DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico Final

Extractores de características y técnicas de agrupamiento

7 de Julio de 2025 — Elementos de reconocimiento visual

Grupo 1

Integrante	LU	Correo electrónico
Padilla, Ramiro	1636/21	ramiromdq123@gmail.com
Cevasco, Jorge	230/23	ccmmshu37@gmail.com
Ranieri, Martina	1118/22	martubranieri@gmail.com

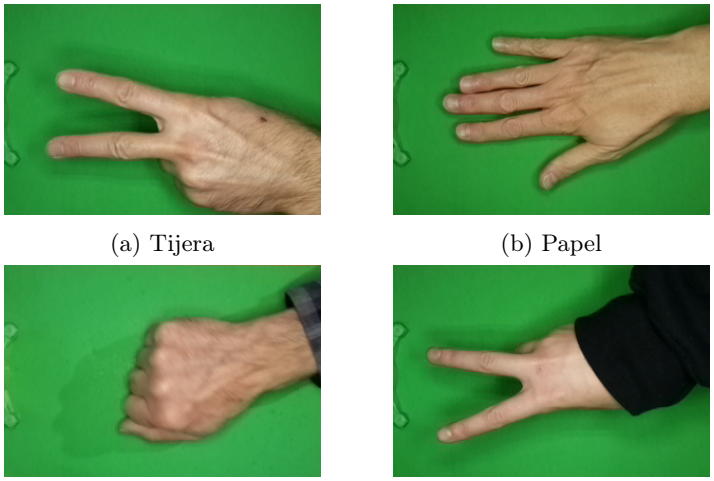
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

Índice

1. Descripción del nuevo dataset elegido

El dataset proviene de Kaggle [1] y se titula 'Rock-Paper-Scissors'. Este dataset fue diseñado para tareas de clasificación de imágenes, donde se entrena un modelo para identificar el gesto de la mano correspondiente a "piedra", "papel" o "tijera". Cada clase cuenta con aproximadamente 700 muestras para entrenamiento y 50 para prueba. Las capturas se realizaron sobre un fondo verde homogéneo, con condiciones relativamente constantes de iluminación y balance de blancos, lo que facilita la segmentación y extracción de características visuales consistentes.

Este conjunto de datos fue seleccionado por su tamaño reducido y su estructura equilibrada, que lo hace especialmente adecuado para implementar y evaluar métodos de reconocimiento visual sin altos costos computacionales. Si bien las imágenes presentan condiciones de iluminación y fondo controladas, lo cual facilita el entrenamiento, la variabilidad en poses, orientación y morfología de las manos introduce un nivel de complejidad realista al problema de clasificación.



(a) Tijera (b) Papel

(c) Roca (d) Tijera

Figura 1: Ejemplos de imágenes de entrenamiento

2. Descripción de los métodos utilizados

2.1. Extractor de características: DAISY

El descriptor DAISY es un método de extracción de características diseñado para imágenes. Genera descriptores robustos y eficientes basados en gradientes. Es útil en contextos donde se necesita saber qué sucede en **toda la imagen**, no solo en puntos interesantes como en SIFT.

La idea principal de DAISY es construir un vector numérico para cada píxel de la imagen, que represente cómo cambian los gradientes (bordes) en su entorno local, usando una estructura espacial en forma de flor. El centro de esta flor es el píxel actual, y alrededor se colocan anillos concéntricos (como pétalos), en los que se calculan histogramas de orientación de gradientes. Cada círculo representa una región del descriptor, y su radio es proporcional a la desviación estándar de los kernels gaussianos usados para suavizar la imagen.

En posiciones específicas de estas regiones, marcadas con un signo '+' en la representación, se muestrean valores de los mapas de orientación previamente convolucionados. El diseño circular con regiones solapadas permite lograr una transición suave entre zonas vecinas del descriptor y otorga robustez frente a rotaciones, ya que los anillos más externos son de mayor tamaño para asegurar un muestreo uniforme.

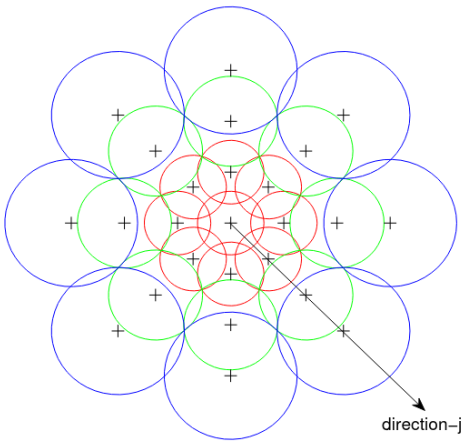


Figura 2: El descriptor DAISY [2]

Para construir el vector DAISY en un píxel primero se calcula la dirección e intensidad del gradiente en cada píxel de la imagen. Esto se hace derivando la imagen en las direcciones horizontal y vertical. Luego, a partir de esos gradientes, se construyen varios mapas de orientación, donde cada mapa representa cuánto gradiente hay en una cierta dirección (por ejemplo, 0°, 45°, 90°, etc.). Cada uno de los mapas de orientación se suaviza utilizando convoluciones gaussianas. Con esto obtenemos mapas convolucionados de orientación para distintas escalas (anillos) y orientaciones. En el centro del píxel se define una región central. Alrededor se colocan anillos concéntricos, y en cada anillo se definen varios puntos equidistantes angularmente. En cada una de las regiones definidas (el centro y los pétalos), se toma una muestra de los mapas de orientación convolucionados en ese punto. Con esa información, se construye un histograma de orientación de gradientes, que indica cuánta intensidad hay en cada dirección. Finalmente, se concatenan todos los histogramas de todas las regiones en un único vector. Este vector es el descriptor DAISY para ese píxel.

En este trabajo, los descriptores fueron extraídos utilizando la función DAISY de scikit image [1] los siguientes parámetros:

- **step = 45:** define la distancia en píxeles entre los puntos del grid donde se computan los descriptores, controlando la densidad del muestreo.
- **radius = 30:** especifica el radio del descriptor, es decir, el tamaño del área de vecindad considerada alrededor de cada punto.
- **rings = 2:** indica la cantidad de anillos concéntricos que rodean el centro del descriptor.
- **histograms = 6:** cantidad de histogramas por anillo.
- **orientations = 8:** número de direcciones de gradiente consideradas por histograma.

Con esta configuración, cada descriptor resultante tiene una dimensión de 104, calculada como $(2\times6+1)\times8$, es decir, un histograma central más doce periféricos, cada uno con ocho orientaciones. En total, se extrajeron 24 descriptores por imagen, correspondientes a 24 centros distribuidos sobre la imagen. Por lo tanto, cada imagen queda representada por una matriz de tamaño 24×104.

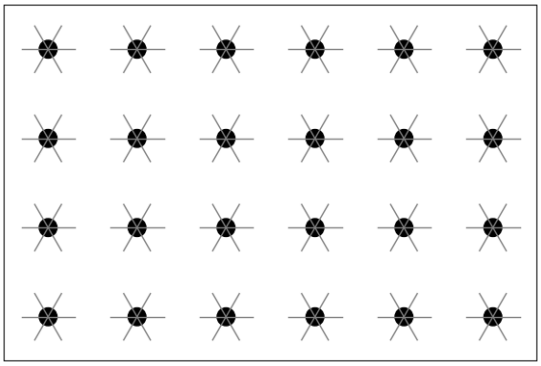


Figura 3: Esquema de la disposición espacial de los 24 descriptores DAISY utilizados.

2.2. Algoritmo de clustering: Agglomerative clustering

El Agglomerative Clustering es un algoritmo de clustering jerárquico que construye una estructura de clústeres en forma de árbol (dendrograma) mediante una estrategia ascendente. Comienza considerando cada punto de datos como un clúster individual y luego fusiona iterativamente los clústeres más similares hasta formar un único clúster que contiene todos los datos.

Al inicio del algoritmo, cada muestra (descriptor visual) comienza como su propio clúster individual. En cada paso, se fusionan dos clústeres más cercanos entre sí según la métrica de distancia y un criterio de enlace. Esto se repite hasta obtener el número deseado de clústeres.

El criterio de enlace usado en este trabajo fue *ward*, el cual fusiona clústeres que minimizan el aumento de la varianza total. La métrica de distancia usada por defecto en la función AgglomerativeClustering de Scikit-Learn [2] es la euclídea y no se puede cambiar.

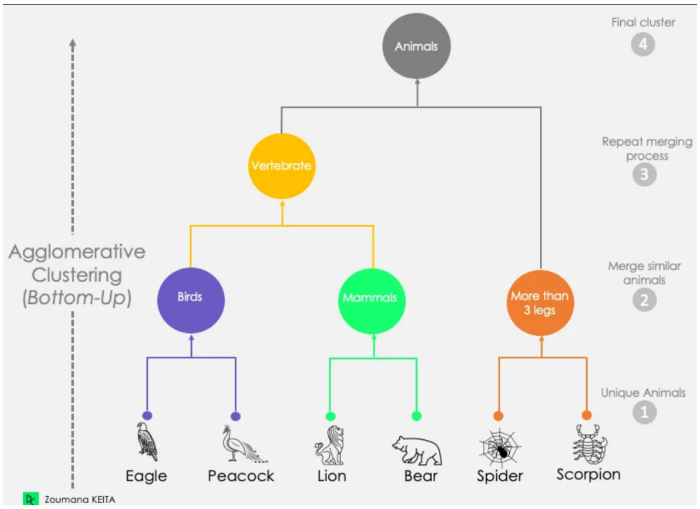


Figura 4: Ejemplo del algoritmo [3]

3. Hipótesis inicial

En primer lugar, se considera que DAISY es apropiado para la extracción de características ya que las imágenes presentan regiones bien definidas donde se concentra la información visual relevante. En particular, las diferencias entre clases es principalmente la posición de los dedos: la presencia de dedos extendidos corresponde a las clases "papel" o "tijera", mientras que su ausencia sugiere la clase "piedra". Esto permite suponer que, DAISY podrá extraer descriptores suficientemente representativos de la clase a partir de esas regiones clave. Además, el fondo uniforme y de color plano (verde) reduce la interferencia de información irrelevante. En este contexto, se espera que DAISY no se vea perjudicado por la falta de textura global o por la uniformidad de los fondos. En comparación con SIFT, que pueden ser más costosos computacionalmente, DAISY generalmente ofrece una alternativa eficiente sin pérdida significativa de rendimiento para este tipo de imágenes controladas.

Por otro lado, Agglomerative Clustering, a pesar de tener una mayor complejidad computacional: $\mathcal{O}(n^3)$, donde n es la cantidad de muestras. Esta elección se justifica por su carácter exhaustivo y determinista, que podría generar vocabularios

[1] DAISY
[2] AgglomerativeClustering

más coherentes y representativos, especialmente en datasets pequeños. En contraposición, MiniBatch K-Means tienen una complejidad mucho más baja: $\mathcal{O}(i * b * k * d)$ donde i es la cantidad máxima de iteraciones, b el tamaño del minibatch, k el número de centroides y d la dimensión de los vectores. Esta diferencia lo vuelve más eficiente para grandes volúmenes de datos. Sin embargo, dado que el presente conjunto de datos es de tamaño moderado y posee imágenes visualmente simples, no se espera que la diferencia en desempeño entre ambos métodos sea sustancial. La elección de Agglomerative Clustering busca explorar si su mayor exhaustividad puede traducirse en una mejora efectiva de la calidad del vocabulario visual.

4. Evaluación y comparativa

k	Batch size	Accuracy BoW	Accuracy TF-IDF	Tiempo Vocab. (s)
5	200	0.765	0.623	1.19
5	500	0.766	0.679	0.45
5	800	0.764	0.604	0.36
5	1000	0.756	0.689	0.36
30	200	0.910	0.916	0.81
30	500	0.897	0.894	0.51
30	800	0.919	0.915	0.49
30	1000	0.910	0.911	0.50
50	200	0.942	0.946	0.80
50	500	0.926	0.939	0.55
50	800	0.937	0.943	0.58
50	1000	0.935	0.940	0.52
100	200	0.955	0.953	1.12
100	500	0.957	0.965	0.86
100	800	0.965	0.968	0.91
100	1000	0.961	0.959	0.91
300	200	0.985	0.984	1.94
300	500	0.984	0.986	1.69
300	800	0.984	0.986	1.90
300	1000	0.988	0.986	1.97

Tabla 1: Desempeño de la combinación SIFT + MiniBatchKMeans con distintos valores.

Clústeres	Accuracy BoW	Accuracy TF-IDF	Tiempo Vocab. (s)
5	0.61	0.546	76.64
10	0.71	0.677	74.58
20	0.86	0.860	79.49
50	0.95	0.953	75.74
100	0.97	0.965	70.96
150	0.98	0.975	73.01
200	0.98	0.983	73.23

Tabla 2: Desempeño del método DAISY + Agglomerative Clustering.

En ambas tablas 1 y 2 se observa que los tiempos de construcción del vocabulario es muy distinta: Con DAISY + Agglomerative Clustering se encuentra entre 70-80 segundos, y para SIFT + MiniBatchKMeans no supera los 2 segundos. En cuanto al desempeño, ambos métodos tienen una efectividad alta y similar, sin mejoras relevantes entre el uso de BoW y TF-IDF.

En cuanto a los parámetros evaluados; el valor de k influyó notablemente pues con k = 30 ya se lograron resultados muy satisfactorios, observándose una tendencia de mejora al aumentar hasta k = 300. En cambio, el parámetro batch size no generó diferencias en el desempeño. Además la cantidad de clústeres en la combinación con Agglomerative Clustering no presentó particular mejora a partir de los 100 clústeres.

Método	Clústeres	k	Batch size	Tiempo Vocab. (s)	Accuracy	F1 Score
MiniBatchKMeans	-	300	1000	1.973	0.9878	0.987719
MiniBatchKMeans	-	300	200	1.936	0.9848	0.984777
MiniBatchKMeans	-	300	800	1.898	0.9838	0.983844
MiniBatchKMeans	-	300	500	1.695	0.9838	0.983796
Agglomerative	200	-	-	73.229	0.9809	0.980833

Tabla 3: Top 5 resultados según precisión (Accuracy) y F1 score. Se comparan métodos de clustering, número de clusters, tamaño de lote ('batch size') y tiempo de construcción del vocabulario.

Entre los mejores resultados, basándonos en accuracy, se destaca MiniBatchKMeans con k = 300 en todas sus configuraciones de batch size. Aunque el mejor resultado obtenido con Agglomerative Clustering presentó un valor de accuracy apenas inferior, el tiempo de construcción del vocabulario fue más de 40 veces mayor, lo que lo hace menos eficiente para grandes volúmenes de datos.

Por este motivo, para la evaluación comparativa final sobre el conjunto de testing se seleccionaron los siguientes parámetros:

- MiniBatchKMeans: k = 300, batch size = 1000
- Agglomerative Clustering: número de clústeres = 200

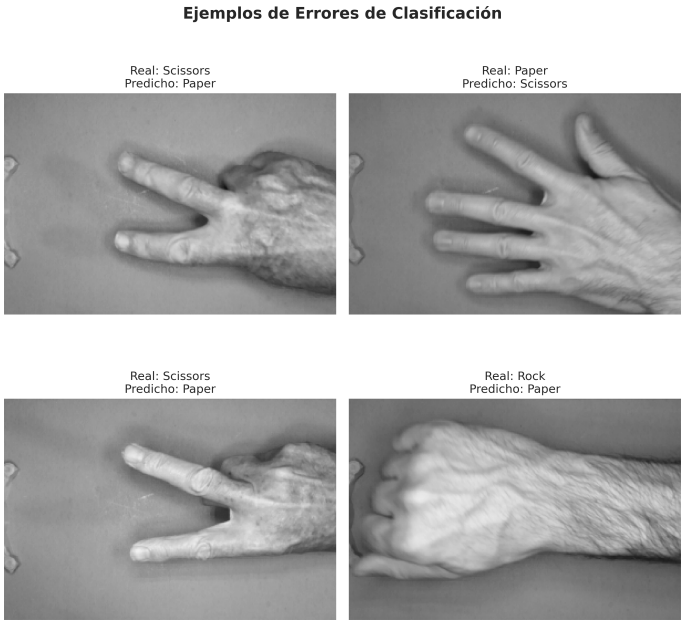
Método	Accuracy	Precisión	Recall	F1 Score
MiniBatchKMeans	0.973	0.974	0.973	0.973
Agglomerative	0.987	0.987	0.987	0.987

Tabla 4: Perfomance de ambos métodos con los mejores parámetros.

Los resultados obtenidos se presentan en la Tabla 4, donde se observan métricas de desempeño muy elevadas en ambos casos. El método basado en DAISY obtuvo una ligera ventaja en todas las métricas, alcanzando una exactitud del 98,7 %, frente al 97,3 % logrado por la configuración con SIFT. Si bien la diferencia porcentual es reducida, en aplicaciones sensibles puede ser significativa.

Los errores evidenciados en la Figura 5 se concentran en clases con alta similitud visual, lo que sugiere que ambos métodos son sensibles a pequeñas variaciones en la forma y orientación de la mano o sombras de la imagen. Sin embargo, DAISY muestra menor dispersión en las confusiones, lo que indica una mejor discriminación de patrones locales.

(a) DAISY + Agglomerative Clustering



(b) SIFT + MiniBatchKMeans

Figura 5: Comparación de errores: DAISY vs SIFT

La Figura 6 muestra las diferencias en la forma de describir la imagen entre SIFT y DAISY. SIFT detecta un conjunto reducido de puntos clave en zonas con alta variación local, lo que permite capturar características distintivas de manera eficiente y robusta frente a cambios de escala y rotación, aunque con cobertura limitada. Además, la cantidad de descriptores varía según el contenido de la imagen. En cambio, DAISY genera descriptores distribuidos de forma regular, manteniendo siempre la misma cantidad independientemente del contenido. Esta estrategia asegura que incluso las áreas con poca textura queden representadas, proporcionando una descripción más completa y detallada de la escena, aunque con un mayor costo computacional.

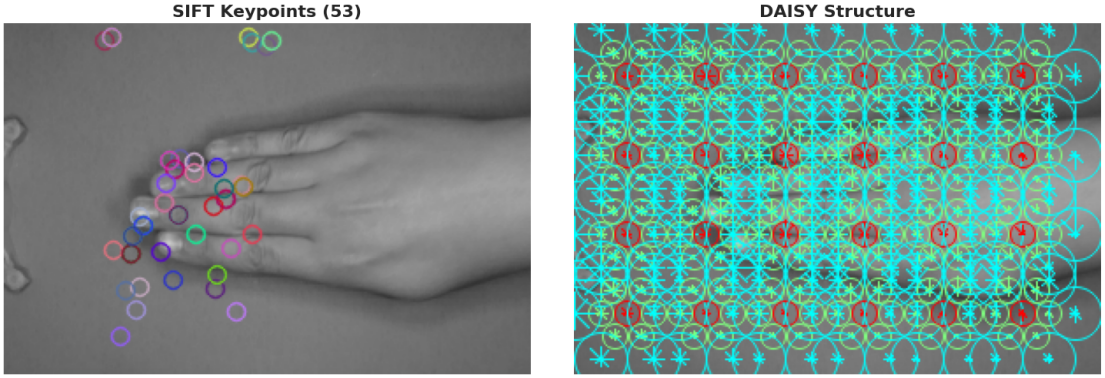


Figura 6

5. Conclusiones

Los experimentos realizados permiten extraer varias conclusiones relevantes. Si bien DAISY presenta un menor costo computacional que SIFT, su combinación con Agglomerative Clustering resultó poco ventajosa. Este método jerárquico incrementó significativamente el tiempo de procesamiento sin aportar mejoras en el rendimiento. La simplicidad del conjunto de datos hizo que un algoritmo de clustering más eficiente, como MiniBatchKMeans, alcanzara resultados positivos sin comprometer la calidad de la clasificación.

Por otra parte, el uso de keypoints en grilla en DAISY generó numerosos descriptores ubicados sobre regiones homogéneas del fondo, lo cual añadió un costo innecesario al proceso de agrupamiento. Esto resalta la importancia de una adecuada selección de puntos de interés, especialmente en escenarios con fondos uniformes.

En términos prácticos, aunque la diferencia en accuracy entre DAISY y SIFT fue de solo 1,4 %, en aplicaciones donde la precisión es crítica *(como sistemas de control gestual para entornos médicos)* esta mejora puede ser significativa. Además, MiniBatchKMeans demostró ser considerablemente más eficiente, reduciendo el tiempo de procesamiento en órdenes de magnitud, convirtiéndolo en una alternativa adecuada para sistemas en tiempo real.

Finalmente, se sugiere como trabajo futuro explorar la reducción de descriptores irrelevantes, y la combinación de diferentes descriptores locales, con el fin de incrementar aún más la precisión sin comprometer la complejidad computacional.

Referencias

[1] URL: https://www.kaggle.com/datasets/drgfreeman/rockpaperscissors?select=scissors (visitado 07-07-2025).

[2] URL: https://www.researchgate.net/publication/42345292_Daisy_An_Efficient_Dense_Descriptor_Applied_to_Wide_Baseline_Stereo (visitado 07-07-2025).

[3] URL: https://medium.com/@prasanth32888/agglomerative-hierarchical-clustering-ahc-e9e7a48cb042 (visitado 07-07-2025).