



Trabajo de Fin de Grado

GRADO DE INGENIERÍA INFORMÁTICA

**Facultad de Matemáticas
Universidad de Barcelona**

**PhotoMeet Barcelona: Sistema de
reconocimiento de edificios**

Guillermo Ruiz Fernández

Directores: Laura Igual i Santi Seguí
Realizado en: Departamento de Matemática
Aplicada i anàlisis. UB
Barcelona, 20 de Junio de 2013

RESUMEN

Hoy en día hacemos fotos que son los recuerdos de nuestros viajes, de celebraciones, de nuestros seres queridos o sitios preferidos. La gran variedad de cámaras y dispositivos a los que tenemos acceso que nos permiten tomar fotografías con facilidad, junto con la gran oferta de redes sociales para compartirlas hacen que tengamos nuestras vidas documentadas en imágenes. Dada esta gran cantidad de imágenes que se genera y que a su vez cada individuo comparte en las redes sociales, es interesante plantear el desarrollo de un sistema que nos permita categorizar y ordenar todas estas fotografías.

En concreto, en este proyecto se presenta un sistema de reconocimiento y clasificación de edificios. El resultado final es un sistema “inteligente” capaz de extraer información de las imágenes y de reconocer algunos de los edificios más emblemáticos de la ciudad de Barcelona.

Este trabajo forma parte de PhotoMeet Barcelona, un proyecto donde se desarrolla una red social de fotografías donde un sistema como el que se ha desarrollado podría potenciar algunas de sus funcionalidades clasificando y geolocalizando automáticamente las imágenes subidas por los usuarios.

Para el desarrollo de este proyecto se utilizan métodos de Visión por Computador e Inteligencia Artificial, que nos permitirán la extracción de características de las imágenes y la creación de un sistema inteligente de aprendizaje capaz de reconocer edificios.

RESUM

Avui en dia fem fotos que on els records de viatges, de celebracions, de les persones que estimem o dels nostres llocs preferits. La gran varietat de càmeres i dispositius als que tenim accés i ens permeten fer fotografies amb facilitat, juntament amb la gran oferta de xarxes socials per compartir-les fan que tinguem les nostres vides documentades en imatges. Donada aquesta gran quantitat d'imatges que es generada i que alhora cada individu comparteix a les xarxes socials, és interessant plantejar el desenvolupament d'un sistema que ens permeti categoritzar i ordenar totes aquestes fotografies.

En concret, en aquest projecte es presenta un sistema de reconeixement i classificació d'edificis. El resultat final, és un sistema “intel·ligent” capaç d'extreure informació de les imatges i de reconèixer alguns dels edificis més emblemàtics de la ciutat de Barcelona.

Aquest treball forma part de PhotoMeet Barcelona, un projecte on es desenvolupa una xarxa social de fotografies on un sistema com el que s'ha desenvolupat podria potenciar algunes de les seves funcionalitats classificant i geolocalitzant de manera automàtica les imatges penjades per els usuaris.

Per dur a terme el projecte, es fan servir mètodes de Visió per Computador i Intel·ligència Artificial, que ens permetran l'extracció de característiques de les imatges i la creació d'un sistema intel·ligent d'aprenentatge capaç de reconèixer edificis.

SUMMARY

Today we take photos for granted which are our memories of holidays and parties, of people and places. The explosion of the cameras and devices that we have today allow us to take pictures easily and the wide range of social networks to share them makes that we have our lives documented with images. With this big number of images that we generate and at the same time are shared in the social networks, is interesting to propose the development of a system that allow us to categorize and sort properly all of this pictures.

Specifically, in this project we present a system of building recognition and classification. The final result is an "intelligent" system able to extract the information of the images and recognize some of the most emblematic buildings from Barcelona.

This work is part of PhotoMeet Barcelona, a project where a social network based in share pictures is developed and a system such as this one presented could be very useful to improve some of the functionalities classifying and geolocating automatically the images uploaded for the users.

During the process of development of this project, Computer Vision and Artificial Intelligence methods are used, which will allow us to extract the images characteristics and the creation of an intelligent learning system able to recognize buildings.

Índice

Índice	4
Índice de Figuras	6
Índice de Tablas	6
1 Introducción	7
1.1 Motivación	10
1.2 Objetivos Generales	11
1.3 Estructura de la memoria	12
2 Arquitectura del Sistema	13
3 Diagramas de bloques	14
3.1 Creación del Sistema de Clasificación	14
4 Extracción de características	15
4.1 Descriptor básico: Histogram of Oriented Gradients (HOG)	17
4.1.1 HOG	18
4.1.2 Descriptor de imágenes basado en HOG	19
4.2 Definición de un diccionario de palabras visuales	20
4.2.1 Clustering K-Means	21
4.2.2 Obtener las palabras con K-Means	22
4.3 Representación de imágenes en el diccionario visual	23
4.3.1 Creación de Histogramas	23
4.3.2 Convoluciones	23
5 Clasificación: Entrenamiento / Test	24
5.1 Máquinas de Soporte de Vectores (SVM)	24
5.1.1 Idea básica y concepto matemático	24
5.1.2 MultiClass Classification: One versus all	26
6 Casos de uso	27
7 Diagrama de Gantt	31
8 Estudio económico	32
9 Matlab	33
10 Experimentos	35
10.1 Método de validación	35
10.1.1 Cross Validation	35
10.2 Medidas de evaluación	36

10.3	Resultados cuantitativos	37
10.3.1	Método Histogramas	37
10.3.2	Método Convoluciones.....	39
10.4	Resultados cualitativos.....	41
11	Conclusiones	45
12	Bibliografía	46

Índice de Figuras

Figura 1. Disciplinas relacionadas con la Visión por Computador (ubsense)	8
Figura 2. Aplicaciones de consumo en Visión por computador (ubsense)	10
Figura 3. Diagrama de bloques	14
Figura 4. Diagrama de bloques del proceso 2	14
Figura 5. Extracción y contabilización de palabras en un texto. (Fei-Fei)	15
Figura 6. Esquema de la extracción de palabras visuales de una imagen	16
Figura 7. Ilustración de la creación de las características de 3 imágenes a partir de un diccionario de palabras visuales	17
Figura 8. Ejemplo de una Imagen y la representación de sus características de HOG I (Andrea Vedaldi)	18
Figura 9. Ejemplo de una Imagen y la representación de sus características de HOG II (Andrea Vedaldi)	19
Figura 10. Representación de la matriz con las características de HOG	20
Figura 11. Representación del algoritmo K-Means	21
Figura 12. Hiperplano que separa puntos de dos clases distintas	25
Figura 13. Vectores de Soporte e hiperplanos	25
Figura 14. Ejemplo de One Versus All para un problema de 3 clases (Fernández, López, Galar, Jesús, & Herrera, 2013)	26
Figura 15. Diagrama de Casos de Uso	27
Figura 16. Diagrama de Gantt (Barashev & Thomas)	31
Figura 17. Esquema del método Cross Validation (Wikipedia, Validación Cruzada)	35
Figura 18. Gráfica del porcentaje de la performance del sistema, según el número de Clústeres (parámetro k)	39
Figura 19. Gráfica del porcentaje de la performance del sistema, según el número de Clústeres (parámetro k) del método de convoluciones	41

Índice de Tablas

Tabla 1. Palabras visuales en 3 clústeres distintos de cada clase	22
Tabla 2. Estudio Económico de los Costes del proyecto	32
Tabla 3. Requisitos Recomendados del Sistema (Mathworks)	33
Tabla 4. Especificaciones del equipo	34
Tabla 5. Esquema de la Matriz de Confusión	36
Tabla 6. Resultados cuantitativos del método Histogramas	38
Tabla 7. Resultados cuantitativos del método convoluciones	40
Tabla 8. Ejemplos de imágenes estimadas incorrectamente por el sistema de clasificación	43
Tabla 9. Ejemplos de imágenes estimadas correctamente por el sistema de clasificación	44

1 Introducción

Con la amplia difusión de smartphones, tablets y cámaras digitales casi todo el mundo tiene acceso a una cámara y por tanto se generan millones de fotografías cada día. Además, con la alta aceptación que tienen las redes sociales existen incentivos que nos llevan a querer hacer fotos de lo que sucede en nuestro entorno, para luego compartirlo. En promedio se suben un total de 300 millones de fotografías diarias a Facebook, según Rick Armbrust, encargado del desarrollo de negocios de esta plataforma. En Instagram, otra red social enfocada a compartir imágenes y aplicarles filtros, se suben cada día 5 millones de fotografías.

Está claro que cada vez más interesa el estudio y el tratamiento de la gran cantidad de imágenes que se producen en el mundo. Una imagen contiene información mucho más allá de la que podemos observar o percibir a simple vista. Si tratamos las diferentes características de una imagen como el color, texturas, intensidad o contornos, podemos extraer información relevante que a priori nos era oculta a simple vista.

La visión artificial es un subcampo de la inteligencia artificial. Se puede decir que la visión artificial busca programar un computador para que “entienda” una escena a partir de sus características. Por eso, mediante esta disciplina podemos conseguir el análisis y posterior resultado de tratar las imágenes.

En este proyecto se quiere crear un sistema inteligente capaz de reconocer diversos edificios emblemáticos de la ciudad de Barcelona. Las principales fases en que se divide el sistema son las siguientes:

- **Obtención de Imágenes:** Para construir un sistema inteligente hay que partir de una base y un entrenamiento previo, por lo tanto, se necesita una Base de Datos de imágenes.
- **Extracción de Características:** A partir de las imágenes obtenidas se extraerán y se tratarán sus características mediante métodos de visión por computador.
- **Entrenamiento del Sistema:** Una vez insertadas las características obtenidas, el sistema las analiza, luego son divididas en los grupos existentes y clasificadas.
- **Test del Sistema:** Finalmente hay que comprobar la efectividad del sistema con nuevas imágenes y observar si éste es capaz de reconocer y clasificarlas correctamente.

Este proyecto está incluido en los campos de procesamiento de imágenes y de visión artificial.

El procesamiento de imágenes es una forma de procesamiento de señales en la que la entrada es una imagen, la salida de la imagen procesada puede ser otra imagen o un conjunto de características o parámetros relacionados a la imagen de entrada.

Se puede clasificar el procesamiento en 3 categorías:

- **Procesamiento de Imágenes** (Imagen -> Imagen)
- **Análisis de Imágenes** (Imagen -> Parámetros)
- **Interpretación de Imágenes** (Imagen -> Descriptores)

Uno de los aspectos interesantes del estudio de esta área es la gran cantidad y diversidad de aplicaciones que hacen uso del procesamiento de imágenes o las técnicas de análisis. Los datos son a menudo multidimensionales y pueden ser transformados en un formato apropiado para su visualización.

Por otro lado, la Visión por Computador es la extracción de información automatizada en imágenes. Se trata de describir el mundo que vemos en una o más imágenes y reconstruirlo a partir de sus propiedades: trata de describir los modelos 3D, perspectivas de la posición de la cámara y objetos o contenido de la imagen. Estos procesos pueden llegar a resultar complejos ya que buscamos recuperar un conocimiento sin suficiente información como para especificar completamente la solución, pues existen muchas ambigüedades.

Existe un gran número de disciplinas relacionadas con la Visión por Computador y sus técnicas. A continuación en la Figura 1 se muestra un esquema de las principales disciplinas que guardan relación con la Visión por Computador.

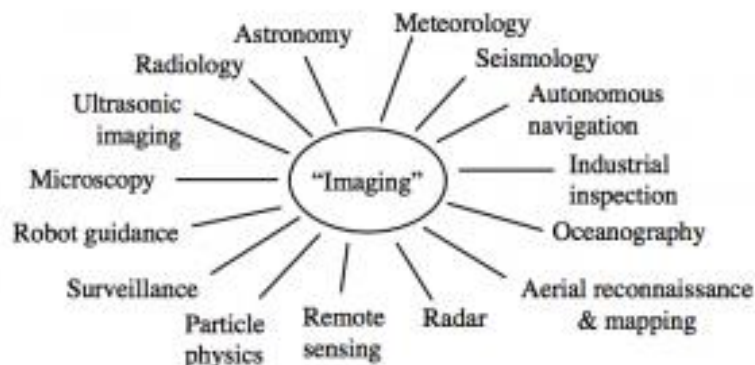


Figura 1. Disciplinas relacionadas con la Visión por Computador (ubsense)

Algunas aplicaciones industriales y ejemplos más concretos de la visión por computador son:

- **Reconocimiento Óptico de Caracteres (OCR):** lectura de códigos postales escritos a mano en cartas y reconocimiento automático del número de placa.
- **Inspección Industrial:** inspección rápida de partes para asegurar la calidad usando visión estéreo con iluminación especializada para medir tolerancias en alas de aviones, partes de robots, o búsqueda de defectos usando rayos-x.
- **Venta al detalle:** reconocimiento de objetos para ventas automáticas.
- **Modelado 3D:** construcción completamente automática de modelos 3D a partir de fotografías aéreas.
- **Seguridad en Automóviles:** detección de obstáculos inesperados.
- **Vigilancia:** monitoreo de intrusos, análisis de tráfico.
- **Reconocimiento de Huellas:** para autenticación de acceso automatizado y aplicaciones forenses.

También existen muchas aplicaciones a nivel de consumidor o masivas. Algunas de ellas son:

- **Combinado de Imágenes (Stitching):** fusión de varias imágenes traslapadas para crear una imagen panorámica.
- **Exposure Bracketing:** fusión de múltiples tomas bajo diferentes condiciones de luz para crear una mejor imagen.
- **Morphing:** crear una imagen de transición usando dos o más imágenes.
- **Modelado 3D:** convertir una o más fotografías en modelos 3D.
- **Detección de movimiento y estabilización:** insertar imágenes 2D u objetos 3D en video o estabilización de video para quitar el temblor propio de la toma.
- **Detección de rostros:** búsqueda de rostros en imágenes.

En la Figura 2 se muestran algunos ejemplos de aplicaciones a nivel de consumidor, muchas de ellas las tenemos presentes en los dispositivos que usamos en nuestro día a día.

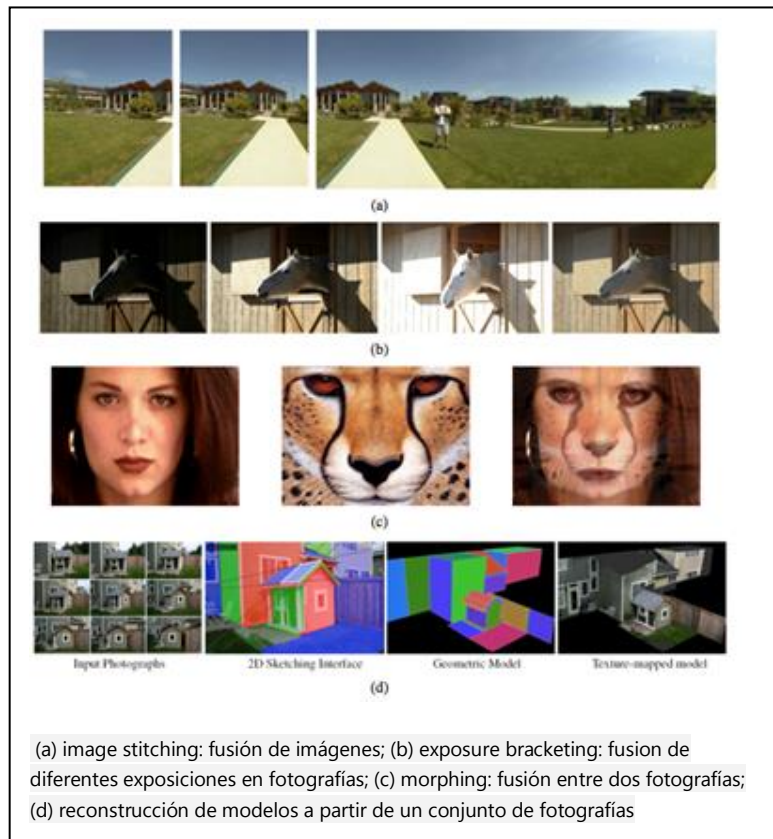


Figura 2. Aplicaciones de consumo en Visión por computador (ubsense)

1.1 Motivación

Durante la carrera he cursado, algunas asignaturas que tienen relación directa con esta temática y que han sido las que me han aportado los primeros conocimientos y la base necesaria para poder realizar un proyecto de estas características. Además han suscitado mi interés en este ámbito. Estas asignaturas son:

- **Visión Artificial:** Se introduce al alumno en un campo muy extenso y en continuo desarrollo como es la visión por computador. Además se enseñan las nociones básicas para entender los fenómenos y los modelos de la visión por computador y sus diferentes impactos en la vida real.
- **Procesamiento de Imágenes:** Se enseñan las técnicas más actuales de procesamiento de señales en el dominio de las imágenes. Se introduce al alumno en el campo del procesamiento de imágenes clásico.

Además, no se debe dejar de mencionar otras asignaturas que aunque no guarden una relación tan directa como las anteriores, me han aportado conocimientos de programación y experiencia en implementación y desarrollo de proyectos. Son las siguientes:

- Programación I: Se introducen elementos básicos de programación.
- Programación II: Se introducen elementos más avanzados en programación.
- Algorítmica: Se introducen conocimientos acerca de qué son los algoritmos, su implementación y su coste computacional.
- Estructura de Datos: Se introducen los conocimientos acerca de qué son i cómo utilizar las diferentes estructuras de Datos.
- Taller de Nuevos Usos de la Informática: Se introducen los nuevos usos de la informática generados en nuestra sociedad y se introducen conceptos de inteligencia artificial i aprendizaje automático.
- Diseño de Software: Se introducen metodologías a seguir para el correcto diseño i posterior desarrollo del Software.
- Proyecto Integrado de Software: Se desarrolla un proyecto de dimensión media, aplicando los conocimientos adquiridos hasta el momento.
- Inteligencia Artificial: Se introduce la disciplina de la Inteligencia Artificial (sus ramas y las técnicas que la forman).

Dado el gran número de aplicaciones que tiene este campo y la experiencia adquirida a lo largo de la carrera me he interesado en éste proyecto, como proyecto de fin de carrera, para seguir aprendiendo e investigando en el campo de la visión por computador.

1.2 Objetivos Generales

Este proyecto sigue la línea de investigación del proyecto de PhotoMeet Barcelona que se inició en 2011 por Santi Seguí, Laura Igual y alumnos de la UB que desarrollaron sus trabajos de Fin de Carrera. La investigación realizada en PhotoMeet Barcelona está dentro del campo de la visión por computador y la inteligencia artificial.

El proyecto PhotoMeet Barcelona, tiene como objetivo general la creación de un mapa virtual de imágenes de la ciudad de Barcelona, empezando por sus edificios más emblemáticos. Hasta ahora, englobando todos los trabajos de los diferentes alumnos que han participado en este proyecto, se han creado las siguientes aplicaciones:

- Una aplicación web. Se trata de una red social de fotografías donde los usuarios comparten sus imágenes y la geo-localización de estas.
- Un sistema capaz de detectar i localizar imágenes de edificios de Barcelona.
- Una aplicación Android, para dispositivos móviles.

Éste trabajo tiene el objetivo de seguir la investigación dentro de la rama de la visión artificial y desarrollar mediante métodos distintos a los anteriores un sistema capaz de reconocer y clasificar imágenes de edificios emblemáticos de Barcelona.

1.3 Estructura de la memoria

El presente documento se organiza en los siguientes capítulos:

Capítulo 1: Introducción.

En este capítulo se introducen cuáles son las motivaciones de este proyecto, en que ámbito está situado y los objetivos generales.

Capítulo 2: Arquitectura del sistema.

Se detalla los principales elementos y fases que conforman el Sistema.

Capítulo 3: Diagramas de bloques.

Se muestra el funcionamiento interno del sistema a partir de su diagrama de bloques.

Capítulo 4: Extracción de características

Se detallan los métodos utilizados para la extracción de las características de las imágenes.

Capítulo 5: Clasificación: Entrenamiento/Test

Se detallan los métodos de entrenamiento y test que sigue el sistema, así como la construcción del clasificador.

Capítulo 6: Casos de Uso

Análisis del sistema mediante los casos de uso.

Capítulo 7: Diagrama de Gantt

Diagrama de Gantt del proyecto.

Capítulo 8: Estudio Económico

Cálculo aproximado de los costes del proyecto.

Capítulo 9: Matlab

Información acerca de la herramienta de Matlab y los requisitos recomendados del sistema.

Capítulo 10: Experimentos

Se exponen y se explican los resultados obtenidos (cualitativos y cuantitativos) y los métodos de validación.

Capítulo 11: Conclusiones

Conclusiones del proyecto. Valoración de los objetivos y posibles líneas de continuación.

Capítulo 12: Bibliografía

Referencias y fuentes de información consultadas.

2 Arquitectura del Sistema

Se enumeran a continuación las principales fases de la Arquitectura del Sistema:

- **Extracción de características:** Extracción de las características de las imágenes usando como descriptor Histogramas de Gradientes Orientados (HOG).
- **Definición de un diccionario de palabras visuales:** Definimos nuestro propio diccionario de palabras visuales a partir de fragmentos de cada una de las imágenes con la técnica de clustering.
- **Representación de imagen en el diccionario visual:** Se determina de qué manera las palabras visuales de nuestro diccionario están presentes en las imágenes. Se han desarrollado dos métodos para obtener los vectores de características de las imágenes.
 - o Creación de Histogramas
 - o Convoluciones
- **Entrenamiento del Sistema:** Implementación de un sistema de clasificación mediante Support Vector Machine (SVM) con los datos de entrenamiento.
- **Test del Sistema:** Comprobación del funcionamiento del sistema con nuevos elementos desconocidos para éste.

En los próximos apartados se desarrollan los puntos anteriores con más detalle.

3 Diagramas de bloques

A continuación se detalla el funcionamiento interno del sistema de reconocimiento de edificios mediante los bloques de procesos y sus relaciones

3.1 Creación del Sistema de Clasificación

En la Figura 3 se muestra el diagrama de bloques del proceso de creación del sistema de clasificación de imágenes:

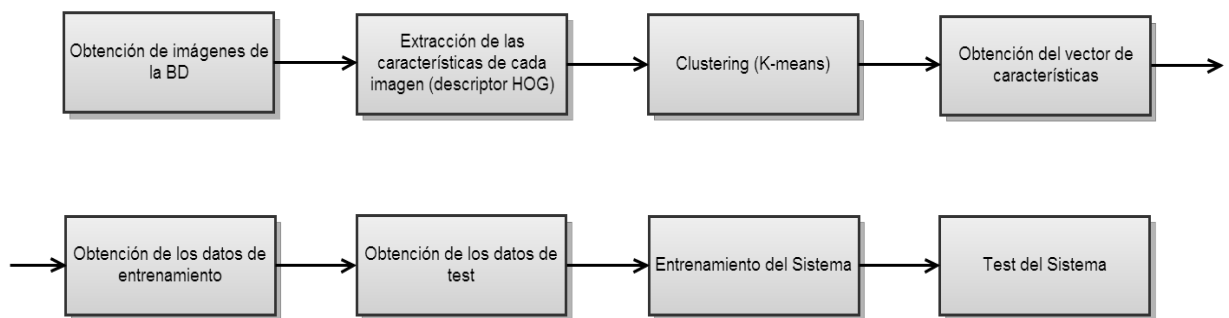


Figura 3. Diagrama de bloques

Como se ha especificado anteriormente, se han desarrollado dos sistemas para la obtención del vector de características de las imágenes Figura 4. El primero está basado en la creación de histogramas y el segundo en realizar convoluciones con filtros en las imágenes. En las secciones siguientes se detallan cada uno de los bloques del sistema y ambos métodos de obtención del vector de características.

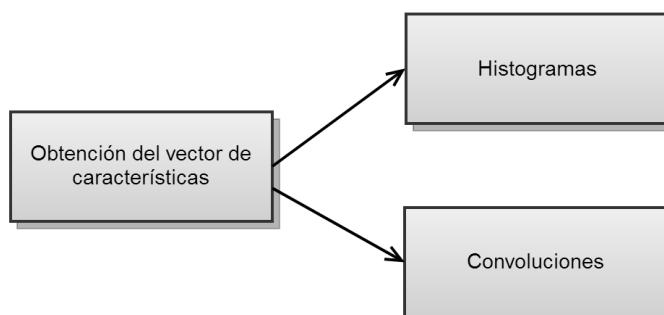


Figura 4. Diagrama de bloques del proceso 2

4 Extracción de características

Las imágenes tienen una gran cantidad de información, quizás no perceptible a simple vista. Necesitamos esta información “extra” para resolver el problema que nos acontece, el de obtener un sistema inteligente que reconozca imágenes.

¿De qué manera tenemos que tratar las imágenes para conseguir esa información? ¿Cómo el sistema puede reconocer una imagen y saber qué tipo de imagen es?

En el mundo de los textos o documentos existen métodos para determinar la clase o el tipo de documento que estamos tratando. Un documento está formado por palabras. Estas palabras definen y conforman el documento. Si lo analizamos y contamos cada una de las palabras que aparecen, nos podemos hacer una idea de cuáles son las palabras que más se repiten. De esta manera podríamos por ejemplo, determinar la temática de un determinado texto, según qué palabras y en qué cantidad aparecen en su interior. En la Figura 5 se ilustra un ejemplo de este tipo de análisis en los documentos.

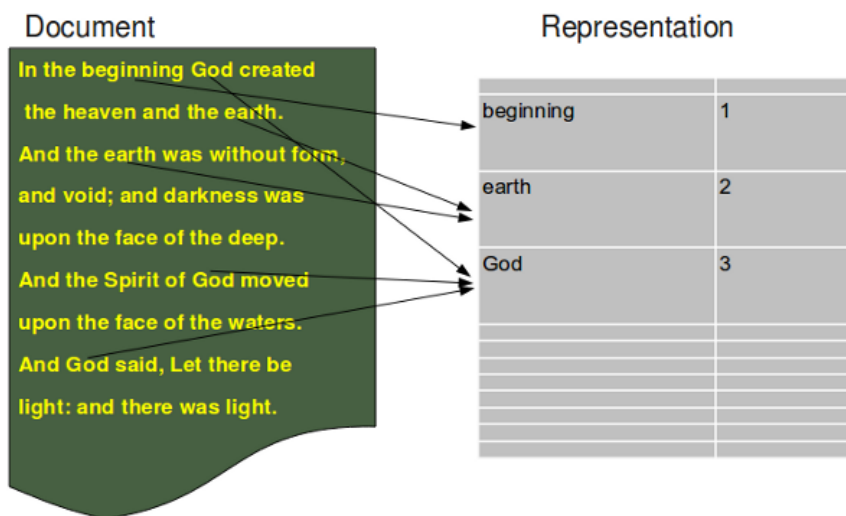


Figura 5. Extracción y contabilización de palabras en un texto. (Fei-Fei)

Podemos entonces tomar la idea anterior y transportarla al mundo de las imágenes. Para empezar consideraremos cada una de las imágenes como una bolsa que contiene “palabras” características de esa imagen. Las palabras son pequeños fragmentos o porciones de una imagen que la definen. Esta idea tiene el nombre de “Bag of Words” o bolsa de palabras. Un ejemplo de palabras visuales de una imagen puede ser el que observamos en la Figura 6: la punta de una de las torres, un trozo del rosetón, un fragmento de cielo o de nube, una rama de un árbol...

De la misma manera que tenemos un diccionario que contiene todas las palabras, y a su vez, los documentos están formados por palabras contenidas en el diccionario podemos aplicar esta idea al mundo de las imágenes.



Figura 6. Esquema de la extracción de palabras visuales de una imagen

Así podremos extraer para cada imagen las palabras visuales que la representan y formar nuestro propio diccionario de palabras visuales.

En primer lugar, nos interesa obtener las características de cada una de las imágenes. En apartados posteriores se explican los métodos que se han utilizado para extraer estas características y cómo se mide la manera en que aparecen las palabras visuales en las distintas imágenes. En la Figura 7 se muestra un ejemplo de un posible diccionario de palabras visuales y la aparición de éstas en imágenes de diferentes edificios. Se ilustra el concepto de “Bag of Words”, puesto que en la bolsa contenemos las palabras visuales de nuestro diccionario y para cada imagen contabilizamos de qué manera está presente cada una de ellas. De esta manera, creamos el vector de características de las imágenes.

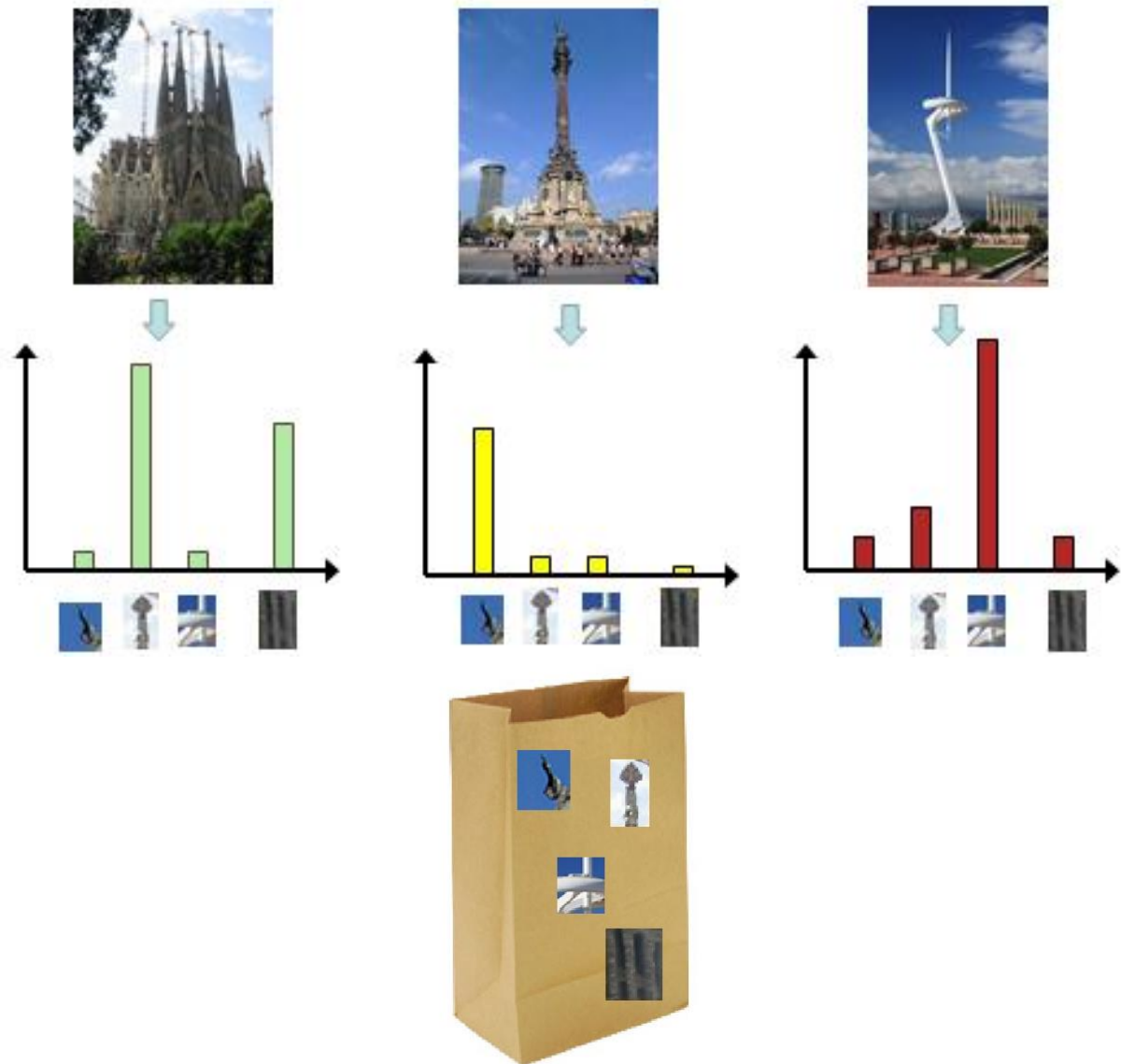


Figura 7. Ilustración de la creación de las características de 3 imágenes a partir de un diccionario de palabras visuales

4.1 Descriptor básico: Histogram of Oriented Gradients (HOG)

Un descriptor de imagen describe características visuales de los contenidos dispuestos en una imagen. Pueden describir características elementales como la forma, el color, la textura, movimiento, entre otros. Esto nos permitirá tener información acerca de las imágenes para luego poder clasificarlas. En los siguientes apartados se detalla que tipo de descriptor se ha usado y de qué manera.

4.1.1 HOG

Como descriptor de características de las imágenes, usaremos Histogram of Oriented Gradients (HOG). Éste método es usado en procesamiento de imágenes y en Visión Artificial, para la detección de objetos. La técnica se basa en contabilizar el número de ocurrencias de los gradientes orientados en diferentes porciones de la imagen.

Navneet Dalal y Bill Triggs, investigadores del French National Institute for Research in Computer Science and Control (INRIA), describieron el Histograma de Gradientes Orientados en Junio del 2005, en un artículo para el CVPR (Conference on Computer Vision and Pattern Recognition). En su trabajo centraban su algoritmo en el problema de la detección de peatones en imágenes estáticas, aunque más adelante expandieron sus pruebas en la detección de humanos en vídeos, así como animales y vehículos en imágenes estáticas.

La idea esencial del HOG es que la apariencia y forma de un objeto en una imagen puede ser descrita como la distribución de la intensidad de los gradientes o la dirección de los contornos.

En el campo del tratamiento de imágenes digitales, el gradiente del píxel de una imagen es un vector que indica la dirección en la cual se produce un mayor cambio en la intensidad o color de la imagen (en imágenes en escala de grises el gradiente se dirige a píxeles de menor valor, es decir, de píxeles blancos a píxeles más negros); y su módulo indica la magnitud de este cambio.

La implementación de los descriptores de HOG se obtiene dividiendo la imagen en pequeñas regiones llamadas celdas. Para cada celda se compila el histograma de las direcciones de los gradientes u orientaciones de los contornos para cada uno de los píxeles que contiene la celda. La combinación de estos histogramas es la representación del descriptor.

A continuación en la Figura 8 y la Figura 9 se muestran ejemplos de la representación de las características de HOG en algunas imágenes a modo de ejemplo.

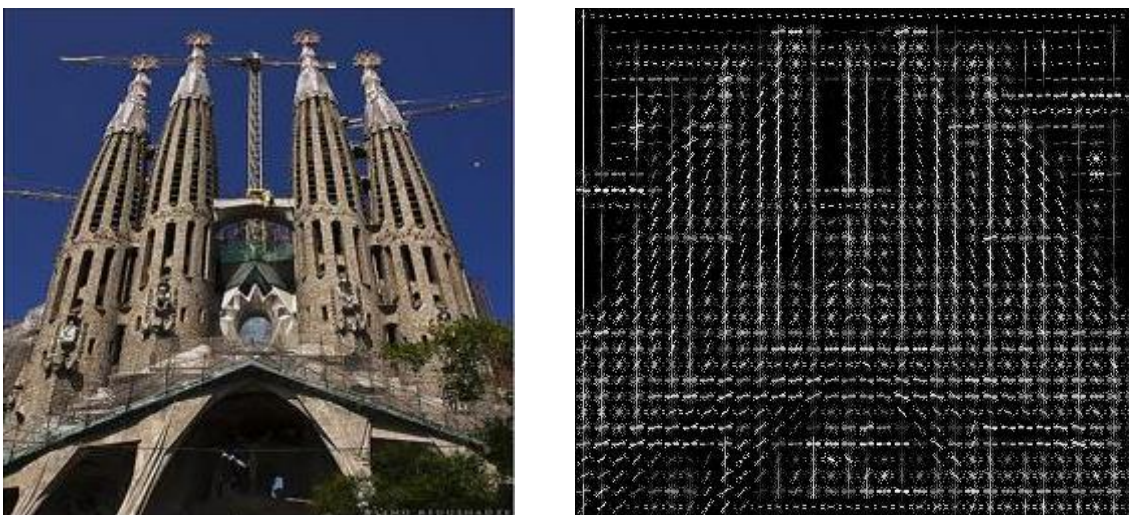


Figura 8. Ejemplo de una Imagen y la representación de sus características de HOG I (Andrea Vedaldi)

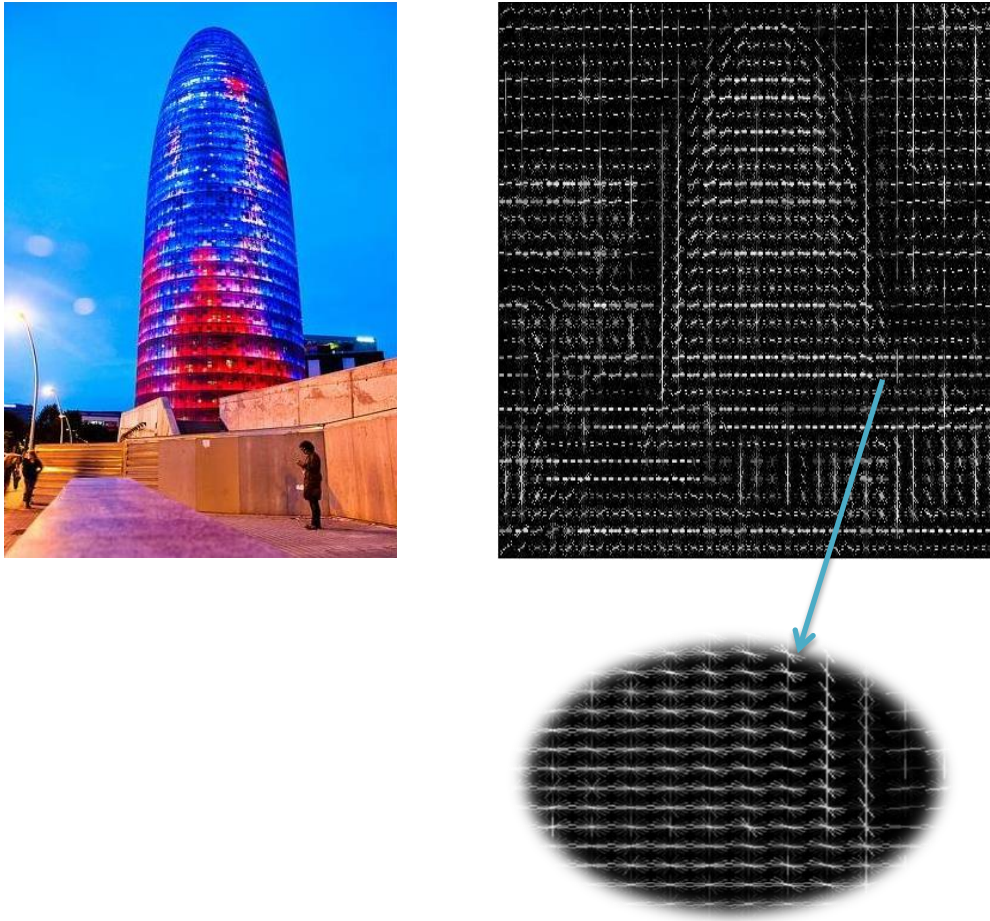


Figura 9. Ejemplo de una Imagen y la representación de sus características de HOG II (Andrea Vedaldi)

4.1.2 Descriptor de imágenes basado en HOG

En este apartado explicaremos como se obtiene el descriptor de imágenes basado en HOG.

Deberemos considerar como parámetro una Imagen I en color (3 canales), cuyos valores sean doubles y como segundo parámetro la medida del bin (celda). Las imágenes son todas re-escaladas a una medida de 128×128 píxeles. Por tanto tendremos n imágenes de $128 \times 128 \times 3$. Como medida de bin tomamos 8 píxeles. Se dividirá la imagen en celdas de 8×8 píxeles y extraerá las características HOG para cada uno de los bloques. Por lo tanto con estos datos, obtendremos una matriz con las características HOG de la imagen con unas dimensiones de $14 \times 14 \times 32$ correspondientes a 16×16 fragmentos de 8×8 píxeles de la imagen descartando los bordes. Por tanto, las características de cada uno de los bloques serán vectores de 32 elementos. En la Figura 10 se ilustra la configuración elegida.

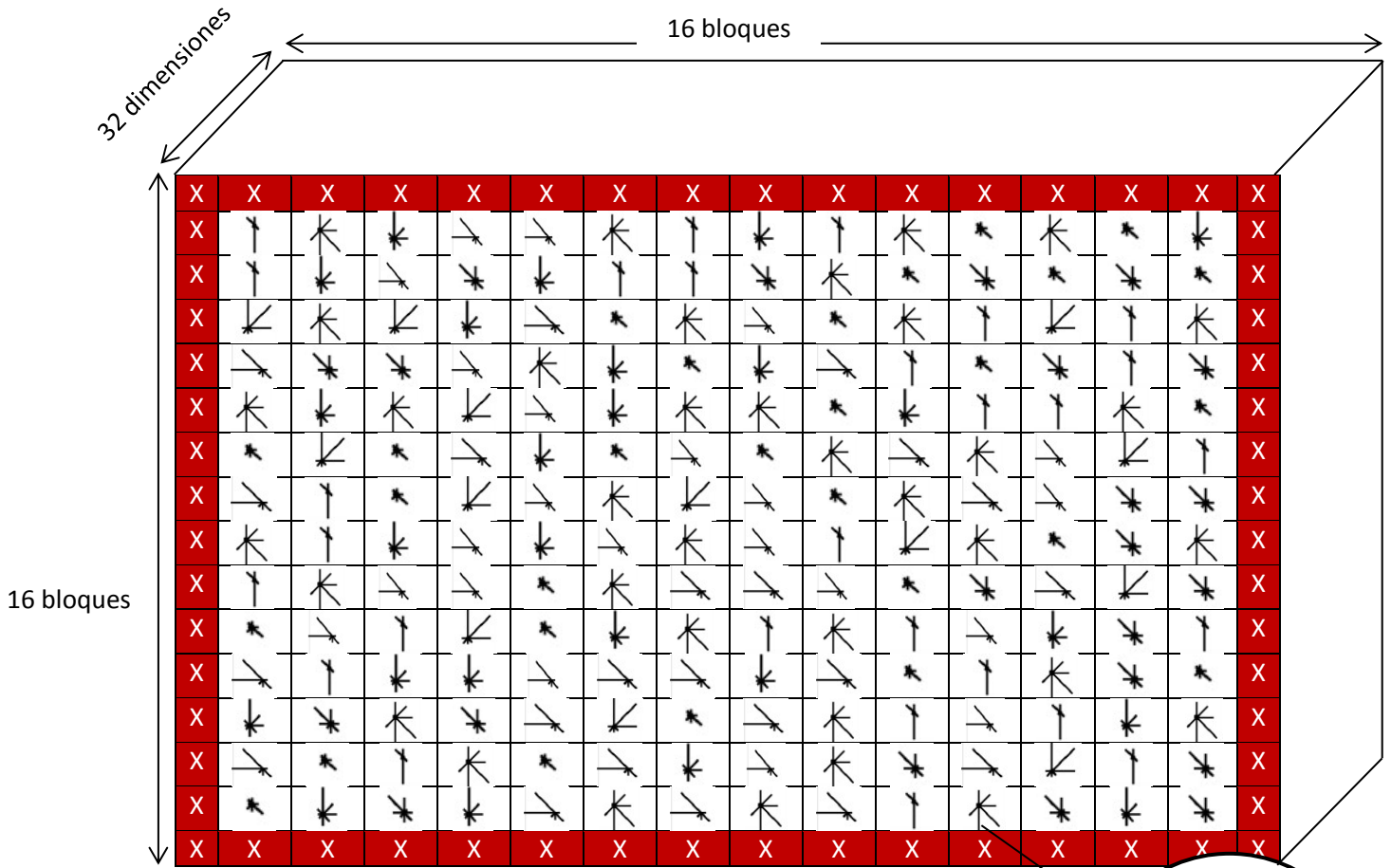
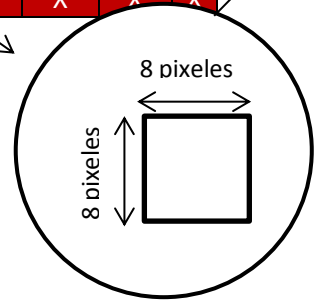


Figura 10. Representación de la matriz con las características de HOG



4.2 Definición de un diccionario de palabras visuales

Tal como se ha explicado antes, en visión artificial se utiliza la idea de “Bag of Words” para representación de imágenes (una imagen se refiere a un objeto particular, como una imagen de un edificio). Por ejemplo, una imagen puede ser tratada como un documento, y las características extraídas en ciertas regiones o puntos de la imagen son consideradas palabras visuales.

Tenemos por tanto, las características de las imágenes usando HOG como descriptor. Cada una de ellas puede ser considerada como una palabra visual en el diccionario. Debido a que podemos obtener un gran nombre de características, el siguiente paso, consiste en crear un diccionario de palabras agrupando los diferentes elementos que hemos obtenido según su similitud, para evitar tener palabras repetidas. Para conseguir esto, recurrimos al **clustering**, que es una técnica que permite la generación automática de grupos en los datos. Podríamos

decir que clustering es el proceso de organización de objetos en grupos cuyos miembros son similares de alguna manera. La manera de encontrar los grupos en una colección de objetos puede tener diversos criterios que por lo general son la distancia o similitud. La cercanía entre objetos se puede definir con una función de distancia (por ejemplo, la distancia euclídea).

Un Clúster por lo tanto es una colección de objetos que son “similares” entre ellos y son “diferentes” a los objetos que pertenecen a otros clústeres.

4.2.1 Clustering K-Means

Para llevar a cabo el clustering, usaremos el método K-Means. Esta función dividirá los descriptores hallados previamente y calcula tantos centroides como se hayan especificado como parámetro de la función.

El proceso de agrupamiento mediante éste método se lleva a cabo de la siguiente manera:

1. Inicialmente se fija el número de grupos K (número de clusters) y se asume el centroide o centro de esos grupos. Los centroides pueden ser determinados de forma aleatoria (K objetos como centroides iniciales) o basándose en algún método heurístico
2. A continuación se asigna a cada elemento el clúster que minimiza la varianza entre éste elemento y el centro del clúster.
3. Se recalculan los centros de los clústeres haciendo la media de todos elementos del clúster.
4. Se repiten los pasos 2 y 3 hasta que se consigue la convergencia (es decir los centroides no se desplazan)

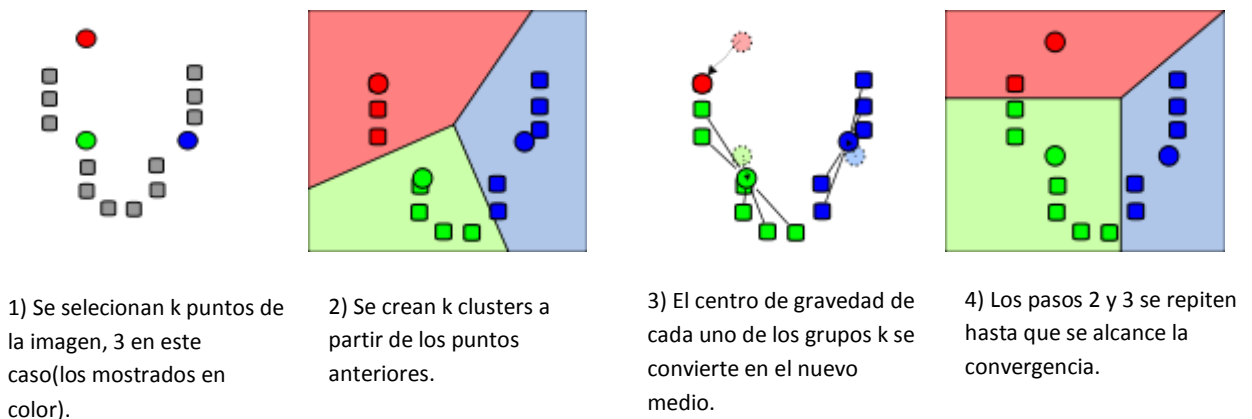


Figura 11. Representación del algoritmo K-Means

En la Figura 11 se ilustra el proceso del agrupamiento. Tras esto obtenemos tantas palabras visuales como se hayan especificado (centroides), las cuales formaran el vocabulario asociado a nuestro problema.

4.2.2 Obtener las palabras con K-Means

Como se ha especificado anteriormente hemos obtenido, por cada una de las imágenes una matriz de 3 dimensiones $14 \times 14 \times 32$ como resultado de la extracción de las características de éstas. Para poder realizar el clustering sobre los datos necesitamos transformar esta matriz tridimensional a una matriz $n \times m$. Se usa el algoritmo de kmeans con los siguientes parámetros:

- hogMatrix: Matriz con los datos resultantes de extraer las características de HOG de las imágenes.
- Clústeres: número de clústeres o divisiones que queremos que el algoritmo realice. Va a indicar el número de palabras visuales que vamos a obtener.
- MaxIter: Máximo número de iteraciones que tiene como valor 100.

En la Tabla 1 se muestra un ejemplo por cada clase de fragmentos de imágenes (palabras visuales) que se han situado en un mismo clúster. Se puede observar cómo las palabras corresponden a diferentes imágenes pero representan la misma zona del edificio.










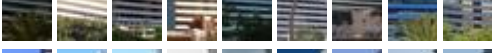
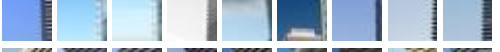

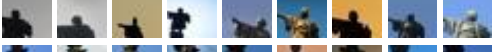



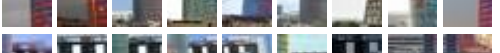

Edificio	Clúster	Palabras visuales en 3 clústeres distintos
Sagrada Familia	1	
	2	
	3	
Montjuïc	1	
	2	
	3	
Hotel Vela	1	
	2	
	3	
Torres Mapfre	1	
	2	
	3	
Colón	1	
	2	
	3	
Torre Agbar	1	
	2	
	3	

Tabla 1. Palabras visuales en 3 clústeres distintos de cada clase

4.3 Representación de imágenes en el diccionario visual

Hemos desarrollado dos variantes para la obtención de los vectores de características y la representación de imágenes en el diccionario visual. A continuación se detallan los dos métodos utilizados.

4.3.1 Creación de Histogramas

El primer método se realiza a partir de Histogramas. El histograma de una imagen digital con L niveles posibles de intensidad en el rango $[0, G]$ se define como una función discreta

$h(r_k) = n_k$ donde r_k es el k -ésimo nivel de intensidad en el intervalo $[0, G]$ y n_k es el número de píxeles de la imagen cuyo nivel de intensidad es r_k . Es decir, contabilizamos cuántos píxeles de una imagen tienen un determinado nivel de intensidad. En un histograma se pueden contabilizar muchos otros parámetros.

En nuestro caso necesitamos determinar cuáles de las palabras visuales obtenidas con el clustering, corresponden a cada descriptor de una imagen. Para ello calcularemos la distancia Euclídea entre cada palabra visual del diccionario y cada uno de los descriptores de una imagen. El que de distancia menor será el asignado a ese descriptor.

Una vez asignada una palabra visual a cada uno de los descriptores, se calcula el histograma, que en este caso contabilizará la aparición de las palabras visuales en una imagen.

4.3.2 Convoluciones

El segundo método se lleva a cabo a partir de convolucionar filtros en las imágenes. Una convolución es el resultado de aplicar una matriz (o filtro) sobre una imagen. En procesamiento de imágenes la aplicación de filtros sobre imágenes mediante convolución tiene muchas utilidades como por ejemplo, modificar el color, aplicar efectos o resaltar los contornos de una imagen. El resultado de una convolución es una nueva imagen que ha sido filtrada.

La convolución se aplica multiplicando un píxel y el de sus píxeles vecinos por el filtro. Este filtro o kernel de convolución se mueve por cada uno de los píxeles de la imagen original y cada píxel que queda bajo la matriz se multiplica por uno de los valores del filtro, el resultado se suma y se divide después por un valor específico. Este valor es el que toma el píxel de la nueva imagen filtrada.

En este caso buscamos otro método distinto al de los histogramas para poder asignar las palabras visuales a cada uno de los descriptores de la imagen. Usaremos cada una de las palabras visuales obtenidas en el clustering como filtros. Aplicaremos una convolución a cada una de las imágenes con cada uno de los filtros. De esta manera podremos observar cuál de los filtros obtiene una respuesta más alta, es decir, obtendremos aquel valor más alto, que resulte de convolucionar un determinado filtro en cada una de las imágenes.

5 Clasificación: Entrenamiento / Test

Una vez representadas todas las imágenes en el diccionario visual, estamos en disposición de realizar el entrenamiento y posteriormente el test del sistema. Para ello se ha usado Máquinas de Soporte de Vectores, que son explicadas a continuación.

5.1 Máquinas de Soporte de Vectores (SVM)

Las máquinas de soporte vectorial o máquinas de vectores de soporte (Support Vector Machines, SVMs) son un conjunto de algoritmos de aprendizaje supervisado desarrollados por Vladimir Vapnik y su equipo en los laboratorios AT&T.

Las SVM fueron presentadas en 1992 y adquirieron fama cuando dieron resultados muy superiores a las redes neuronales en el reconocimiento de letra manuscrita, usando como entrada píxeles. Pretenden predecir a partir de lo ya conocido. Es decir, dado un conjunto de ejemplos de entrenamiento (de muestras) podemos etiquetar las clases y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. Intuitivamente, una SVM es un modelo que representa a los puntos de muestra en el espacio, separando las clases por un margen lo más amplio posible. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de su proximidad pueden ser clasificadas como pertenecientes a una u otra clase.

Más formalmente, una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá una clasificación correcta.

5.1.1 Idea básica y concepto matemático

Dado un conjunto de puntos, subconjunto de un conjunto mayor (espacio de dimensión d), en el que cada uno de ellos pertenece a una de dos posibles categorías, un algoritmo basado en SVM construye un modelo capaz de predecir si un punto nuevo (cuya categoría desconocemos) pertenece a una categoría o a la otra.

Para construir el clasificador tenemos como entrada un par de datos:

- Un vector $x_i \in R^n, i = 1, \dots, l$
- Una etiqueta $y_i \in \{+1, -1\}$

Un hiperplano separa las muestras positivas (+1) de las negativas (-1). Los puntos x_i que están en el hiperplano satisfacen $w \cdot x + b = 0$.

La Figura 12 ilustra este hiperplano de separación entre clases.

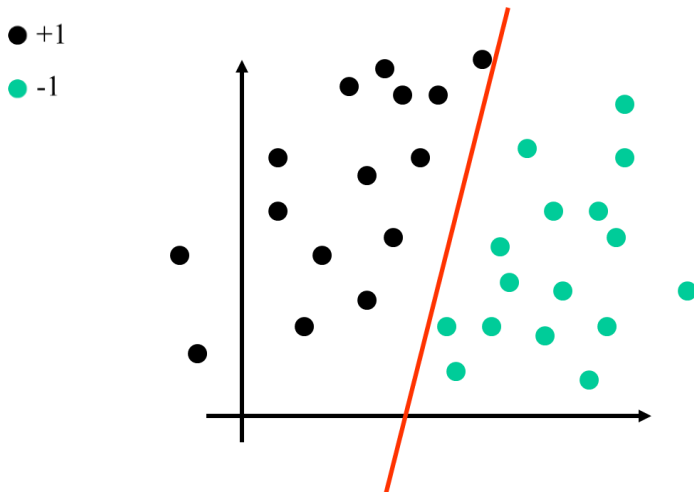


Figura 12. Hiperplano que separa puntos de dos clases distintas

La SVM busca un hiperplano que separe de forma óptima a los puntos de una clase de la de otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior.

En ese concepto de “separación óptima” es donde reside la característica fundamental de las SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia (margen) con los puntos que estén más cerca de él mismo. Por eso también a veces se les conoce a las SVM como *clasificadores de margen máximo*. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán al otro lado.

Al vector formado por los puntos más cercanos al hiperplano se le llama vector de soporte. En la Figura 13 se observa un esquema de dos hiperplanos y los márgenes entre los puntos más cercanos.

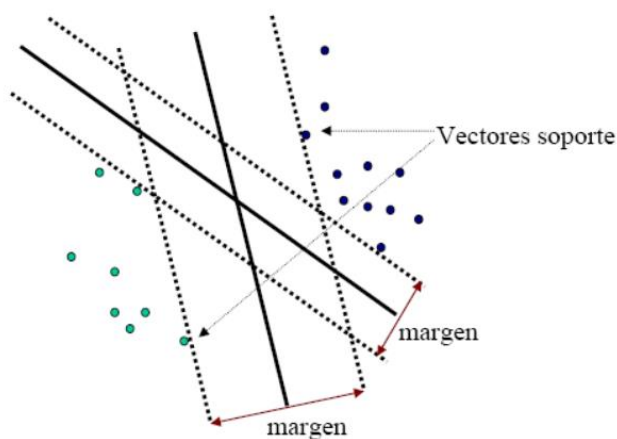


Figura 13. Vectores de Soporte e hiperplanos

5.1.2 MultiClass Classification: One versus all

Para realizar el clasificador de edificios, vamos a usar la técnica “One versus all”. Es decir, construir un SVM por cada clase (edificio), el cual será entrenado para distinguir las palabras visuales de su clase concreta del resto de clases.

Por lo tanto, como tenemos datos de 6 edificios crearemos un SVM por cada uno de ellos.

Construiremos cada uno de los SVM de la siguiente manera:

Para el i -ésimo clasificador asignaremos con una etiqueta de valor positivo a los puntos de la clase i , y asignaremos un valor negativo a los puntos que no pertenecen a la clase i . En la Figura 14 se presenta un esquema sobre la técnica de binarización para un problema de 3 clases, donde las clases representadas en color se les asignaría un valor positivo y las representadas en gris un valor negativo. Obtendríamos así 3 clasificadores (1 por cada clase) que a continuación se usarían para la resolución del problema general.

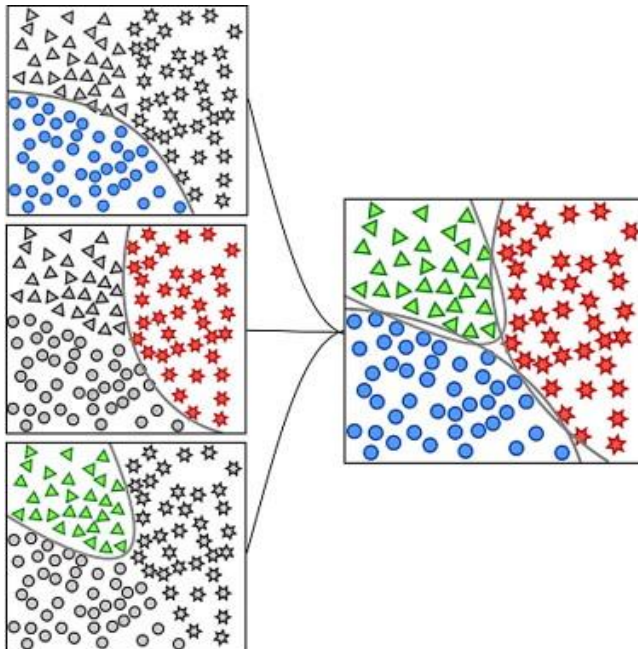


Figura 14. Ejemplo de One Versus All para un problema de 3 clases (Fernández, López, Galar, Jesús, & Herrera, 2013)

6 Casos de uso

Los casos de uso describen los pasos o las actividades que deben realizarse para llevar a cabo el proceso. Documentan el comportamiento del sistema desde el punto de vista del usuario o actor. Representan las funciones que el sistema puede ejecutar.

Un actor en un diagrama de casos de uso representa un rol que alguien puede estar jugando, no un individuo particular, por lo tanto puede haber personas particulares que puedan estar usando el sistema de formas diferentes en diferentes ocasiones. En el caso de este proyecto el actor principal es el desarrollador que tiene la necesidad de ejecutar las distintas funciones del sistema.

Se detallan a continuación en la Figura 15 cuáles son las funciones que puede llevar a cabo el usuario mediante el sistema desarrollado:

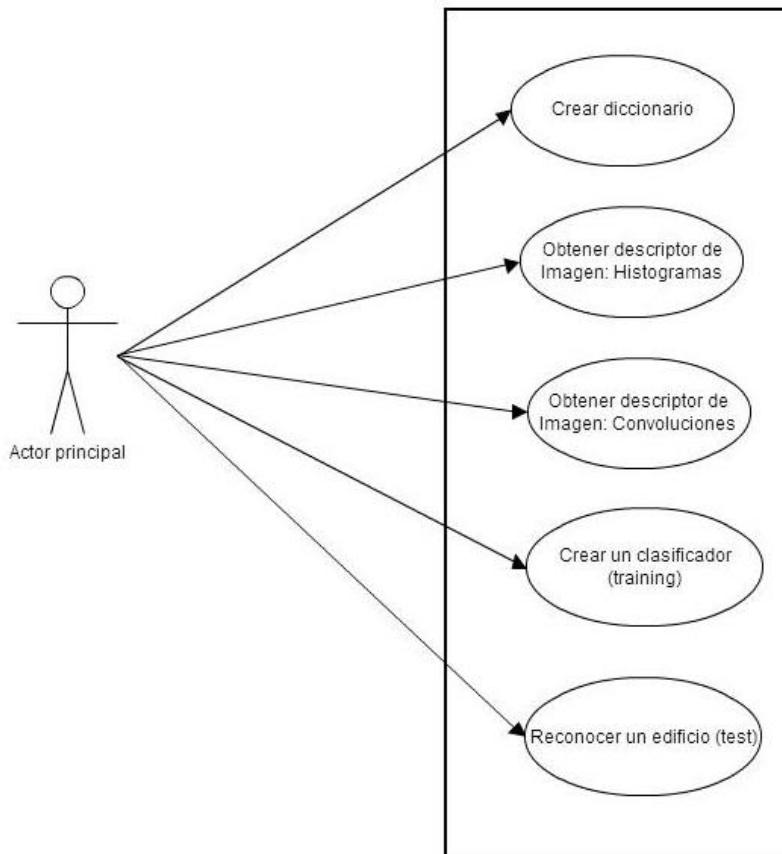


Figura 15. Diagrama de Casos de Uso

Caso de uso 1 (UC1)

Crear un diccionario

Actor principal: Desarrollador

Personal involucrado e intereses: El usuario desea extraer las características de un conjunto de imágenes, usando como descriptor de imagen HOG y aplicar clustering a los datos extraídos para conseguir tener las palabras visuales.

Pre condiciones: Se dispone de una base de datos con múltiples imágenes de un edificio en concreto. El usuario introduce los parámetros correctamente: ruta del directorio donde se almacenan las imágenes, el edificio sobre el que quiere trabajar, el número de clusters y las divisiones que desea realizar al obtener las características.

Escenario principal:

1. Se leen cada una de las imágenes en el directorio indicado por el usuario, y se reescalan para que todas tengan el mismo tamaño.
2. Se extraen sus características, usando HOG como descriptor de imagen.
3. Se realiza clustering, con los datos anteriormente obtenidos, mediante k-means con el número de clusters indicado por el usuario.

Post condiciones: Si el algoritmo de k means converge correctamente, se almacena la información obtenida como resultado de ejecutar el algoritmo.

Caso de uso 2 (UC2)

Obtener descriptor de Imagen: Histogramas

Actor principal: Desarrollador

Personal involucrado e intereses: El usuario desea crear los histogramas del conjunto de imágenes de un edificio dado para determinar cuáles de las palabras visuales obtenidas con anterioridad corresponden a cada uno de los descriptores de una imagen.

Pre condiciones: Se ha ejecutado con anterioridad el algoritmo de clustering y se han obtenido las palabras visuales correspondientes al edificio solicitado. El usuario introduce los parámetros correctamente: el edificio para el cual se quieren crear los histogramas de cada una de sus imágenes, el número de clusters con el que se ha ejecutado el clustering y las divisiones que se han realizado al obtener las características.

Escenario principal:

1. Se recupera la información obtenida anteriormente en el clustering. Los centroides o palabras visuales y la matriz con las características de las imágenes del edificio solicitado por el usuario.

2. Se contabiliza el número de apariciones de cada una de las palabras visuales en cada imagen para crear así los histogramas.
3. Los histogramas son guardados.

Post Condiciones: -

Caso de uso 3 (UC3)

Obtener descriptor de Imagen: Convoluciones

Actor principal: Desarrollador

Personal involucrado e intereses: El usuario desea convolucionar las palabras visuales obtenidas con anterioridad, sobre cada una de las imágenes de un edificio para obtener la información de qué palabras dan una similitud más alta y en qué zona de las imágenes.

Pre condiciones: Se ha ejecutado con anterioridad el algoritmo de clustering y se han obtenido las palabras visuales correspondientes al edificio solicitado. El usuario introduce los parámetros correctamente: la ruta donde se almacenan las imágenes, el edificio para el cual se quieren crear los histogramas de cada una de sus imágenes, el número de clusters con el que se ha ejecutado el clustering y las divisiones que se han realizado al obtener las características.

Escenario principal:

1. Se recuperan las palabras visuales o centroides obtenidos con anterioridad.
2. Se transforman los datos para pasarlos a forma de matrices para poder ser usados como filtros en las convoluciones.
3. Se cargan las imágenes y para cada una, se convoluciona cada uno de los filtros.
4. Se almacenan aquellos valores que dan una respuesta más alta (más similitud).
5. Se almacenan los datos.

Post Condiciones: -

Caso de uso 4 (UC4)

Crear un clasificador

Actor principal: Desarrollador

Personal involucrado e intereses: El usuario desea realizar el entrenamiento. Para realizar el entrenamiento se desea usar Support Vector Machines (SVM).

Pre condiciones: Se han obtenido con anterioridad los vectores de características, bien por el método de los histogramas o bien, por el método de las convoluciones. El usuario introduce los parámetros correctamente: Número de clusters con el que se realizó el clustering y divisiones que se realizaron a la hora de obtener las características.

Escenario principal:

1. Se obtienen los vectores de características de cada una de las clases (edificios).
2. Se obtienen aleatoriamente los datos de training 90% .
3. Se crea un clasificador por cada una de las clases y se entrena con los datos de entrenamiento.
4. Se repite el proceso desde 2 un total de 10 iteraciones para generar conjuntos distintos de training.
5. Se repite el proceso desde 2 por cada una de las clases.

Post Condiciones: Si el método de (SVM) entrena correctamente y converge, se obtienen los clasificadores deseados.

Caso de uso 5 (UC5)

Reconocer un edificio

Actor principal: Desarrollador

Personal involucrado e intereses: El usuario desea realizar el test del sistema.

Pre condiciones: Se han creado con anterioridad clasificadores pertenecientes a cada una de las clases. Los datos de test son nuevos para el clasificador, no han sido usados para entrenarlo.

Escenario principal:

1. Se obtienen los vectores de características de cada una de las clases (edificios).
2. Se obtienen aleatoriamente los datos de test 10% .
3. Se testea el clasificador de la clase, creado con anterioridad, con los datos de test.
4. Se repite el proceso desde el paso 2 un total de 10 iteraciones para generar conjuntos distintos de test.
5. Se repite el proceso desde el paso 2 por cada una de las clases.

Post Condiciones: Se obtienen los resultados de aciertos y errores como resultado de realizar las pruebas de test.

7 Diagrama de Gantt

En esta sección se detalla la organización del proyecto desarrollado. El diagrama de Gantt es la representación gráfica del tiempo que dedicamos para cada una de las tareas o actividades que forman parte de un proyecto concreto, a lo largo de un tiempo total determinado.

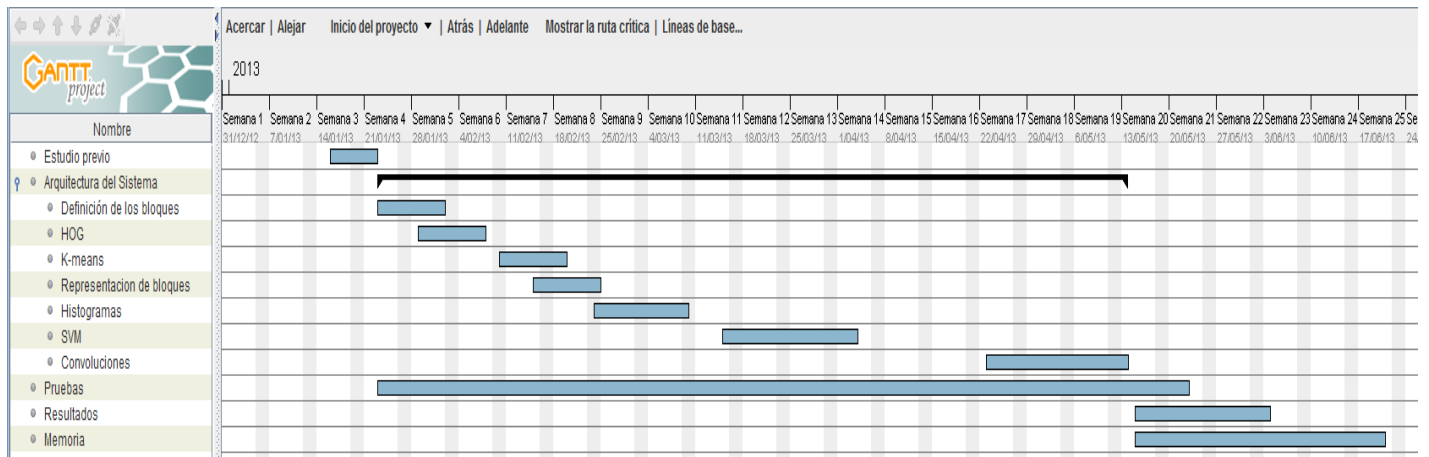


Figura 16. Diagrama de Gantt (Barashev & Thomas)

A continuación se detallan las distintas etapas de desarrollo de este proyecto, contempladas en el diagrama de Gantt, con una breve explicación:

1. Estudio previo: Como se ha especificado en apartados anteriores, PhotoMeet Bcn forma parte de una línea de investigación activa en la UB y con varias alternativas o direcciones a escoger para seguir con el proyecto. En los días iniciales se hace un estudio previo sobre los trabajos anteriores y la dirección en qué se va a trabajar en este proyecto.
2. Arquitectura del sistema: En las semanas posteriores se empieza a diseñar y a desarrollar el sistema. Es la tarea principal del proyecto en la que se desarrollan los distintos métodos que tendrá el sistema de clasificación de edificios:
 1. Definición de los bloques.
 2. Extracción de las características mediante HOG.
 3. Clustering mediante K-means.
 4. Representación de los bloques.
 5. Creación de Histogramas.
 6. Clasificador (SVM)
 7. Convoluciones.
3. Pruebas del sistema: Esta fase está presente durante la mayor parte del desarrollo del proyecto y es paralela a la tarea de Arquitectura del sistema. Durante todo el tiempo de diseño y desarrollo del sistema se realizan pruebas parciales para testear y mejorar los distintos aspectos del sistema.
4. Resultados: Tarea de selección de los parámetros óptimos para analizar los resultados obtenidos por el sistema final.
5. Memoria: Tarea de recopilación y escritura de documentación del trabajo realizado durante los meses anteriores.

8 Estudio económico

En este apartado se hace un cálculo aproximado del presupuesto ficticio de desarrollar este proyecto.

Se desglosa en forma de tabla mostrada a continuación:

Software	Precio estimado
Licencia Matlab	2000 €

Hardware	Precio estimado
Ordenador con los requisitos recomendados	350-800 €

RRHH	Horas estimadas	Precio/H en €	Precio estimado
Programador	200-220	35	7000-7700€

Precio total estimado	9350-10500 €
------------------------------	---------------------

Tabla 2. Estudio Económico de los Costes del proyecto

9 Matlab

Todo el proyecto ha sido desarrollado con Matlab. Matlab (Matrix Laboratory) es un lenguaje de alto nivel y un entorno interactivo para el cálculo numérico, la visualización y la programación. Mediante Matlab, es posible analizar datos, desarrollar algoritmos y crear modelos o aplicaciones. El lenguaje, las herramientas y las funciones matemáticas incorporadas permiten explorar diversos enfoques y llegar a una solución antes que con hojas de cálculo o lenguajes de programación tradicionales, como pueden ser C/C++ o Java.

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware.

Matlab se puede utilizar en una gran variedad de aplicaciones, tales como procesamiento de señales y comunicaciones, procesamiento de imagen y vídeo, sistemas de control, pruebas y medidas, finanzas computacionales y biología computacional. Más de un millón de ingenieros y científicos de la industria y la educación utilizan MATLAB, el lenguaje del cálculo técnico. Es un software muy usado en universidades y centros de investigación y desarrollo.

Para desarrollar este proyecto se ha usado la versión de Matlab R2011 a. A continuación se muestran los requisitos recomendados para utilizar esta versión de Matlab en la Tabla 3.

Sistemas operativos	Procesadores	Espacio en Disco	RAM
32-Bit y 64-Bit MATLAB y Simulink Product Families			
Windows XP Service Pack 3	Cualquier Intel o AMD x86 procesador que soporte instrucciones SSE2	1 GB for MATLAB only, 3–4 GB para instalación típica	1024 MB (Al menos 2048 MB recomendado)
Windows XP x64 Edition Service Pack 2			
Windows Server 2003 R2 Service Pack 2			
Windows Vista Service Pack 2			
Windows Server 2008 Service Pack 2 or R2			
Windows 7			
Windows 8			

Tabla 3. Requisitos Recomendados del Sistema (Mathworks)

Las características del equipo en que se ha desarrollado el proyecto son las especificadas en la Tabla 4.

Sistema Operativo	Windows 8
Procesador	Intel(R) Core™ i7-3630QM CPU @ 2.40Ghz
Disco duro	500 GB
Memoria Ram	8 GB
Tipo de Sistema	64 bits, procesador x64

Tabla 4. Especificaciones del equipo

10 Experimentos

En este apartado se presentan los resultados obtenidos mediante los dos métodos de reconocimiento de edificios desarrollados en el proyecto y el método de validación de dichos resultados.

10.1 Método de validación

Una vez desarrollado el sistema y entrenado con un subconjunto de la base de datos, se pasa a testarlo con imágenes nuevas para el clasificador, es decir, que no han sido utilizadas en la fase de entrenamiento. Deseamos obtener resultados que sean fiables, y para evitar caer en una determinada configuración dada por el componente aleatorio en que se seleccionan los datos de entrenamiento y test se ha utilizado el método de Cross Validation.

10.1.1 Cross Validation

Cross Validation es una técnica utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba. Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones. Es una técnica muy usada en proyectos de inteligencia artificial para validar modelos generados. Como se puede observar en la Figura 17 para cada iteración se seleccionan unos datos de prueba o test diferentes. Por lo tanto, en cada iteración se obtiene una configuración distinta de los datos.

Para la construcción de cada uno de los clasificadores se genera aleatoriamente a partir de la muestra general, los datos de training y los de test. Las distintas configuraciones de datos se realizan cogiendo aleatoriamente el 90% de las muestras para training y el 10% para test.

Se obtiene así un porcentaje de acierto de ese clasificador con esa configuración concreta de training y test. El proceso se repite hasta 10 veces obteniendo así, 10 configuraciones de training y test aleatorias y distintas entre sí. Asimismo se obtendrán resultados de los porcentajes de acierto del clasificador en las 10 iteraciones. El resultado final resulta ser la media de los resultados de las 10 iteraciones.

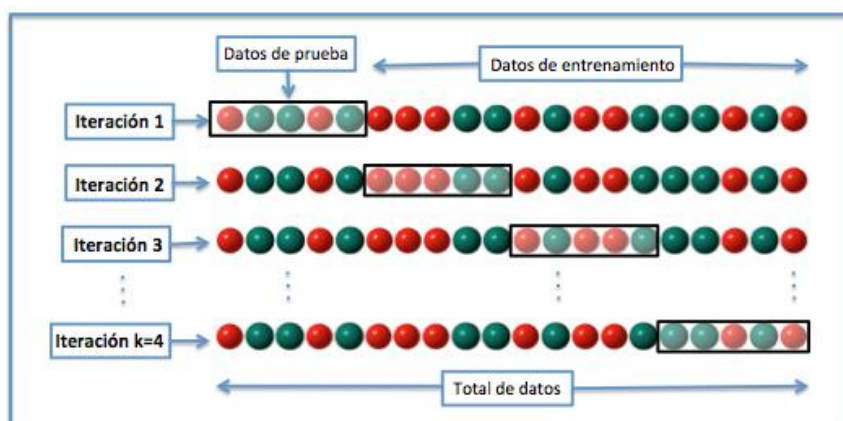


Figura 17. Esquema del método Cross Validation (Wikipedia, Validación Cruzada)

10.2 Medidas de evaluación

A medida que vamos realizando tests, proporcionamos al clasificador una imagen y éste tendrá que decidir (clasificar) a cuál de las clases pertenece. El clasificador podrá acertar, asignando correctamente la clase a la imagen proporcionada o bien asignarle otra clase confundiendo. Para visualizar de una manera clara estos aciertos y errores y ver si el clasificador confunde alguna clase con otra, los resultados son mostrados mediante una matriz de confusión. Es un diseño de tabla específica que permite la visualización de la ejecución de un algoritmo, por lo general un aprendizaje supervisado. Es una matriz de un clasificador de dos o más clases. Contiene información acerca de las clasificaciones actuales y predicciones hechas por el sistema de clasificación. Las matrices son $n \times n$, donde n es el número de clases. Cada columna de la matriz representa los casos que el algoritmo predijo, mientras que cada fila representa los casos en una clase real. Una de las ventajas de las matrices de confusión es que nos permiten observar con facilidad si el clasificador está confundiendo dos clases concretas. En la Tabla 5 se ilustra el esquema que siguen las matrices de confusión.

	Clase 1(p) estimado	Clase2(n) estimado
Clase1(P) real	TP	FP
Clase2(N) real	FN	TN

Tabla 5. Esquema de la Matriz de Confusión

Diagonal de aciertos

- TP: Son los casos que pertenecen a la clase y el clasificador los definió en esa clase.
- FN: Son los casos que sí pertenecen a la clase y el clasificador no los definió en esa clase.
- FP: Son los casos que no pertenecen a la clase pero el clasificador los definió en esa clase.
- TN: Son los casos que no pertenecen a esa clase y el clasificador definió que no pertenecen a esa clase.

10.3 Resultados cuantitativos

En esta sección se muestran los resultados obtenidos. Se divide la sección en dos apartados para mostrar las pruebas y resultados obtenidos entrenando al sistema mediante el método de obtención del vector de características basado en Histogramas y a continuación se muestran los obtenidos con el método basado en convoluciones.

Los resultados cuantitativos mostrados a continuación son visualizados en forma de matrices de confusión. Se muestran las distintas matrices de confusión obtenidas con diferentes valores de k. Los valores de la tabla están representados en porcentajes de estimación sobre el total de datos de test de cada clase. El parámetro k corresponde al número de clústeres utilizado en el k-means, es decir, indica el número de palabras que queremos tener de cada clase en nuestro diccionario. Dado que según este parámetro pueden variar los resultados de acierto y error a la hora de clasificar los edificios y el coste computacional de todo el proceso, se han hecho pruebas con distintos valores del parámetro k.

10.3.1 Método Histogramas

En este apartado observamos los resultados que hemos obtenido del sistema utilizando el primer método implementado. Los vectores de características que se han usado para entrenar al clasificador son formados por los Histogramas de cada una de las imágenes.

Los números en la Tabla 6 corresponden a los porcentajes de acierto de cada uno de los clasificadores. Los números de 1-6 corresponden a: 1 (clasificador Sagrada Família), 2 (clasificador Hotel Vela), 3 (clasificador Montjuïc), 4 (clasificador Torre Agbar), 5 (clasificador Colón), 6 (clasificador Torres Mapfre).

k	Class	Sagrada Família	Hotel Vela	Montjuïc	Torre Agbar	Colón	Torres Mapfre
50	1	91.55	0.22	1.55	1.33	2.22	3.11
	2	1.77	84.88	4.44	2.88	3.33	2.66
	3	1.33	2.66	85.55	1.55	3.11	5.77
	4	1.11	2.88	2.44	87.11	2.66	3.77
	5	2.88	1.77	5.55	1.77	80.88	7.11
	6	2.44	3.77	4.66	2.22	5.55	81.33
100	1	93.77	1.11	1.33	1.11	0.88	1.77
	2	2.66	85.11	3.11	1.55	3.33	4.22
	3	1.11	3.11	86.88	2.66	3.33	2.88
	4	0.88	1.33	1.55	90.66	3.33	2.22

	5	2.66	4.22	5.33	2.88	81.11	3.77
	6	4.22	8.44	3.33	4.88	4	75.11
150	1	94.88	0.88	2	0.44	1.11	0.66
	2	0.66	90.22	3.55	1.33	2.44	1.77
	3	1.11	3.11	90.22	1.33	1.55	2.66
	4	2.44	1.11	1.11	89.55	3.11	2.66
	5	1.11	2.22	2.44	2.66	87.11	4.44
	6	3.11	3.11	4.44	3.77	4.88	80.66
200	1	94.8	0.88	2	0.44	1.11	0.66
	2	0.66	90.22	3.55	1.33	2.44	1.77
	3	1.11	3.11	90.22	1.33	1.55	2.66
	4	2.44	1.11	1.11	89.55	3.11	2.66
	5	1.11	2.22	2.44	2.66	87.11	4.44
	6	3.11	3.11	4.44	3.77	4.88	80.66
400	1	97.33	0.44	0.66	0.44	0.22	0.88
	2	0.66	91.33	1.55	1.55	2.44	2.44
	3	1.33	0.88	90.66	0.88	2.66	3.55
	4	1.33	2.66	3.33	89.11	1.77	1.77
	5	2	3.11	1.77	2	88.22	2.88
	6	2.22	3.55	2.88	1.77	5.33	84.22

Tabla 6. Resultados cuantitativos del método Histogramas

Como se ha explicado anteriormente, debemos fijarnos en la diagonal de cada una de las tablas para observar el porcentaje de los aciertos de cada uno de los clasificadores.

Los otros valores nos indican el porcentaje de error de cada una de las clases respecto a las otras. De esta manera podemos observar si el clasificador confunde una clase con otra.

A continuación se muestra una gráfica de las performances según el número de clústeres (parámetro k). Los resultados de las performances se calculan haciendo la media de la diagonal de aciertos, con el objetivo de observar con qué número de clústeres obtenemos unos resultados de acierto más elevados teniendo en cuenta las 6 clases.

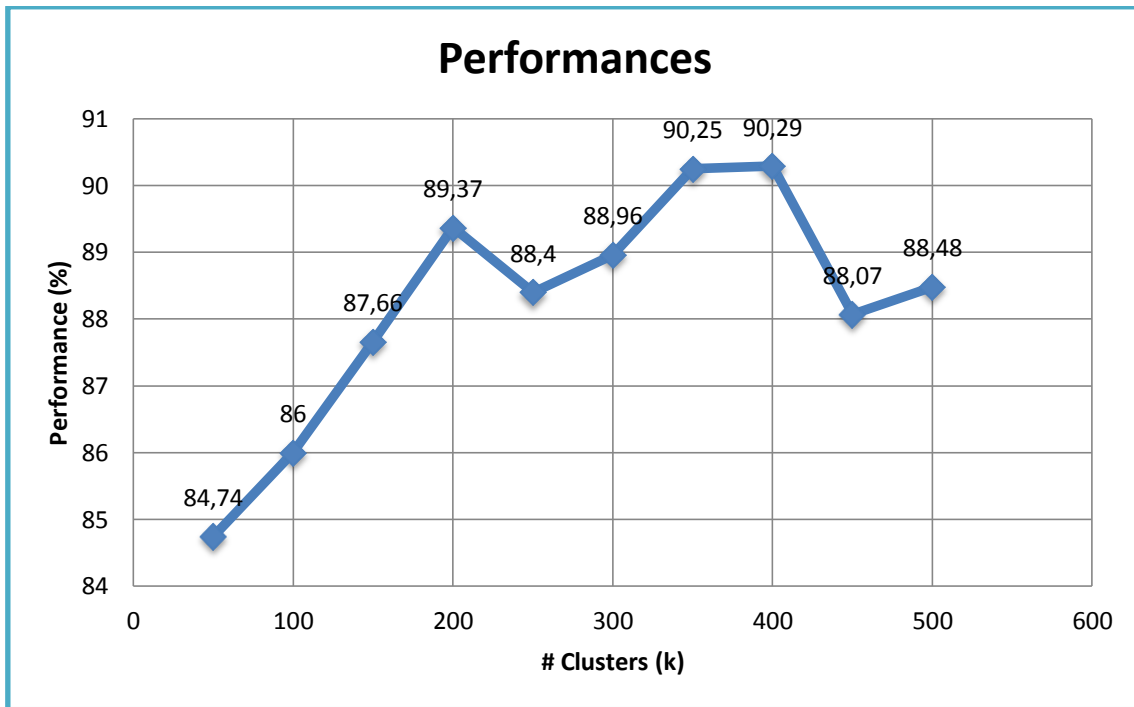


Figura 18. Gráfica del porcentaje de la performance del sistema, según el número de Clústeres (parámetro k)

Se han realizado pruebas variando el número de clústeres, empezando desde 50 hasta 500.

Observamos como la performance tiene una tendencia a incrementarse considerablemente, según se va aumentando el número de clústeres, alcanzando su valor máximo con 400 clústeres. A partir de ahí la performance del sistema empieza a descender. Determinamos pues, que el número de clústeres óptimo es 400, obteniendo una media de acierto del 90,2% entre las 6 clases de edificios.

10.3.2 Método Convolutiones

En este apartado observamos los resultados que hemos obtenido del sistema utilizando el método de obtención de los vectores de características mediante convoluciones como ya se ha explicado anteriormente.

Los números en la Tabla 7 corresponden a los porcentajes de acierto de cada uno de los clasificadores. Los números de 1-6 corresponden a: 1 (clasificador Sagrada Família), 2 (clasificador Hotel Vela), 3 (clasificador Montjuïc), 4 (clasificador Torre Agbar), 5 (clasificador Colón), 6 (clasificador Torres Mapfre).

k	Class	Sagrada Familia	Hotel Vela	Montjuïc	Torre Agbar	Colón	Torres Mapfre
50	1	83.77	4.44	1.11	3.11	4	3.55
	2	5.55	47.33	11.55	11.55	6.88	17.11
	3	0.88	14.22	69.33	3.55	4.66	7.33
	4	2.88	10.44	3.55	72.66	5.11	5.33
	5	2	7.11	6	7.55	71.55	5.77
	6	9.11	15.33	9.7	8.22	8.44	49.11
200	1	85.33	3.55	0.44	3.33	1.33	6.00
	2	2.88	64.44	7.33	8.22	4.88	12.22
	3	0.88	6.66	78.44	3.33	3.55	7.11
	4	2.44	11.11	3.33	72.44	2.44	8.22
	5	0.88	4	6.66	4.66	77.55	6.22
	6	5.77	12	6.22	9.11	5.77	61.11
400	1	88.66	1.77	1.55	2.22	1.55	4.22
	2	4.66	67.55	6	6.44	5.55	9.77
	3	1.55	4.22	82.88	2.22	5.11	4
	4	2.66	8.88	3.55	73.33	4.66	6.88
	5	0.88	2.44	4.66	3.55	85.55	2.88
	6	8.66	12.66	5.77	5.11	6.22	61.55
800	1	92	2.22	0.88	1.33	0.88	2.66
	2	2.44	72.88	3.55	8.44	4.22	8.44
	3	0.88	5.11	86	1.77	3.55	2.66
	4	1.11	7.55	1.55	80.66	2.88	6.22
	5	0.44	4.44	4.22	2.22	85.77	2.88
	6	4.22	10.66	4.66	8.88	4.44	67.11

Tabla 7. Resultados cuantitativos del método convoluciones

A continuación se muestra la gráfica de las performances según el número de clústeres (parámetro k) correspondiente a los resultados de éste método.

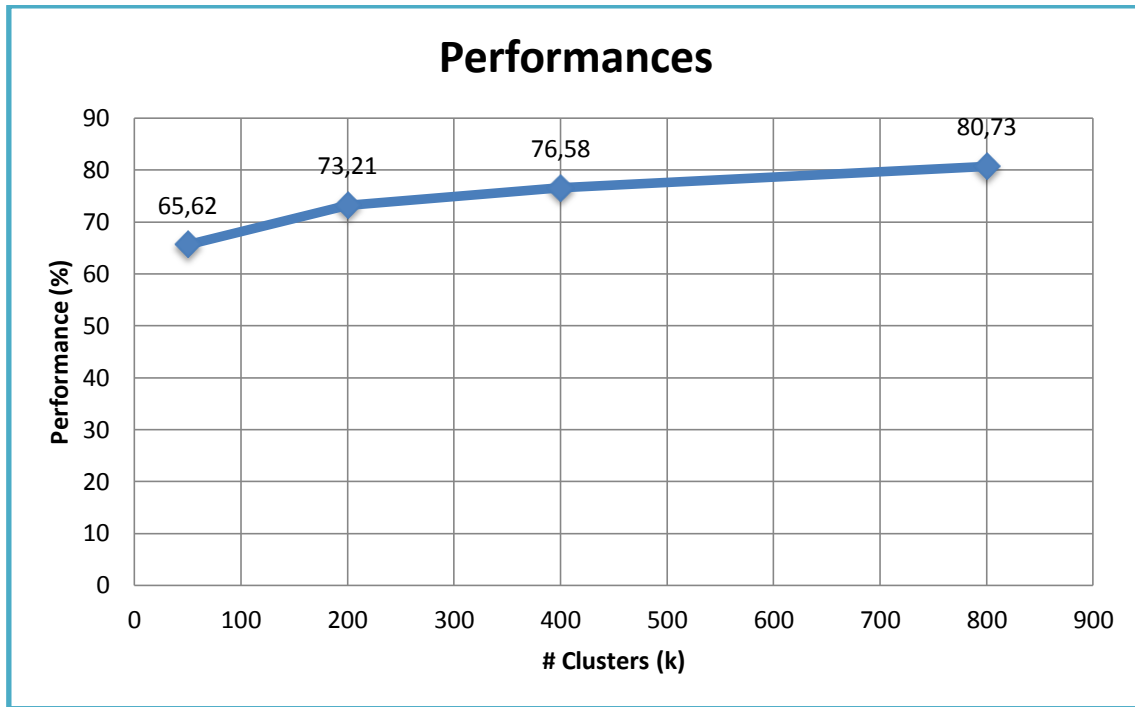


Figura 19. Gráfica del porcentaje de la performance del sistema, según el número de Clústeres (parámetro k) del método de convoluciones
















Se observa que con el método de convoluciones no obtenemos una media de acierto tan alta como en el de histogramas. Además para obtener una mejor performance del sistema se tiene que incrementar de manera considerable el número de clústeres haciendo que el proceso se vuelva mucho más costoso computacionalmente. Se han realizado pruebas con un valor k de como máximo 800 ya que un valor más alto no resultaba factible.

10.4 Resultados cualitativos

Se presentan en este apartado una serie de ejemplos para poder observar de forma cualitativa algunas imágenes que han sido clasificadas correcta e incorrectamente por el clasificador. Los ejemplos mostrados a continuación han sido extraídos a partir de las clasificaciones realizadas por el sistema mediante el método de los Histogramas.

En primer lugar, en la Tabla 8 se muestran 3 ejemplos por cada una de las clases de imágenes clasificadas de manera incorrecta y cuál ha sido la estimación del clasificador. Al observar estas imágenes podemos deducir si presentan alguna dificultad para el sistema. Algunas de estas dificultades pueden ser la perspectiva de la imagen, la aparición de mucho trozo de cielo o










elementos que forman parte del ecosistema como mar, playa, personas u otros elementos que puedan alterar la decisión del clasificador.

Edificio	Ejemplo 1	Ejemplo2	Ejemplo 3
Sagrada Família			
Estimado como	Colón	Torre Agbar	Hotel Vela
Hotel Vela			
Estimado como	Montjuic	Torre Agbar	Colón
Montjuïc			
Estimado como	Sagrada Família	Colón	Colón
Torre Agbar			
Estimado como	Colón	Montjuic	Colón
Colón			
Estimado como	Torre Mapfre	Torre Agbar	Torre Agbar

Torres Mapfre			
Estimado como	Sagrada Família	Torre Agbar	Colón

Tabla 8. Ejemplos de imágenes estimadas incorrectamente por el sistema de clasificación

En segundo lugar, en la Tabla 9 se muestran 3 ejemplos por cada una de las clases de imágenes clasificadas de manera correcta. Se puede observar como suelen ser imágenes donde se tiene una visión general del edificio donde hay una probabilidad alta de que aparezcan las palabras más características de esa clase y sean detectadas con facilidad.

Edificio	Ejemplo 1	Ejemplo 2	Ejemplo 3
Sagrada Família			
Estimado como	Sagrada Família	Sagrada Família	Sagrada Família
Hotel Vela			
Estimado como	Hotel Vela	Hotel Vela	Hotel Vela
Montjuïc			
Estimado como	Montjuïc	Montjuïc	Montjuïc

Torre Agbar			
Estimado como	Torre Agbar	Torre Agbar	Torre Agbar
Colón			
Estimado como	Colón	Colón	Colón
Torres Mapfre			
Estimado como	Torres Mapfre	Torres Mapfre	Torres Mapfre

Tabla 9. Ejemplos de imágenes estimadas correctamente por el sistema de clasificación

11 Conclusiones

Este proyecto tenía como objetivo principal analizar, diseñar e implementar un sistema reconocimiento y clasificación de edificios, el cual permitiera ser probado con imágenes nuevas para el sistema y que éstas fueran clasificadas con una tasa de acierto razonable.

Se han utilizado diferentes técnicas de visión por computador para finalmente desarrollar un sistema basado en el aprendizaje con máquinas de vectores de soporte y que permite dos configuraciones posibles para la obtención de los vectores de características usados en el aprendizaje.

Los resultados obtenidos durante las pruebas realizadas han sido satisfactorios, sobretodo en la obtención de los vectores de características mediante histogramas donde se ha obtenido una tasa de acierto media del 90.2% en la clasificación de edificios.

El proceso de análisis, diseño e implementación final del proyecto ha comportado una curva de aprendizaje, puesto que se han ampliado conocimientos tanto en el lenguaje de programación Matlab usado para el desarrollo del proyecto, como en métodos de visión por computador y procesamiento de imágenes, extendiendo así los conceptos y la base ya adquirida durante el cursado de las asignaturas. Estos conocimientos corresponden a las distintas técnicas usadas en el proyecto como la extracción de características de imágenes, descriptores de imagen, definición de un diccionario de palabras visuales, representación de imágenes en el diccionario visual, la clasificación y entrenamiento de un sistema mediante SVM y métodos de validación de resultados estadísticos.

Concluye así con éxito el análisis, el diseño y la implementación del sistema de reconocimiento y clasificación de edificios.

Se comentan a continuación algunas de las mejoras o posibles futuras líneas de investigación del proyecto:

- Eliminar las palabras visuales del diccionario demasiado frecuentes en las imágenes y con información poco relevante acerca de los edificios, como por ejemplo trozos de cielo.
- Tener en cuenta el orden o la posición dónde suelen aparecer las palabras visuales.
- Acoplar el sistema de reconocimiento y clasificación de edificios a la aplicación web.
- Aumento del número de fotografías en la base de datos y distintos porcentajes para las muestras de entrenamiento y test del sistema.

12 Bibliografía

Referencias

- Andrea Vedaldi, B. F. (s.f.). *VLF org*. Obtenido de <http://www.vlfeat.org/>
- Fei-Fei, L. (s.f.). *Stanford Vision Lab*. Obtenido de <http://vision.stanford.edu/publications.html>
- Fernández, A., López, V., Galar, M., Jesús, M. J., & Herrera, F. (2013). Analysing the classification of imbalanced data-sets with multiple classes. 14.
- Jabardo, J. M. (s.f.). <http://www.imf-formacion.com/blog/marketing/>. Obtenido de El Blog de Marketing Online: <http://www.imf-formacion.com/blog/marketing/>
- Jonathan Milgram, M. C. (s.f.). "One Against One" or "One Against All".
- MathWorks. (s.f.). Matlab.
- Mathworks. (s.f.). *System Requeriments - Release 2011a*. Obtenido de <http://www.mathworks.es/support/sysreq/release2011a/index.html>
- Matteucci, M. (s.f.). *A Tutorial on Clustering Algorithms*. Obtenido de <http://extraccionrecuperacionnosupervisada.50webs.com/clustering.html>
- Science, U. o. (s.f.). *Confusion Matrix*. Obtenido de http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html
- ubsense. (s.f.). <http://blog.ubsense.com/>.
- Wikipedia. (s.f.). *Histogram of oriented gradients*. Obtenido de http://en.wikipedia.org/wiki/Histogram_of_oriented_gradients
- Wikipedia. (s.f.). *Validación Cruzada*. Obtenido de https://es.wikipedia.org/wiki/Validaci%C3%B3n_cruzada
- Works, M. (s.f.). *Documentation center*. Obtenido de <http://www.mathworks.com/help/>

Aplicaciones Web

- Gliffy. (s.f.). Obtenido de <http://www.gliffy.com/>
- Barashev, D., & Thomas, A. (s.f.). *Gantt Project*. Obtenido de <http://www.ganttproject.biz/>

