Prefacio

Título

Hola, soy... y voy a defender mi trabajo ...

Índice

La presentación sigue la misma estructura que el artículo escrito: Intro con arte Conjunto de datos Metodología, se explica el sistema implementado Resultados Conclusiones

Introducción

Contexto

Empezamos con un poco de contexto.

El grafiti no autorizado es un fenómeno urbano conocido por todos y que requiere un esfuerzo constante por parte de las autoridades para su control.

El seguimiento y clasificación manual de los grafitis es un proceso costoso e inconsistente, lo que hace que la automatización de cualquier parte de este proceso sea de gran interés.

Problema y objetivo

Para ser más concretos, el problema que nos ocupa es: dado un banco de fotografías de grafitis, ¿cómo podemos agruparlo por similitud de forma automática?

Para esto, se desarrolla un sistema modular que permite la combinación de diferentes técnicas de segmentación, extracción de características, técnicas de reducción de dimensionalidad y algoritmos de agrupamiento. Con este sistema, se busca encontrar la combinación de técnicas que mejor se adapte a este problema, evaluando la calidad y coste de cada elemento del sistema.

Trabajos relacionados

No existe mucha literatura sobre la aplicación de técnicas informáticas al estudio de grafitis. Algunos de los pocos trabajos que se han encontrado cubren puntos similares a este: - Fogaca trata la segmentación de grafitis y la clasificación de los mismos en grafiti o no grafiti. - Tokuda propone un sistema para detectar grafitis automáticamente desde Street View para cuantificar la presencia de grafitis en las ciudades. - Miguel García desarrollan un sistema para agrupar grafitis según sus colores dominantes. Este trabajo se diferencia de los anteriores en que se centra en la agrupación de grafitis por similitud, con un énfasis en la evaluación de la calidad y coste de cada elemento del sistema.

Datos

Conjunto de datos

Para el proyecto, disponemos de un banco principal de 6600 fotografías de grafiti en Salamanca, capturadas principalmente en 2022. Las fotografías son bastante distintas entre si: hay bastante variedad de perspectivas, iluminación y escenas, algunos grafitis son sobre muros, otros sobre papeleras o postes. Muchas de las fotografías contienen varios grafiti.

Además, se cuenta con un banco auxiliar de más de 1000 fotografías de Cuenca, cedidas por la iniciativa StopGrafiti. Estas se utilizan solo para el ajuste fino para evitar el solapamiento entre datos de entrenamiento y evaluación.

Anotaciones y subconjuntos

Ninguno de los bancos cuenta con anotaciones, por lo que ha sido necesario crear varios conjuntos anotados para poder evaluar la calidad de los resultados. Puesto que muchas de las fotografías contienen varios grafitis, se ha

decidido recortar los grafitis de las fotografías y crear un conjunto de datos de grafitis anotados.

Los recortes se anotan según dos listas de etiquetas: una de ellas para el estilo del grafiti, que contiene las 4 categorías mostradas aquí y otra para el "autor" del grafiti, que contiene unos ~80 autores distintos. Aquí en la tabla se muestran la procedencia y tamaño de cada uno de los conjuntos utilizados.

Metodología

Sistema: dos fases acopladas por el almacenamiento

Ahora pasamos a la metodología. El sistema desarrollado consta de dos fases bien diferenciadas, que se acoplan mediante el paso por el almacenamiento.

La primera fase, la ingesta, se realiza de forma incremental y consiste en la segmentación y obtención de la representación numérica de cada grafiti para luego almacenarlos.

La segunda fase, la búsqueda por similitud, se realiza de forma puntual y consiste en la obtención de la representación numérica de cada grafiti para agruparlos y buscar los más similares.

A continuación se explican con más detalle cada una de las fases.

Segmentación

La etapa de segmentación consiste en obtener de manera automática los recortes de los grafitis a partir de las fotografías. Para esto, se utiliza un modelo de YOLO ajustado sobre algunos de los recortes anotados para detectar específicamente grafitis.

La fase procede como se muestra en el diagrama: los grafitis se detectan, se fusionan los recortes que cumplen ciertos criterios, se añade un ligero margen a los recortes y se almacenan en el banco de grafitis.

Esta fase es opcional, y en el caso de prescindir de ella se hace uso de la fotografía completa.

Extracción de características

Los recortes de grafiti se alimentan a una red neuronal de la que luego se extrae una representación numérica de cada grafiti. Para esto, se utilizan distintos tipos de redes neuronales pre-entrenadas como ResNet, MobileNet o transformers como DINOv2 o CLIP.

Extracción de características: ajuste al dominio

Para mejorar la calidad de las representaciones numéricas, se realiza un ajuste fino de algunas de las redes neuronales. Esto se hace entrenando una pequeña red neuronal que se situa al final de la red pre-entrenada y que se entrena sobre los conjuntos anotados de grafitis. Se entrenan 4 cabezas: 2 para estilo y 2 para autoría, una en MobileNet y otra en DINOv2.

Reducción de dimensionalidad

Los vectores de características obtenidos por la red neuronal llegan a contener hasta 2048 dimensiones. Esto hace que el coste de agrupamiento y búsqueda por similitud sea muy alto, además de que la calidad de los resultados se ve afectada por un suavizado en la distancia entre vectores, conocido como la maldición de la dimensionalidad como se puede ver en el gráfico.

Para esto, la etapa opcional de reducción reduce el tamaño de estos vectores a uno más manejable mediante distintas técnicas como PCA, Isomap o UMAP.

Agrupamiento

Por último, para el agrupamiento de los grafitis, se utilizan distintos algoritmos de agrupamiento de varias familias:
...

Experimentos y resultados

Diseño experimental

Pasando a los experimentos, se han realizado varios barridos de combinaciones de técnicas, intentando siempre variar un solo eje para poder evaluar el efecto de cada técnica de forma aislada. Podemos distinguir tres grupos de experimentos: los que evalúan la calidad general de cada componente del sistema, los que evalúan el coste de cada componente y los experimentos supervisados.

Como métricas de calidad, distinguimos entre métricas intrínsecas, que evalúan la calidad de los resultados sin necesidad de anotaciones, que son las que se utilizarán para la mayoría de experimentos y métricas extrínsecas, que comparan los resultados con anotaciones. Para el coste, se mide el tiempo de ejecución y el consumo de memoria.

Resultados: segmentador (E5)

Siguiendo el flujo de la metodología, en este experimento se evalúa la calidad del segmentador en el pipeline completo. Aunque los resultados del segmentador son aparentemente buenos viendo las fotografías segmentadas, el efecto que tiene sobre el sistema completo es negativo.

El uso de segmentador empeora la calidad de los resultados, se use la configuración de segmentador que se use. Viendo los resultados del entrenamiento del segmentador, recall X y mAP X, se puede deducir que la razón de este efecto es la variabilidad de los recortes obtenidos. El modelo pierde 1 de cada 4 grafitis, y los recortes obtenidos no suelen ajustarse correctamente al grafiti. Esta variabilidad extra provoca naturalmente un aumento del ruido, una peor agrupación y una ingesta mucho más costosa.

Por esto, se decide prescindir del segmentador y trabajar con las fotografías completas.

Resultados: extractores de características (E1)

En este experimento se evalúa qué extractor produce las mejores representaciones para el agrupamiento. La métrica principal es la silueta, que mide la separación entre clústeres, puesto que no se disponen de anotaciones para el banco completo.

Algo sorprendentemente, las CNN clásicas lideran el ranking: MobileNet y ResNet50 obtienen las mejores siluetas, mientras que las cabezas ajustadas al dominio del grafiti quedan en las últimas posiciones. La cabeza de estilo DINOv2, que es la que intuitivamente debería hacerlo mejor, queda décima de doce.

Esto no significa necesariamente que las CNN sean mejores para este problema. Lo que puede estar pasando es que estas redes estén agrupando por características de escena —el muro, la perspectiva, el entorno— y no por similitud visual del grafiti en sí. Para despejar esta duda, se lleva a cabo la validación supervisada.

Resultados: validación supervisada (E8) y elección del extractor

Con las anotaciones de estilo, se evalúan los extractores de forma supervisada: se mide en qué medida los clústeres resultantes se corresponden con las etiquetas de estilo reales. La cabeza de estilo DINOv2 prácticamente dobla el ARI del mejor extractor sin ajustar. Un ARI de 0.72 frente a valores en torno a 0.33 para ResNet50 o DINOv2 base. Esto confirma que las CNN sí estaban agrupando por escena, no por grafiti.

La gráfica de la derecha muestra la correlación entre silueta y ARI por celda. Para estilo con número de clústeres fijo, la correlación es buena —r de 0.78—, lo que significa que la silueta sí es un indicador útil dentro de un mismo extractor. Pero no sirve para comparar entre extractores distintos.

Para autoría —E8b, con 87 autores— los resultados son muy bajos, en torno a 0.04-0.08. Esto se debe a que el conjunto de evaluación tiene de media menos de dos recortes por autor. Con tan poca muestra, la tarea se convierte en recuperación abierta, no en partición, y ningún algoritmo de agrupamiento puede hacer nada con esto.

Se elige, por tanto, la cabeza de estilo DINOv2 como extractor base del pipeline.

Resultados: reducción de dimensionalidad (E2)

En este experimento se evalúan las distintas técnicas de reducción. El gráfico muestra a la izquierda la calidad y a la derecha el coste de cada opción.

Este experimento es un ejemplo de como las métricas aisladas pueden llegar a engañar: sin reducción se obtiene la mejor silueta, pero a costa de un 60 % de ruido. La mayoría de puntos se quedan sin asignar a ningún clúster. Además, el coste de reducir tampoco desaparece, solo se traslada del paso de reducción al de agrupamiento.

PCA produce resultados degenerados: en la versión de 10 componentes aparecen solo 2 clústeres con silueta nula, y en la de 50 el ruido sube al 70 %.

UMAP es el claro ganador: ofrece el mejor equilibrio entre silueta y ruido, a un coste de reducción muy bajo. Comparando UMAP-10 y UMAP-50, los resultados de calidad son prácticamente idénticos, pero UMAP-50 casi dobla el coste. Se elige UMAP-10 como configuración base.

Resultados: algoritmo de agrupamiento (E3)

El gráfico muestra el frente de Pareto calidad-coste por familia de algoritmos. La conclusión más importante: las métricas aisladas engañan. Silueta y Calinski-Harabasz premian particiones triviales de dos clústeres, por lo que hay que leerlas siempre en conjunto con el ruido y el balance de tamaños.

HDBSCAN es el mejor candidato: silueta competitiva, ruido moderado, tamaños razonablemente balanceados y un coste de agrupamiento de 0.35 segundos.

OPTICS iguala la calidad de HDBSCAN pero multiplica el tiempo por 25 y la memoria por 2500. No hay razón para usarlo.

K-medias es más rápido que HDBSCAN y es la mejor opción cuando no se quiere ruido, pero tiene una silueta casi 0.1 puntos inferior.

Con esto queda definido el pipeline base: DINOv2 estilo, UMAP-10 y HDBSCAN con mcs igual a 5.

Resultados: coste del sistema completo (E7)

En este experimento se mide el coste de cada etapa del pipeline a distintos tamaños del banco, de 500 a 6416 imágenes. La gráfica está en escala log-log, de modo que la pendiente de cada línea indica el orden de crecimiento.

La ingesta tiene pendiente cercana a 1, es decir, crece linealmente con N. A esta escala es con diferencia la etapa más costosa en tiempo.

La reducción con UMAP tiene pendiente ligeramente superior a 1 pero es prácticamente despreciable en tiempo absoluto: menos de 5 segundos en todo el rango.

Para el agrupamiento, K-medias crece sub-linealmente, HDBSCAN algo más, y el aglomerativo ya empieza a mostrar costes cuadráticos. En este caso más que el tiempo, lo que preocupa es la memoria: OPTICS consume 763 MB y el aglomerativo 314 MB a escala completa.

El verdadero cuello de botella es la búsqueda por similitud en SQLite, con pendiente de 2.1 —casi cuadrática— y un tiempo de 54 minutos a N igual a 6416. Esto es lo que motiva el siguiente experimento.

Resultados: búsqueda por similitud (E6, E9)

La tabla muestra el rendimiento de la búsqueda por similitud a distintos tamaños de banco y con tres backends distintos.

SQLite hace la búsqueda en Python con un barrido lineal, lo que lo hace muy lento. pgvector exacto elimina el round-trip a Python y llega a unas 600 consultas por segundo a N pequeño, pero su throughput cae con N.

El índice HNSW, que es un índice aproximado, mantiene un throughput prácticamente constante en torno a 560 consultas por segundo independientemente del tamaño del banco. A N igual a 6416, es 3.2 veces más rápido que el exacto y 240 veces más rápido que SQLite, y todo esto sin ninguna pérdida de recall: R@5 es 1.000 en todos los casos.

El índice tarda menos de medio segundo en construirse, por lo que se amortiza en una sola consulta. Y su ventaja sobre el exacto sigue creciendo con N, por lo que para bancos más grandes la diferencia sería aún más pronunciada.

Conclusiones

Conclusiones

Para cerrar, recapitulo los hallazgos principales del trabajo.

Se ha diseñado un sistema modular que permite comparar extractores, técnicas de reducción y algoritmos de agrupamiento bajo una misma interfaz. El pipeline base resultante combina la cabeza de estilo de DINOv2 con UMAP-10 y HDBSCAN.

Un resultado contraintuitivo importante: las métricas intrínsecas no son suficientes para elegir el extractor. Las CNN de ImageNet lideran la silueta pero agrupan por escena, no por grafiti. Hace falta validación supervisada para confirmar que el extractor captura la similitud correcta.

En cuanto al coste, la ingesta domina el tiempo a esta escala. El cuello de botella real es la búsqueda por similitud, que en SQLite crece de forma cuadrática. El índice HNSW la elimina sin perder recall, y su ventaja crece con el tamaño del banco.

La reducción de dimensionalidad con UMAP es indispensable: sin ella el agrupamiento produce un 68 % de ruido y el coste se traslada al paso de agrupamiento.

Como limitaciones, el rango de N que se ha podido explorar es estrecho y dificulta confirmar las pendientes teóricas. Los datos de autoría son insuficientes para una evaluación real de esa tarea. Como trabajo futuro, sería interesante explorar el preentrenamiento no supervisado antes de las cabezas supervisadas, y pasar de una ingesta en batch a una ingesta incremental que actualice el índice y el espacio reducido sobre la marcha.