Detección automática de patrones rutinarios con imágenes egocéntricas

Tomás Federico García[1][0000−0001−6841−4136]

Facultad de Ciencias Exactas - Ingeniería de Sistemas
Universidad Nacional del Centro de la Provincia de Buenos Aires, Tandil BA 7000, Argentina

Resumen En los últimos años se ha comprobado que el registro de las actividades diarias de una persona compuesto principalmente por un conjunto de imágenes y otras magnitudes provenientes de sensores como GPS y acelerómetros, conforma un volumen de información extenso, el cual puede contribuir a tratamientos contra enfermedades o condiciones relacionadas a la pérdida de memoria [10]. Una forma de capturar estos eventos es mediante el uso de cámaras egocéntricas, un tipo de cámaras que intentan imitar el campo visual de un individuo. Gracias a los avances en el área de la inteligencia artificial, los algoritmos de *Deep Learning* para la clasificación de imágenes convierten la tarea de clasificar estas imágenes en una tarea semi-automática. En base a esta premisa, se pretende estudiar la viabilidad de una solución que sea capaz de detectar e identificar actividades rutinarias a partir de un conjunto de imágenes tomadas a lo largo de un período de tiempo determinado.

Keywords: Rutina · Reconocimiento de patrones · Minería de grafos.

1. Introducción

Las enfermedades o trastornos relacionados a la pérdida inusual de memoria ocasionan una disminución en la calidad de vida no solo del paciente, sino también de sus personas allegadas. En los últimos años, el análisis de la rutina diaria en seres humanos se ha comenzado a profundizar, especialmente en el campo de la medicina [13]. Numerosos trabajos han demostrado que el registro de eventos y el análisis de estos registros provoca efectos prometedores en el tratamiento y la rehabilitación de pacientes con pérdidas de memoria [21] [4] [14].

El área conocida como *Life-logging* consiste en el proceso de creación de un diario de vida, ya sea automático o de forma manual [21]. Uno de los avances que han presentado los dispositivos móviles en los últimos años es la capacidad de incorporar sensores fotográficos de tamaño reducido, sin perjudicar la calidad de imagen. Esto ha permitido el desarrollo y la comercialización de cámaras *wearables*, que se especializan en emular una visión en primera persona. El análisis de datos que expresan la experiencia vivida desde la perspectiva del usuario tiene sus orígenes en el trabajo [16], donde se analizan potenciales herramientas

construidas a partir de dispositivos portables sin reducir el movimiento del individuo. Sin embargo, el apogeo de estos estudios no se dio hasta la introducción de la cámara SenseCam de la empresa Microsoft en el año 2004, utilizada en su mayor parte, para el registro diario de eventos. Estas cámaras intentan aproximar el campo de visión de quien las usa, dando lugar a imágenes que muestran la experiencia del usuario. Gracias al foco en primera persona de las imágenes se puede obtener una representación fiel del día de un individuo, la cual nos permite estudiar patrones de comportamiento.

Este trabajo se enfoca en el desarrollo de una técnica de minado para la detección y clasificación de rutinas a partir de un conjunto de imágenes egocéntricas. Estas técnicas podrán ser utilizada potencialmente por pacientes, e incluso personas que decidan ganar consciencia de forma objetiva sobre su estilo de vida. Una rutina ha sido previamente descrita como una costumbre o hábito adquirido de hacer las cosas por mera práctica y con cierta regularidad [12].

El resto del documento se estructura de la siguiente manera. En la Sección 2 se presentan los trabajos relacionados existentes. En la Sección 3 se aborda la metodología propuesta. A continuación, en la Sección 4 los resultados obtenidos. Finalmente, se exponen las conclusiones en la Sección 5.

2. Estado del arte

En esta sección se abordan diferentes soluciones que están, en mayor o menor medida, relacionadas con este trabajo.

El primer antecedente para analizar las regularidades en el comportamiento humano de un conjunto de datos a gran escala de manera no supervisada se basa en el análisis de información recopilada a través de teléfonos móviles. En [11], Eagle y Pentland recolectaron datos de teléfonos móviles, pudiendo desarrollar un clasificador que es capaz de percibir estructuras y rutinas en la vida de una persona. Por otro lado, en [2], se exploró el reconocimiento de rutina mediante la aplicación de sensores. A través de unos pocos sensores y un pequeño número de clasificadores, alcanzaron un valor de *recall* y *accuracy* de 79,75 % y 83,40 % respectivamente en la distinción entre diferentes rutinas.

Varios trabajos aplican Redes Neuronales Convolucionales (CNNs) al reconocimiento de clases bien definidas. Por ejemplo, Castro et al. [6] presenta reconocimiento de actividades basado en CNNs combinando imágenes, tiempo e información global de la imagen, utilizando más de cuarenta mil imágenes etiquetadas para obtener 19 clases que alcanzan un valor de *accuracy* de 83,07 %. En este sentido, Cartas et al. [5] explora la clasificación de actividades diarias desde cámaras egocéntricas usando un enfoque CNN. Ellos muestran que el valor de *accuracy* puede ser mejorado en un 8 % cuando añaden información de diferentes capas de una red CNN. Sin embargo, estos trabajos no son capaces de identificar patrones ni comportamiento más allá de lo definido durante su entrenamiento.

Recientemente, el análisis de imágenes egocéntricas para descubrir comportamiento humano ha sido analizado en [25] y [17]. En [25] se propuso un método

basado en el análisis de conceptos detectados en secuencias con métodos de procesamiento de lenguaje natural. En [17] se propuso una estrategia de tipo *greedy* para la identificación de patrones a lo largo de los días. Si bien estos métodos constituyen el estado del arte más cercano a este trabajo, es necesario mencionar que presentan ciertas limitaciones. Un aspecto relevante es que los métodos [17] y [25] definen un valor de tiempo para los *frames* (*time-frame*) y proponen un valor fijo de distancia para agrupar los mismos. Si bien en el trabajo que se presenta aquí se define un parámetro de longitud de ventana, es necesario destacar que los algoritmos que se utilizan no son tan costosos desde el punto de vista computacional como los que se utilizan en dichos trabajos. En [20], los autores propusieron un *framework* para segmentar y organizar automáticamente un conjunto de vídeos de imágenes egocéntricas de escenas de la vida diaria. A través de una técnica de segmentación temporal no supervisada, cada vídeo se divide en escenas considerando las características generadas por una CNN. El punto débil de este método es que no se considera el contenido o actividad de las escenas para el cálculo de la similitud. Esto hace imposible describir los patrones encontrados y puede producir grupos de escenas que visualmente parecen similares pero que sean muy diferentes en un nivel semántico y conceptual.

En contraste con los métodos descritos en esta sección, en este trabajo se propone un método no supervisado basado en grafos para el descubrimiento de patrones. La principal diferencia está en la técnica de segmentación, compuesta por el análisis de *features* obtenidas de una red neuronal, los histogramas de las imágenes egocéntricas y el lugar donde fueron tomadas las imágenes. Otro aspecto a destacar es es el *fine-tunning* realizado en una red neuronal para el reconocimiento de escenas, pudiendo mejorar su desempeño general en imágenes de tipo egocéntricas.

3. Metodología propuesta

En esta sección se describe el método para identificar y detectar la rutina de una persona. Como datos de entrada se utilizan una serie de imágenes egocéntricas provenientes de una cámara *wearable*. Posteriormente se agrupan las imágenes por clases y se construyen segmentos de actividades ordenados cronológicamente. Para culminar, se aplica una transformación de los datos para representar cada día de la persona en un grafo dirigido, seguido de la aplicación de técnicas de minería para encontrar patrones frecuentes entre ellos.

3.1. Recolección de datos

Para poder detectar e identificar patrones rutinarios en el día a día de una persona, se utilizó durante dos semanas una cámara Narrative Clip de primera generación. Este tipo de cámaras resulta muy útil a la hora de representar el campo visual de su portador, debido a la posición donde se viste, la apertura focal y la velocidad de obturación. De esta forma, el conjunto de datos de

entrada estaba compuesto por aproximadamente unas 15.000 capturas, en promedio unas 1.000 fotos por día. Adicionalmente, se utilizó el conjunto de datos EgoRoutine-Dataset [25] de la Universidad de Barcelona, compuesto por imágenes de 7 usuarios distintos a lo largo de 104 días, brindando un total de más de 100.000 imágenes. En la figura A.1 es posible apreciar un conjunto de imágenes pertenecientes a un usuario de este *dataset*.

3.2. Armado de segmentos

Tomando el conjunto de imágenes de entrada y basándose en el método LPaMI: A Graph-Based Lifestyle Pattern Mining Application Using Personal Image Collections in Smartphones[15], se construyeron, para cada día, segmentos compuestos por todas las imágenes que representen una actividad o escena en particular, representando cada segmento un conjunto de imágenes que comparten la misma actividad. A diferencia de LPaMI, este proceso consideró además, los horarios de inicio y finalización de cada segmento, facilitando la búsqueda de patrones frecuentes entre los distintos segmentos de los días de un usuario.

3.3. Similitud entre imágenes

El análisis de similitudes entre imágenes basado en el histograma es uno de las soluciones más comunes a la hora de clasificar, por lo que resultó natural comenzar por este método. Partiendo de la premisa que si se tienen histogramas similares, entonces las imágenes son similares y, por lo tanto, de la misma escena representada, se construyeron los segmentos a partir del algoritmo 1.

Algorithm 1: Crear segmentos usando histogramas

Input: Conjunto finito de imágenes *images* de un día determinado para un usuario y un umbral numérico positivo *threshold* de valor 0,7
Output: Conjunto de segmentos generados *segments*

```
images ← input
segments ← [ ]
segment ← [ ]
for image in images do
    histogram_1 ← average_histogram(images)
    histogram_2 ← histogram(image)
    value ← cross_correlation(histogram_1, histogram_2)
    if value > threshold then
        segments.append(image)
    else
        segments.append(segment)
        segment ← [ ]
        segment.append(image)
return segments
```

El orden temporal de las imágenes es relevante, una escena es un conjunto de imágenes consecutivas en el tiempo que refieren a un lugar o una actividad, por lo tanto la búsqueda de similitud de imágenes es entre inmediatas adyacentes. Se calcula y promedia el histograma de las imágenes anteriores y se evalúa con respecto al histograma de la imagen a analizar utilizando correlación cruzada[3]. En consecuencia, se obtienen segmentos de imágenes de diverso tamaño, siendo algunos muy similares, lo cual se puede observar en la Figura 1.

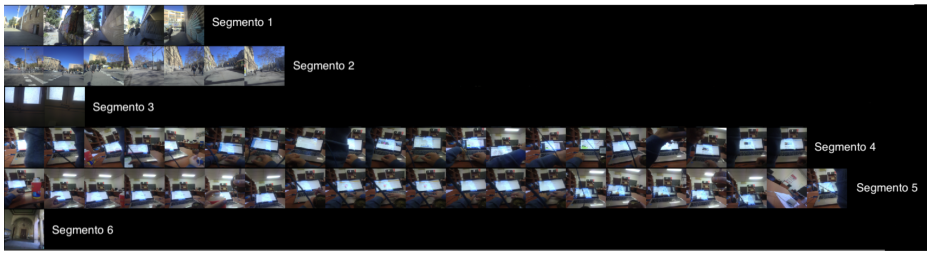

Figura 1. Ejemplo de segmentos resultantes de la agrupación de imágenes por histograma.

3.4. Reducción de segmentos

Para reducir los segmentos muy pequeños y similares entre si, fue necesario agrupar los segmentos de características similares. En el trabajo [1] se emplean *features* de una red neuronal para determinar la similitud entre escenas. Basados en ese trabajo, se calcularon las *features* de cada una de las imágenes del usuario evaluadas en 3 redes neuronales (Densenet, Inception V3 y Xception). Estas *features* son modeladas como vectores de 1920, 2048 y 2048 elementos, respectivamente. Estos vectores fueron concatenados, obteniendo así un vector de 6016 elementos que representa el contenido de la imagen (ver Figura 2) y de esta manera es posible entonces, interpretar a cada segmento de imágenes como un segmento de vectores y promediar los mismos para obtener un vector representativo del segmento $(V_1 + V_2 + V_3 ... + V_n)/n$. El último paso consiste en comparar cada uno de los segmentos con su consecutivo mediante similitud del coseno. En caso de ser mayor o igual a un umbral determinado se concatenan los segmentos.

3.5. Clasificación de segmentos

Para la clasificación de segmentos en clases o escenas, se utilizó la red neuronal convolucional VGG16 [23], entrenada a partir del conjunto de datos Places365, un *dataset* formado por 365 clases y más de 1.8 millones de imágenes [13]. Gracias a esta aplicación se obtienen para cada imagen las *features softmax*

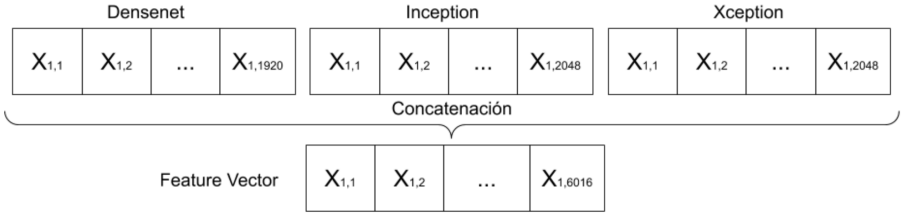

Figura 2. *Feature vector* correspondiente a una imagen.

a partir de la última capa de la red neuronal. En un principio, se optó por la clasificación independiente de cada una de las imágenes. Sin embargo, esta solución aportaba demasiada variación entre imágenes, resultando en secuencias de actividades cortas y con poca relación entre sí, por lo que fue descartada. Por el contrario, la clasificación de las imágenes agrupadas en segmentos aportó resultados significativos. Para cada segmento, se sumaron las probabilidades de cada clase *features softmax*, y luego se dividió cada valor por la cantidad de imágenes que componen cada segmento. En una primera instancia, las 365 clases propuestas por Places365 provocaban resultados extremadamente dispersos, por lo que se optó por una reducción de categorías. Finalmente, cada segmento posee un vector de probabilidades, siendo la clase más representativa la que posea el mayor valor de probabilidad. De esta manera, queda armado un segmento con su conjunto de imágenes, un horario de inicio y final proporcionado por su primer y última imagen respectivamente, y su escena más representativa. Un ejemplo de estos segmentos se encuentra en la Figura 3.

Figura 3. Segmento *Pathways*, representado por un conjunto de imágenes. Además de las imágenes, este segmento está caracterizado por su hora de inicio y fin.

3.6. Aplicación mejorada de VGG16-Places-365

En una primer prueba con VGG16-Places-365 [23], se obtuvo un promedio de 39 % de *F1-Score*. Este valor bajo en la métrica puede deberse a que VGG16-Places fue entrenada con imágenes cuya apertura focal es mayor, y en su mayoría, son imágenes de paisajes a diferencia de las imágenes utilizadas en este trabajo, las cuales son imágenes egocéntricas, con una apertura focal mucho menor y en su mayoría, primeros planos o acercamientos. Teniendo una cantidad importante

de imágenes, se optó por realizar un *fine tuning*, una técnica muy común en el campo de *deep learning*.

Para poder mejorar el reconocimiento de imágenes y por consecuencia, la generación de segmentos, se optó por ajustar la red neuronal utilizando como conjunto de datos de entrenamiento una cantidad de 600 imágenes por clase y el framework *Keras* [7]. El conjunto de entrenamiento presentaba una disparidad entre las clases considerablemente grande, yendo desde las 8000 imágenes para la clase *Office* hasta las 451 imágenes de la clase *Transportation*, o las pocas 20 imágenes para la clase *Beach*. Para balancear las clases, se optó por la generación de datos de forma sintética, implementando una técnica de *image augmentation* aplicando diversas transformaciones. Como resultado, se incrementó el valor de *F1-Score* hasta un 73 %. Las métricas para cada una de las clases se puede observar en el cuadro A.4 del apéndice. En la sección de resultados se comparan los resultados obtenidos para cada usuario antes y después del *fine-tunning*.

3.7. Reducción de ruido

Previamente a la ejecución de algoritmos de minería para poder encontrar patrones que se repitan en los segmentos, se pre-procesaron los datos para poder eliminar *outliers*. Para ello, se consideró que toda actividad menor a una duración de tres minutos no aportaría información relevante para este trabajo, y por lo tanto, deberían ser descartadas. Por ejemplo, una persona puede tomarse una pausa en su horario laboral para ir al baño, atender un llamado en una habitación diferente o ir al auto a buscar alguna pertenencia. Este tipo de actividades no aportan a la búsqueda de rutinas, ya que no solo son de poca duración, sino que a priori, no representan un comportamiento rutinario y, a fin de cuentas, la actividad "trabajo" es la que sí contribuye con información. Sin embargo, existe la posibilidad de que un sujeto pueda tener actividades de corta duración en su día a día. Un ejemplo claro de este caso sería ir al cajero automático a retirar dinero todos los días a la mañana, de camino al trabajo. Si aporta o no información dependerá de cada caso en particular, por lo que este pre-procesamiento es configurable mediante la parametrización mediante un umbral de duración. Luego de limpiar los segmentos bajo este criterio, se unen los segmentos consecutivos cuyas clases sean iguales. En la Figura del apéndice A.2 se muestra un gráfico con los diferentes días de un usuario con sus respectivos segmentos.

3.8. Distinción entre días rutinarios y no rutinarios

Para poder establecer una distinción entre un día rutinario de uno no rutinario, se realizó una alineación de secuencia por pares usando la técnica algorítmica de programación dinámica desarrollada por Needleman–Wunsch [18], tomando como entrada los segmentos generados en la sección anterior. Este algoritmo maximizará el número de coincidencias entre secuencias en toda su longitud. Para dos secuencias $a = a_1a_2...a_n$ y $b = b_1b_2...b_n$ dónde $S_{(i,j)} = S(a_1a_2...a_i, b_1b_2...b_j)$, se cumple que:

$$S_{i,j} = \begin{cases} S_{i-1,j-1} + S(a_i b_j) \\ max(S_{i-x,j} - w_x), x \geq 1 \\ max(S_{i,j-y} - w_y), y \geq 1 \end{cases} \quad (1)$$

Dónde S_i, j es el *score* en la posición i en la secuencia a y la posición j. w_x es la penalización por una distancia de longitud x en la secuencia a, y w_y es la penalización por una distancia de longitud y en la secuencia b.

Para la implementación, se relacionó a cada escena, un caracter de un diccionario. Por ejemplo, un día puede estar compuesto por las clases *forest*, *office* y *pathways*, y se les asigna un caracter F, O y P respectivamente. Una vez codificados los días en secuencias de caracteres, estos pueden ser alineados mediante BioPython [8], un conjunto de herramientas orientadas al campo de la bioinformática. Entre ellas, se encuentra el algoritmo de Needleman–Wunsch mencionado anteriormente.

Para poder realizar una comparación entre los días de un usuario, se debió establecer un criterio que permita representar el alineamiento entre dos días mediante un valor numérico. Para este fin, por cada imagen que compone una escena se concatenaron las *features* correspondientes previamente calculadas. De esta manera, se forma un vector de *features* representativo de cada escena. El siguiente paso consistió en comparar cada par de vectores (cada uno de un día distinto) mediante la similitud del coseno. Esta medida analiza la similitud existente entre dos vectores en un espacio, y su resultado se expresa con un valor dentro del intervalo [1,-1], siendo 1 el valor que representa similitud completa. Se propuso $1 - similitud$, con el objetivo de expresar la mayor similitud entre dos vectores cuando este resultado se acerca a 0. Para considerar el diferente tamaño de los días, a las no coincidencias modeladas como '-' por BioPython, se les asignó el valor de máxima distancia posible. En la Figura 4 se encuentra un ejemplo reducido, donde se puede ver el resultado de aplicar la diferencia de $1 - similitud$ para cada escena entre dos días. La suma de las diferencias indicará el parecido entre los días, siendo más similares en valores más cercanos a 0. A continuación, se generó una matriz de distancias con el objetivo de relacionar todos los días de un mismo usuario, tal como se observa en la Figura 5.

3.9. Minado de segmentos

Finalmente, se aprovecharon los segmentos generados para poder extraer información útil, buscando patrones frecuentes entre ellos. Siguiendo un proceso de KDD (Knowledge Discovery in Databases), luego de pre-procesar los datos de entrada, el siguiente paso en un algoritmo KDD es la transformación. En este estadio, los segmentos son transformados en grafos dirigidos, estructuras consideradas ideales para representar una secuencia lineal de actividades, dado el amplio repertorio de algoritmos de minería en este tipo de estructura. Siendo un grafo $G_i(V)$ dónde i corresponde al día del conjunto de datos; V es el conjunto de nodos, representando las actividades; y E un conjunto de arcos, que representan la transición de una actividad a la siguiente. Fue fundamental el

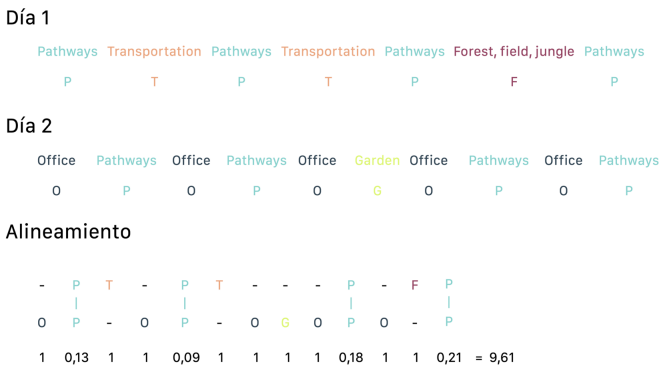

Figura 4. Alineamiento entre dos días de un usuario, aplicando similitud del coseno.

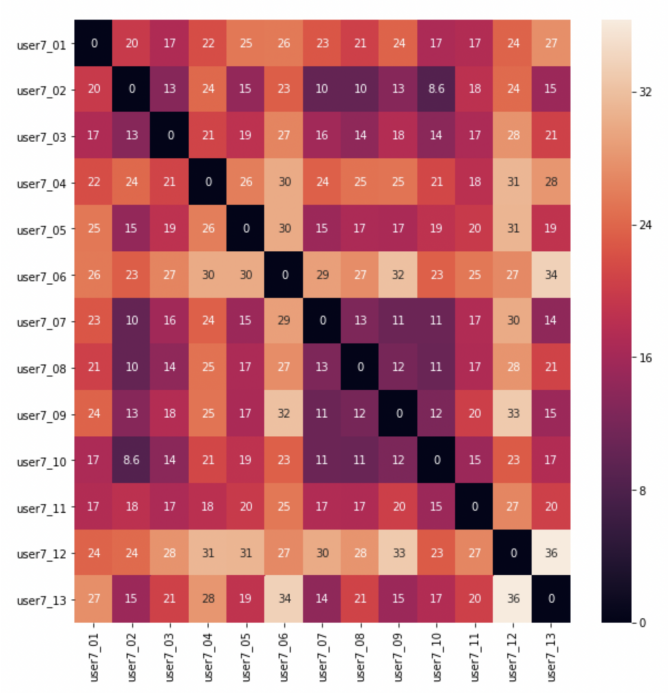

Figura 5. Matriz de distancias entre los días del usuario 7.

etiquetado de cada componente del grafo para su correcta identificación y para poder proporcionar información extra al algoritmo de minería. En concreto, se etiquetó a cada nodo con el nombre de la actividad que representa, y cada arco recibió la etiqueta del rango horario de la hora de inicio del nodo entrante. De esta manera se pudo diferenciar una misma secuencia de segmentos que se re-

piten en diferentes días del conjunto de datos pero en diferentes horarios. Este rango horario se determinó para poder reducir posibles ruidos, tal y como se presenta en *Sr-clustering: Semantic regularized clustering for egocentric photo streams segmentation* [9], y se define a continuación: 00am a 07am, 07am a 9am, 09am a 11am, 11am a 2pm, 2pm a 5pm, 5pm a 7pm, 7pm a 9pm. y 9pm a 00am. Cabe destacar que a mayor cantidad de rangos horarios, mayor será la precisión de este algoritmo. En la Figura 6 se puede observar una representación de un día completo transformado a una estructura de grafo dirigido, con los horarios de conmutación en los arcos.

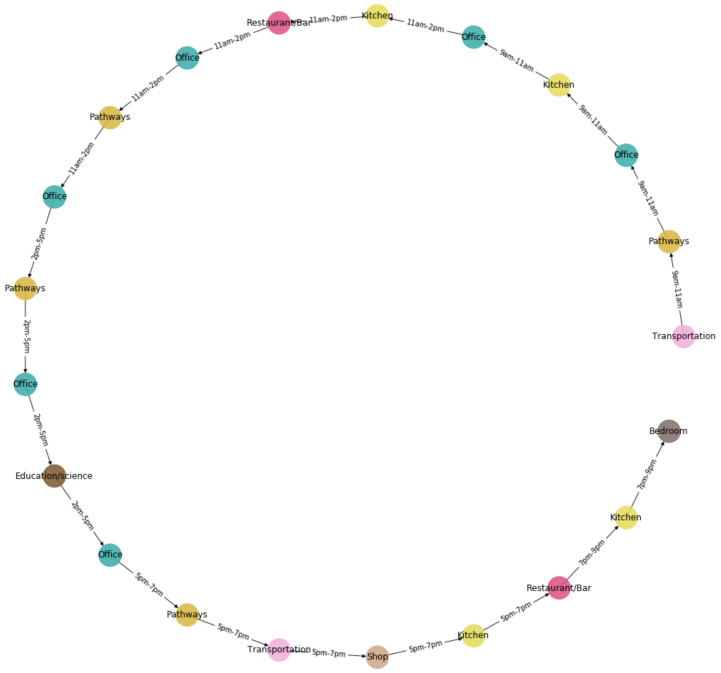

Figura 6. Ejemplo de un día del conjunto de datos transformado a una estructura de tipo grafo para su posterior análisis.

Con los datos de entrada pre-procesados y transformados, se aplicó el algoritmo de minería de datos en grafos gSpan [26]. La elección del algoritmo gSpan fue por su compatibilidad con grafos dirigidos, dejando de lado otros similares, como GASTON [19]. En pocas palabras, este algoritmo toma de entrada un conjunto de grafos y retorna todo sub-grafo común entre los grafos de entrada. A este algoritmo también se le puede proporcionar un umbral de frecuencia mínimo por parámetro. De esta manera, se pueden considerar únicamente todas las secuencias rutinarias que se repitan en al menos cantidad de días.

Sin embargo, gSpan no genera los sub-grafos maximales. Esto quiere decir que dentro del conjunto resultante del proceso de minado, un grafo puede ser sub-grafo de otro del mismo conjunto. Para esto, se realizó un post-procesamiento de la información eliminando los sub-grafos del conjunto arrojado por gSpan y así poder obtener las actividades rutinarias correspondientes. Cabe destacar que estos grafos resultantes pueden resultar de utilidad para poder relacionar días que son rutinarios entre sí. Partiendo de una secuencia *pathways*, *transportation*, *office*, *shop*, con una frecuencia de 5, un posible sub-grafo podría ser *pathways*, *transportation*, *office* con una frecuencia de 3 y ser completamente válido.
Finalmente, el resultado de este proceso es un conjunto de estructuras basadas en grafos, que representan secuencias de actividades que se repiten al menos x veces en el conjunto de imágenes de entrada. Resulta de interés destacar el grado de ajuste de esta última etapa del proceso propuesto. Pudiendo modificar el rango horario de las actividades, descartar actividades dada una duración en particular y la cantidad mínima de frecuencia de las secuencias de actividades.

4. Resultados

El método propuesto en la sección anterior, se probó en una computadora con un procesador Intel de séptima generación, modelo Core i7-6700; una tarjeta gráfica Nvidia GTX 1080 y se contó con una cantidad de 16 GB de memoria principal DDR4 a 2166 MHz. Este equipo se utilizó para todo el camino del pipeline propuesto, tanto la clasificación de las imágenes como la minería de resultados. Para poder evaluar la eficacia de este algoritmo, se debió generar un conjunto de referencia (*ground truth*, en inglés), para todos los conjuntos de datos de entrada. Cabe destacar que se agregó a las 25 clases la clase denominada *noise*, para poder identificar el ruido en el conjunto de datos. Este ruido fue principalmente generado de forma involuntaria por los usuarios, ya que al tener la cámara colgando como si se tratara de un collar, se tiende a cubrirla con abrigos, gestos de las manos, etc. Se consideró también comparar los resultados obtenidos con *Organizing egocentric videos of daily living activities* [20], pero dado que utiliza un método de clasificación diferente, se descartó esta posibilidad.

4.1. Algoritmo vs. Ground Truth

Luego de generar un conjunto *ground truth*, se procedió a comparar cada día de cada usuario contra los resultados arrojados por el algoritmo propuesto en la sección anterior. Cabe destacar que, a continuación, se muestran los resultados para el usuario número 6. En la Figura 7 se encuentran los segmentos generados por el algoritmo para el conjunto de datos correspondiente al usuario número 6. Concretamente, la precisión del algoritmo propuesto para cada clase se puede encontrar en la Figura del apéndice A.3 en el apéndice de este documento, en la cual se puede observar un desempeño más que alentador, teniendo en cuenta la cantidad de clases planteadas para este problema.

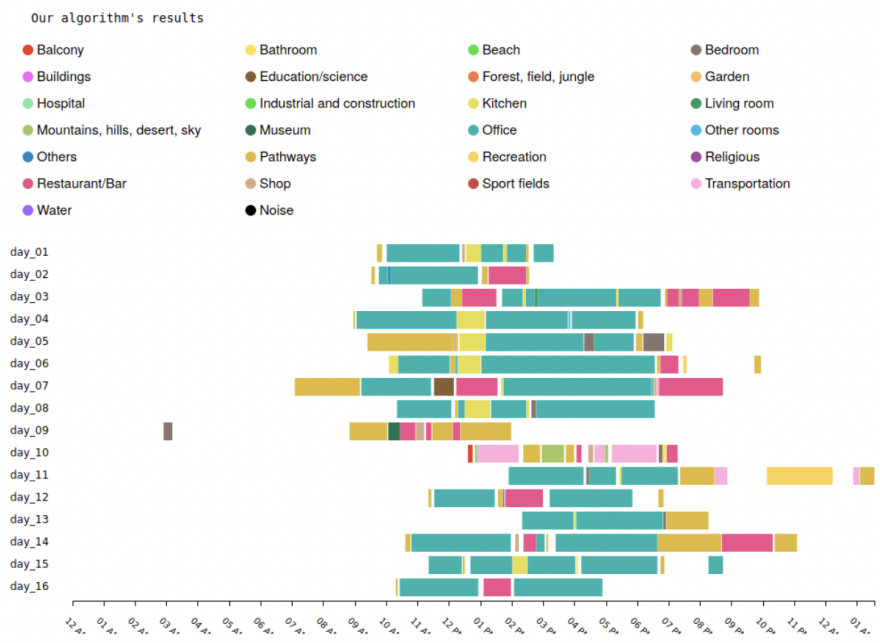

Figura 7. Segmentos resultantes aplicando el método propuesto.

4.2. Otras métricas del algoritmo propuesto

Gracias a la utilización de la librería scikit-learn [22] de Python para Machine Learning, se pudo obtener otras métricas para el algoritmo propuesto. En la siguiente tabla se encuentran los resultados arrojados luego del análisis de la salida del algoritmo. En estos se encuentran los valores de *accuracy*, *precision*, *recall* y *F1-Score*. En el cuadro del apéndice A.1 se pueden apreciar las métricas obtenidas y, en el cuadro del apéndice A.2, las mismas métricas luego de realizar un *fine-tunning* a la red.

4.3. Distinción entre días rutinarios y no rutinarios

Utilizando nuevamente la librería scikit-learn [22] se pudo evaluar los distintos algoritmos de agrupamiento según las métricas anteriormente propuestas. Los resultados obtenidos se pueden apreciar en el cuadro A.3 en el apéndice de este documento. Como se puede observar, el algoritmo *Agglomerative Clustering*, fue el que mejor desempeño tuvo, con una exactitud promedio de 0,77 entre los usuarios del conjunto de datos. A partir de los resultados y, observando con atención las matrices de confusión de los usuarios y contrastándolas con el *ground truth*, se pudo determinar que el algoritmo propuesto genera mejores resultados para aquellos usuarios cuyos días no rutinarios se diferencian con mayor intensidad de los rutinarios.

4.4. Minería de grafos

Por último, se analizan los resultados obtenidos de la minería de grafos para cada usuario. Se logró identificar las actividades para cada uno de los usuarios, y mediante la representación en forma de grafo y un posterior proceso de minería, la cantidad de veces que se repite una actividad en el período evaluado. Como se puede observar en la Figura 8, la cantidad de rutinas detectada para cada usuario por parte del algoritmo propuesto frente a las rutinas obtenidas a partir del minado del conjunto de referencia es menor. Esto se debe principalmente a las disparidades encontradas entre los segmentos para ciertos usuarios. En particular, se puede observar una disminución drástica de rutinas detectadas para aquellos usuarios cuya matriz de confusión es más dispar. Resulta importante destacar que las rutinas arrojadas por el algoritmo propuesto no son maximales.

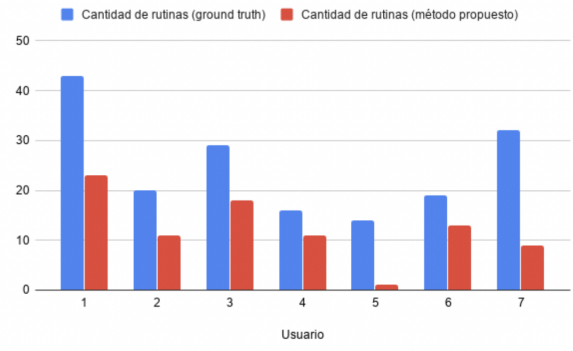

Figura 8. Total de rutinas detectadas sobre el ground truth y los segmentos generados por el algoritmo propuesto.

5. Conclusiones

En esta sección se comentan y discuten los resultados presentados en la sección anterior.

La conclusión más evidente es el desarrollo de un algoritmo no supervisado que permite segmentar escenas de la vida de un usuario a partir de un conjunto de imágenes egocéntricas tomadas a lo largo de un período de tiempo determinado.

La red neuronal VGG16-Places-365 no fue entrenada a partir de imágenes egocéntricas, sino con imágenes cuyo campo de visión es mucho más amplio. Esto produjo resultados inesperados para varias imágenes del conjunto de datos de entrada. Es por esto que se realizó un *fine-tunning* de la red para poder evitar este problema, obteniendo un 0.70 de *accuracy* para el conjunto de datos de validación. Se logró diferenciar los días rutinarios de los no rutinarios de cada

usuario con un 0.77 de *accuracy*. Si bien en Talavera Et. Al. [24] se reconocieron los días rutinarios para el mismo *dataset* con 0,76 de *accuracy*, el método propuesto en este trabajo es computacionalmente más sencillo obteniendo resultados similares en un menor tiempo. Gracias a la minería de datos en grafos, se logró reconocer de forma no supervisada cuáles son los patrones rutinarios de cada usuario junto con su frecuencia. Además, resulta importante destacar la necesidad de un conjunto de datos de entrada uniforme, especialmente en la amplitud temporal de los días. Es recomendable que los días comiencen y terminen en el mismo horario para poder obtener resultados favorables.

Para concluir, se pretende continuar con el trabajo propuesto en este documento. Por un lado, se planifica re-entrenar la red VGG16-Places-365 en un clúster de procesamiento junto con una mayor cantidad de datos para lograr un mejor *accuracy*. Además se evaluará hacer esto con otras redes neuronales y por último, se considerará la incorporación de datos de ubicación de las fotografías al momento de la creación de segmentos para poder mejorar su clasificación.

Referencias

1. Berhe, A., Barras, C., Guinaudeau, C.: Video scene segmentation of tv series using multimodal neural features. Series-International Journal of TV Serial Narratives **5**(1), 59–68 (2019)
2. Blanke, U., Schiele, B.: Daily routine recognition through activity spotting. In: International Symposium on Location-and Context-Awareness. pp. 192–206. Springer (2009)
3. Bourke, P.: Cross correlation. Cross Correlation", Auto Correlation—2D Pattern Identification (1996)
4. Buso, V., Hopper, L., Benois-Pineau, J., Plans, P.M., Mégret, R.: Recognition of activities of daily living in natural "at home" scenario for assessment of alzheimer's disease patients. In: 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). pp. 1–6. IEEE (2015)
5. Cartas, A., Marín, J., Radeva, P., Dimiccoli, M.: Recognizing activities of daily living from egocentric images. In: Iberian Conference on Pattern Recognition and Image Analysis. pp. 87–95. Springer (2017)
6. Castro, D., Hickson, S., Bettadapura, V., Thomaz, E., Abowd, G., Christensen, H., Essa, I.: Predicting daily activities from egocentric images using deep learning. In: proceedings of the 2015 ACM International symposium on Wearable Computers. pp. 75–82 (2015)
7. Chollet, F., et al.: Keras. https://keras.io (2015)
8. Cock, P.J.A., Antao, T., Chang, J.T., Chapman, B.A., Cox, C.J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., de Hoon, M.J.L.: Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics **25**(11), 1422–1423 (03 2009). https://doi.org/10.1093/bioinformatics/btp163, https://doi.org/10.1093/bioinformatics/btp163
9. Dimiccoli, M., Bolaños, M., Talavera, E., Aghaei, M., Nikolov, S.G., Radeva, P.: Sr-clustering: Semantic regularized clustering for egocentric photo streams segmentation. Computer Vision and Image Understanding **155**, 55–69 (2017)

10. Doherty, A.R., Hodges, S.E., King, A.C., Smeaton, A.F., Berry, E., Moulin, C.J., Lindley, S., Kelly, P., Foster, C.: Wearable cameras in health: the state of the art and future possibilities. American journal of preventive medicine **44**(3), 320–323 (2013)
11. Eagle, N., Pentland, A.S.: Reality mining: sensing complex social systems. Personal and ubiquitous computing **10**(4), 255–268 (2006)
12. Fatima, I., Fahim, M., Lee, Y.K., Lee, S.: A unified framework for activity recognition-based behavior analysis and action prediction in smart homes. Sensors **13**(2), 2682–2699 (2013)
13. Giovagnoli, R.: From habits to we-intentionality: Rituals as social habits. In: The Logic of Social Practices, pp. 185–199. Springer (2020)
14. Karaman, S., Benois-Pineau, J., Dovgalecs, V., Mégret, R., Pinquier, J., André-Obrecht, R., Gaëstel, Y., Dartigues, J.F.: Hierarchical hidden markov model in detecting activities of daily living in wearable videos for studies of dementia. Multimedia tools and applications **69**(3), 743–771 (2014)
15. Khan, K.U., Alam, A., Dolgorsuren, B., Uddin, M.A., Umair, M., Sang, U., Duong, V.T., Xu, W., Lee, Y.K.: Lpami: A graph-based lifestyle pattern mining application using personal image collections in smartphones. Applied Sciences **7**(12), 1200 (2017)
16. Mann, S.: Wearable computing: A first step toward personal imaging. Computer **30**(2), 25–32 (1997)
17. Menchón, M., Talavera, E., Massa, J., Radeva, P.: Behavioural pattern discovery from collections of egocentric photo-streams. In: European Conference on Computer Vision. pp. 469–484. Springer (2020)
18. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology **48**(3), 443–453 (1970)
19. Nijssen, S., Kok, J.N.: The gaston tool for frequent subgraph mining. Electronic Notes in Theoretical Computer Science **127**(1), 77–87 (2005)
20. Ortis, A., Farinella, G.M., D'Amico, V., Addesso, L., Torrisi, G., Battiato, S.: Organizing egocentric videos of daily living activities. Pattern Recognition **72**, 207–218 (2017)
21. Pauly-Takacs, K., Moulin, C.J., Estlin, E.J.: Sensecam as a rehabilitation tool in a child with anterograde amnesia. Memory **19**(7), 705–712 (2011)
22. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
23. Qassim, H., Verma, A., Feinzimer, D.: Compressed residual-vgg16 cnn model for big data places image recognition. In: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). pp. 169–175. IEEE (2018)
24. Talavera, E., Petkov, N., Radeva, P.: Unsupervised routine discovery in egocentric photo-streams. In: International Conference on Computer Analysis of Images and Patterns. pp. 576–588. Springer (2019)
25. Talavera, E., Wuerich, C., Petkov, N., Radeva, P.: Topic modelling for routine discovery from egocentric photo-streams. Pattern Recognition p. 107330 (2020)
26. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: 2002 IEEE International Conference on Data Mining, 2002. Proceedings. pp. 721–724. IEEE (2002)

A. Apéndice

Este apéndice contiene figuras y tablas adicionales a las presentadas en el presente documento

A.1. Figuras

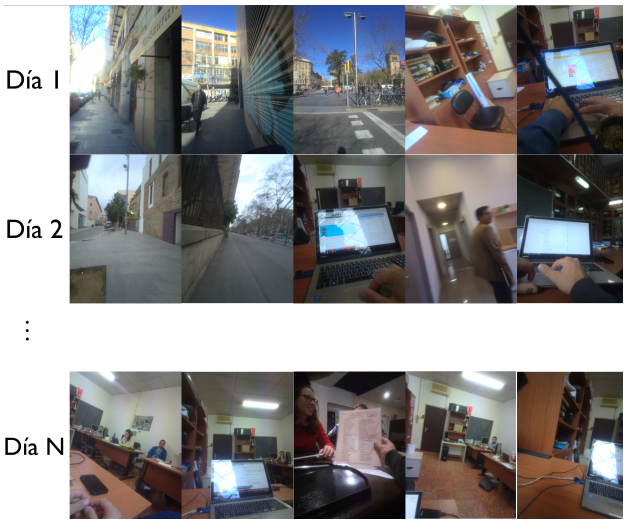

Figura A.1. Ejemplo de imágenes grabadas por uno de los usuarios de la cámara. Cada fila corresponde a un día del usuario

A.2. Tablas comparativas

Cuadro A.1. Métricas obtenidas en la clasificación de imágenes antes de realizar el fine-tunning de VGG-Places-365.

User	Accuracy	Recall	Precision	F1 Score
user 01	0.2544	0.2544	0.6624	0.2339
user 02	0.1612	0.1612	0.4156	0.1649
user 03	0.6461	0.6461	0.6955	0.6533
user 04	0.2764	0.2764	0.3796	0.3165
user 05	0.3526	0.3526	0.3700	0.3043
user 06	0.5020	0.5020	0.7410	0.5614
user 07	0.3894	0.3894	0.7267	0.4173
Avg	0.3673	0.3673	0.5870	0.3919

Figura A.2. Segmentos de un usuario obtenidos a partir del algoritmo propuesto.

Cuadro A.2. Métricas obtenidas en la clasificación de imágenes después de realizar el fine-tunning de Places-365.

User	Accuracy	Recall	Precision	F1 Score
user 01	0.7946	0.7946	0.8828	0.8281
user 02	0.6057	0.6057	0.7002	0.6303
user 03	0.7953	0.7953	0.8673	0.8245
user 04	0.5258	0.5258	0.7222	0.5942
user 05	0.6264	0.6264	0.7334	0.6583
user 06	0.6932	0.6932	0.7964	0.7285
user 07	0.6585	0.6585	0.7810	0.6950
Avg	0.6932	0.6932	0.7964	0.7285

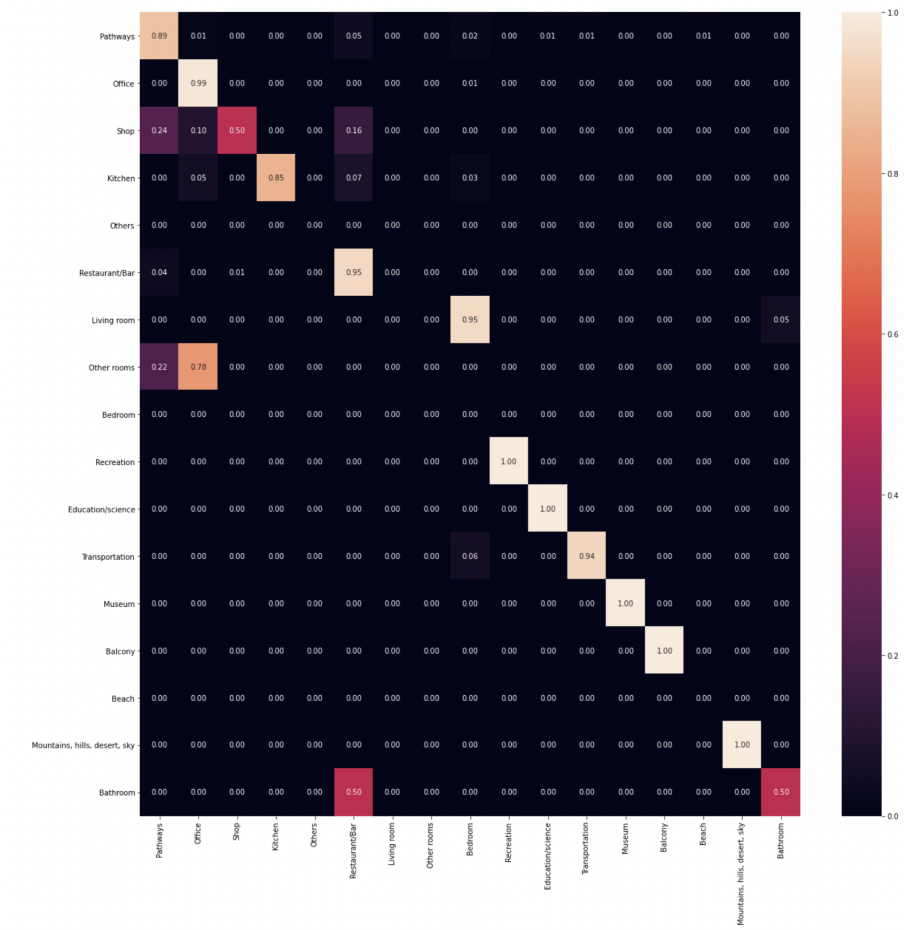

Figura A.3. Matriz de confusión normalizada después de realizar el fine-tunning de VGG-Places-365.

Cuadro A.3. Métricas por usuario en la distinción entre día rutinario y no rutinario.

User	Accuracy	Recall	Precision	F1 Score	Clustering
user 01	0.64	0.64	0.87	0.67	Agglomerative Clustering
user 02	0.60	0.60	0.47	0.53	Agglomerative Clustering
user 03	0.75	0.75	0.71	0.72	Agglomerative Clustering
user 04	0.80	0.80	0.80	0.80	Agglomerative Clustering
user 05	0.77	0.77	0.78	0.77	Agglomerative Clustering
user 06	0.88	0.88	0.88	0.88	Agglomerative Clustering
user 07	0.92	0.92	0.93	0.92	Agglomerative Clustering
Avg	0.77				
user 01	0.71	0.71	0.60	0.65	IsolationForest
user 02	0.60	0.60	0.47	0.53	IsolationForest
user 03	0.75	0.75	0.71	0.72	IsolationForest
user 04	0.60	0.60	0.41	0.49	IsolationForest
user 05	0.54	0.54	0.77	0.44	IsolationForest
user 06	0.88	0.88	0.88	0.88	IsolationForest
user 07	0.54	0.54	0.29	0.38	IsolationForest
Avg	0.66				
user 01	0.64	0.64	0.87	0.67	KMeans
user 02	0.50	0.50	0.44	0.47	KMeans
user 03	0.75	0.75	0.71	0.72	KMeans
user 04	0.80	0.80	0.80	0.80	KMeans
user 05	0.77	0.77	0.78	0.77	KMeans
user 06	0.88	0.88	0.88	0.88	KMeans
user 07	0.92	0.92	0.93	0.92	KMeans
Avg	0.75				
user 01	0.57	0.57	0.65	0.60	Spectral Clustering
user 02	0.50	0.50	0.54	0.52	Spectral Clustering
user 03	0.63	0.63	0.63	0.63	Spectral Clustering
user 04	0.55	0.55	0.60	0.56	Spectral Clustering
user 05	0.77	0.77	0.78	0.77	Spectral Clustering
user 06	0.50	0.50	0.78	0.58	Spectral Clustering
user 07	0.85	0.85	0.88	0.84	Spectral Clustering
Avg	0.62				

Cuadro A.4. Métricas obtenidas de la aplicación del modelo al *dataset* de validación para cada clase.

	Precision	Recall	F1-Score
Balcony	0.78	0.79	0.78
Bathroom	0.62	0.73	0.67
Beach	0.79	0.86	0.82
Bedroom	0.77	0.73	0.75
Buildings	0.56	0.57	0.56
Education	0.80	0.68	0.74
Forest Field Jungle	0.74	0.78	0.76
Garden	0.71	0.74	0.72
Hospital	0.90	0.86	0.88
Kitchen	0.82	0.75	0.78
Livingroom	0.63	0.74	0.68
Mountains Hills Desert Sky	0.74	0.77	0.75
Museum	0.74	0.64	0.69
Noise	0.67	0.71	0.69
Office	0.79	0.86	0.82
Other Rooms	0.68	0.57	0.62
Others	0.66	0.57	0.61
Pathways	0.51	0.59	0.55
Recreation	0.87	0.89	0.88
Restaurant	0.67	0.80	0.73
Shop	0.80	0.74	0.77
Sportfields	0.92	0.86	0.89
Transportation	0.75	0.71	0.73
Water	0.00	0.00	0.00