Trabajo Fin de Máster

Agrupador de imágenes multimodal no supervisado

Jesús Pérez Hernando

Índice de contenido

1. Introducción....2
2. Estado del arte en el agrupamiento de imágenes en la visión artificial....4
En qué consiste el problema....4
Qué aplicaciones tiene....6
Descriptores de imágenes....10
SIFT....11
HOG....13
GIST....15
CENTRIST....18
LBP....20
3. Integración de los descriptores de imagen en el algoritmo MMSC....23
MMSC....23
Agrupamiento espectral....24
Agrupamiento espectral multimodal....26
Algoritmo MMSC....27
4. Tecnología utilizada y detalles de implementación....29
OpenCV....29
Decisiones generales....30
SIFT....31
HOG....32
GIST....33
CENTRIST....33
LBP....35
5. Pruebas y resultados obtenidos....36
Métricas de calidad de agrupamiento....36
NMI (*Normalized Mutual Information*)....36
Purity....37
Pruebas efectuadas....38
MSRC-v1....38
CALTECH 101....41
Prueba 1: 7 clases....41
Prueba 2: 20 clases....44
6. Conclusiones. Futuros trabajos....47
7. Bibliografía....49

1. Introducción

La visión artificial es la ciencia y tecnología de hacer máquinas que vean. Está relacionada con la teoría, diseño e implementación de algoritmos que pueden procesar automáticamente datos visuales para reconocer escenas y objetos, hacer seguimiento a estos últimos con cámaras (*tracking*) y obtener sus contornos y formas. Actualmente es un campo en pleno desarrollo, sustentado por los grandes avances en la disciplina de Inteligencia Artificial y alimentado por el deseo cada vez más insistente de "enseñar a las máquinas a ver".

Por tanto, la visión artificial está en un punto extraordinario de su desarrollo. Se ha estado investigando desde los años 60, pero solo en los últimos tiempos ha sido posible el construir sistemas lo suficientemente potentes para poner en práctica las teorías ideadas. El avance se debe a dos causas principales: los ordenadores tienen cada vez más capacidad de proceso y los sistemas de imagen son cada vez más económicos. No hace mucho, estos sistemas costaban miles de euros, mientras que ahora valen unos pocos cientos como mucho, por lo que es mucho más fácil y económico el realizar pruebas e investigaciones. Otro aliciente radica en el hecho de que la captura de imágenes es un proceso no destructivo y por lo general discreto, lo que posibilita que se registren cantidades incalculables de imágenes cada segundo.

Nuestra comprensión de la geometría y de la física sobre las que se basa la visión, y más importante aún, que conseguir con ellas, ha mejorado de forma significativa. Estamos comenzando a ser capaces de resolver problemas planteados desde hace mucho tiempo, pero aún quedan grandes problemas por resolver, y multitud de problemas relativamente simples a los que todavía no se les ha encontrado solución; por tanto, es la época más atractiva para investigar en el ámbito de la visión artificial.

El gran truco de la visión artificial es el de extraer descriptores del mundo, obteniendo su información útil, a partir de imágenes o de una secuencia de las mismas en forma de vídeo. Sin embargo, la tarea de extraer esta información útil es una tarea sorprendentemente difícil: después de cinco décadas de investigación constante seguimos lejos de ser capaces de construir una máquina con visión de

propósito general.

Parte del problema radica en la complejidad de los datos visuales. Normalmente hay cientos de objetos en una escena y casi ninguno de ellos se presenta de una forma "típica". Casi todos ellos pueden estar parcialmente ocultos. Para un algoritmo de visión artificial, ni siquiera es fácil establecer donde termina un objeto y comienza otro.

Podríamos desanimarnos respecto a nuestras posibilidades de desarrollar algoritmos de visión artificial útiles y eficientes si no fuera por un motivo: es un hecho que la visión es posible, dado que nuestros propios sistemas visuales son capaces de extraer información de imágenes complejas. Podemos incluso determinar con precisión la localización de una imagen extrayendo pistas sutiles de acuerdo a su contexto.

Por tanto, la visión artificial no es imposible, pero es muy compleja, lo que quizás no se apreció en un primer momento, debido a que lo que percibimos cuando miramos a una escena ya ha sido procesado en gran medida. El cerebro visual realiza muchos trucos de este tipo, y desafortunadamente, cuando nosotros construimos algoritmos de visión artificial, no tenemos el beneficio de este preproceso.

No obstante, recientemente ha habido importantes progresos en la comprensión de la visión artificial, y la última década ha visto el primer despliegue a gran escapa de tecnologías de visión artificial para el usuario. Por ejemplo, la mayoría de las cámaras digitales tienen ahora algoritmos preinstalados para la detección de caras, e incluso se están comercializando a gran escala dispositivos que permiten el seguimiento en tiempo real del cuerpo humano, como es el caso de Microsoft Kinect.

2. Estado del arte en el agrupamiento de imágenes en la visión artificial

En qué consiste el problema

La categorización de escenas y el reconocimiento visual son tareas clave en la investigación de la visión artificial. No obstante, debido a la variabilidad dentro de las imágenes, su ambigüedad y su amplio rango de iluminación, esta clasificación es complicada. La forma más popular de solventar estos problemas es utilizar características de la imagen a bajo nivel, tales como color global, histogramas de texturas, contornos de objetos, etc.

Una de las causas de esta dificultad se debe a que, en contra de la lógica, conocer los objetos en una imagen no resuelve automáticamente la categoría de la escena. Muchas fuentes de información útiles pueden estar contenidas en regiones de la imagen que no son objetos.

Otro de los motivos por la que la categorización es compleja se debe a que pueden existir enormes variaciones dentro de una misma categoría de escenas. Como consecuencia, no podemos encontrar un patrón común usando los métodos aplicados en la mayoría de tareas de reconocimiento de objetos. Por ejemplo, una cama puede aparecer en una imagen tomada de un dormitorio. No obstante, la imagen puede ser tomada desde cualquier perspectiva, y la cama puede aparecer de muy diferentes maneras (colchón, sofá-cama, colchón de agua, etc.). Los dormitorios pueden ser decorados con cualquier estilo o color, con muebles muy variados, así como pueden serlo las condiciones de luminosidad.

Tampoco nos podemos basar en la disposición de los objetos en una imagen, ya que estos pueden estar desordenados, pudiendo ser dispuestos en una infinidad de posiciones diferentes, lo que es otro obstáculo para la clasificación correcta de una escena.

Para complicar aún más las cosas, desde que existe interés en las colecciones de datos autónomos, las imágenes puede que no sean representativas de la categoría

de la escena. En el ejemplo del dormitorio, una cama podría no verse en muchas de las imágenes.

A todo esto debemos añadirle los problemas derivados de las ambigüedades semánticas: la clasificación de escenas es más arbitraria que la de objetos, porque no hay un consenso claro sobre lo diferentes que deberían ser las etiquetas. Algunas etiquetas son muy claras y fáciles de asignar de forma precisa, por ejemplo "cocina", "dormitorio", y otros nombres de habitaciones en una casa. Sin embargo, otras etiquetas son ambiguas: ¿se debería distinguir entre "caminos forestales" y "prados"? ¿o simplemente se deberían etiquetar a todas esas imágenes como "exteriores"? Los humanos también tenemos estos problemas a la hora de etiquetar. No obstante, en los conjuntos de imágenes que sirven de prueba, cada uno de ellos viene con su propio conjunto de etiquetas, por lo que los métodos pueden ser comparados y evaluados.

Los diferentes métodos ideados para clasificar imágenes funcionan bien bajo unas condiciones específicas, degradando su rendimiento de forma significativa cuando dichas condiciones no se cumplen. La solución por tanto será la de encontrar métodos complementarios que den buenos resultados donde los otros fallan, y combinarlos con el objetivo de aprovechar sus ventajas al mismo tiempo que se minimizan sus debilidades.

Qué aplicaciones tiene

La clasificación no supervisada de escenas tiene de forma directa varias aplicaciones:

- **Seguir un proceso de selección o diagnóstico:** Ya sea con productos en una cadena de envasado, fabricación de piezas, discriminando aquellos productos o piezas defectuosos. También puede ser apropiado en medicina para hacer un rápido diagnóstico basándose en alguna imagen médica.

  Un ejemplo existente de esta aplicación es el de los sistemas usados por la empresa E2M (Estudis Electro-mecánics, S.L.), donde sus diferentes aplicaciones orientadas al envasado de líquidos (VisioLevel, VisioCap, VisioLabel, etc.) se encargan de verificar el nivel de llenado de las botellas, si hay ausencia de tapón o problemas con él, fallos de etiquetado, etcétera.

- **Clasificación de las imágenes de grandes bancos de datos:** A las entidades y usuarios les puede ser útil el clasificar las imágenes almacenadas en sus equipos informáticos de una forma rápida y sin interacción ninguna.

  Una aplicación existente que ejecuta una clasificación automática (aunque sólo en base a rostros humanos) es iPhoto de Apple, que asocia entre sí las imágenes en las que aparecen las mismas personas

- **Detección de imágenes explícitas:** Existen numerosas razones para detectar imágenes violentas, de desnudos o contenido sexual:

  - En algunas jurisdicciones, la posesión o distribución de esas imágenes podría ser ilegal.

  - Muchos empleadores quieren asegurarse de que lo trabajadores no ven ni guardan imágenes de este tipo en sus equipos; de hecho, los vendedores de *software* de filtrado de imágenes han intentado persuadir a los empleadores de que podrían ir a juicio si no toman esas medidas.

- Los anunciantes quieren asegurarse de que sus anuncios no aparezcan cerca de imágenes que perturben a sus potenciales clientes.

- A las compañías de búsqueda en Internet les gustaría permitir a sus usuarios el poder evitar los tipos de imágenes que éstos encuentren perturbadoras.

Es difícil incluso para los juristas el estar seguros de qué imágenes son aceptables y cuáles no lo son. Para la mayoría de aplicaciones es suficiente con filtrar las imágenes que muestran desnudos o contenido sexual, y esto puede conseguirse con un clasificador.

Todos los métodos publicados referentes a estos clasificadores se basan en encontrar piel en las imágenes, y algunos de estos métodos, en las características de la piel. Lo más lógico es que haya dos pasos en la clasificación: se necesita clasificar los píxeles en piel y no-piel; y también se necesita clasificar las imágenes en explícitas y no explícitas basándose en las características de la piel.

- **Segmentación de imágenes de vídeos:** Puede aplicarse a todos los fotogramas para así segmentar el contenido, categorizando todas las imágenes, lo que permitiría rápidas búsquedas semánticas a lo largo todo el contenido.

- **Clasificación de materiales:** Normalmente, materiales diferentes producen texturas diferentes, por lo que las características naturales para usar con este problema de clasificación serían las características de las texturas. No obstante, los materiales tienden a tener algún relieve en su superficie (por ejemplo, los poros en la superficie de una naranja, los surcos en las cortezas de los árboles, los pequeños granos que conforman la arena), y estos relieves causan comportamientos de sombreado complejos. Los cambios en la dirección de la iluminación pueden causar que las sombras vinculadas a los relieves cambien enormemente, por lo que la textura cambia de forma completa. Además, las características de las texturas pueden indicar de qué material está hecho un objeto, pero hay objetos que pueden tener la misma

textura y estar hechos de diferentes materiales.

Hasta la fecha sólo hay aproximaciones teóricas para lograr esta aplicación, no existiendo ningún producto comercial.

- **Encontrar imágenes duplicadas:** Es muy útil el buscar imágenes que sean muy parecidas a otra imagen dada, preferiblemente ignorando los cambios en tamaño, resolución, orientación, recortes, ruidos debidos a la compresión y efectos similares. Por ejemplo, el logo de una marca necesita ser único, y un usuario que esté intentado registrar un logo puede de esta manera buscar logos similares que ya se encuentren registrados. La detección de duplicados puede ser usada para proteger imágenes o logos con derechos de autor, buscando copias robadas de la imagen en Internet.

  Alternativamente, si se intenta hacer que la gente use imágenes determinadas (algo parecido a una campaña viral de *marketing*), se podría saber si se ha obtenido éxito haciendo búsquedas de duplicados. Asimismo, si se mantiene una gran cantidad de imágenes en un ordenador personal, sería posible el liberar espacio en disco gracias a la detección de duplicados.

  Actualmente existen varios buscadores que implementan esta funcionalidad, como Google y TinEye, realizando la búsqueda por el contenido de las imágenes y no en base a sus metadatos. Estos buscadores permiten realizar la búsqueda a partir de la url de una imagen, o bien enviándola a su servidor.

- **Búsquedas semánticas:** Otras aplicaciones requieren criterios de búsqueda más complejos. Por ejemplo, una librería comercial de imágenes sobrevive gracias a las ventas de los derechos de uso sobre determinadas imágenes.

  Un método automático para la realización de estas búsquedas necesitará una profunda comprensión de la solicitud y de las imágenes en la librería. Las búsquedas de imágenes en Internet muestran que se puede obtener una resultado útil sin tener que recurrir a un reconocimiento de objetos exhaustivo, basándose principalmente en los metadatos. Estos sistemas

parecen ser útiles, aunque es difícil saber cuánto o para quién.

Estas son los tipos búsquedas que llevan más tiempo implementadas por Google o Bing (por ejemplo), y que realmente no conllevan un gran esfuerzo en cuanto a visión artificial.

- **Seguir la evolución de procesos:** En la minería de datos, se usan simples análisis estadísticos para hallar tendencias. Se pueden obtener muchos beneficios aplicando la minería de datos al contenido de las imágenes. Por ejemplo, con las imágenes vía satélite de nuestro planeta podríamos responder a preguntas como las siguientes: ¿Hasta dónde se extiende la urbanización? ¿cuánto de grande será la cosecha de maíz? ¿cuánto queda de selva?, etcétera.

  Del mismo modo, se podrían extraer datos de imágenes médicas para tratar de encontrar señales visuales para los resultados de un tratamiento a largo plazo.

De forma indirecta, el agrupamiento no supervisado puede ser adecuado como proceso previo a una clasificación supervisada. Esto se debe a que la obtención de datos sin etiquetar es un proceso relativamente fácil, mientras que el etiquetado es difícil, caro, tedioso o propenso a errores, ya que suele estar basado en las decisiones de un humano.

Por tanto, en muchos problemas es deseable tener algoritmos de aprendizaje en los que se pueda incorporar un gran número de datos sin etiquetar a un pequeño número de datos etiquetados a la hora de hacer clasificadores.

Descriptores de imágenes

Oliva, en 1999, definió la categorización de escenas en base a tres niveles de abstracción: el nivel subordinado, el nivel básico y el nivel de orden superior. El objetivo del reconocimiento de escenas varía entre estos tres niveles, lo que tiene un impacto directo en la categorización y los métodos de etiquetado supervisados de los conjuntos de escenas de imágenes.

En el nivel subordinado, la coincidencia de una escena con otra se define con respecto a la coincidencia de objetos u otros tipos de estructuras locales. Debido a la naturaleza local de este nivel de reconocimiento, las imágenes de escenas son a menudo segmentadas en pequeñas regiones que se describen de forma específica por su textura y su color. Por tanto, la coincidencia de escenas se basa en la coincidencia a nivel de objeto: este método requiere la búsqueda de escenas similares en términos de similitud de regiones y configuración. Así, a este nivel de reconocimiento, las semánticas de la escena completa no son tan importantes como las identidades de los detalles locales, y las imágenes de escenas obtenidas pueden no pertenecer a la misma categoría semántica. Por tanto, a este nivel normalmente no hay necesidad de etiquetar de forma supervisada.

El nivel básico de reconocimiento requiere una identificación más amplia y holística de las imágenes de escenas respecto al nivel subordinado. En este nivel, el reconocimiento de objetos locales se sustituye por la identificación de la función de la escena. Para este fin, se usa normalmente una descripción más global de las escenas para la categorización de las mismas, y se debería dar preferencia a la configuración a alto nivel de las diferentes grandes regiones de la escena en vez de a las estructuras locales detalladas. Ya que este nivel de categorización de escenas requiere la comprensión de sus semánticas, las imágenes son a menudo clasificadas de acuerdo a sus significados globales: playa, montaña, rascacielos, etc.

En el nivel de orden superior, incluso se ignora el nivel semántico básico de una escena. Este alto nivel de abstracción sólo se preocupa de las estructuras de la escena a resolución completa. En cierto sentido, se corresponde con la percepción visual humana de una escena estando a una distancia razonable. A este nivel de abstracción, las variaciones de las propiedades espaciales dentro de cada categoría

son más pronunciadas que las que existen en los niveles más bajos. Por ejemplo, basándose en este nivel, se ha investigado en la separación de escenas naturales y aquellas construidas por el hombre; mientras que otras lo han hecho para distinguir entre escenas interiores y exteriores.

En años recientes, se han propuesto una variedad de métodos de representación de características para ayudar a describir las escenas y objetos visuales en imágenes diferentes. Las singularidades de cada uno de estos métodos describen diferentes aspectos de las características visuales, siendo cada uno de ellos más apropiado para ciertas condiciones (o propiedades) en particular. Algunos de estos métodos se centran en la información local, otros en los descriptores holísticos.

Entre todos los descriptores de características locales, los más utilizados son SIFT (*Scale-Invariant Feature Transform*), HOG (*Histogram of Oriented Gradients*), SURF (*Speeded-up Robust Features*), los cuales se utilizan para solventar la variabilidad en la imagen causada por los cambios de perspectiva, las oclusiones y la variación en la luminosidad. El descriptor LBP (*Local Binary Patterns*) se propuso como un buen descriptor de las texturas, basándose en un histograma de las ocurrencias de los patrones binarios locales. GIST y CENTRIST, por otra parte, son dos representantes de los descriptores holísticos.

A continuación se hace una breve reseña de estos descriptores:

SIFT

SIFT fue diseñado originalmente para reconocer al mismo objeto apareciendo bajo diferentes condiciones y ha sido ampliamente usado en visión artificial y en la recuperación del contenido de imágenes. Como descriptor local, es invariante a la rotación de las imágenes, así como a los cambios de escala. Es también robusto frente a un amplio rango de ruido y cambios en la iluminación. Permite incluso identificar los objetos de forma fiable incluso estando desordenados o parcialmente ocultos

El algoritmo fue publicado por David Lowe en 1999 y está patentado en los Estados Unidos, poseyendo los derechos la Universidad de British Columbia.

Su funcionamiento se basa en que, para cualquier objeto en una imagen, se pueden extraer puntos de interés sobre el objeto con el objetivo de obtener una "descripción de características" del objeto. Esta descripción, extraída de una imagen de entrenamiento, puede ser entonces usada para identificar el objeto cuando se intenta localizarlo en otra imagen de prueba que contiene muchos otros objetos. Para realizar un reconocimiento fiable, es importante que las características extraídas de la primera imagen sean detectables incluso aunque sufran cambios en la escala, en la iluminación y les afecte el ruido. Estos puntos de interés normalmente se encuentran en regiones de alto contraste, tales como los contornos de los objetos.

Otra propiedad importante de estas características radica en que las posiciones relativas entre dichas características en la escena original no deberían cambiar de una imagen a otra. Por ejemplo, si solamente las cuatro esquinas de una puerta fueran usadas como características, podrían reconocerse independientemente de la posición de la puerta, pero si se usara como característica también la hoja de la puerta, el reconocimiento podría fallar si la misma se encuentra abierta o cerrada. De forma similar, las características localizadas en objetos articulados o flexibles normalmente no permitirán el reconocimiento si hay cualquier cambio en su geometría interna entre dos imágenes del conjunto de imágenes que está siendo procesado. No obstante, en la práctica SIFT detecta y usa un número mucho más grande de características, lo que reduce la contribución de los errores provocados por estas variaciones locales al error promedio de las comparaciones de las características.

Ilustración 1: Puntos de interés SIFT (keypoints)

HOG

HOG es un buen descriptor local para describir las formas de la imagen. En contraste a SIFT, el cual describe la característica en un punto de interés (*keypoint*), HOG describe la característica que describe a toda una región dada.

El método se basa en contar las ocurrencias de la orientación de los gradientes en porciones localizadas de una imagen. Este método es similar en cierto sentido a SIFT, pero difiere en que es calculado sobre una malla de celdas uniformes y usa una normalización local de contraste solapada para obtener una mayor precisión.

Navneet Dala y Bill Triggs, investigadores del INRIA, describieron por primera vez el Histograma de Gradientes Orientados (*Histogram of Oriented Gradient*, HOG) en 2005. En este trabajo se enfocaron sobre el problema de detección de peatones en imágenes estáticas, aunque después incluyeron en sus pruebas la detección humana en imágenes y vídeos, además de una variedad de animales comunes y

vehículos en imágenes estáticas.

La idea esencial en la que se basan los descriptores de HOG es que la apariencia y forma de los objetos locales dentro de una imagen puede ser descrita por la distribución de los gradientes de intensidad o las direcciones de los contornos. La implementación de estos descriptores puede lograrse dividiendo la imagen en pequeñas regiones conectadas, llamadas celdas, y por cada celda compilar un histograma de direcciones de gradientes u orientaciones de contornos para los píxeles dentro de la misma. La combinación de estos histogramas representa entonces el descriptor.

Para obtener una mayor precisión, los histogramas locales pueden ser normalizados por contraste calculando una medida de la intensidad a lo largo de una región más grande de la imagen, llamada bloque, y usar entonces este valor para normalizar todas las celdas dentro del bloque. Esta normalización produce una mejor invariabilidad a cambios en iluminación o sombreados.

El descriptor HOG mantiene algunas ventajas clave sobre otros métodos de extracción de descriptores. Ya que el descriptor HOG opera en celdas localizadas, el método se beneficia de ser invariable frente a transformaciones fotométricas y geométricas, aunque tiene el inconveniente de no ser invariable a la orientación de los objetos. Estos cambios podrían aparecer solamente en regiones espaciales de mayor tamaño.

Las características del descriptor HOG hacen que sea particularmente adecuado para la detección humana en las imágenes.

Ilustración 2: Descriptores HOG

GIST

GIST codifica la geometría base y las estructuras espaciales dentro de una imagen y suprime las texturas detalladas, centrándose en la información holística. Alcanza una gran precisión en el reconocimiento de categorías de escenas naturales, tales como montañas o playas. Su inconveniente radica en que no tiene un buen desempeño en el reconocimiento de imágenes en escenas interiores.

GIST usa descriptores globales. La idea es desarrollar una representación de baja dimensionalidad de la escena, por lo que no se requiere ningún tipo de segmentación.

Los autores de GIST, Oliva y Torralba, proponen un conjunto de dimensiones perceptuales (naturalidad, apertura, rugosidad, expansividad, aspereza) que representan la estructura espacial dominante de una escena.

Estas dimensiones perceptuales se definen como sigue:

- **Grado de naturalidad:** el grado de naturalidad es un importante indicador que separa las estructuras construidas por el hombre de los hábitats naturales. Se observa que la mayoría de las escenas en las que aparecen estructuras construidas por el hombre están caracterizadas principalmente por sus líneas verticales y horizontales a través de la imagen. En contraste, los contornos de los paisajes naturales son más flexibles y se extienden cambiando de dirección continuamente.

  En el caso de que una escena exhiba entornos naturales y estructuras construidas por el hombre al mismo tiempo, el grado de naturalidad proporcionará una medida del dominio de una de las dos características.

- **Grado de apertura:** Esta propiedad refleja cuánto de cerrada tiene una escena. En algunas categorías de escenas, los espacios abiertos dominan la textura de todas las imágenes, tales como las escenas de playas o de campo abierto, mientras que otras escenas están caracterizadas por estar confinadas en estructuras que albergan la mayor parte de las imágenes. Por ejemplo, en la escena de un cuarto de baño, el lavabo y la ducha están rodeados de los muros de la estancia; y en la escena de un bosque, los árboles normalmente están agrupados entre sí y se propagan a lo largo de la escena, limitando el área abierta (el cielo) a una pequeña región de la imagen.

- **Grado de aspereza:** representa un concepto similar al de la granularidad en los análisis de texturas. Se ve muy afectado por la configuración de los elementos más básicos en una escena, especialmente por la complejidad de su configuración. Además, también influyen las relaciones entre estos elementos básicos y su habilidad para formar estructuras más complejas. Debe reseñarse que el grado de aspereza está altamente relacionado con la escala espacial en la que los elementos son medidos. Si una escena presenta un alto grado de aspereza, es que la configuración general de la escena es bastante compleja. Cuanto más pequeña sea la escala espacia, más se pueden perder detalles estructurales de la escena.

- **Grado de expansividad:** Esta propiedad se refiere a cuánto de extensas parecen ser las líneas de una escena. Por ejemplo, en la escena de una autopista, las líneas verticales parecen extenderse hasta el infinito. En una imagen de dos dimensiones, este fenómeno se observa en términos del grado de convergencia de las líneas verticales u horizontales. Si todas las líneas principales de una escena tienen tendencia a converger a un puto dado, el observador podrá cerciorarse de que se cubre una distancia considerable en la imagen. Por otro lado, si la mayoría de las líneas de una escena se propagan a través de la misma en paralelo, se dice que la escena tiene un bajo grado de expansividad.

- **Grado de rugosidad:** El grado de rugosidad depende de cómo emerge el horizonte de una escena. Se dice que una escena presenta un alto grado de rugosidad si la línea horizontal es ambigua o bien inexistente. Por el contrario, las escenas con un bajo grado de rugosidad exhiben un horizonte bien definido. Por ejemplo, en la imagen de una montaña, la línea horizontal es casi invisible, mientras que en la escena de una calle, los edificios se ajustan perfectamente al horizonte, lo que sugiere que la imagen de la calle presenta un bajo grado de rugosidad.

Los autores muestran que estas dimensiones perceptuales pueden ser estimadas de una forma fiable usando información espectral y localizada sin estar detallada. La imagen se divide en una malla de 4x4 sobre la que se extraen histogramas de orientación. Obsérvese que el descriptor es similar en espíritu al descriptor local SIFT.

Para calcular la descripción GIST en color de una imagen, ésta es segmentada en una malla de 4x4 para la que se extraen los histogramas de orientación. La implementación toma como entrada una imagen cuadrada de tamaño fijo y produce un vector de dimensión 960. La mayoría de los trabajos que utilizan el descriptor GIST cambian el tamaño de la imagen en una etapa preliminar, produciendo una pequeña imagen cuadrada. Su anchura normalmente varía entre los 32 y 128 píxeles. Esto es suficiente debido a la baja dimensionalidad del descriptor, ya que no representa los detalles de una imagen. Las imágenes son reescaladas a ese tamaño independientemente de sus dimensiones.

Oliva y Torralba apuestan por el uso de la "forma de una escena", una estructura subyacente y similar que seguramente existe dentro de las imágenes de escenas que pertenecen a la misma categoría, para reconocer categorías de escenas. GIST calcula la información espectral de una imagen a través de la transformada discreta de Fourier (DFT). Las señales espectrales son entonces comprimidas por la transformada de Karhunen-Loeve (KLT), un homólogo del método de análisis de componentes principales discreto (PCA).

Los descriptores GIST son normalmente comparados usando la distancia euclídea.

CENTRIST

CENTRIST es un descriptor holístico que captura la estructura espacial estable dentro de las imágenes que refleja la funcionalidad de la localización, y es especialmente bueno en la categorización de escenas en interiores.

CENTRIST (*CENsus TRansform hiSTogram)*, está diseñado especialmente para el reconocimiento de categorías de lugares o escenas. Se muestra que en el reconocimiento de lugares y escenas, especialmente para los entornos en interiores, se requiere un descriptor visual que tenga propiedades que sean diferentes de otros dominios de la visión artificial (por ejemplo el reconocimiento de objetos).

Este método se basa en la percepción de que las imágenes tomadas de una misma categoría de escenas pueden parecer muy diferentes. De forma similar, las imágenes tomadas desde una parte diferente o desde otra perspectiva de la misma localización, también contendrán grandes variaciones. CENTRIST toma la idea de que en lugar de capturar la información de texturas detalladas de los objetos en la escena, se requiere capturar la estructura espacial estable dentro de las imágenes que reflejan la funcionalidad de una localización.

CENTRIST codifica las propiedades estructurales de una imagen y suprime la información detallada de las texturas. Las grandes ventajas de CENTRIST radican en que es fácil de implementar, no tiene casi ningún parámetro para ajustar, y su

evaluación es extremadamente rápida.

Esta codificación de las propiedades estructurales de una imagen se basa en la transformada de censo (*Census Transform,* CT). CT es una tranformación local no paramétrica que fue originalmente diseñada para establecer correspondencias entre regiones locales. La CT compara el valor de intensidad de un píxel con el de sus ocho píxeles vecinos, tal como se muestra en la ilustración 3. Si el píxel central es mayor o igual que uno de sus vecinos, un bit 1 es establecido en su posición correspondiente. Si el píxel central es menor, se establece un bit 0.

$$\begin{array}{c|c|c} 32 & 64 & 96 \\ \hline 32 & \mathbf{64} & 96 \\ \hline 32 & 32 & 96 \end{array} \Rightarrow \begin{array}{ccc} 1 & 1 & 0 \\ 1 & & 0 \\ 1 & 1 & 0 \end{array} \Rightarrow (11010110)_2 \Rightarrow \mathbf{CT} = 214$$

Ilustración 3: Ejemplo de transformación CT

Los ocho bits generados con las comparaciones de los valores de intensidad pueden ser agrupados en cualquier orden (en este ejemplo se agrupan de izquierda a derecha y de arriba a abajo), y el número binario obtenido se convierte entonces a base decimal, lo que nos da un valor entre 0 y 255. Este es el valor CT para el píxel central.

De forma similar a otras transformaciones locales no paramétricas que se basan en las comparaciones de los valores de intensidad, la transformada del censo es robusta frente a los cambios de iluminación, variaciones de gamma, etc.

Como método de visualización, se crea una imagen CT reemplazando cada píxel con su valor CT. Como se ve en la ilustración 4, la imagen CT retiene las estructuras globales de la imagen (especialmente las discontinuidades), además de capturar las estructuras locales para las que estaba diseñada la transformación.

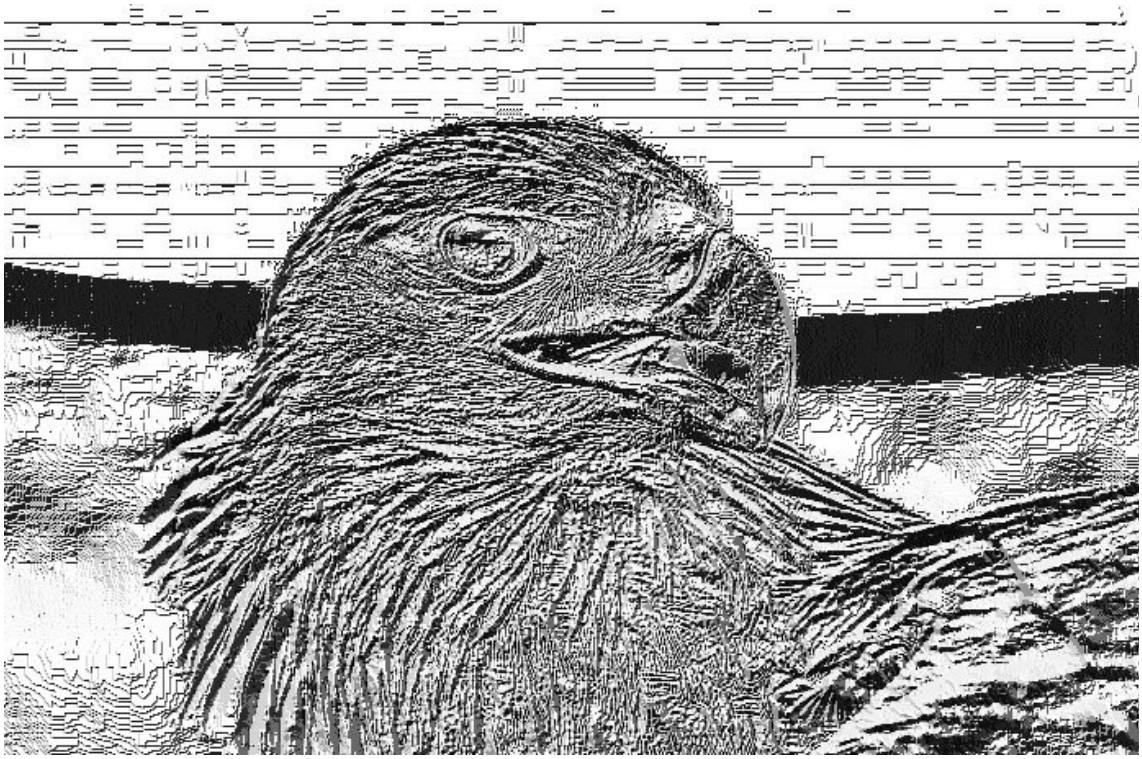


Ilustración 4: Imagen CT sobre la que se basa CENTRIST

LBP

LBP es un potente descriptor de texturas basado en el histograma de ocurrencias de patrones binarios locales. Hace énfasis en la estructura local y es famoso por su robustez frente a la rotación y a la iluminación no uniforme.

El algoritmo LBP está entre los descriptores más eficientes usados hoy en día en el análisis de texturas. Cuando se compara con algoritmos más sofisticados de análisis de texturas se confirma que el descriptor LBP es muy eficiente en cuanto a su cálculo, además de fácilmente paralelizable, por lo que su rendimiento puede ser mejorado aún más al aprovechar la potencia computacional de las unidades gráficas de proceso (GPUs).

Puede ser tomado como un método que unifica los modelos estructurales y estadísticos de análisis de texturas, que han sido tradicionalmente divergentes. Quizás la propiedad más importante del operador LBP en las aplicaciones del mundo

real sea su robustez frente a los cambios en la escala monotónica de grises causada, por ejemplo, por variaciones en la iluminación. Otra propiedad importante es la de su sencillez computacional, lo que hace que sea posible el analizar imágenes en entornos de tiempo real.

La idea básica para el desarrollo del operador LBP radica en que las superficies de texturas bidimensionales pueden ser descritas por dos medidas complementarias: patrones espaciales locales y contraste a escala de grises. El operador LBP original, creado por Ojala en 1996, formaba etiquetas para los píxeles de la imagen al establecer un umbral en el vecindario 3x3 para cada píxel con el valor central y considerando el resultado como un número binario. El histograma de estas 256 etiquetas diferentes puede ser usado como un descriptor de textura (obsérvese que este es el mismo método que se aplica en CENTRIST). Este operador usado junto a una simple medida de contraste local suministra un rendimiento muy bueno en la segmentación de texturas no supervisada.

El operador LBP fue extendido para usar tamaños de vecindarios de diferentes tamaños. Usando un vecindario circular e interpolando de manera bilineal los valores en coordenadas de píxeles en coma flotante permite utilizar cualquier radio y número de píxeles en el vecindario. La variabilidad de la escala de grises en el vecindario local puede ser usada como una medida de contraste complementaria. A partir de ahora, la notación (P, R) será usada para los vecindarios de píxeles, lo que significa P puntos de muestreo sobre un círculo de radio R.

LBP es un descriptor enormemente parecido a CENTRIST, salvo que tiene un enfoque local, al contrario que el enfoque global (holístico) de CENTRIST. Asimismo, su tamaño de histograma es menor, ya que utiliza los patrones binarios para reducir su dimensionalidad además de hacerse invariable frente a las rotaciones por medio de la extensión del método basada en los patrones uniformes.

Esta extensión del operador original consistente en los patrones uniformes fue inspirada por el hecho de que algunos patrones binarios aparecen más que otros en las texturas de las imágenes. Un patrón binario local es llamado uniforme si el patrón binario contiene como mucho dos transiciones a nivel de bit del 0 al 1 o viceversa cuando el patrón de bits es recorrido circularmente. Por ejemplo, los patrones 00000000 (0 transiciones), 01110000 (2 transiciones) y 11001111 (2

transiciones) son uniformes mientras que los patrones 11001001 (4 transiciones) y 0010010 (6 transiciones), no lo son. En el cálculo de las etiquetas LBP, se usan los patrones uniformes tal que hay una etiqueta asignada a cada patrón uniforme y todos los patrones que no son uniformes están etiquetados con la misma etiqueta. Por ejemplo, cuando se usa un vecindario (8, R), hay un total de 256 patrones, 58 de los cuales son uniformes, lo que genera 59 etiquetas diferentes.

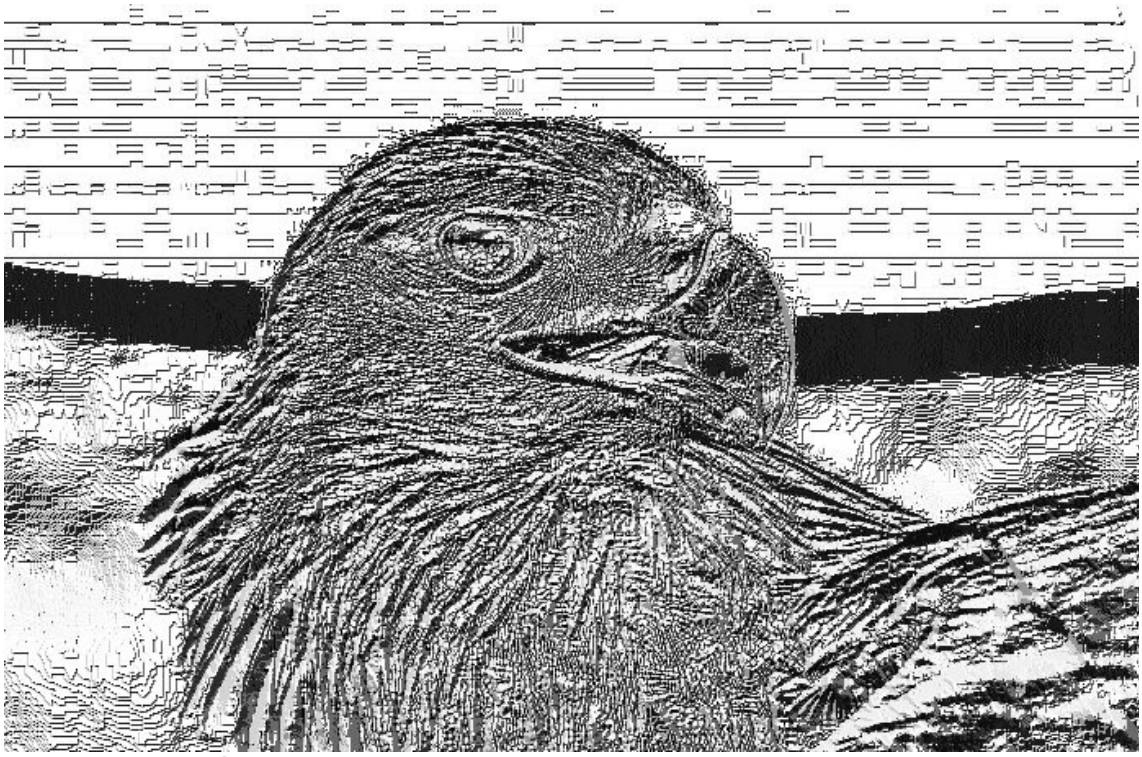


Ilustración 5: Imagen CT obtenida con los patrones uniformes de LBP

En la ilustración 5 se comprueba como no existen diferencias apreciables entre la imagen CT tomando en cuenta los 256 valores del histograma de CENTRIST y la imagen obtenida con los 59 valores del histograma de LBP.

Los autores de LBP observaron en sus experimentos con imágenes con texturas que los patrones uniformes alcanzaban poco menos del 90% de todos los patrones posibles cuando se usaba el vecindario (8, 1) y alrededor de un 70% para el vecindario (16, 2). Cada contenedor (código LBP) puede ser tomando como un microtexton.

3. Integración de los descriptores de imagen en el algoritmo MMSC

MMSC

MMSC es un método para la integración no supervisada de las características heterogéneas de imágenes sin etiquetar y sin segmentar basado en un agrupamiento espectral multimodal que fue descrito por Xiao Cai, Feiping Nie, Heng Huang, Farhad Kamangar en la Universidad de Texas.

Considerando cada tipo de característica (extraídas por los diferentes descriptores) como un modo, el algoritmo de agrupamiento espectral multimodal (*Multi-modal Spectral Clustering*, MMSC) calcula una sola matriz laplaciana para todos ellos, ya que unifica los diferentes descriptores de las imágenes. Después añade una relajación no negativa para mejorar su fiabilidad y eficiencia en el agrupamiento de imágenes.

En este Trabajo Fin de Máster se ha aplicado el método MMSC para integrar cinco tipos de descriptores de imágenes ampliamente usados: SIFT, HOG, GIST, CENTRIST y LBP.

En comparación con los métodos no supervisados existentes de categorización de objetos y escenas, el método siempre alcanza un mejor rendimiento que utilizando cada uno de ellos de manera individual.

Ya que diferentes descriptores describen diferentes aspectos de las características visuales, es cierto que un descriptor puede tener mejor desempeño que los otros bajo ciertas circunstancias. Si entonces integrásemos todos los descriptores de acuerdo a un método de aprendizaje adecuado, se podría crear un descriptor más preciso y fiable que cualquiera por separado, lo cual se asemejaría al escenario de utilizar una vista múltiple para observar un objeto, en el que podríamos advertir mejor sus detalles. Esto es lo que consigue el algoritmo MMSC.

Se han propuesto varios métodos para la integración de características heterogéneas, tales como HFM (*Heterogeneus Features Machine*), para unir de forma supervisada los distintos tipos de características para efectuar las

clasificaciones visuales. Sin embargo, la categorización no supervisada de imágenes (o sus partes) es necesaria, ya sea como un agrupamiento definitivo o como parte de un preproceso para una clasificación supervisada posterior.

Siendo éste un problema difícil de resolver, la gran mayoría de intentos realizados, se han basado en la utilización de un único descriptor de las características de la imagen, sin la ayuda de otros descriptores.

Un paso adelante frente a estos intentos anteriores es el método de agrupamiento espectral multimodal (MMSC) para integrar las características heterogéneas de una imagen en tareas de visión artificial no supervisadas.

En el agrupamiento espectral multimodal, cada modo se vincula a un tipo de descriptor de características. Para mejorar la efectividad limitada a las condiciones iniciales y mejorar el rendimiento, se añade una constante no negativa a la función objetivo. Esto permite alcanzar una mejor categorización de escenas que si se realizara el agrupamiento espectral sobre únicamente un descriptor visual.

Agrupamiento espectral

dadas n imágenes $X=\{x_1, x_2, \ldots, x_n\}$, la tarea de categorización de imágenes no supervisada es la de particionar X en k grupos. Por convenio, llamamos a la matriz de indicador de grupo como $Q=[q_1, q_2, \ldots, q_n]^T \in \mathbb{R}^{n x k}$, donde $q_i \in \mathbb{R}^{k x 1}$ $(1 \leqslant i \leqslant n)$ es el vector de indicador de grupo para la imagen x_i . El elemento j-iésimo de q_i es 1, si la imagen x_i es asignada a la j-iésima categoría; 0 en los demás casos. El agrupamiento espectral va a calcular entonces la matriz de indicador de grupo Q a través de un grafo ponderado no dirigido $G=\{V, A, W\}$ con un conjunto de vértices V , un conjunto de aristas A y una matriz de afinidad $W \in \mathbb{R}^{n x n}$.

W es una matriz simétrica donde cada entrada W_{ij} representa la similitud o afinidad entre los pares de vértices si el punto x_i está entre los k vecinos más cercanos de x_j o si x_j está entre los k vecinos más cercanos de x_i . A

menudo se utiliza un *kernel* gaussiano para la construcción de W_{ij} con $\exp(-\|x_i - x_j\|^2 / \sigma^2)$, si x_i y x_j son vecinos; 0 en los demás casos. σ es el parámetro que controla la dispersión de los vecinos en el grafo de los k vecinos más próximos.

Estos son los pasos a seguir para realizar un agrupamiento espectral:

Entrada:	Matriz de similitud W , número de k grupos a construir
Procedimiento:	
1)	Sea L la matriz laplaciana no normalizada de W
2)	Calcular los primeros k eigenvectores $u_1, u_2, \ldots, u_k$ de L (los eigenvectores de menor valor)
3)	Sea $U \in \mathbb{R}^{n x n}$ la matriz que contiene los vectores $u_1, u_2, \ldots, u_k$ como columnas
4)	Para $i=1, \ldots, n$ sea $y_i \in \mathbb{R}^k$ el vector correspondiente a la fila i-ésima de U
5)	Agrupar los puntos $(y_i)_{i=1,\ldots,n}$ en $\mathbb{R}^k$ con el algoritmo k-means en los grupos $C_1, \ldots, C_k$
Salida:	Grupos $A_1, \ldots, A_k$ con $A_i = \{ j \mid y_j \in C_i \}$

Tabla 1: Algoritmo de agrupamiento espectral

Para calcular la matriz laplaciana de un grafo ponderado $G = \{V, E, \omega\}$, donde G es simplemente un grafo no dirigido $G = \{V, E\}$ con la función $\omega : E \rightarrow \mathbb{R}$ donde $\mathbb{R}$ denota el conjunto de números reales positivos. La matriz de adyacencia (similitud) A_G de un grafo ponderado es dada por:

$$A_G(i, j) = \begin{Bmatrix} \omega(i, j) & si\,(i, j) \in E \\ 0 & en\ los\ demás\ casos \end{Bmatrix}$$

La matriz de grados D_G de un grafo ponderado G es la matriz diagonal tal que:

$$D_G(i,i)=\sum_j A_G(i,j)$$

La matriz laplaciana L_G de un grafo ponderado G se calcula como sigue:

$$L_G=D_G-A_G$$

Agrupamiento espectral multimodal

Hay dos mecanismos principales para encontrar las soluciones al problema del aprendizaje no supervisado multimodal:

- El primero está basado en algoritmos centralizados, que hacen uso de múltiples perspectivas simultáneamente para averiguar los patrones ocultos en los datos.

- La segunda intenta resolver el problema de agrupamiento multimodal no supervisado de manera distribuida, esto es, averiguar los patrones ocultos de forma individual en cada representación y entonces averiguar los patrones ocultos desde estos patrones múltiples.

Para integrar de manera natural estas características heterogéneas de las imágenes, se propone una función objetivo unificada para la optimización simultánea de los resultados de agrupamiento de cada descriptor y sus combinaciones. Es otras palabras, se minimiza el error de agrupamiento espectral de cada descriptor y se minimizan también las distancias entre la matriz indicadora del agrupamiento multimodal y la matriz indicadora del agrupamiento de cada descriptor en particular. De esta manera, la función objetivo del agrupamiento espectral multimodal es

$$\min_{G \geqslant 0, G^T G = I, G^i} \sum_i O(G, G^i)$$

donde $O(G, G^i) = Tr(G^i)^T L^i G^i + \alpha Tr(G - G^i)^T (G - G^i)$

L^i es la matriz laplaciana correspondiente a cada descriptor en particular, G^i es la matriz indicadora de grupo de cada descriptor en particular, α es la constante de penalización y G es la matriz final indicadora de grupo multimodal.

Así, dada la matriz laplaciana de cada descriptor en particular, se utiliza la ecuación anterior para calcular de forma simultánea la matriz indicadora de grupo para cada descriptor y la matriz indicadora de grupo multimodal.

Algoritmo MMSC

Entrada:	Dado un conjunto de imágenes con n imágenes, el número de grupos c y el número de k vecinos más próximos para construir el grafo de los k-vecinos más próximos:
Salida:	Matriz indicadora de grupo G
Procedimiento:	
1)	Calcular las cinco matrices de afinidad (similitud) $\{W^1, W^2, W^3, W^4, W^5\}$ correspondientes a los cinco descriptores diferentes de las imágenes, donde $W^i \in \mathbb{R}^{n x n}$
2)	Calcular cinco matrices laplacianas, $L = \{L^1, L^2, L^3, L^4, L^5\}$, donde $L^i = D^i - W^i$
3)	Calcular la matriz laplaciana multimodal, $L_{multimodal} = \sum_i^5 (L^i + \alpha I)^{-1}$
4)	Calcular los primeros c eigenvectores $u_1, u_2, \ldots, u_c$ de $L_{multimodal}$
5)	Sea $U \in \mathbb{R}^{n x c}$ la matriz que contiene los vectores $u_1, u_2, \ldots, u_n$ como columnas

6)	Formar la matriz $T \in \mathbb{R}^{n x c}$ desde U al normalizar las filas de U para ser norm 1. (Por ejemplo $T_{ij} = \frac{U_{ij}}{\sqrt{\sum_j x_{ij}^2}}$)
7)	Para $i = 1, 2, \ldots, n$ sea $g_i \in \mathbb{R}^c$ el vector correspondiente a la fila i-ésima de T
8)	Agrupar los puntos $(g_i)_{i=1,2,\ldots,n}$ con agrupamiento espectral para obtener G_0
9)	Usar $G_0 + 0.2$ como entrada y realizar el algoritmo iterativo con la relajación no negativa, para obtener la matriz final de asignación de grupo G

El algoritmo iterativo con relajación no negativa consiste en la siguiente ecuación:

$$G_{ij} \leftarrow G_{ij} \sqrt{\frac{(JG)_{ij}}{(G\alpha)_{ij}}} \quad \text{donde} \quad \alpha \equiv G^T J G \quad \text{y} \quad J = \sum_i (L^i + \alpha I)^{-1}$$

Tabla 2: Algoritmo MMSC

4. Tecnología utilizada y detalles de implementación

Para el desarrollo del algoritmo MMSC se ha decidido utilizar la librería de visión artificial OpenCV, eligiendo como lenguaje de programación C++, el cual tiene la ventaja de producir los binarios más eficientes para su ejecución sobre los diferentes equipos de cómputo.

OpenCV

OpenCV es una librería de visión artificial y de *machine learning* de código abierto originalmente desarrollada por Intel en el año 1999.

Su licencia es de tipo BSD, lo que permite que pueda ser usada libremente tanto para para propósitos comerciales como de investigación, con las condiciones en ella expresadas. Uno de los objetivos de la publicación bajo este tipo de licencia ha sido el de promover y acelerar el uso de la percepción computacional en los productos comerciales, además de promover una infraestructura común para facilitar su uso académico.

OpenCV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X, Windows, iOS y Android. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, tales como reconocimiento de objetos, reconocimiento facial, calibración de cámaras, visión estéreo, visión robótica, etcétera.

OpenCV pretende proporcionar un entorno de desarrollo fácil de utilizar y altamente eficiente, por lo que es adecuada para su utilización en entornos de tiempo real. Consigue esta eficiencia gracias a estar programado de manera optimizada en los lenguajes C y C++, aprovechando además las capacidades de los procesadores multinúcleo, así como las ventajas de las unidades gráficas de proceso, soportando tanto CUDA como OpenCL. OpenCV puede además utilizar el sistema de primitivas de rendimiento integradas de Intel, un conjunto de rutinas de bajo nivel específicas para procesadores Intel.

La librería tiene más de 2500 algoritmos optimizados, que incluyen un amplio conjunto de algoritmos clásicos y los exponentes del *state-of-the-art* sobre visión

artificial y *machine learning*.

Con seis millones totales de descargas y una comunidad de 47 mil personas, es claramente la librería de visión artificial de alto rendimiento más utilizada a nivel global.

Decisiones generales

Se ha optado por utilizar el método de auto-ajuste para obtener el parámetro σ que se va a utilizar con el *kernel* gaussiano con el objetivo de realizar las comparaciones entre los descriptores de las imágenes, al igual que en el artículo original de Xiao Cai.

Según este método, $\sigma_i = d(s_i, s_k)$, donde s_k es el k vecino más cercano al punto s_i . Si se utiliza un parámetro k igual o mayor a siete en la llamada a los descriptores o el algoritmo MMSC se utilizará $k=7$; y si el número es menor, se utilizará k .

Por tanto, con este método se propone calcular un parámetro local σ_i por cada punto s_i , en vez de seleccionar un único parámetro σ para todas las comparaciones.

La afinidad (o similitud) $\hat{A}_{ij}$ entre un par de puntos i y j puede ser escrita como:

$$\hat{A}_{i,j} = \exp\left(\frac{-d^2(s_i, s_j)}{\sigma_i \sigma_j}\right)$$

Donde d es la función distancia entre los puntos s_i y s_j .

SIFT

Se ha utilizado el extractor de características de la imagen ya implementado en OpenCV. Como procesado previo de las imágenes, se crea una copia de las mismas en escala de grises y redimensionadas a un tamaño de 128 x 128 píxeles.

Para la obtención de la matriz de similitud de todas las imágenes con este descriptor se han seguido los pasos siguientes:

Paso	Descripción
1	Crear una instancia de un comparador por fuerza bruta (`BFMatcher`) de todos los descriptores obtenidos en las imágenes.
2	Crear una instancia de un comparador destinado a eliminar las coincidencias de baja calidad entre las imágenes (`RobustMatcher`). Sus parámetros son: • `radio: 1.05` • `nivel de confianza: 0.98` • `distancia: 3.0`
3	Por cada para de imágenes almacenadas:
3.1	Encontrar las dos mejores coincidencias, ordenadas de menor a mayor en cuanto a distancia, para cada descriptor y para los dos pares de imágenes (imagen1, imagen2) y (imagen2, imagen1).
3.2	Para las coincidencias obtenidas en el paso 3.1, se efectúa un test de ratio para eliminar las coincidencias de un mismo par cuya distancia entre ellas supera cierto umbral (`radio = 1.05`)
3.3	A las coincidencias que han superado el filtro del paso 3.2, se les aplica un test de simetría entre ellas, para eliminar aquellas que no son simétricas.
3.4	Las coincidencias que superan el filtro del paso del paso 3.3 son las que se toman para medir la distancia entre los descriptores de las imágenes: • si no hay coincidencias, se toma la máxima distancia posible entre

	las dos imágenes (`std::numeric_limits<double>::max()`) • si hay coincidencias, éstas se ordenan ascendentemente en cuanto a distancia y se suman las distancias de las catorce primeras
4	Crear la matriz de σ auto-ajustadas para cada par de imágenes.
5	Calcular la similitud con el *kernel* gaussiano para los descriptores de cada par de imágenes con sus σ determinadas, formando así la matriz de similitud para el descriptor SIFT.

Tabla 3: Obtención de la matriz de similitud de SIFT

HOG

Se ha utilizado el extractor de características de la imagen ya implementado en OpenCV. Como procesado previo de las imágenes, se crea una copia de las mismas en escala de grises y redimensionadas a un tamaño de 128 x 128 píxeles.

Para la obtención de la matriz de similitud de todas las imágenes con este descriptor se han seguido los pasos siguientes:

Paso	Descripción
1	Crear la matriz de distancias entre cada par de imágenes utilizando la distancia euclídea.
2	Crear la matriz de σ auto-ajustadas para cada par de imágenes.
3	Calcular la similitud con el *kernel* gaussiano para los descriptores de cada par de imágenes con sus σ determinadas, formando así la matriz de similitud para el descriptor HOG.

Tabla 4: Obtención de la matriz de similitud de HOG

GIST

El desarrollo de este descriptor en el presente trabajo está basado en la implementación de Lear, desarrollada en INRIA por Matthijs Douze y otros. En este Trabajo Fin de Máster se ha conseguido integrar el descriptor con la librería OpenCV y mejorar levemente su rendimiento por medio de PCA. Como procesado previo de las imágenes, se crea una copia de las mismas en color y redimensionadas a un tamaño de 128 x 128 píxeles.

Para la obtención de la matriz de similitud de todas las imágenes con este descriptor se han seguido los pasos siguientes:

Paso	Descripción
1	Realizar una proyección PCA de los descriptores de todas las imágenes (los cuales constan de 960 valores cada uno). Seleccionar las 40 componentes principales.
2	Formar la matriz de distancias calculando la distancia euclídea entre las 40 componentes principales de los descriptores de las imágenes.
3	Crear la matriz de σ auto-ajustadas para cada par de imágenes.
4	Calcular la similitud con el *kernel* gaussiano para los descriptores de cada par de imágenes con sus σ determinadas, formando así la matriz de similitud para el descriptor GIST.

Tabla 5: Obtención de la matriz de similitud de GIST

CENTRIST

El desarrollo de este descriptor se ha realizado desde cero. Como procesado previo de las imágenes, se crea una copia de las mismas en escala de grises y redimensionadas a un tamaño de 128 x 128 píxeles.

Para la obtención de la matriz de similitud de todas las imágenes con este descriptor se han seguido los pasos siguientes:

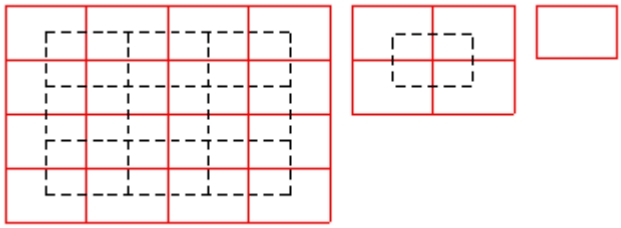
Paso	Descripción
1	Realizar una división de la imagen en 3 niveles:  *Ilustración 6: Nivel 2, nivel 1 y nivel 0 de la división de la imagen* El nivel 0 se conforma de la imagen original; el nivel 1 está formado por cuatro regiones y una quinta que se solapa con las demás en el centro; y el nivel 2 está formado por 16 regiones y 9 que se solapan en sus centros. Un total de 31 regiones a tratar.
2	Por cada una de las regiones:
2.1	Calcular la imagen del gradiente Sobel de la región.
2.2	Redimensionar la imagen Sobel al tamaño de la imagen original (128x128)
2.3	Calcular la imagen CT de la región redimensionada en el paso 2.2.
2.4	Extraer el histograma de la imagen CT obtenida en el paso 2.3 y desechar los valores 0 y 255.
2.5	Concatenar el histograma obtenido a los anteriores.
3	Normalizar el descriptor obtenido al concatenar todos los histogramas.
4	Formar la matriz de distancias calculando la distancia euclídea entre los histogramas concatenados de las imágenes.
5	Crear la matriz de σ auto-ajustadas para cada par de imágenes.
6	Calcular la similitud con el *kernel* gaussiano para los descriptores de cada par de imágenes con sus σ determinadas, formando así la matriz de similitud para el descriptor CENTRIST.

Tabla 6: Obtención de la matriz de similitud de CENTRIST

LBP

El desarrollo del descriptor LBP también se ha realizado por completo. Como procesado previo de las imágenes, se crea una copia de las mismas en escala de grises y redimensionadas a un tamaño de 128 x 128 píxeles.

Paso	Descripción
1	Realizar una división de la imagen en 64 regiones del mismo tamaño (una cuadrícula de 8 x 8).
2	Por cada una de las regiones:
2.1	Calcular la imagen uniforme a partir de los patrones uniformes utilizando 8 vecinos y un radio 1.
2.2	Obtener el histograma de la imagen uniforme obtenida en el paso 2.1.
2.3	Concatenar el histograma obtenido a los anteriores.
3	Realizar proyección PCA de los descriptores de todas las imágenes (consistentes en los histogramas concatenados). Seleccionar las 40 componentes principales.
4	Formar la matriz de distancias calculando la distancia euclídea entre los vectores formados por las 40 componentes principales de los descriptores de las imágenes.
5	Crear la matriz de σ auto-ajustadas para cada par de imágenes.
6	Calcular la similitud con el *kernel* gaussiano para los descriptores de cada par de imágenes con sus σ determinadas, formando así la matriz de similitud para el descriptor LBP.

Tabla 7: Obtención de la matriz de similitud de LBP

5. Pruebas y resultados obtenidos

Métricas de calidad de agrupamiento

Para tener una medida objetiva de los resultados obtenidos es imperativo el utilizar métricas estándar con el fin de evaluar la calidad de los agrupamientos obtenidos. En el caso de este Trabajo Fin de Máster se han utilizado las métricas NMI (*Normalized Mutual Information*) y *Purity*.

NMI (*Normalized Mutual Information*)

NMI es una métrica robusta para evaluar la calidad de un agrupamiento. Su definición formal viene dada por la siguiente ecuación:

$$NMI(\Omega, \mathbb{C}) = \frac{\sum_{g=1}^{K} \sum_{c=1}^{C} n_c^g \log\left(\frac{N \cdot n_c^g}{|G_g||C_c|}\right)}{\sqrt{\left(\sum_{g=1}^{K} |G_g| \log\frac{|G_g|}{N}\right)\left(\sum_{c=1}^{C} |C_c| \log\frac{|C_c|}{N}\right)}}$$

Donde N es el número total de elementos etiquetados, $\Omega = \{\omega_1, \ldots, \omega_k\}$ es el conjunto de grupos formados, $\mathbb{C} = \{c_1, \ldots, c_c\}$ es el conjunto de las clases de los datos etiquetados, K es el total de grupos formados, C es el número total de clases, $n_c^g = |\omega_g \cap c_c|$ es el número de objetos del grupo g que son de la clase c, y $|G_g|$ y $|C_c|$ son los cardinales del grupo g y de la clase c respectivamente.

Cuanto más alto sea el valor (qué estará en el intervalo (0, 1]), mayor será la calidad del agrupamiento.

Purity

Purity es una medida simple y transparente de calidad de un agrupamiento. La pureza de un agrupamiento es la fracción de los elementos asignados a la clase mayoritaria en cada grupo. Para calcular *purity*, cada grupo es asignado a la clase que es más frecuente en el mismo, y entonces se mide la precisión de esta asignación contando el número de elementos correctamente asignados y dividiéndolo por el número total de elementos:

$$purity(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

Donde $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ es el conjunto de grupos, y $\mathbb{C} = \{c_1, c_2, \ldots, c_J\}$ es el conjunto de clases. Se interpreta a ω_k como el conjunto de elementos en ω_k y a c_j como el conjunto de elementos en c_j .

Al igual que en NMI, cuanto más alto sea el valor (qué estará en el intervalo (0, 1]), mayor será la calidad del agrupamiento.

Se puede ver más claro con un ejemplo. Sean cuatro clases de escritos (ciencia, novelas, economía, noticias) y cuatro grupos:

	grupo1	**grupo2**	**grupo3**	**grupo4**
ciencia	0	4	**10**	4
novelas	**10**	0	1	0
economía	0	10	5	**12**
noticias	1	**12**	5	10

Tabla 8: Ejemplo de cálculo de Purity

En este caso la pureza sería $purity = \frac{1}{84}(10+12+10+12) = \frac{44}{84} \simeq 0.52$

Purity es una medida útil si el número de grupos está fijado y es mucho más pequeño que el número de elementos totales.

Pruebas efectuadas

En el artículo original del agrupamiento espectral se utilizan dos conjuntos de imágenes para realizar las pruebas, los correspondientes a CALTECH 101 y MSRC-v1. Dado que en las pruebas relativas a CALTECH 101 no se especifica un conjunto de imágenes (sino un número de ellas a seleccionar entre ciertas clases), y que en la prueba realizada con MSRC-v1 sí se especifica qué imágenes son las utilizadas, se va a operar sobre MSRC-v1 en primer lugar. Con ello se va a lograr establecer una comparación fiable con los resultados obtenidos por Xiao Cai y compañía, con la ventaja adicional de encontrar el valor óptimo para el valor de penalización α del algoritmo MMSC implementado en este trabajo. Utilizando los valores óptimos para este parámetro, los resultados de MMSC respecto al conjunto de imágenes CALTECH 101 serán más fiables.

Para todas las pruebas se va a establecer un número de grupos igual al número de clases etiquetadas en cada banco de imágenes (para MSRC-v1, siete clases; para CALTECH 101, siete y veinte clases respectivamente), estableciéndose $k=9$, con lo que se van a utilizar los nueve vecinos más cercanos a cada imagen para calcular su matriz de afinidad.

MSRC-v1

MSRC-v1 es un conjunto de imágenes categorizadas publicado por Microsoft. Consta de ocho clases, cada una con 30 imágenes, de las que se van a seleccionar las clases de árboles, edificios, aeroplanos, vacas, caras, coches y bicicletas, exactamente las mismas destinadas a las pruebas en el artículo original, con un total de 210 imágenes.

Ilustración 7: Muestra de imágenes de MSRC-v1

Estos son los resultados obtenidos por cada uno de los descriptores, en comparación con los presentados en el artículo de Xiao Cai:

descriptor	NMI	*Purity*	NMI Cai	*Purity* Cai
SIFT	**0.2704**	**0.4381**	0.2604	0.4358
HOG	**0.5843**	0.6238	0.5209	**0.6522**
GIST	**0.6539**	0.7047	0.6393	**0.7148**
CENTRIST	0.6008	0.6761	**0.6027**	**0.7374**
LBP	**0.5315**	0.6047	0.4719	**0.6136**

Tabla 9: Agrupamiento de los descriptores para MSRC-v1

A continuación se van a mostrar los resultados obtenidos por el algoritmo MMSC con diferentes valores del parámetro de penalización alfa, lo que nos servirá para

seleccionar el adecuado. En todas las ejecuciones se han empleado 1000 iteraciones:

alfa	NMI	*Purity*
0.1	0.7059	0.7762
0.2	**0.8150**	**0.9**
0.3	**0.7116**	**0.8**
0.4	0.6754	0.6857
0.5	**0.7210**	**0.8095**
0.6	**0.7233**	**0.8095**
0.7	**0.7088**	**0.8**
0.8	0.6721	0.6809
0.9	0.7598	0.7904
1.0	0.7110	0.7904
1.1	0.7207	0.7666
1.2	0.6932	0.7857
1.3	0.7019	0.7809
1.4	0.6868	0.7761
1.5	0.6927	0.7761
1.6	0.6792	0.7666
1.7	0.6792	0.7666
1.8	0.6739	0.7619
1.9	0.6603	0.7571
2.0	0.6535	0.7523

Tabla 10: Agrupamiento de MMSC para MSRC-v1

Como se observa, el algoritmo MMSC mejora ampliamente los resultados obtenidos por cada descriptor de manera individual.

Se han resaltado los mejores resultados correspondientes a los valores de alfa 0.2, 0.3, 0.5, 0.6 y 0.7. Estos van a ser los valores que se van a utilizar en la evaluación de rendimiento con el banco de pruebas CALTECH 101. Como se puede comprobar, el parámetro de penalización del algoritmo MMSC afecta en gran medida al resultado del agrupamiento.

descriptor	NMI	*Purity*	NMI Cai	*Purity* Cai
MMSC	**0.8150**	**0.9**	0.7745	0.8547

Tabla 11: Comparación resultados MMSC para MSRC-v1

CALTECH 101

Prueba 1: 7 clases

Se va a efectuar la primera prueba escogiendo siete categorías del banco de imágenes CALTECH 101. Las siete categorías son *Faces*, *Motorbikes*, *Dollar-Bill*, *Garfield*, *Snoopy*, *Stop-Sign* y *Windsor-Chair*; con un total de 441 imágenes elegidas aleatoriamente, intentando equilibrar al máximo el número de imágenes por cada categoría:

Categoría	Número de imágenes
Faces	100
Motorbikes	100
Dollar-Bill	52
Garfield	34
Snoopy	35
Stop-Sign	64
Windsor-Chair	56

Tabla 12: Distribución de las clases para la prueba 1 de CALTECH 101

Estos son los resultados obtenidos por cada uno de los descriptores, en comparación con los presentados en el artículo de Xiao Cai:

descriptor	NMI	*Purity*	NMI Cai	*Purity* Cai
SIFT	**0.6189**	**0.7211**	0.2744	0.7011
HOG	**0.5695**	0.6417	0.5466	**0.7078**
GIST	**0.5834**	0.6621	0.5636	**0.7250**
CENTRIST	**0.5551**	0.6326	0.5150	**0.6891**
LBP	0.4091	0.5170	**0.4188**	**0.5735**

Tabla 13: Agrupamiento de los descriptores en la prueba 1 de CALTECH 101

Ilustración 8: Muestra de imágenes de las 7 clases de la prueba 1 de CALTECH

A continuación se van a mostrar los resultados obtenidos por el algoritmo MMSC para los parámetros de penalización alfa que fueron obtenidos en la prueba del MSRC-v1 como óptimos. En todas las ejecuciones se han empleado 1000 iteraciones:

alfa	NMI	*Purity*
0.2	0.7302	0.7551
0.3	0.7230	0.7505
0.5	0.6707	0.7142
0.6	0.6758	0.7392
0.7	0.6766	0.7460

Tabla 14: Agrupamiento de MMSC en la prueba 1 de CALTECH 101

De nuevo se puede observar como el algoritmo MMSC mejora ampliamente los resultados obtenidos por cada descriptor de manera individual y como el parámetro de penalización α influye enormemente en el resultado.

descriptor	NMI	*Purity*	NMI Cai	*Purity* Cai
MMSC	**0.7302**	0.7551	0.6792	**0.7802**

Tabla 15: Comparación resultados MMSC en la prueba 1 de CALTECH 101

Prueba 2: 20 clases

En la segunda prueba se escogen 20 categorías del banco de imágenes CALTECH 101. Estas categorías son *Faces*, *Leopards*, *Motorbikes*, *Binocular*, *Brain*, *Camera*, *Car-Side*, *Dollar-Bill*, *Ferry*, *Garfield*, *Hedgehog*, *Pagoda*, *Rhino*, *Snoopy*, *Stapler*, *Stop-Sign*, *Water-Lilly*, *Windsor-Chair*, *Wrench* y *Yin-Yang*; con un total de 1230 imágenes, intentando equilibrar al máximo el número de imágenes por cada categoría. Esta es la distribución del número de imágenes:

Categoría	Número de imágenes
Faces	116
Leopards	100
Motorbikes	92
Binocular	33
Brain	98
Camera	50
Car-Side	92
Dollar-Bill	52
Ferry	67
Garfield	34
Hedgehog	54
Pagoda	47
Rhino	59
Snoopy	35
Stapler	45
Stop-Sign	64
Water-Lilly	37
Windsor-Chair	56
Wrench	39
Yin-Yang	60

Tabla 16: Distribución de las clases para la prueba 2 de CALTECH 101

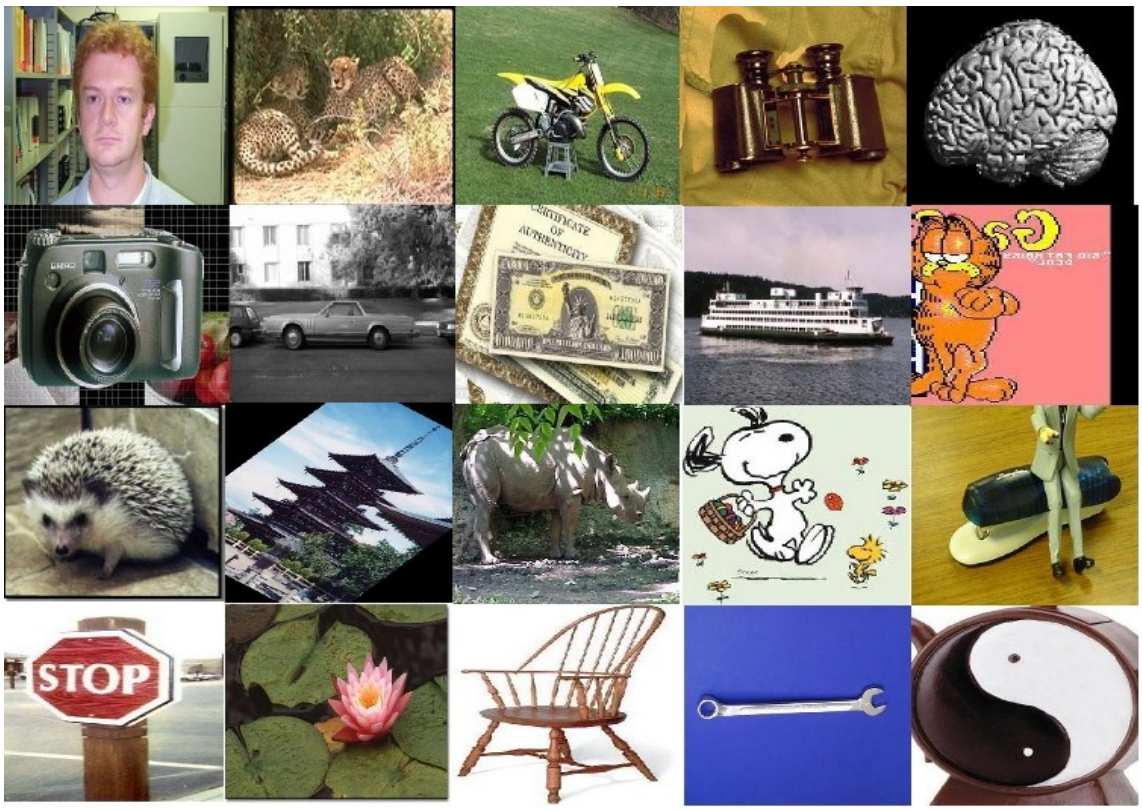


Ilustración 9: Muestra de imágenes de las 20 clases de la prueba 2 de CALTECH

Estos son los resultados obtenidos por cada uno de los descriptores, en comparación con los presentados en el artículo de Xiao Cai:

descriptor	NMI	*Purity*	NMI Cai	*Purity* Cai
SIFT	**0.3565**	**0.3991**	0.2876	0.3122
HOG	**0.5259**	**0.4975**	0.4326	0.4227
GIST	**0.5469**	**0.5333**	0.4692	0.4499
CENTRIST	**0.5639**	0.5382	0.5514	**0.5942**
LBP	**0.4636**	**0.4609**	0.3802	0.3640

Tabla 17: Agrupamiento de los descriptores en la prueba 2 de CALTECH 101

A continuación se van a mostrar los resultados obtenidos por el algoritmo MMSC para los parámetros de penalización alfa que fueron obtenidos en la prueba del MSRC-v1 como óptimos. En todas las ejecuciones se han empleado 1000 iteraciones:

alfa	**NMI**	***Purity***
0.2	0.6656	0.6699
0.3	0.6555	0.6658
0.5	0.6312	0.6382
0.6	0.6209	0.6398
0.7	0.6193	0.6073

Tabla 18: Agrupamiento de MMSC en la prueba 2 de CALTECH 101

De nuevo los resultados de MMSC mejoran significativamente los obtenidos por cada descriptor por separado.

descriptor	**NMI**	***Purity***	**NMI Cai**	***Purity* Cai**
MMSC	**0.6656**	**0.6699**	0.6329	0.6132

Tabla 19: Comparación resultados MMSC en la prueba 2 de CALTECH 101

6. Conclusiones. Futuros trabajos.

De los resultados obtenidos se extrae que el algoritmo MMSC de agrupamiento multimodal no supervisado mejora visiblemente la eficiencia del agrupamiento no supervisado de imágenes (tomando como referencia la métrica NMI):

- En el banco de pruebas de MSRC-v1 mejora en un 24,64% al descriptor individual con mejor desempeño en la prueba (GIST).
- En la primera prueba realizada con el conjunto de imágenes de CALTECH 101, mejora en un 17.98% al descriptor con mejor desempeño en la prueba (SIFT).
- En la ejecución de la segunda prueba efectuada con el conjunto de imágenes de CALTECH 101, mejora en un 18.039% al descriptor con mejor desempeño (CENTRIST).

Respecto a la comparación con el artículo original del algoritmo de MMSC propuesto, las mejoras experimentadas en base a la métrica NMI son de un 5.23%, 7.51% y 5.17% respectivamente para cada una de las tres pruebas; si bien es cierto que el resultado más fiable es el de la prueba con el conjunto de imágenes de MSRC-v1, ya que existe certeza de que se han utilizado exactamente las mismas imágenes en el artículo de Xiao Cai y en este Trabajo Fin de Máster.

Esta leve ventaja en rendimiento seguramente esté basada en una mejor obtención y una comparación más refinada de los descriptores de características, lo que permite mejores resultados obtenidos de forma independiente por cada uno de ellos (en forma de la matriz de similitud) que se ven reflejados en el desempeño final del algoritmo MMSC.

Como tareas a realizar en el futuro, se puede mejorar aún más la eficiencia del algoritmo, tanto en términos de calidad de agrupamiento como en los de tiempo de procesado. Para conseguir un mejor agrupamiento se deben añadir, probar y crear nuevos descriptores de características de imágenes, así como refinar sus comparaciones, con el objetivo de extraer características más distintivas y/o con mayor contenido semántico *per se*.

Conseguir mejoras en el tiempo de proceso es una tarea mucho más sencilla que la anterior: en la implementación realizada en este Trabajo Fin de Máster no se ha aprovechado la programación multinúcleo, así como las unidades gráficas de proceso. Los algoritmos en los que se basa la extracción de descriptores de las imágenes son fácilmente paralelizables, por lo que se pueden escalar al máximo, permitiendo ofrecer resultados en tiempo real.

7. Bibliografía

Davies E. R. (2012). *Computer & Machine Vision. Fourth edition*. Elsevier.

Laganiére R. (2011). *OpenCV Computer Vision Application Programming Cookbook*. Packt Publishing.

Forsyth D.; Ponce J. (2012). *Computer Vision. A modern approach. Second edition*. Pearson.

Artículos y webs:

- **Cai X.; Nie F.; Huang H.; Kamangar F.** (2011). "Heterogeneus Image feature integration via multi-modal spectral clustering". *The 24th IEEE conference on Computer Vision and Patter Recognition, CVPR 2011. 01/2011*; páginas 1977-1984.
- **Chang-Yeon J.** (2008). "Face Detection using LBP features". *CS 229 Final Project Report*.
- **Spielman D.** (2009). "Spectral Graph Theory. The Laplacian". Lecture 2.
- **Ding C.; Li T.; Jordan M.** (2008). "Nonnegative Matrix Factorization for Combinatorial Optimization Spectral Clustering, Graph Matching, and Clique Finding". *2008 Eighth IEEE International Conference on Data Mining*; páginas 183-192.
- **Nie F.; Xu D.; Tsang I.; Zhang C.** (2009). "Spectral Embedded Clustering". *IJCAI 2009*; páginas 1181-1186.
- **Meyer C.; Wessel C.** (2010). "A Comparison of Methods for Creating Similarity Matrices". *34th SIAM Southeastern-Atlantic Section Conference, March 21, 2010*.
- **Lowe D.** (2004). "Distinctive image features from scale-invariant keypoints". *International Journal of Computer Vision, 60, 2*; páginas 91-110.
- **Chung F.; Richardson M.** (2006). "Weighted Laplacians and th Sigma Function of a Graph". *Contemporary mathematics*; páginas 93-108.
- **von Luxburg U.** (2007). "A Tutorial on Spectral Cluistering". *Statistics and Computing*, 17 (4); páginas 395-416.
- **Zolynski G.; Braun T.; Berns K.** (2008). "Local Binary Pattern Based Texture Analysis in Real Time using a Graphics Processing Unit". *VDI-Bericht*; páginas 321-

324.

- **Wu J.; Rehg. J.** (2011). "CENTRIST: A Visual Descriptor for Scene Categorization". *IEEE transactions on pattern analysis and machine intelligence*; páginas 1489-1501.
- **Donze M.; Jegóu H.; Sandhawalia H. y otros.** (2009). "Evaluation of GIST descriptors for web-scale image search". *CIVR Proceedings of the ACM International Conference on Image and Video Retrieval*. Artículo número 19.
- **Yun W.; Kim D.; Song B.; Yoon H.** (2009). "Block comparison based face identification using HOG feature". *The 18th IEEE International Symposium on Robot and Human Interactive Communication*; páginas 484-487.
- **Huynh T.; Min R.; Dugelay J.** (2013). "An efficient LBP-based descriptor for Facial Depth Images applied to Gender Recognition using RGB-D Face Data". *Journal on data semantics*; páginas 133-145.
- **Hu J.; Guo P. (2012).** "Spatial Local Binary Patterns for Scene Image Classification". *Science of Electronics, Technologies of Information and Telecommunications (SETIT), 2012 6th International Conference*; páginas 326-330.
- **Zelnik-Manor L.; Perona P. (2004).** "Self-tuning spectral clustering". *NIPS*.
- http://docs.opencv.org
- http://www.scolarpedia.org/article/Local_Binary_Patterns