

# Proyecto 3: Clustering



## **Integrantes:**

Chumbe Ñahui, Herbert (10%)  
Cespedes Zevallos, Adrian (100%)  
Machaca Soloaga, Leandro (100%)  
Maguiña Aranda, Paola (100%)

**Docente:** Cristian López del Álamo  
**Curso:** Machine Learning  
**Grupo de Proyecto:** 3

## CONTENTS

<b>I</b>	<b>Introducción</b>	<b>1</b>
<b>II</b>	<b>Dataset</b>	<b>1</b>
II-A	Descripción . . . . .	1
II-B	Exploración y Análisis del Conjunto de Datos . . . . .	1
II-C	Visualización . . . . .	1
II-D	Transformación . . . . .	2
<b>III</b>	<b>Metodología</b>	<b>2</b>
III-A	Recopilación y Preparación de Datos . . . . .	2
III-B	Extracción de Características . . . . .	3
III-C	Reducción de Dimensionalidad . . . . .	3
III-D	Aplicación de Métodos de Clustering . . . . .	3
III-D1	<b>Clustering con KMeans</b> . . . . .	3
III-D2	<b>Clustering con Agglomerative Clustering</b> . . . . .	3
III-E	Evaluación y Comparación de Resultados . . . . .	3
<b>IV</b>	<b>Implementación</b>	<b>3</b>
<b>V</b>	<b>Experimentación</b>	<b>3</b>
V-A	<b>Exploración y Análisis del Conjunto de Datos</b> . . . . .	3
V-B	<b>Extracción y Reducción de Dimensionalidad</b> . . . . .	3
V-C	<b>Aplicación de técnicas de Clustering</b> . . . . .	4
<b>VI</b>	<b>Discusión</b>	<b>6</b>
VI-A	Eficacia de la Extracción y Reducción de Características . . . . .	6
VI-B	Análisis de Métodos de Reducción de Dimensionalidad . . . . .	6
VI-C	Evaluación de Métodos de Clustering . . . . .	6
VI-D	Limitaciones y Áreas de Mejora . . . . .	6
VI-E	Implicaciones y Futuras Direcciones . . . . .	6
<b>VII</b>	<b>Conclusión</b>	<b>6</b>

## I. INTRODUCCIÓN

El siguiente proyecto esta orientado en el reconocimiento de acciones humanas (*Human Action Recognition, HAR*) cuyo fin es identificar acciones humanas a partir de videos. Para ello se nos proporcionó un conjunto de datos a gran escala y de alta calidad, disponible en la plataforma Kaggle, que incluye enlaces URL encontrados en Youtube, y contienen hasta 650,000 videoclips, que cubren entre 400 y 700 clases de acciones humanas, dependiendo de la versión del conjunto de datos. Los videos presentan varios tipos de interacciones, como tocar instrumentos (humano-objeto), darse la mano y abrazarse (humano-humano) entre más. Cada clase de acción contiene al menos 400 videoclips, cada uno anotado manualmente con una única clase de acción y con una duración aproximada de 10 segundos.

Nuestro objetivo principal es analizar las características del conjunto de datos que se nos ha sido proporcionado, extraer características relevantes de los vídeos y aplicar métodos de reducción de dimensionalidad y *clustering* para identificar patrones y estructuras en los datos.

Este proyecto no solo proporcionará una comprensión profunda del conjunto de datos y sus características, sino que también permitirá evaluar la eficacia de diferentes técnicas de reducción de dimensionalidad y clustering en el contexto del reconocimiento de acciones humanas (*Human Action Recognition, HAR*).

## II. DATASET

### A. Descripción

El conjunto de datos de reconocimiento de acciones humanas (*Human Action Recognition, HAR*) fue proporcionado por medio de Kaggle. Para este proyecto, se utilizaran los archivos proporcionados, ya que contienen información necesaria para el análisis y la extracción de características de los vídeos, que como ya se mencionó, trabajaremos con data de gran escala, pues contiene un aproximado de 650,000 videoclips que deben de pasar por el proceso de extracción. Los archivos con los que abordaremos este proyecto son:

- **test.csv**: Contiene los IDs de YouTube para el conjunto de prueba. Estos videos se utilizarán para evaluar el rendimiento de los modelos de *clustering* desarrollados en este proyecto.
- **train.csv**: Contiene los IDs de YouTube y las etiquetas correspondientes para el conjunto de entrenamiento. Las etiquetas indican la clase de acción que se está realizando en cada vídeo. Este conjunto de datos se utilizará para entrenar los modelos de *clustering*.
- **val.csv**: Similar al archivo *train.csv*, este archivo contiene los IDs de YouTube y las etiquetas para el conjunto de validación. Este conjunto de datos se utilizará para ajustar los modelos y validar su rendimiento antes de evaluarlos en el conjunto de prueba.
- **videourls.txt**: Incluye enlaces a los archivos zip que contienen los vídeos correspondientes a los conjuntos de entrenamiento, validación y prueba. Cada enlace permite

descargar los vídeos necesarios para el análisis y la extracción de características.

### B. Exploración y Análisis del Conjunto de Datos

- El análisis del conjunto de datos se centrará en la revisión y comprensión de las características proporcionadas. Utilizaremos diversas bibliotecas para la extracción de características de los vídeos, evaluando opciones como Gluon y Video Features Documentation.
- Se implementarán métodos de reducción de dimensionalidad como PCA (Análisis de Componentes Principales), JL (Proyección Aleatoria), LDA (Análisis Discriminante Lineal), SVD (Descomposición en Valores Singulares), entre otros.
- Para el *clustering*, se aplicarán métodos como los métodos de particionamiento, métodos jerárquicos y métodos basados en densidad.
- Se reportarán métricas de clasificación como la puntuación de *silhouette*, el índice de Rand (RI) y la información mutua (MI). Los resultados obtenidos de cada método serán analizados y discutidos para evaluar la efectividad de los métodos aplicados.

### C. Visualización

Se tuvo que realizar la extracción de los vídeos, por lo que se utilizó la librería Video Feature Documentation, la cual nos permitió extraer características de los videoclips. Esta librería presenta diversos modelos para la extracción de características, y en este caso, optamos por usar modelos de reconocimiento de acción afines a nuestro proyecto. Dentro de estos modelos, existen varias opciones como: S3D (Kinetics 400), R(2+1)d RGB (IG-65M, Kinetics 400) e I3D-Net RGB + Flujo (Kinetics 400). Cada uno de estos modelos tiene una forma particular de extraer las características de los vídeos. En este proyecto, optamos por probar con dos modelos: S3D y R(2+1)d RGB (r2plus1d-34-32-ig65m-ft-kinetics). La finalidad de esta elección fue comparar cómo afecta la extracción de características en los resultados obtenidos al aplicar técnicas de clustering. Durante el proceso de extracción de características de los vídeos ‘.mp4’, obtuvimos archivos ‘.npy’ que contienen las características procesadas de cada videoclip. Para comprender mejor la distribución y las características del conjunto de datos se realizó un gráfico de barras que muestra la distribución de las etiquetas en el conjunto de datos de entrenamiento (*Y\_train*). En el gráfico de la (Figura 1) se hizo

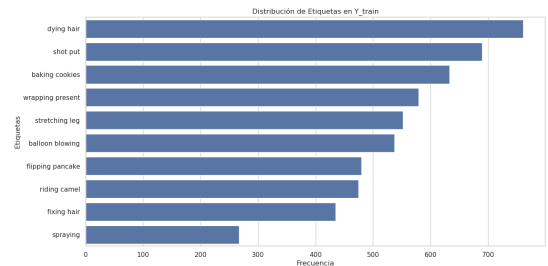


Fig. 1. Distribución de Etiquetas en *Y\_train* con extracción R(2+1)d

la extracción de los vídeos con R(2+1)d y se puede observar que las etiquetas representan diferentes clases de acciones humanas, este gráfico nos permitirá identificar las clases de acciones más y menos representadas.

La barra más larga corresponde a la etiqueta *dying hair*, lo que indica que esta clase de acción tiene la mayor frecuencia. Por otro lado, la etiqueta *spraying* tiene la menor frecuencia entre las etiquetas mostradas. En el gráfico de la (Figura 2)

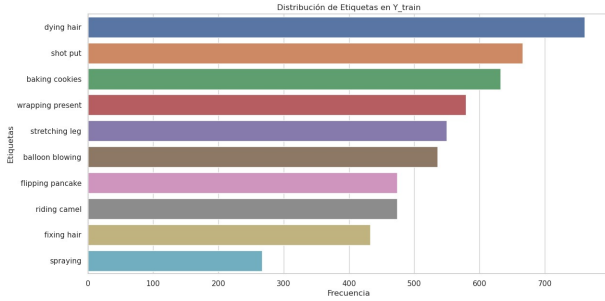


Fig. 2. Distribución de Etiquetas en  $Y_{train}$  con extracción S3D

se hizo la extracción de los vídeos con S3D, la proporción de las acciones fueron parecidas al de las acciones más y menos frecuentes que en la (Figura 1).

Esto se debe a que cada modelo extrae features de distinta manera y en distintos segmentos del vídeo. El modelo R(2+1)d no podía procesar vídeos de bastante corta duración.

#### D. Transformación

Una vez obtenidas las características procesadas de cada videoclip se procedió a reducir la dimensionalidad de las características. La reducción de dimensionalidad es un paso importante en el procesamiento de datos, ya que nos ayuda a evitar características redundantes que no aportan información adicional, nos facilita a la hora tener que visualizar y analizar los datos, elimina el ruido y mejora la calidad de los datos. En este caso, se experimentó con ciertos métodos que nos ayudan a la reducción de la dimensionalidad, como es el caso de PCA, t-SNE, UMAP, TriMAP, PaCMAP. En la figura (Figura

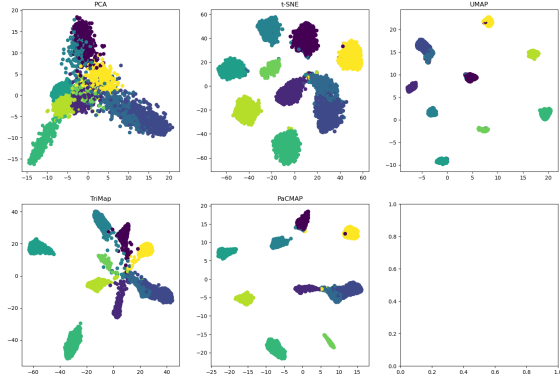


Fig. 3. Métodos de reducción de dimensionalidad con extracción R(2+1)d

??), podemos observar la reducción de dimensionalidad con la

extracción del método R(2+1)d, en el gráfico de PCA muestra una estructura más dispersa y menos agrupada en comparación con otros métodos no lineales. Aunque se pueden observar algunas agrupaciones, los datos no están tan claramente separados como en otros métodos. El gráfico de t-SNE muestra agrupaciones bien definidas y claramente separadas. UMAP también muestra agrupaciones bien definidas y separadas, similar a t-SNE. TriMAP presenta agrupaciones bien definidas, aunque la estructura es un poco diferente a la de t-SNE y UMAP. PaCMAP muestra agrupaciones bien definidas y separadas, similares a UMAP y t-SNE.

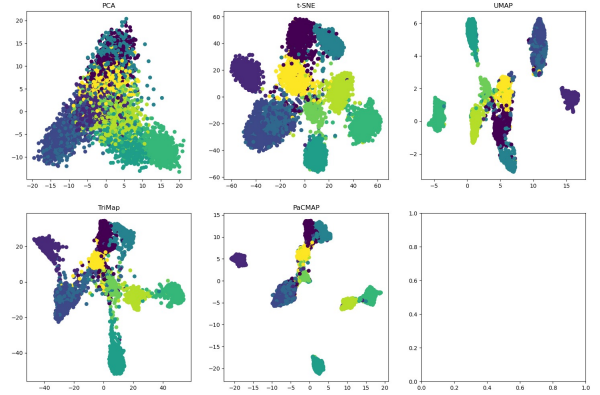


Fig. 4. Métodos de reducción de dimensionalidad con extracción S3D

De mismo modo, al analizar la reducción de dimensionalidad con la extracción del método S3N, podemos observar lo siguiente, el gráfico de PCA muestra una estructura más dispersa; el gráfico de t-SNE muestra agrupaciones un tanto definidas y separadas; UMAP muestra agrupaciones definidas pero a la vez no tan separadas; lo mismo ocurre en TriMAP, y por último PaCMAP presenta agrupaciones definidas y separadas pero no a la perfección. Como conclusión, después de analizar las gráficas obtenidas, optamos por trabajar con el grupo de características extraídas mediante el modelo R(2+1)d. Al comparar este conjunto con las características extraídas y reducidas por S3D, encontramos que R(2+1)d ofrece mejores resultados en términos de reducción de dimensionalidad. En especial, los métodos de reducción de dimensionalidad UMAP y PaCMAP demostraron ser efectivos cuando se aplicaron a las características extraídas por R(2+1)d, proporcionando agrupaciones más claras y bien definidas. Por lo que estos dos métodos de reducción de dimensionalidad nos van a poder facilitar la posterior aplicación de los métodos de clustering.

### III. METODOLOGÍA

En este proyecto de reconocimiento de acciones humanas seguimos una metodología estructurada que incluyó los siguientes pasos:

#### A. Recopilación y Preparación de Datos

Inicialmente, recopilamos el conjunto de datos proporcionado, que consiste en videoclips de acciones humanas con sus respectivas etiquetas. Este conjunto se dividió en

conjuntos de entrenamiento, validación y prueba (*train.csv*, *val.csv*, *test.csv*), y se descargaron los vídeos correspondientes a través de enlaces proporcionados en *videourls.txt*.

#### B. Extracción de Características

Utilizamos la biblioteca Video Feature Documentation para extraer características de los vídeos. Se exploraron dos modelos de extracción: R(2+1)d RGB (r2plus1d-34-32-ig65m-ft-kinetics) y S3D (Kinetics 400). Cada modelo proporciona características específicas que afectan directamente los resultados de los métodos de clustering.

#### C. Reducción de Dimensionalidad

Aplicamos técnicas de reducción de dimensionalidad como PCA, t-SNE, UMAP, TriMAP y PaCMAP a las características extraídas. El objetivo fue reducir la complejidad de los datos manteniendo la estructura subyacente y facilitando la visualización y el clustering posterior.

#### D. Aplicación de Métodos de Clustering

Implementamos varios métodos de clustering, incluyendo K-Means, DBSCAN, Agglomerative Clustering y MeanShift. Cada método fue evaluado con las características reducidas para identificar patrones y estructuras en los datos de acciones humanas.

1) **Clustering con KMeans:** El algoritmo KMeans es una técnica de clustering simple y eficaz. Se ha implementado el algoritmo de KMeans con las siguientes consideraciones:

- **Inicialización de centroides con KMeans++:** Esta técnica mejora la convergencia del algoritmo al elegir centroides iniciales que están más dispersos en el espacio de los datos.
- **Cálculo eficiente de distancias:** Se utiliza la función `np.linalg.norm` para calcular la distancia euclidiana entre puntos.

2) **Clustering con Agglomerative Clustering:** El algoritmo Agglomerative Clustering es una técnica de clustering jerárquico que construye una jerarquía de clusters mediante la fusión iterativa de los pares de clusters más cercanos. Se ha implementado el algoritmo con algunas optimizaciones:

- **Uso de la librería `heapq`:** Para gestionar eficientemente las distancias entre clusters y encontrar los pares de clusters más cercanos de manera rápida. Antes usábamos una versión donde se recalaban las distancias innecesariamente.
- **Cálculo eficiente de distancias:** Se utiliza la función `pdist` de SciPy para calcular la matriz de distancias de manera eficiente.

#### E. Evaluación y Comparación de Resultados

Se utilizaron métricas como la puntuación de Silhouette, el Índice de Rand y la Información Mutua para evaluar la calidad de los clusters obtenidos por cada método de clustering. Se compararon los resultados para determinar cuál método y conjunto de características proporcionaba los clusters más coherentes y significativos, el análisis de esta misma será visto en el apartado de Experimentación.

### IV. IMPLEMENTACIÓN

La implementación del proyecto se llevó a cabo usando un repositorio de Github, usando librerías como Pandas, Seaborn, Numpy Y Matplot para la visualización de datos y operaciones con vectores y matrices multidimensionales. Para realizar la reducción de dimensionalidad se uso librerías como PCA, TSNE, UMAP, TRIMAP, PACMAP. Asimismo otra librerías de métodos de clustering ya implementadas como: KMeans, DBSCAN, AgglomerativeClustering, MeanShift, para la experimentación. Adjuntamos el enlace donde se realizó el trabajo: [Link de la implementación en Github](#)

### V. EXPERIMENTACIÓN

En esta sección se describe detalladamente el enfoque para abordar el proyecto. Se dividió en las siguientes etapas:

#### A. Exploración y Análisis del Conjunto de Datos

Se realizó un análisis exhaustivo del conjunto de datos proporcionado, que incluye los archivos **train.csv**, **val.csv** y **test.csv**. Se revisó la distribución de etiquetas, la cantidad de videos disponibles y se exploraron las características de los vídeos mediante la extracción con modelos específicos como S3D y R(2+1)d RGB. Pero como ya se mencionó, optamos con trabajar con las características extraídas con **R(2+1)d**

#### B. Extracción y Reducción de Dimensionalidad

Se extrajeron características de los vídeos utilizando modelos como S3D y R(2+1)d RGB. Posteriormente, se aplicaron técnicas de reducción de dimensionalidad como PCA, t-SNE, UMAP, TriMAP y PaCMAP para visualizar y reducir la complejidad de los datos obtenidos.

A continuación se muestra una tabla con los Silhouette Scores para diferentes métodos de reducción de dimensionalidad.

Método	Silhouette score
PCA	0.5148
t-SNE	0.6210
UMAP	0.8039
TriMap	0.6693
PaCMAP	0.7132

El análisis de esto revela los siguiente:

- **PCA:** Con un Silhouette Score de 0.5148, muestra una capacidad moderada en cuanto a la reducción de datos, pero es menos efectivo.
- **t-SNE:** Con un Silhouette Score de 0.6210, t-SNE es más eficaz que PCA, capturando mejor las relaciones de los datos.
- **UMAP:** Con el mayor Silhouette Score de 0.8039, UMAP es el método más efectivo, formando clusters bien definidos.
- **TriMap:** Con un Silhouette Score de 0.6693, TriMap es más efectivo que PCA y t-SNE.
- **PaCMAP:** Con un Silhouette Score de 0.7132, PaCMAP es altamente efectivo, comparado con PCA, t-SNE y Trimap.

También, presentamos los gráficos respectivos de cada método de reducción de dimensionalidad.

En conclusión, UMAP y PaCMAP son las opciones ideales para la reducción de dimensionalidad en este conjunto de datos, ya que nos proporcionan clusters bien definidos y coherentes.

### C. Aplicación de técnicas de Clustering

Decidimos aplicar técnicas de clustering como KMeans y Agglomerative. Ambos métodos son efectivos para la agrupación de datos, pero cada uno tiene sus propias características y ventajas. KMeans es conocido por su simplicidad y eficiencia en la partición de datos en clusters esféricos. Por otro lado, Agglomerative es un método jerárquico, flexible en cuanto a la forma de los clusters y puede capturar estructuras de datos más complejas.

Al evaluar nuestra implementación propia, obtuvimos varias métricas de evaluación para ambos métodos: el Silhouette Score, el Adjusted Rand Index (ARI) y el Adjusted Mutual Information (AMI).

- El **Silhouette Score** es una métrica que mide la calidad de los clusters formados. Un puntaje está entre un rango de -1 y 1. Si es 1 indica que los puntos están bien agrupados en sus respectivos clusters, mientras que si es cercano o igual -1 indica que los puntos pueden estar asignados al cluster incorrecto.
- El **Adjusted Rand Index (ARI)** evalúa la similitud entre dos agrupaciones. Un ARI cercano a 1 indica una alta similitud, mientras que un valor cercano a 0 indica una baja similitud.
- El **Adjusted Mutual Information (AMI)** mide la cantidad de información compartida entre las dos agrupaciones. Un AMI cercano a 1 indica que las agrupaciones comparten mucha información, mientras que un valor cercano a 0 indica que no comparten mucha información.

Para verificar la efectividad de nuestra implementación, decidimos comparar el score obtenido con los mismos métodos ya implementados por la librería 'sklearn', lo cual nos daría una referencia confiable al evaluar nuestras implementaciones.

A continuación se muestran los resultados de la evaluación de diferentes métodos de clustering utilizando varias métricas. Estos resultados incluyen los Silhouette Scores, Adjusted Rand Index y Adjusted Mutual Information para nuestros métodos propios implementados y los que ya nos proporciona **Sklearn**, evaluados en los conjuntos de entrenamiento, validación y prueba.

- Silhouette Score

Técnica	Train	Val	Test
KMeans own	0.8529	0.8161	0.8445
Agglomerative own	0.8520	0.8628	0.8501
KMeans sklearn	0.8537	0.8319	0.8463
Agglomerative sklearn	0.8531	0.8320	0.8441

Acerca de esta tabla basada en el Silhouette Score podemos concluir que tanto nuestras implementaciones propias como las de **Sklearn** muestran un rendimiento comparable. Las puntuaciones son muy similares con una diferencia mínima en todos los conjuntos de datos (entrenamiento, validación y prueba), lo que indica que ambos enfoques son efectivos para la agrupación de datos.

- Adjusted Rand Index

Técnica	Train	Val	Test
KMeans own	0.9508	0.8231	-
Agglomerative own	0.9484	0.8524	-
KMeans sklearn	0.9520	0.8806	-
Agglomerative sklearn	0.9491	0.8811	-

Acerca de esta tabla basada en el Adjusted Rand Index, podemos observar que las implementaciones propias y las de **Sklearn** muestran una alta similitud en la agrupación de datos. Donde se presenta puntuaciones cercanas son en el conjunto de entrenamiento, esto nos indica que ambas implementaciones son capaces de identificar grupos similares en los datos.

- Adjusted Mutual Information

Técnica	Train	Val	Test
KMeans own	0.9531	0.9035	-
Agglomerative own	0.8524	0.9204	-
KMeans sklearn	0.9534	0.9000	-
Agglomerative sklearn	0.9517	0.9049	-

Acerca de esta tabla basada en el Adjusted Mutual Information, podemos concluir que nuestras implementaciones propias y las de **Sklearn** son altamente efectivas en la preservación de la información mutua entre las agrupaciones. Las puntuaciones indican que ambas implementaciones mantienen una alta consistencia en los conjuntos de datos de entrenamiento y validación.

Además mostraremos gráficos obtenidos con el método de implementación propia de **Kmeans** y cómo esta definida en nuestros datos de entrenamiento, validación y prueba.(Figura 5) y (Figura 6).

Al analizar las gráficas, podemos observar que existen diferencias significativas entre el uso de UMAP y PaCMAP al aplicar KMeans. En particular, en el conjunto de datos de prueba (*TestData*), se puede ver que con UMAP los clusters están más separados y definidos en comparación con PaCMAP. Del mismo modo, esto se observa también en los conjuntos de datos de entrenamiento (*TrainData*) y validación (*ValData*). En general, UMAP proporciona clusters más separados y definidos.

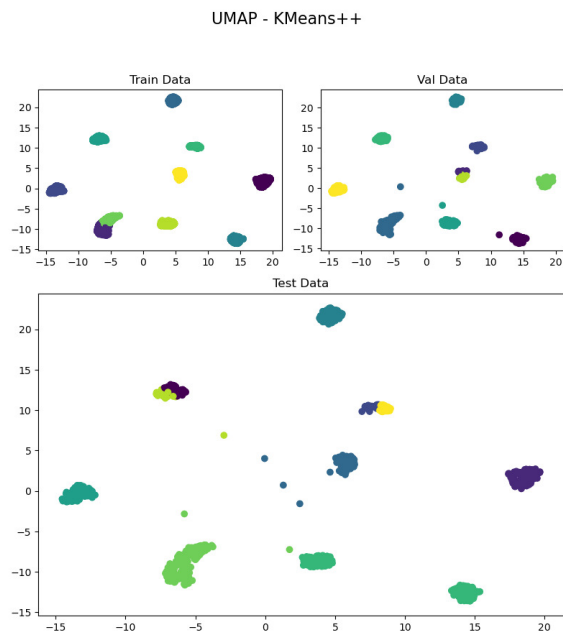


Fig. 5. Resultados de KMeans en el conjunto de prueba, entrenamiento y validación - UMAP

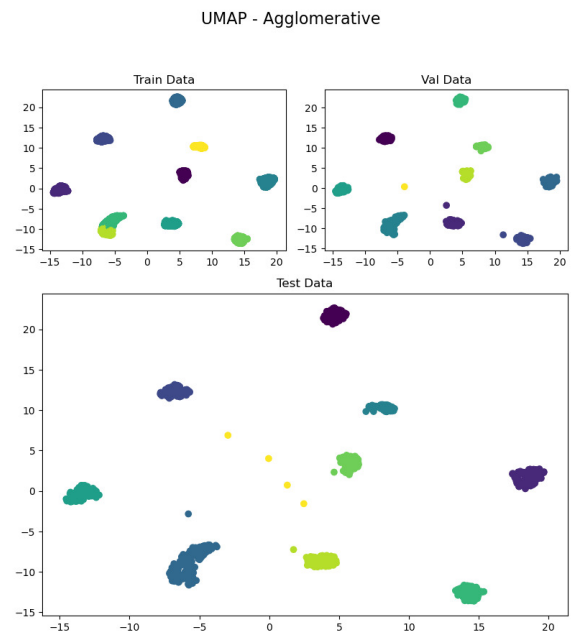


Fig. 7. Resultados de Agglomerative en el conjunto de prueba, entrenamiento y validación - UMAP

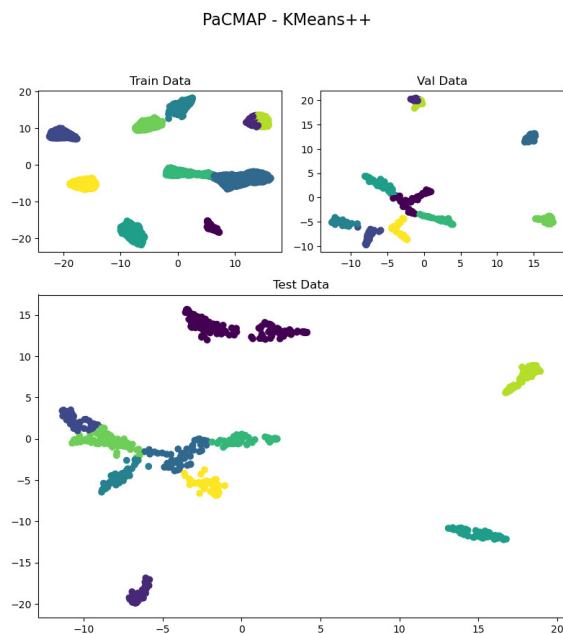


Fig. 6. Resultados de KMeans en el conjunto de prueba, entrenamiento y validación - PaCMAP

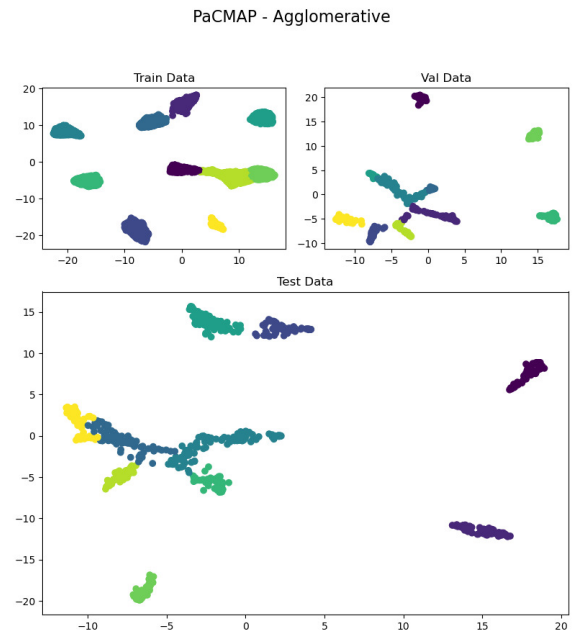


Fig. 8. Resultados de Agglomerative en el conjunto de prueba, entrenamiento y validación - PaCMAP

Por lo tanto, se concluye que UMAP es más eficaz que PaCMAP cuando se usa en combinación con KMeans para la agrupación de datos.

Y también tenemos los gráficos correspondientes del método de implementación de **Agglomerative Clustering**.(Figura 7) y (Figura 8).

Al analizar las gráficas, podemos observar que la diferencia entre usar UMAP y PaCMAP al aplicar Agglomerative no

fue tan notoria. En su mayoría, los resultados fueron similares entre ambos métodos. Por lo tanto, se puede concluir que para Agglomerative se pueden usar ambos métodos de reducción de dimensionalidad, ya que ambos proporcionarán buenos resultados.



## VI. DISCUSIÓN

En esta sección, discutimos los resultados obtenidos y las observaciones derivadas de la implementación y experimentación realizadas en el proyecto de reconocimiento de acciones humanas.

### A. Eficacia de la Extracción y Reducción de Características

Inicialmente, evaluamos la eficacia de los modelos de extracción de características, R(2+1)d RGB y S3D. Observamos que las características extraídas por R(2+1)d RGB mostraron una mejor capacidad para representar las acciones humanas en comparación con S3D. Esto se evidenció en la mayor claridad y separación de los clusters obtenidos después de aplicar técnicas de reducción de dimensionalidad como UMAP y PaCMAP.

### B. Análisis de Métodos de Reducción de Dimensionalidad

Durante la experimentación con métodos de reducción de dimensionalidad (PCA, t-SNE, UMAP, TriMAP, PaCMAP), observamos que UMAP y PaCMAP proporcionaron las proyecciones más significativas y separadas de las características de los videos. Estos métodos no lineales fueron preferidos sobre PCA, que mostró una separación menos clara entre los clusters.

### C. Evaluación de Métodos de Clustering

Se implementaron varios métodos de clustering como K-Means, DBSCAN, Agglomerative Clustering y MeanShift sobre las características reducidas. Observamos que K-Means y Agglomerative Clustering fueron efectivos para identificar grupos coherentes de acciones humanas en los datos, especialmente cuando se combinaron con UMAP para la reducción de dimensionalidad. Para reforzar nuestra decisión decidimos realizar gráficos comparativos de como se comportan nuestros métodos implementados, de acuerdo al método de reducción de dimensionalidad escogido UMAP y PacMap.

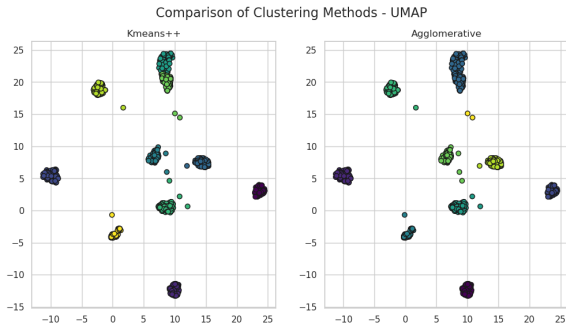


Fig. 9. Comparación de métodos de clustering UMAP

En los gráficos (Figura 9) y (Figura 10) se puede observar la similitud de separación de clustering con ambos métodos, proporcionando así un buen desempeño. Ambas gráficas muestran cómo los datos, reducidos con el método de UMAP y PaCMAP, respectivamente, se agrupan en clusters distintos y bien definidos.

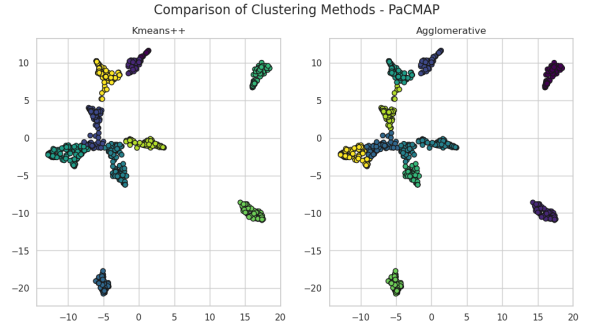


Fig. 10. Comparación de métodos de clustering PaCMAP

### D. Limitaciones y Áreas de Mejora

Es importante destacar que el rendimiento de los métodos de clustering puede verse afectado por la calidad de las características extraídas y la elección de parámetros. En nuestro caso, notamos que algunas clases de acciones menos representadas o más ambiguas podrían beneficiarse de técnicas de mejora de características o ajustes adicionales en los algoritmos de clustering.

### E. Implicaciones y Futuras Direcciones

Este proyecto proporcionó una comprensión profunda del proceso de reconocimiento de acciones humanas utilizando técnicas de aprendizaje automático y análisis de datos.

Para futuras investigaciones, se sugiere explorar métodos avanzados de aprendizaje profundo para la extracción de características y el clustering en grandes conjuntos de datos de videos. Además, sería beneficioso considerar la integración de información temporal y contextual para mejorar la precisión del reconocimiento de acciones en videos de secuencias largas y complejas.

## VII. CONCLUSIÓN

El proyecto demostró la importancia crítica de la selección adecuada de modelos de extracción de características, destacando la efectividad de R(2+1)d RGB para mejorar la representación de acciones humanas en videoclips. La reducción de dimensionalidad utilizando técnicas como UMAP y PaCMAP permitió una visualización y análisis efectivos de las características extraídas, facilitando el clustering y la interpretación de patrones.

Los métodos de clustering como K-Means y Agglomerative mostraron ser eficaces para agrupar videoclips en clusters coherentes de acciones similares. Sin embargo, se identificaron limitaciones en la representación de acciones menos frecuentes o ambiguas, así como en la sensibilidad a los parámetros de los algoritmos utilizados. Para mejorar, se sugiere explorar técnicas avanzadas de aprendizaje profundo y considerar la integración de información temporal y contextual en futuras investigaciones.