SVM en Sklearn [4]

[3] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html
[4] https://scikit-learn.org/stable/modules/svm.html

Capítulo 4

Evaluación en un entorno real

Debido al carácter innovador del presente trabajo, se ha realizado una adquisición de datos propia, en un escenario realista de estudios del sueño. Esto se debe a que a pesar de que existen conjuntos de datos típicos de reconocimiento de acciones, no existe un conjunto de datos que se ajuste a las necesidades del estudio que se realiza y por lo tanto, se ha decidido realizar una captura propia. Los datos adquiridos son acordes al objeto de estudio, estudios del sueño con cámaras de eventos.

El *dataset* construido se ha adquirido en un escenario de sueño realista, utilizado para otro tipo de estudios médicos sobre trastornos del sueño (en los laboratorios de la empresa Bitbrain[1]), pero con sujetos que no son pacientes reales. El objetivo es conseguir una cantidad significativa de movimientos que ocurren típicamente durante el sueño para permitir la evaluación de las capacidades de la cámara de eventos en este tipo de situaciones.

4.1. Adquisición de datos

El conjunto de datos consta de 42 grabaciones (14 participantes, cada uno grabado en 3 configuraciones diferentes), cada una de las cuales dura unos 3 minutos aproximadamente. Cada configuración considera diferentes niveles de visibilidad del sujeto (Figura 4.1):

- **Configuración 1:** El sujeto se encuentra cubierto por un edredón en condiciones de oscuridad total ($\leq$ 0,1 lux).
- **Configuración 2:** El sujeto se encuentra cubierto por un edredón y hay iluminación parcial (0,2 lux).

[1] `https://www.bitbrain.com/es`

- **Configuración 3:** El sujeto se encuentra descubierto en condiciones de oscuridad total ($\leq$ 0,1 lux).

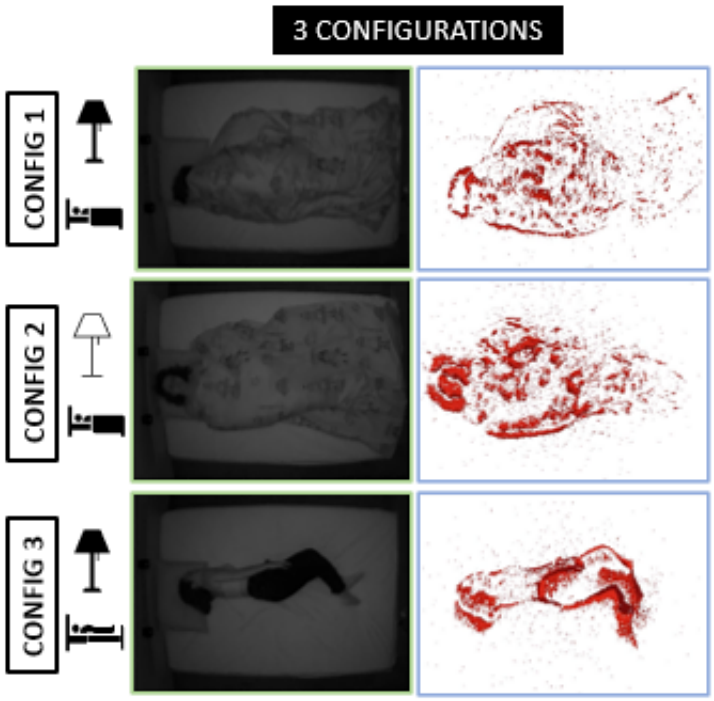


Figura 4.1: Configuraciones de los sujetos en las grabaciones

La configuración de oscuridad total consiste en un dormitorio con todas las luces apagadas, la puerta y la ventana cerradas, y unas persianas que impiden la entrada de luz. La configuración de oscuridad parcial incluye una lamparita encendida en el suelo, lejos de la cabeza del sujeto pero que añade una pequeña iluminación a toda la escena.

Los experimentos se han grabado con dos cámaras diferentes: Eventos (cámara DVXplorer, resolución 640 x 480) e Infrarrojos (cámara digital ELP HD). Las dos cámaras se colocaron juntas, sujetas a una barra metálica, orientadas hacia abajo y situadas a 2 metros por encima del centro de la cama, lo que garantiza una observación completa de los movimientos de los participantes (Figura 1.2). Cabe destacar que en el *setup* se muestra una cámara de profundidad que se utilizó en unas grabaciones de prueba que finalmente fueron descartadas del *dataset*. Se decidió descartar los datos adquiridos con esta cámara debido al ruido que introduce en las demás debido al patrón infrarrojo utilizado por la cámara de profundidad.

Se realizaron grabaciones continuas en las que se indica a los sujetos que realicen distintas acciones o que se mantengan tumbados.

Tabla 4.1: Detalles de los sujetos: Sexo (M: Hombre, F: Mujer), Edad (1: 20-30, 2: 30-40, 3: 40-50), Altura (1: 160-170cm, 2: 170-180cm, 3: 180-190cm).

Subject	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
Sex	M	F	M	M	M	M	F	F	M	M	F	M	M	M
Height	1	1	2	2	2	2	2	1	1	3	2	2	3	2
Age	3	3	2	2	3	1	1	1	1	1	1	3	2	1

4.1.1. Participantes en el experimento

El conjunto de datos registrados incluye datos de un grupo diverso de 14 participantes, de los cuales 4 son mujeres y 10 hombres. La edad de los participantes oscila entre los veinte y los cuarenta años, lo que refleja un amplio espectro demográfico. Sus estaturas oscilan entre los 150 cm y los 190 cm, lo que contribuye a la diversidad del conjunto de datos. Los detalles de cada sujeto pueden observarse en la tabla.

Los participantes son voluntarios. Todos los sujetos fueron advertidos previamente sobre el consentimiento de grabación y se les informó acerca de los objetivos del estudio.

Se instruyó cuidadosamente a los participantes sobre las principales acciones presentes en los trastornos del sueño y el objetivo de nuestro estudio. Aunque se orientó a los participantes sobre los movimientos que debían realizar, no se les indicó específicamente que debían realizar cada movimiento de una manera predefinida. Este enfoque les permitió la libertad de ejecutar las acciones de forma natural, garantizando la diversidad y capturando una amplia gama de variaciones en el conjunto de datos.

4.1.2. Etiquetado de datos

El etiquetado de datos consiste en separar los instantes temporales grabados con ambas cámaras asignándoles la acción a la que corresponden.

Para obtener las distintas secuencias de acciones es necesario sincronizar ambas cámaras y de esta manera obtener sincronía entre la cámara de Infrarrojos y la representación de los eventos.

La sincronización de ambas cámaras, la de eventos y la de infrarrojos, es esencial para un análisis preciso de los datos. Mientras que las cámaras de eventos guardan las marcas de tiempo en una referencia global, la cámara de infrarrojos registra las marcas de tiempo en un marco de referencia local, relativo al inicio del vídeo.

Para lograr la sincronización, se ha desarrollado un programa especializado. Este programa detecta automáticamente el momento en que se apaga la luz. En las grabaciones con infrarrojos, este momento se indica mediante una disminución sustancial de los valo-

res medios de los píxeles. Del mismo modo, en las grabaciones de la cámara de eventos, el apagado de la luz se identifica por una disminución en el número de eventos que se generan.

Al detectar con precisión este punto de transición, el programa alinea las marcas de tiempo de ambas cámaras. A continuación, ambas grabaciones se recortan para que comiencen en este preciso momento y, por tanto, contengan la misma información.

Una vez ambas cámaras se encuentran sincronizadas, se procesan *frame a frame* anotando el inicio y final de cada acción. Una vez se tienen todas las grabaciones anotadas, se cortan tanto las secuencias de Infrarrojos como las secuencias de eventos separando los distintos *clips* de acciones.

Capítulo 5

Resultados

En este capítulo se presentan los distintos resultados obtenidos en el trabajo.

5.1. Diseño de experimentos

El conjunto de datos se ha dividido en tres subconjuntos: *Train*, *Validation* y *Test*. Esta estructura de separación de los datos es la realizada típicamente en los experimentos de clasificación. Para ello, se han tomado como datos de entrenamiento 9 de los 14 sujetos, como validación 1 de los 14 sujetos y como test 4 sujetos. La separación por sujetos se encuentra en la tabla 5.1.

Tabla 5.1: Separación de datos en tres conjuntos: *Train*, *Validation* y *Test*.

Subject	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
Train	X	X	X	X	X	X	X	X		X				
Validation											X			
Test									X			X	X	X

Como parte del experimento, primero se ha realizado un ajuste de hiper-parámetros para las cuatro combinaciones de clasificadores elegidos: kNN con interpolación, SVM con interpolación, kNN con ventana y SVM con ventana. Para el ajuste de los parámetros se han tomado los datos de entrenamiento y validación. Se entrena y evalúa con todas las configuraciones.

En el caso de kNN, el parámetro a ajustar es el número óptimo de vecinos. Por otro lado, en el caso de SVM, se busca el mejor grado del polinomio que ajuste a los datos. Además, entre los parámetros se encuentra el tamaño óptimo de los descriptores. Por lo

tanto, se ajustan a la vez, los parámetros propios del clasificador y el número óptimo de características.

Los experimentos realizados son:

1. Ajuste de hiper-parámetros
2. Encontrar el mejor clasificador
3. Validación cruzada por configuraciones

El *dataset* contiene dos niveles de etiquetas y se realizan los experimentos tanto con las etiquetas *coarse* como con las etiquetas *fine-grained*.

Por otro lado, se ajustan los parámetros utilizados para generar blobs. Comenzando por los filtros, el umbral de tiempo mínimo que se impone entre dos eventos consecutivos que ocurren en el mismo píxel $T_{ref} = 50$ms. En cuanto al filtro de vecinos, el tiempo umbral que se utiliza es $T_{NNb} = 5$ms. Del mismo modo, el hiper-parámetro que ajusta las características en función de los eventos que se reciben $\alpha = 0{,}9$. Además, se ajustan las condiciones de máximo y mínimo de los *clusters*, $R_{\text{mín}} = 50$ y $R_{\text{máx}} = 130$. Por último, se establece $R_{multiple} = 3$ como factor multiplicador que establece el espacio de búsqueda al triple de su tamaño.

5.2. Métricas

La métrica utilizada para evaluar los distintos clasificadores es *accuracy*. El *accuracy* mide la tasa de aciertos por el total de datos contemplados.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\text{correct classifications}}{\text{all classifications}} \tag{5.1}$$

A continuación de presentan las medidas obtenidas con los distintos clasificadores.

5.3. Resultados

5.3.1. Ajuste de hiper-parámetros

Se han ajustado los hiper-parámetros de los cuatro clasificadores para ambos niveles de etiquetas.

Como se puede observar en la Figura 5.1 y en la Figura 5.2, con 100 muestras de obtienen valores de *accuracy* más elevados para todos los clasificadores. Por lo tanto, se fija el tamaño del descriptor a 100 muestras. Esto se debe a que cuantas más muestras se toman, mejor funciona el clasificador. Por otro lado, si se toman muestras excesivas puede producirse sobreajuste. Es por eso, que se han tomado estos tres valores, y el mejor de ellos es el más alto. Además, se ajusta para cada clasificador su hiper-parámetro óptimo. En cada gráfica corresponde con su valor más alto.

En la Tabla 5.2 se muestra un resumen de los hiper-parámetros obtenidos.

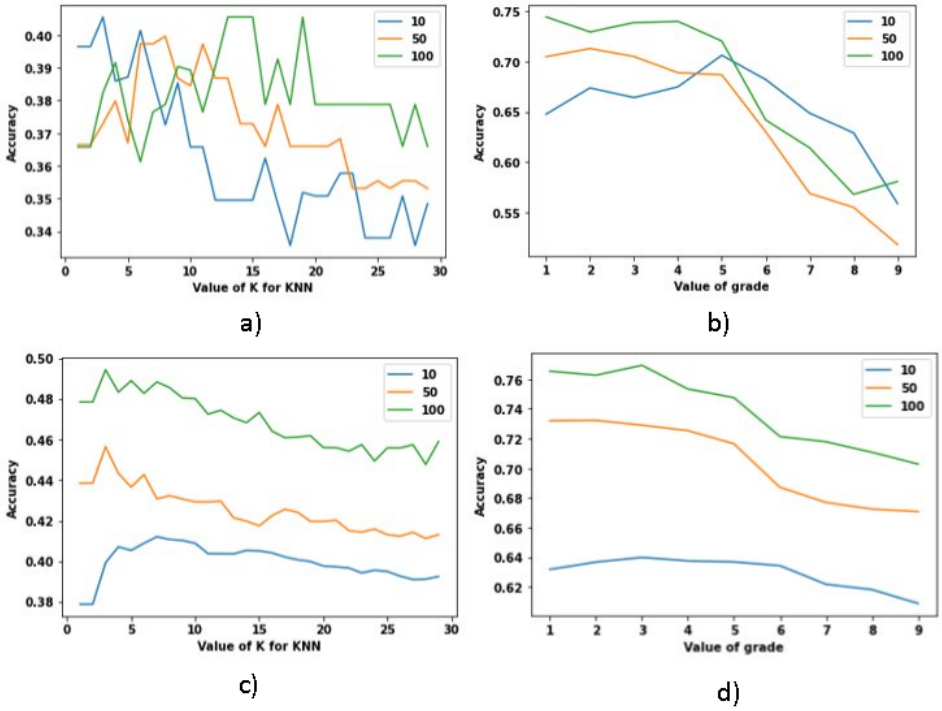


Figura 5.1: Hiper-parámetros calculados con las etiquetas *coarse*. a) estudio de hiper-parámetros del clasificador KNN con interpolación de datos. b) estudio de hiper-parámetros del clasificador SVM con interpolación de datos. c) estudio de hiper-parámetros del clasificador KNN con ventana de datos. d) estudio de hiper-parámetros con ventana de datos

La principal diferencia entre ambos niveles de etiquetas es que la versión reducida de las mismas obtiene valores de *accuracy* considerablemente más altos. Esto se debe a que el problema es mucho más simplificado. Por otro lado, se puede observar que ambos niveles de etiquetas se comportan de forma similar para cada uno de los clasificadores ya que los datos elegidos, son los mismos.

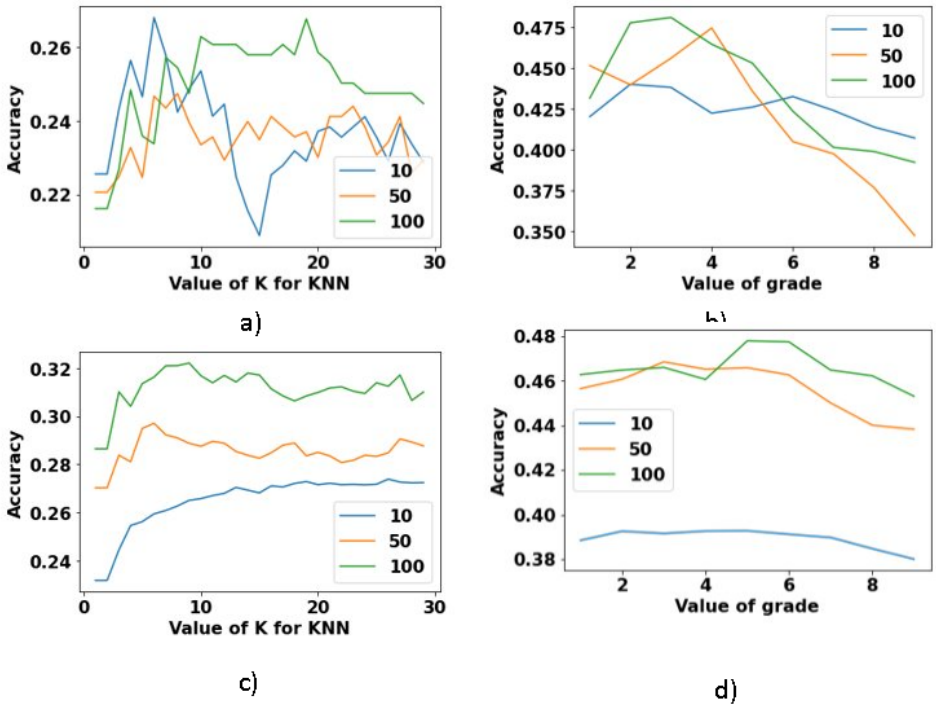


Figura 5.2: Hiper-parámetros calculados con las etiquetas *fine-grained*. a) estudio de hiper-parámetros del clasificador KNN con interpolación de datos. b) estudio de hiper-parámetros del clasificador SVM con interpolación de datos. c) estudio de hiper-parámetros del clasificador KNN con ventana de datos. d) estudio de hiper-parámetros con ventana de datos

Tabla 5.2: Resumen de hiper-parámetros

Nivel de etiquetas	Clasificador	k vecinos	grado
Coarse labels	kNN interpolación	13	
	SVM interpolación		1
	kNN ventana	3	
	SVM ventana		3
Fine-grained labels	kNN interpolación	19	
	SVM interpolación		3
	kNN ventana	9	
	SVM ventana		5

5.3.2. Mejor clasificador

A continuación se evalúa el mejor clasificador con la medida de *accuracy* y los parámetros ya elegidos.

Para elegir el mejor clasificador se utilizan los hiper-parámetros óptimos y para cada

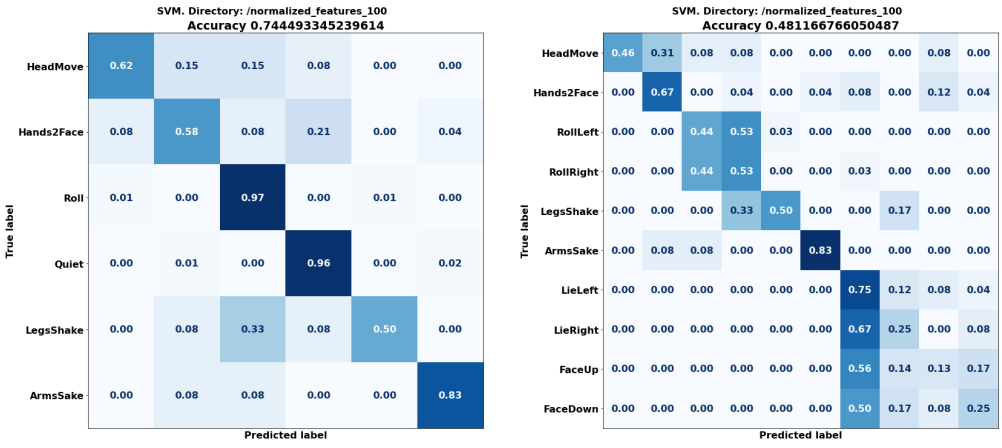
clasificador, se entrena con los datos de entrenamiento y se valora con los datos de *test*.

En la tabla 5.3 se puede observar el *accuracy* obtenido con cada uno de los clasificadores. Como se puede observar en ambos casos el mejor clasificador obtenido es SVM con interpolación de datos. Esto se debe a que con interpolación de datos se toman muestras equidistantes de toda la secuencia y por lo tanto, las características elegidas representan toda la acción. Por el contrario, el método de ventana toma subacciones a las que les asigna la misma etiqueta, lo que hace que funcione peor.

Tabla 5.3: Medidas de los clasificadores

Clasificador	accuracy *coarse*	accuracy *fine-grained*
kNN interpolación	0.40545	0.26763
SVM interpolación	**0.74449**	**0.48117**
kNN ventana	0.38230	0.24744
SVM ventana	0.73879	0.45323

Como se puede observar en 5.3 se incluyen las matrices de confusiones con ambas etiquetas. Por un lado, en las etiquetas *coarse*, predice muy bien las clases *Roll* y *Quiet*. Sin embargo, confunde en numerosas ocasiones *LegsShake* con *Roll*. Esto se debe a que en ambas se realiza movimiento de las piernas. Por otro lado, en las etiquetas *fine-grained*, se puede observar que confunde *RollLeft* con *RollRight*. Esto se debe a que en ambas la acción es girar y es muy difícil distinguir hacia qué lado gira el sujeto. También confunde *LieRight*, *FaceUp* y *FaceDown* con *LieLeft*. Esto se debe a que en todas ellas la acción consiste en estar quieto y debido a la similitud de las acciones las confunde. Por último, confunde de nuevo *LegsShake* con *RollRight*. Esto es debido a que en ambas acciones se mueven las piernas.



a) b)

Figura 5.3: a) Matriz de confusión calculada con las etiquetas *coarse*. b) Matriz de confusión calculada con las etiquetas *fine-grained*

La principal diferencia entre ambos niveles de etiquetas es de nuevo, las medidas de

accuracy que se obtienen. En el caso de las etiquetas *fine-grained* se obtienen resultados más limitados debido a la complejidad del problema al que se afronta. Las etiquetas *fine-grained* tratan de distinguir acciones que son muy similares. En concreto, trata de distinguir distintas acciones que consisten en estar quieto en dos posiciones distintas. La cámara de eventos no produce eventos cuando hay ausencia de movimiento. Por lo tanto, las regiones de interés extraídas para realizar la clasificación de dichas acciones, son muy confusas ya que son muy similares. En las matrices de confusión, Figura 5.3, se puede observar el fenómeno de confusión que ocurre en todas las acciones que consisten en estar quieto.

5.3.3. Validación cruzada por configuraciones

El último de los experimentos consiste en realizar una validación cruzada por configuraciones con el mejor de los clasificadores obtenido (SVM interpolado). Para realizar la validación cruzada, se entrena con cada una de las configuraciones y se evalúa con cada una de ellas (una por una) y por último se calcula la media de las medidas obtenidas. De esta manera se puede observar cuál de las configuraciones generaliza mejor.

Como se puede observar en la Tabla 5.4 se encuentran los resultados del accuracy entrenando con las ditintas configuraciones. Se puede observar como la configuración que mejor generaliza es la 3 (y la que mejores resultados obtiene). Esto se debe a que hay mayor visibilidad del sujeto y por lo tanto las acciones se observan mejor.

Tabla 5.4: Tabla resumen del accuracy balanceado de los distintos clasificadores teniendo en cuenta las configuraciones como datos de entrenamiento con las etiquetas *coarse* y *fine-grained*

Etiquetas	Config. de entrenamiento	Accuracy config 1	Accuracy config 2	Accuracy config 3	Accuracy average
coarse labels	configuración 1	0.76019	0.68611	0.55509	0.66713
	configuración 2	0.64444	0.62159	0.55138	0.60581
	configuración 3	0.81204	0.68270	0.78426	0.75966
fine-grained labels	configuración 1	0.40819	0.33369	0.31997	0.35395
	configuración 2	0.44928	0.36440	0.30273	0.37214
	configuración 3	0.50833	0.36000	0.46250	0.44361

En este caso, no hay diferencias entre las etiquetas. En ambos casos generaliza mejor la configuración 3 debido a que se obtiene mayor visibilidad del sujeto. Cada uno de los niveles de etiquetas obtiene valores del *accuracy* acordes a su nivel de dificultad.

Capítulo 6

Conclusiones

En este trabajo se ha estudiado la importancia de los estudios del sueño y se ha diseñado un sistema capaz de reconocer acciones realizadas por personas mientras duermen. Se ha planteado un uso innovador de las cámaras de eventos en este tipo de estudios. Estas cámaras, además de proporcionar privacidad al paciente, obtienen información de manera más precisa en condiciones de baja iluminación que las cámaras convencionales. Esto se debe a las características de alto rango dinámico y alta sensibilidad de captura que poseen particularmente las cámaras de eventos. Estas propiedades las convierten en sensores más atractivos que otros dispositivos convencionales de captura de datos típicamente utilizados en los estudios del sueño.

Por un lado, en el TFG se ha propuesto e implementado un extractor de regiones de interés a partir de eventos. Para ello, se ha desarrollado un algoritmo en varias etapas. En la extracción de estas regiones de interés, se han ajustado diversos parámetros que ayudan a que el *blob* extraído sea lo más preciso al movimiento posible. Con las regiones de interés, se han extraído características para entrenar diversos clasificadores de acciones. De entre los clasificadores de acciones, se ha discutido el mejor.

Además, como resultado de este TFG, se incluye un *dataset* construído con el mísmo objetivo, monitorización de pacientes en estudios del sueño. En este *dataset* se incluyen diversos sujetos grabados en distintas condiciones de visibilidad e iluminación. Los sujetos realizan acciones seleccionadas que se asemejan a acciones que se realizan típicamente durante el sueño o que tienen especial interés para determinar determinados trastornos del sueño.

Como resultado de la realización de este TFG se ha alcanzado el objetivo de clasificación de acciones en estudios del sueño se ha obtenido un sistema completamente operativo, capaz de realizar predicciones de acciones considerablemente precisas.

6.1. Líneas de trabajo futuro

Como posibles líneas futuras de investigación abiertas tras los resultados obtenidos, se proponen las siguientes:

1. Realizar reconocimiento de acciones basado la representación de eventos en un fotograma. Es decir, realizar reconocimiento de acciones basado en su representación en una imagen. De este modo se pueden usar métodos de clasificación más complejos como pueda ser una red neuronal convolucional.

2. Aumentar el tamaño del *dataset* con más cantidad de sujetos. De esta manera, se dispone de mayor variabilidad en los datos y se evita sobreajuste a los datos de entrenamiento.

3. Realizar predicciones con sistemas de clasificación más complejos como pueden ser las redes neuronales.

4. Extraer más regiones de interés y con mayor adaptabilidad para considerar el movimiento de múltiples extremidades a la vez. De esta manera, se puede distinguir las diferentes extremidades que mueve el sujeto durante la acción y realizar predicciones más precisas.

Bibliografía

[1] G. Gallego, "Event-based robot vision: Sampling in time vs in range," https://sites.google.com/view/guillermogallego/teaching/event-based-robot-vision, 2020.

[2] Efros, Berg, Mori, and Malik, "Recognizing action at a distance," in *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 726–733.

[3] T. Åkerstedt and P. M. Nilsson, "Sleep as restitution: an introduction," *Journal of internal medicine*, vol. 254, no. 1, pp. 6–12, 2003.

[4] S. K. Yadav, K. Tiwari, H. M. Pandey, and S. A. Akbar, "A review of multimodal human activity recognition with special emphasis on classification, applications, challenges and future directions," *Knowledge-Based Systems*, vol. 223, p. 106970, 2021.

[5] C. Lustenberger, M. L. Ferster, S. Huwiler, L. Brogli, E. Werth, R. Huber, and W. Karlen, "Auditory deep sleep stimulation in older adults at home: a randomized crossover trial," *Communications medicine*, vol. 2, no. 1, p. 30, 2022.

[6] G. Matar, J.-M. Lina, and G. Kaddoum, "Artificial neural network for in-bed posture classification using bed-sheet pressure sensors," *IEEE journal of biomedical and health informatics*, vol. 24, no. 1, pp. 101–110, 2019.

[7] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis *et al.*, "Event-based vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, pp. 154–180, 2020.

[8] Q. Wang, Y. Zhang, J. Yuan, and Y. Lu, "Space-time event clouds for gesture recognition: From rgb cameras to event cameras," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 1826–1835.

[9] M. Litzenberger, C. Posch, D. Bauer, A. Belbachir, P. Schon, B. Kohn, and H. Garn, "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *2006 IEEE 12th Digital Signal Processing Workshop 4th IEEE Signal Processing Education Workshop*, 2006, pp. 173–178.

[10] Y. Kong and Y. Fu, "Human action recognition and prediction: A survey," *International Journal of Computer Vision*, vol. 130, no. 5, pp. 1366–1401, 2022.

[11] J. G. Klinzing, N. Niethard, and J. Born, "Mechanisms of systems memory consolidation during sleep," *Nature neuroscience*, vol. 22, no. 10, pp. 1598–1610, 2019.

[12] X. Zhou, Y. Cui, G. Xu, H. Chen, J. Zeng, Y. Li, and J. Xiao, "Sleep action recognition based on segmentation strategy," *Journal of Imaging*, vol. 9, no. 3, p. 60, 2023.