# Categorización automática de Imágenes

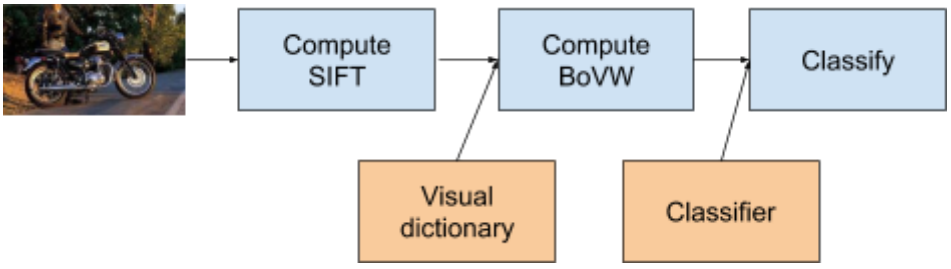
## Objetivo

El objetivo de esta práctica es desarrollar un sistema basado en visión para categorizar de forma automática los contenidos visuales de las imágenes.

## Definición del problema

Dada una imagen de entrada y un conjunto de categorías predefinidas, el sistema de visión tiene que ser capaz de clasificar una imagen de entrada en la categoría correcta.
A continuación se resumen los principales pasos de una sistema básico para conseguir este objetivo:

1. Extracción de características locales: se utilizará el detector/descriptor SIFT para detectar y describir los puntos característicos en las imágenes.
2. Sólamente durante la fase de entrenamiento, a partir de un conjunto de imágenes, calcular un agrupamiento (clustering) de los descriptores SIFT extraídos en K palabras visuales para obtener nuestro diccionario visual. Se sugiere utilizar el algoritmo K-means empezando con 100 palabras.
3. Representación global de la imagen. Dada una simple imagen, crear el descriptor global BoVW (Bag of Visual Words) consistente en un histograma de frecuencia de las palabras visuales presentes en la imagen, a partir de asignar a cada descriptor SIFT detectado en la imagen una palabra visual del diccionario. Se sugiere utilizar un clasificador k-NN con k=1 para realizar la asignación.
4. Clasificación: a partir de los descriptores BoVW extraídos de un conjunto de imágenes de entrenamiento, entrenar un clasificador k-NN. En la fase de Test, usar este clasificador entrenado para asignar categorías a un conjunto de imágenes de Test (distinto de las imágenes de Entrenamiento).

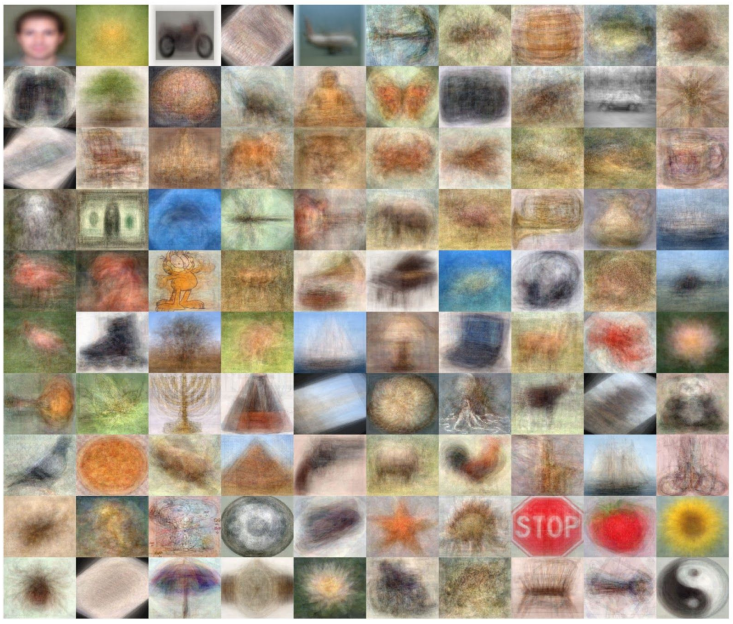



Nótese que la secuencia que se acaba de describir define un sistema básico para empezar. Se espera que los estudiantes utilicen aproximaciones más sofisticadas para mejorar los

resultados tanto como sea posible. Sugerencias para realizar esto serán proporcionadas durante las clases teóricas.

# Dataset

Para entrenar y comprobar nuestro sistema, vamos a utilizar el dataset “Caltech-101” es cual puede ser descargado desde la URL:
http://www.vision.caltech.edu/Image_Datasets/Caltech101/




Este Dataset contiene 101+1(fondo) categorías de objetos, como puede ser visto en la figura anterior. Para más detalles sobre el Dataset consultar el sitio Web.
Con la intención de acelerar los experimentos, se sugiere redimensionar las imágenes de tal manera que el ancho de todas las imágenes sea 300 pixeles, manteniendo su ratio de aspecto original (esto es si originalmente W/H = 1.5, en la versión redimensionada también). Se sugiere utilizar la función opencv, cv::resize.

# Protocolo de evaluación

El conjunto de Train contiene un muestra aleatoria de N imágenes de cada categoría y el conjunto de Test contiene una muestra aleatoria de 50 imágenes (o las restantes si no hubiera suficientes) de cada categoría, distintas a las seleccionadas para el conjunto de Train. Se evaluarań casos para valores de N={5, 15, 30}.
Se proporcionan varios subconjuntos de categorías del Dataset para facilitar el desarrollo y evaluación del sistema. Estos subconjuntos estarán disponibles en el Moodle en forma de ficheros de configuración.

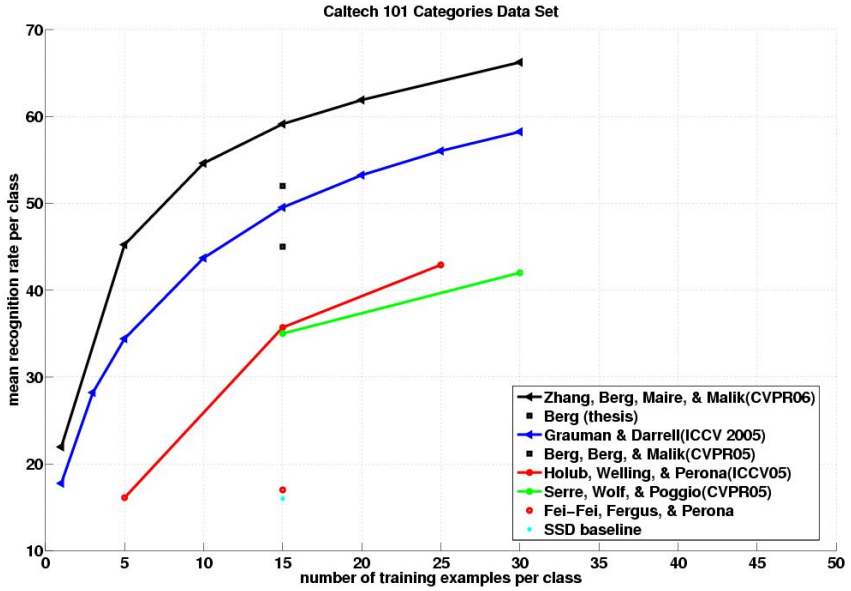
# ¿Cómo medir el rendimiento del sistema?

Para cada categoría c, será calculado el ratio de reconocimiento $r_c$. Después, se calculará el ratio promedio sobre todas las clases $R_{all}$. Este ratio es conocido como “ratio de reconocimiento por clase medio”. El proceso de “entrenamiento+prueba” se repetirá 10 veces generando aleatoriamente 10 conjuntos distintos de Train/Test y el rendimiento del sistema desarrollado será definido como el valor promedio de los ratios:

$$\overline{R} = \frac{1}{10} \sum_{i=1}^{10} R_{all}^{i}$$

junto con la desviación estándar correspondiente.

Los estudiantes deberán generar un informe de sus pruebas añadiendo una gráfica de líneas para mostrar los resultados finales obtenidos para los distintos valores de N. A continuación se muestra un gráfico similar con valores obtenidos por distintos investigadores sobre el Dataset al completo:




# ¿Qué entregar?

Los estudiantes tienen que desarrollar, al menos, los siguientes dos programas:

- train_bovw: este programa es el encargado de entrenar el sistema y seleccionando la configuración de parámetros adecuada. Los modelos entrenados (diccionario visual y clasificador) deberán almacenarse en disco (usando formato YML) para ser utilizados posteriormente en la fase de prueba. Además este programa será el que genere los valores de “ratio de reconocimiento por clase medio” para generar el informe.
- test_bovw.cpp: este programa es el encargado de aplicar un modelo entrenado a un conjunto de imágenes indicando sus categorías. Opcionalmente permitirá un flujo de video en vivo.

Los estudiantes subirán a Moodel un sólo fichero zip con nombre "apellido1_apellido2_bovw.zip". Este fichero debe contener la siguiente estructura de directorios y ficheros:

- `README.txt`: Texto plano describiendo cómo compilar y ejecutar los programas.
- `CMakeFiles.txt`: Configurarción para compilar. Nótese que al menos dos ejecutables son esperados: `train_bovw` and `test_bovw`.
- `src`: directorio con las fuentes (`.h`, `.cpp`);
- `models`: modelos entrenados por el estudiante para los distintos valores de número de categoría/N. Estos modelos después serán cargados por 'test_bovw'. Sugerencia de nombres: `cat2_N15_dictionary.yml` y `cat2_N15_kNN_k1.yml`;
- `data`: pequeño conjunto de imágenes (p.e. 10) descargadas desde Internet de las clases entrenadas y que serán usadas para validar el sistema desarrollado.
- `docs`: un fichero pdf con nombre `results.pdf` debe ser incluido resumiendo tanto gráficamente como en tablas los resultados obtenidos con los modelos finales (los mejores). Se espera que también haya una sección dedicada a realizar una discusión de los resultados y conclusiones.

Además, una página Web en el Moodle será disponible para que los alumnos puedan ir publicando sus mejores resultados y compartirlos con el resto de estudiantes.

# ¿Cuándo entregar?

Esta práctica se divide en dos partes y cada parte tendrá un fecha de entrega. Una vez pasada la fecha de entrega correspondiente, la nota máxima se reducirá por semana de retraso.

# Orientación para la calificación

La siguiente tabla sirve de orientación para saber la calificación que puede obtenerse en función de lo entregado.

**PARTE I:**

| ***Points (up to)*** | ***Item*** | ***Details*** |
|---|---|---|
| 7 | Funcionalidad básica (obligatorio) | ● El usuario puede seleccionar el descriptor que será usado para describir los puntos característicos. Al menos dos tipos de descriptores son esperados, p.e., SIFT, SURF...<br>● El número de vecinos K usados para clasificar (kNN) puede ser seleccionado durante entrenamiento/testeo.<br>● Desarrollar un programa llamado test_bovw que recibe como entrada una imagen y un modelo entrenado y devuelve la categoría asignada o un |

| | | |
|---|---|---|
| | | ranking de categorías con sus correspondientes scores.<br>La cli sería:<br>`test_bovw --img <input_image> --classifier <path_classifier> --dict <path_dictionary> --config_file <path_classes_config>` |
| +1.25 | Dense SIFT | El sistema permite usar SIFT denso multiescala. |
| +1.25 | PHOW | El sistema permite usar PHOW. |
| +0.5 | Interfaz gráfica. | El sistema produce resultados gráficos tanto para las fases de entrenamiento y prueba, como para cualquier otra facilidad GUI. P.ej. en la fase de test se muestra una matriz de confusión visual de la prueba, y el programa de test_bovw muestra la imagen de entrada y muestra encima de ella la categoría inferida. |

**PARTE II:**

| ***Points (up to)*** | ***Item*** | ***Details*** |
|---|---|---|
| 7 | Basic functionality (compulsory) | ● The system allows using SVM as the classifier. The parameters of the SVM (at least, kernel and margin) can be chosen during training. The possible SVM kernels must include: linear, radial, polinomial.<br>● The system is able to recognize objects in (almost) real-time by using the web camera of the computer and/or a video file. |
| +1 | Alternative classifiers | In addition to *k*-NN and SVM, the system gives the option of using alternative classifiers. E.g. Random Forests, Boosting,... |
| +1 | Fisher Vectors | The system gives the option of using Fisher Vectors for encoding. |
| +1 | VLAD | The system gives the option of using VLAD for encoding. |