Caracterización del desempeño y el coste computacional en algoritmos de similitud en bancos de imágenes de grafiti

Alberto Luengo Román

Máster en Sistemas Inteligentes
Universidad de Salamanca

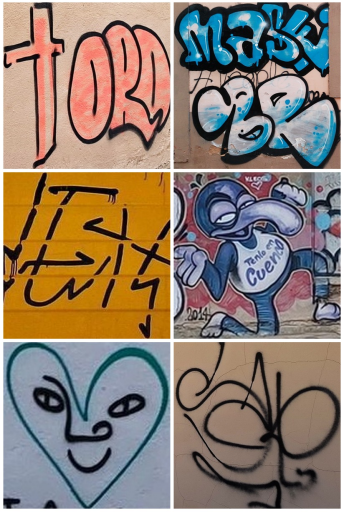
Junio 2026

VNiVERSiDAD D SALAMANCA

Índice

Contexto

- El grafiti no autorizado es un fenómeno urbano recurrente que requiere atención continua por parte de las administraciones
- El seguimiento manual es **costoso e inconsistente**
- Un agrupamiento automático por similitud visual facilitaría la trazabilidad de autores o la priorización de retirada basada en el contenido visual



Problema y objetivo

Problema: dado un banco de fotografías de grafiti sin etiquetar, agruparlas por similitud visual de forma automática

Objetivo: diseñar y evaluar un **sistema modular** que permita:

- Comparar combinaciones de extractores de características, técnicas de reducción y algoritmos de agrupamiento
- Caracterizar el **coste computacional** por etapa en función del tamaño del banco
- Evaluar la calidad del agrupamiento con métricas intrínsecas y extrínsecas

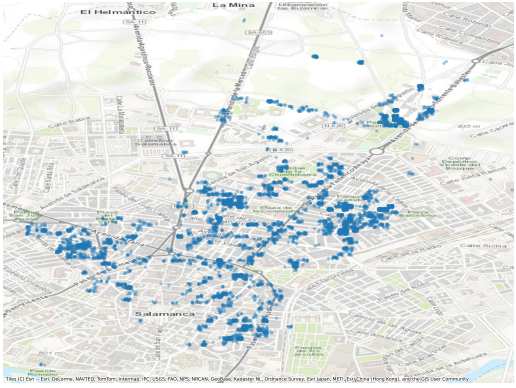
Trabajos relacionados

Trabajo	**Tarea**	**Limitación**
Fogaça et al. [1]	Detección y clasificación	Supervisado, sin agrupamiento
Tokuda et al. [2]	Cuantificación geográfica	Sin similitud visual
García García et al. [3]	Agrupamiento por color	Sin descriptores profundos
Este trabajo	**Agrupamiento + coste**	—

Ningún trabajo previo combina **descriptores profundos**, agrupamiento clásico y caracterización del **coste computacional** sobre imágenes de grafiti.

Conjunto de datos

- **Banco principal:** 6681 fotografías de grafiti en **Salamanca**. Provenientes de un proyecto de la Dra. González Arrieta
- Capturadas con un teléfono móvil ($\sim$10.6 MP), sin postprocesado. Captura masiva en otoño de 2022
- Alta variabilidad entre fotografías: iluminación, perspectiva, oclusión, escena, etc. Muchas de las fotografías contienen **múltiples grafitis**
- **Banco auxiliar:** 1106 imágenes de **Cuenca** cedidas por StopGrafiti. Utilizadas solo para ajuste fino, evitando solapamiento entrenamiento/evaluación



Ubicación de las imágenes con GPS

Anotaciones y subconjuntos

Ninguno de los bancos está etiquetado. Para poder obtener métricas que representen la calidad real del agrupamiento, se lleva a cabo un proceso de recorte y anotación manual

Cuatro categorías estilísticas: *tag*, *throw-up*, *piece*, *character*

tag

throw-up

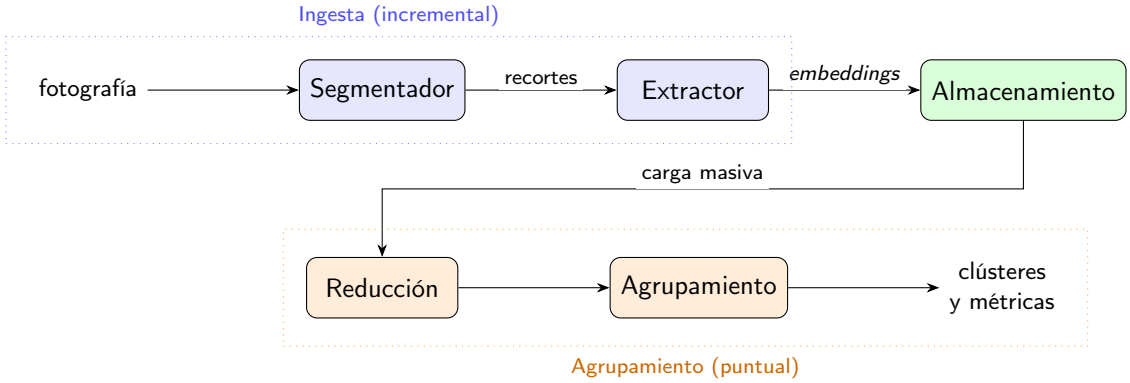
piece

character

Subconjunto	**Procedencia**	**Tamaño**
Estilo (entr.)	StopGrafiti + Salamanca	385
Estilo (eval.)	Banco completo	294
Autoría (entr.)	StopGrafiti	321
Autoría (eval.)	Salamanca	185
Detección	Ambos bancos	447
No superv.	Salamanca (disjunto)	6416

Tamaño en recortes, salvo detección y no superv. (imágenes).
Entrenamiento y evaluación nunca se solapan.

Sistema: dos fases acopladas por el almacenamiento



Cinco etapas **intercambiables** tras una interfaz común

Segmentación

- Etapa **opcional** para recortar grafitis de la escena
- Detector **YOLO ajustado** sobre grafiti (447 imágenes, ∼650 cajas; partida de pesos COCO)
- **Fusión** de cajas por solapamiento y proximidad
- **Margen** de contexto configurable antes de recortar
- Se trata de una **extentsión** a lo especificado en la propuesta original con el objetivo de obtener mejores resultados del dataset dado

Original → detecciones → fusión → margen → recortes

Extracción de características

Modelos **preentrenados y congelados**: se toma su representación latente como vector de características.

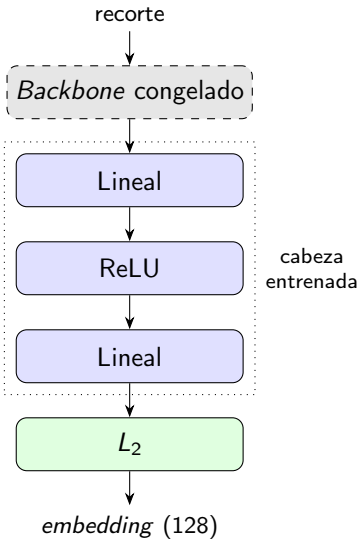
- CNN: se elimina la capa de clasificación
- *Transformers* (DINOv2, CLIP): ya devuelven un vector
- Vectores de entre 384 y 4096 dimensiones

Modelo	**Tipo**	**Dim.**
ResNet50	CNN residual	2048
VGG16	CNN profunda	4096
InceptionV3	CNN multi-rama	2048
MobileNetV3	CNN ligera	1280
DINOv2 ViT-S/14	*Vis. Trans.*	384
CLIP ViT-B/32	*Vis. Trans.*	512

Extracción de características: ajuste al dominio

Cabezas de proyección entrenadas sobre grafiti utilizando *transfer learning* [4]:

- *Backbone* **congelado**, solo se entrena una cabeza ligera (Linear→ReLU→Linear→L_2)
- Bases: **MobileNetV3** y **DINOv2**
- Tareas: **autoría** (pérdida de tripletes) y **estilo** (*SupCon*)
- En total 4 cabezas

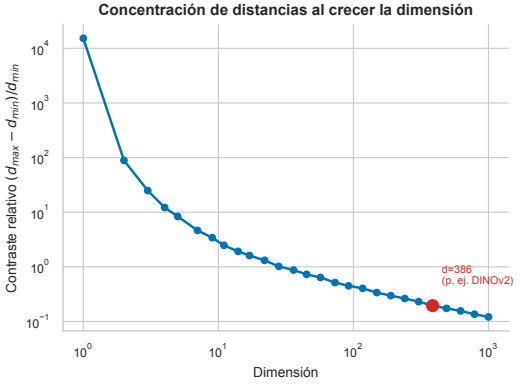


Reducción de dimensionalidad

Paso **opcional** entre almacenamiento y agrupamiento: reduce coste y mitiga la maldición de la dimensionalidad [5].

- **PCA** (lineal): direcciones de máxima varianza
- **Kernel PCA** (no lineal): PCA en un espacio de características implícito
- **Isomap** (no lineal): preserva distancias geodésicas sobre el grafo de vecinos
- **UMAP** (no lineal): preserva la estructura local de vecindad

Las métricas de calidad se calculan sobre los *embeddings* **originales** (sin reducir).

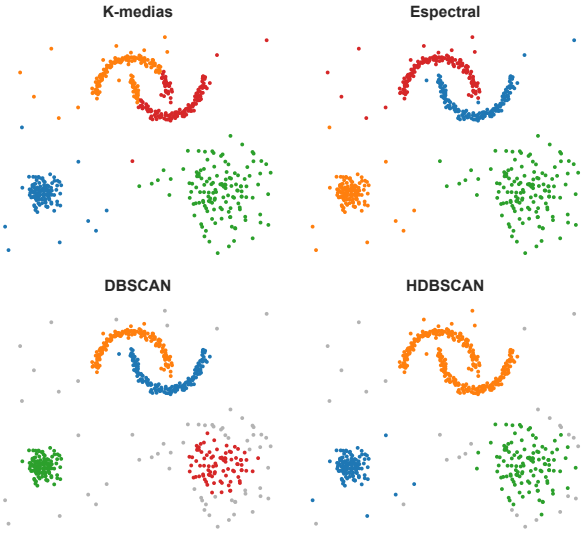


Conforme **aumenta** el número de dimensiones (hacia la derecha) la diferencia entre la distancia más cercana y la más lejana tiende a 0.

Agrupamiento

Ocho algoritmos de varias familias tras una interfaz común:

- **Particional:** K-medias
- **Densidad:** DBSCAN, OPTICS, HDBSCAN
- **Jerárquico:** Aglomerativo
- **Grafo:** Espectral
- **Probabilístico:** GMM
- **Mensajes:** Affinity Propagation



Diseño experimental

Diez barridos; cada uno varía **un solo eje** sobre una configuración base, para comparaciones justas.

Objetivo	Exp.	Eje variado
Calibración	E1	Extractor
	E2	Reducción
	E3–E4	Agrupamiento / `mcs`
	E5	Segmentador
Coste	E7	Tamaño del banco N
	E6, E9	*Backend* / índice HNSW
Supervisado	E8a	Estilo (4 clases)
	E8b	Autoría (87 autores)

N hasta 6416; K=3–5 repeticiones por celda.

Métricas de calidad

- **Intrínsecas** (sin etiquetas): silueta, Calinski–Harabasz, Davies–Bouldin, ruido, balance de tamaños
- **Extrínsecas** (con etiquetas): ARI, NMI, F1 balanceada, MAP, Recall. Calculadas **sobre recortes, sin segmentador**.
- **Coste**: tiempo de pared y memoria pico por etapa

Configuración base resultante

DINOv2 cabeza-estilo + UMAP-10 + HDBSCAN (`mcs`=5) + segmentador identidad

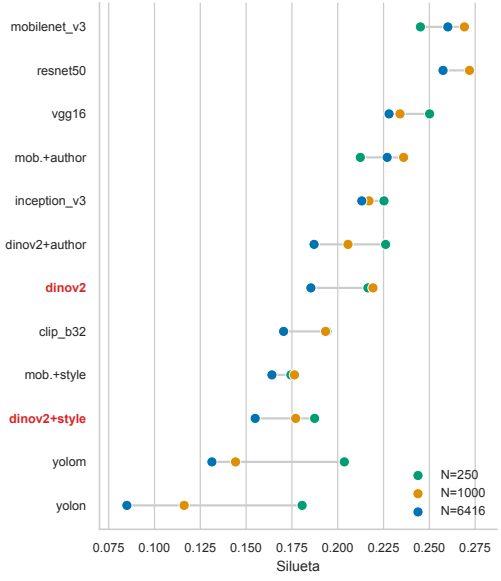
Las métricas intrínsecas se leen **en conjunto**: aisladas premian particiones degeneradas.

Resultados: segmentador (E5)

Segmentador	r/img	ruido↓	sil↑	ingesta
identidad	1.00	**0.199**	**0.151**	**1236**
YOLO11s	1.03	0.225	0.127	1777
YOLO11m	1.03	0.230	0.131	1812
YOLO11m *ft*	2.39	0.298	0.122	2152

- **Identidad gana en todo**: mejor silueta, menor ruido y menor coste de ingesta
- Detectores base (YOLO11s/m): ∼1 caja/imagen, reproducen el banco original; **calidad marginal o peor** además de ser más costosos
- YOLO11m *ft*: lidera a N=1000, pero a escala completa produce 2,4 recortes/imagen y **degenera** (30 % ruido, 1049 clústeres)
- El detector alcanza un recall 0,74 y mAP@.5–.95 0,60. El segmentador **pierde uno de cada cuatro grafitis**, y los que detecta no los ajusta bien del todo. Las cabezas de estilo **no toleran bien la variabilidad** de los recortes.

Resultados: extractores de características (E1)



E1: silueta por extractor a tres tamaños de banco.

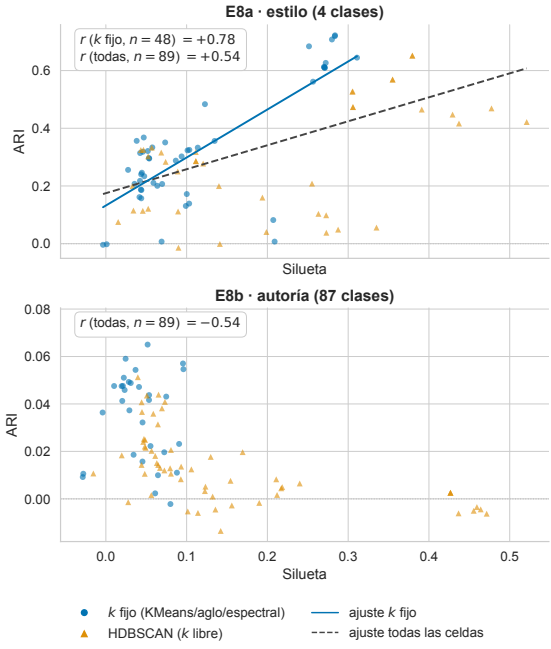
Extractor	sil	cls	ruido	balance
mobilenet_v3	**0.260**	820	**0.078**	**0.52**
resnet50	0.257	794	0.086	0.71
vgg16	0.228	773	0.095	0.54
inception_v3	0.213	724	0.118	0.64
dinov2_vits14	0.185	632	0.123	0.92
clip_vit_b32	0.170	734	0.150	0.74
mob. estilo	0.164	629	0.214	0.58
dinov2_graffiti_style	0.155	562	0.202	0.74
yolom	0.131	480	0.253	0.74
yolon	0.085	348	0.341	0.81

- Las CNN realizan la mejor agrupación, pero no hay seguridad de que estas redes midan similitud entre grafitis y no entre escenas
- La silueta se utiliza como criterio de ordenación principal **dentro** del barrido, pero no decide el extractor final

Resultados: validación supervisada (E8) y elección del extractor

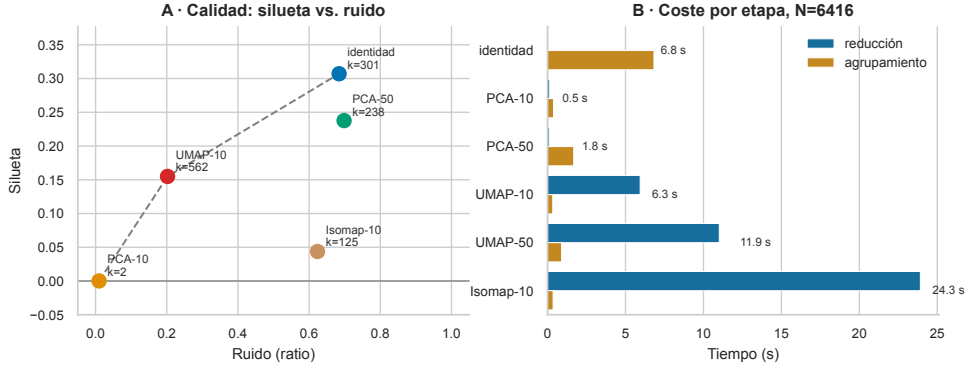
Extractor	ARI	NMI	F1
DINOv2 estilo	**0.722**	**0.678**	**0.806**
MobileNet estilo	0.627	0.539	0.740
CLIP ViT-B/32	0.368	0.392	0.556
DINOv2 (base)	0.356	0.374	0.558
ResNet50	0.334	0.277	0.548

- La cabeza de estilo DINOv2 **dobla** el ARI de cualquier extractor sin ajustar.
- **E8b (autoría, 87 autores):** ARI $\approx$ 0,04–0,08. Con $\sim$2 recortes por autor, 87 clústeres no son detectables; es una tarea de *recuperación abierta*, no de partición

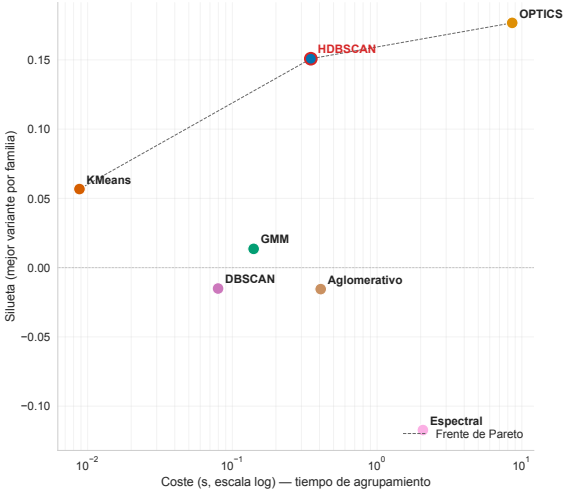


Resultados: reducción de dimensionalidad (E2)

- Sin reducción: presenta la mejor silueta del experimento, pero a costa de un 68 % de ruido. El coste de memoria y tiempo se trasladan de la etapa de reducción al agrupamiento
- PCA solo produce agrupamientos degenerados: dos clústeres y silueta nula en PCA-10 y un 69,8 % de ruido en PCA-50
- **UMAP**: Mejor equilibrio entre silueta y ruido. UMAP-50 da resultados prácticamente idénticos a UMAP-10, pero prácticamente doblando el coste de memoria y tiempo



Resultados: algoritmo de agrupamiento (E3)



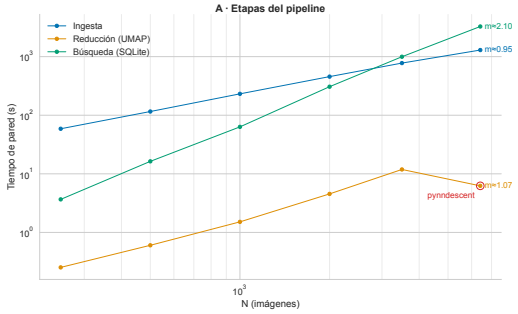
E3: frente de Pareto calidad-coste por familia (N=6416).

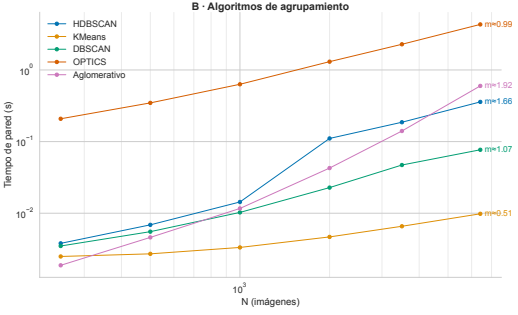
- **HDBSCAN**: silueta competitiva, ruido moderado, tamaños balanceados, coste despreciable (0,35 s)
- **OPTICS** iguala en calidad, $\sim$25$\times$ más tiempo, $\sim$2 500$\times$ más memoria
- **K-medias**: más rápido, pero silueta 0.094 puntos inferior. Mejor opción **sin ruido**
- Métricas **aisladas** engañan: silueta y CH premian k=2 trivial $\Rightarrow$ criterio conjunto

Pipeline base

DINOv2 estilo + UMAP-10 + HDBSCAN (`mcs=5`) + segmentador identidad

Resultados: coste del sistema completo (E7)





Pendiente log–log observada (vs. teórica):

- **Ingesta** (DINOv2): 0,95 $\approx O(n)$; dominante a esta escala
- **Reducción** (UMAP): 1,15, **despreciable** (< 5 s)
- **Agrupamiento**: K-medias 0,43, HDBSCAN 1,53, aglomerativo 1,77. A esta escala es más aparente el **coste de memoria** que el tiempo para OPTICS (763 MB) y aglomerativo (314 MB, $O(n^2)$)
- **Cuello**: búsqueda SQLite 2,10 $\approx O(n^2)$ por pasada, **54.4 min** a N=6416

Rango de N estrecho: pendientes *observadas*.

Resultados: búsqueda por similitud (E6, E9)

N	***Backend***	**cons./s**	**R@5**
500	SQLite	32.8	–
500	pgvector exacto	619	–
500	pgvector + HNSW	608	1.000
2500	SQLite	5.9	–
2500	pgvector exacto	427	–
2500	pgvector + HNSW	604	1.000
6416	SQLite	2.3	–
6416	pgvector exacto	173	–
6416	**pgvector + HNSW**	**560**	**1.000**

E6: *throughput* (consultas/s) y *recall*@5 por *backend*.

- **SQLite**: Búsqueda completa $O(n^2)$, **inviable** a escala (54,4 min a N=6416)
- **PostgreSQL** + **pgvector**: $\sim$2 órdenes de magnitud sobre SQLite (elimina el *round-trip* de Python)
- **HNSW**: único *throughput* que se **mantiene *constante* con** N ($\sim$560 cons./s) mientras el exacto y SQLite se desploman; a N=6416, 3,2$\times$ sobre el exacto y 240$\times$ sobre SQLite, **sin** pérdida de *recall* (R@5 = 1,000). La construcción del índice se amortiza en **una** consulta.

Conclusiones: Hallazgos principales

- Se ha desarrollado un sistema **modular** que compara extractores, reducciones y agrupadores bajo una misma interfaz
- **Pipeline base**: cabeza de estilo DINOv2 + UMAP-10 + HDBSCAN; valida la cabeza por su calidad **supervisada**, no la intrínseca
- El **coste** lo dominan ingesta y, sobre todo, la búsqueda $O(n^2)$; **HNSW** la elimina sin perder calidad
- La reducción de dimensionalidad es **indispensable** para calidad y coste

Conclusiones: Limitaciones y trabajo futuro

- Rango de N estrecho (≤ 6416): difícil separar $O(n \log n)$ de $O(n^2)$
- Datos de **autoría** insuficientes $\Rightarrow$ ampliar y anotar el banco
- Preentrenamiento no supervisado antes de las cabezas supervisadas
- Pasar de *batch* a **ingesta incremental** (índice y espacio reducido)

Gracias

Alberto Luengo Román
luengor@usal.es

Referencias I

[1] Joana Fogaça, Tomás Brandão y João C. Ferreira. «Deep Learning-Based Graffiti Detection: A Study Using Images from the Streets of Lisbon». Inglés. En: *Applied Sciences* 13.4 (ene. de 2023), pág. 2249. ISSN: 2076-3417. DOI: 10.3390/app13042249. URL: https://www.mdpi.com/2076-3417/13/4/2249.

[2] Eric K. Tokuda, Roberto M. Cesar y Claudio T. Silva. «Quantifying the presence of graffiti in urban environments». Inglés. En: *2019 Ieee International Conference on Big Data and Smart Computing (bigcomp)*. Num Pages: 4 Series Title: International Conference on Big Data and Smart Computing Web of Science ID: WOS:000469779800067. New York: IEEE, 2019, págs. 405-408. ISBN: 978-1-5386-7789-6. DOI: 10.1109/bigcomp.2019.8679113.

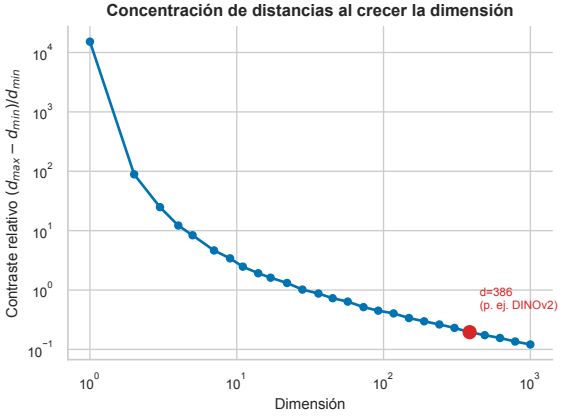
[3] Miguel García García et al. «Graffiti Identification Using Color Analysis: An Approach Based on K-Means Clustering». Inglés. En: *Distributed Computing and Artificial Intelligence, Special Sessions II, 21st International Conference*. Ed. por Goreti Marreiros et al. Cham: Springer Nature Switzerland, 2025, págs. 323-328. ISBN: 978-3-031-80946-0. DOI: 10.1007/978-3-031-80946-0_35.

[4] Sinno Jialin Pan y Qiang Yang. «A Survey on Transfer Learning». En: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (oct. de 2010), págs. 1345-1359. ISSN: 1558-2191. DOI: 10.1109/TKDE.2009.191. URL: https://ieeexplore.ieee.org/document/5288526.

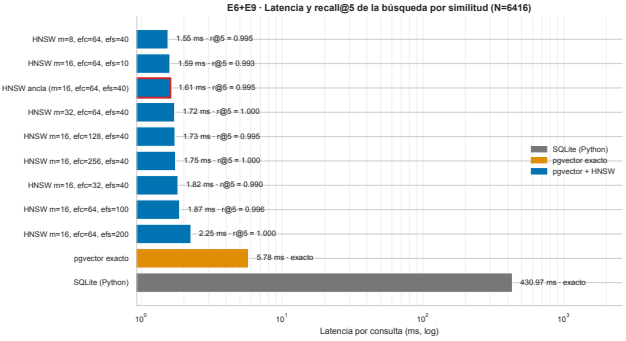
[5] Charu C. Aggarwal, Alexander Hinneburg y Daniel A. Keim. «On the Surprising Behavior of Distance Metrics in High Dimensional Space». Inglés. En: *Database Theory — ICDT 2001*. Ed. por Jan Van den Bussche y Victor Vianu. Berlin, Heidelberg: Springer, 2001, págs. 420-434. ISBN: 978-3-540-44503-6. DOI: 10.1007/3-540-44503-X_27.

Respaldo: la maldición de la dimensionalidad

- Con cosenos sobre vectores unitarios la distancia sigue siendo informativa, pero al crecer d el contraste vecino cercano/lejano se desvanece [5]
- Por eso la reducción (UMAP-10) **mejora** el agrupamiento y abarata las etapas siguientes
- Las métricas de calidad se calculan sobre los *embeddings* **originales** (L2-normalizados) para comparar entre familias
- A la derecha, la relación entre la diferencia de distancia entre el vecino más cercano y más lejano y la distancia al más cercano, que tiende a cero a medida que d crece.

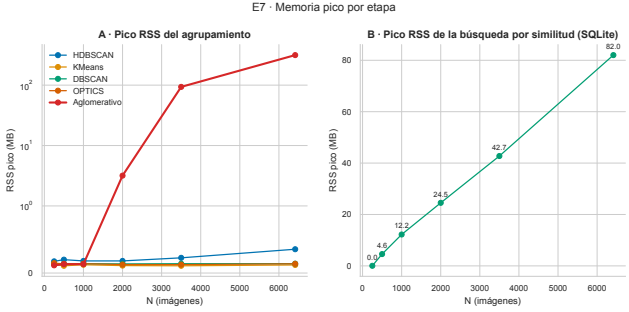


Respaldo: *backend* de búsqueda y configuración HNSW



- Por debajo de $N \approx 1000$, pgvector exacto y HNSW son indistinguibles
- Por encima, HNSW se separa: su latencia se mantiene plana mientras la del exacto crece; a N=6416 es 3,2× el exacto y 240× SQLite
- E9: ninguna configuración baja de *recall* 0,99; subir `m` o `ef_construction` llega a 1,0 sin penalizar latencia

Respaldo: memoria pico por etapa



- El agrupamiento **aglomerativo** materializa la matriz de distancias $O(n^2)$ y se descarta por memoria antes que por tiempo
- HDBSCAN, DBSCAN y OPTICS se mantienen sublineales
- Refuerza la elección de HDBSCAN: coste y memoria mínimos

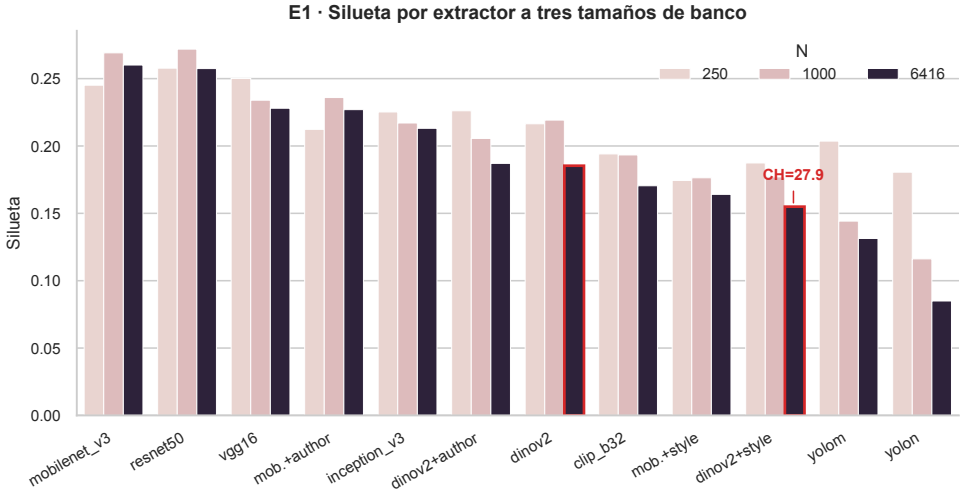
Respaldo: ¿por qué silueta?

- **Ninguna métrica se lee sola**. La comparación de *extractores* se ancla en métricas *extrínsecas* (ARI/NMI), que no asumen forma; la silueta es el *proxy* para cuando no hay etiquetas.
- **Por qué la silueta como eje único**: es el único índice interno *acotado* ($[-1, 1]$) y comparable entre distintos k y entre familias de algoritmos. Calinski–Harabasz crece con n y d; Davies–Bouldin no está acotado $\Rightarrow$ malos para comparar entre ejecuciones.
- Aún penalizada por la silueta, HDBSCAN llega a las mejores configuraciones $\Rightarrow$ el ranking es robusto al sesgo.
- El ruido (-1) se **excluye** de la silueta: corta en ambos sentidos (puede *inflar* la de densidad); se reporta `noise_ratio` aparte para que sea visible.

Respaldo: ¿por qué silueta? (¿sesga SupCon la métrica?)

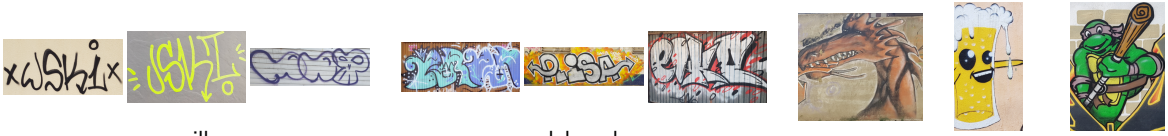
- **Distinción clave**: el ARI mide *directamente* el objetivo (acuerdo con las etiquetas reales); la silueta es un *proxy* geométrico de validez condicional.
- Optimizar hacia el objetivo es legítimo (es aprender). Sólo sería «Goodhart» si el proxy subiera *sin* subir el objetivo (ARI).
- **Evidencia (E1)**: las CNN *sin ajustar* lideran la silueta pero *no* la calidad supervisada $\Rightarrow$ la silueta no favorece a los modelos ajustados; si acaso, lo contrario.
- Por eso el ranking de extractores descansa en el ARI, no en márgenes de silueta: no comparables entre geometrías moldeadas por objetivos distintos.
- (DBCV es un índice interno apto para densidad, pero sólo puntúa *clusterers* de densidad $\Rightarrow$ no sirve de eje común con KMeans/GMM.)

Respaldo: silueta por extractor a tres escalas



Las CNN de ImageNet lideran la silueta *intrínseca*, pero no la calidad *supervisada* (E8a). La métrica intrínseca mide apariencia, no semántica de grafiti.

Respaldo: ejemplos de clústeres de estilo (E8a)



sencillo elaborado *characters*

HDBSCAN colapsa los 4 estilos en 3 grupos coherentes: une *tag*/*throw-up* en «sencillo».

Respaldo: autoría (E8b) bajo recuperación, no partición

Extractor	MAP@5	Rec@5	MRR
CLIP ViT-B/32	0.252	0.268	0.262
DINOv2 (base)	0.248	0.268	0.251
ResNet50	0.246	0.236	0.256
DINOv2 autor	0.228	0.254	0.232
MobileNet autor	0.217	0.233	0.222

E8b: mejor celda por extractor (185 recortes, 87 autores).

- Bajo recuperación sí hay señal: MRR $\approx$ 0,25, 8$\times$ el azar ($\approx$ 0,03)
- Pero esa señal es **similitud genérica** (misma sesión, muro, color): las cabezas de autor **no superan** a los extractores sin ajustar, incluso quedan algo por debajo
- El ajuste fino entrenado *solo en Cuenca* no transfiere a Salamanca (cambio de dominio + autores no vistos). Asimetría con estilo, que se entrenó mixto y se evalúa en dominio