

---

## Miniproyecto 3:

# Clasificación supervisada y no supervisada por medio de descriptores de color, textura y forma.

---

## 1. Objetivos

- Familiarizarse con los principales descriptores utilizados en la visión por computador para representar las imágenes.
- Aprender a usar los métodos más comunes para la clasificación tanto supervisada como no supervisada.
- Adquirir las habilidades de realizar una correcta metodología experimental y plasmarla asertivamente en las secciones de abordaje, experimentos y discusión de un artículo científico en formato CVPR.
- Resolver la tarea de clasificación en un problema con imágenes naturales.

## 2. Parámetros de entrega

No se recibirá ningún archivo por algún medio diferente a Brightspace.

### ENTREGAS:

- Adjuntar un archivo con el desarrollo principal de los códigos de cada entrega de la siguiente forma:  
*main\_CódigoEstudiante1\_CódigoEstudiante2.py*
  - **NO** se recibirán archivos con otra extensión. Su entrega conllevará una penalización sobre la calificación final de la entrega.
  - **La entrega de un archivo .py es obligatoria.**

- El código enviado debe tener una longitud máxima de 500 líneas. Se generará una penalización sobre la calificación final de cada entrega por cada 10 líneas adicionales. Esto con el objetivo de promover el uso de estructuras iterativas y operaciones vectoriales, que aumenten la eficiencia de sus códigos (aspecto fundamental en tareas de programación).
- Adjuntar un archivo con las clases y funciones de soporte de cada entrega de la siguiente forma: *utils\_CódigoEstudiante1\_CódigoEstudiante2.py*.
  - **NO** se recibirán archivos con otra extensión. Su entrega conllevará una penalización sobre la calificación final de la entrega.
  - **La entrega de un archivo .py es obligatoria.**
- Adjuntar una copia de **TODO** el código .py de los ítems anteriores en formato .txt (Este debe contener tanto el código de su `main` como el de su `utils`). Este debe llamarse de la siguiente manera: *CódigoEstudiante1\_CódigoEstudiante2.txt*. Esto con el fin de poder evaluar en Brightspace automáticamente cualquier intento de copia, plagio, o similitud entre los algoritmos. **Cualquier intento de copia/plagio será tratado de acuerdo al reglamento de la Universidad. Aquel grupo que no incluya este ítem en cualquiera de sus entregas tendrá una nota de 0 en la entrega que corresponda.**
- Adjuntar un único archivo con el informe en PDF. El informe debe presentarse con el formato **CVPR** (para más información del informe, ver sección 4). Este debe ser nombrado de la siguiente forma: *CódigoEstudiante1\_CódigoEstudiante2.pdf*.
- En el caso de que no tenga compañero, debe utilizar un cero como el código de su compañero, i.e. *main\_CódigoEstudiante\_0.py* para el archivo de código, *CódigoEstudiante\_0.pdf* para el informe y *CódigoEstudiante\_0.txt* para el archivo de texto.
- **NO** se permiten archivos comprimidos tales como *zip*, *rar*, *7z*, *tar*, *gz*, *xz*, *iso*, *cbz*, etc. Aquel grupo que envíe su entrega como un archivo comprimido tendrá una penalización sobre la calificación de la entrega.
- Este proyecto tiene una duración de cuatro semanas y está compuesto por tres entregas. Cada entrega del presente miniproyecto recibirá una calificación por separado. Estas se entregarán en las siguientes fechas:
  - Entrega 1 (3 %): Viernes 22 de abril de 2022
  - Entrega 2 (3 %): Viernes 29 de abril de 2022
  - Entrega 3 (4 %): Viernes 06 de mayo de 2022

El plazo de entrega será hasta las 9:59 p.m. de las fechas establecidas. Sin embargo, el vínculo para el envío de los documentos estará disponible hasta las 10:59 p.m. De 10:00 p. m a 10:29 p.m la entrega se calificará sobre 4.0. Y de 10:30 p.m a 10:59 p.m la entrega se calificará sobre 3.0 (tenga en cuenta que la hora queda registrada en el envío de Brightspace).

- Se tendrá flexibilidad únicamente ante las circunstancias extraordinarias y estipuladas en el reglamento de la Universidad. Ante esta situación, les pediremos se comuniquen con una anticipación moderada (no a última hora antes de las entregas) con los dos asistentes docentes por correo explicando su situación y así poder establecer un nuevo plazo para su entrega por Brightspace.
- Cada estudiante tiene varios intentos en Brightspace para cada entrega, pueden hacer entregas parciales y **se tendrá en cuenta únicamente el último intento** para la calificación.
- Teniendo en cuenta que se recibirán archivos después de la hora estipulada con una penalización, **no se recibirán documentos o archivos después de la hora máxima de entrega (10:59 p.m) ni por Brightspace, ni por ningún otro medio.**

### 3. Reglas Generales

- La asistencia a la sección de laboratorio es **obligatoria**. De acuerdo con el **Reglamento General de Estudiantes de Pregrado de la Universidad de los Andes**, la inasistencia a más del 20 % de las clases de laboratorio resultará en la reprobación de la totalidad de la materia (laboratorio y magistral).
- Los informes deben realizarse **únicamente** con la pareja. Esto quiere decir que, aunque es válido discutir los problemas generales con sus compañeros, la solución y el código deben ser de su completa autoría. **Está prohibido** copiar literalmente el algoritmo y/o procedimiento desarrollado por otro grupo.
  - Si llega a obtener un código de Internet, asociado al problema a resolver, este debe estar **debidamente referenciado** y usted debe entenderlo por completo. Sin embargo, todos los ejercicios que requieren una implementación propia, deben ser de su completa autoría. En estos casos, no podrán tomar implementaciones (o partes) de un autor externo así estén adecuadamente referenciadas.

### 4. Parámetros de Calificación

#### Resultados:

1. Toda figura debe estar numerada y debe llevar su título y descripción en el informe.
2. Todos los códigos deben mantener orden y coherencia en la ejecución de comandos. Cada vez que se muestre una figura, el programa debe esperar para que se presione una tecla, para así continuar con la siguiente y así sucesivamente (para esto utilice en Python `input("Press Enter to continue...")`). Si quiere contrastar dos o más imágenes utilice un `subplot`.
3. Independientemente del entorno de desarrollo integrado (IDLE) que utilicen para implementar sus códigos con extensión `.py`, solo utilicen funciones y librerías propias de python. Por lo que, se debería poder correr su código desde cualquier compilador o terminal.

4. Dado que esta es una materia de programación, es igual de importante tanto la sintaxis como la ejecución de sus códigos. Por este motivo, la nota correspondiente al código será distribuida entre la correcta ejecución del código y el desarrollo del mismo.
5. El código debe estar adecuadamente documentado (comentado).
6. Para no tener inconvenientes a la hora de realizar la entrega, asuma que al mismo nivel de ejecución del código se encuentra una carpeta con TODOS los archivos necesarios para el laboratorio. Esta carpeta SIEMPRE será proveída por lo que no deben cambiar ni el nombre de la carpeta ni de los archivos dentro de ella.
7. No utilice rutas absolutas para leer o guardar archivos. Este es el error más común en la ejecución de los códigos.

Para generar rutas utilice `os.path.join` en Python, ya que puede que corramos los laboratorios usando Linux o Windows y los separadores de archivos cambian dependiendo del sistema operativo.

**Ejemplo:** Dentro del código principal, el estudiante quiere leer la imagen `im.png` que está dentro de una carpeta de imágenes en la misma ruta que el `main`.

7.1. Forma incorrecta:

```
skimage.io.imread('C:/Estudiante/docs/ElProyecto/ims/im.png')
```

7.2. Forma correcta:

```
skimage.io.imread(os.path.join('ims', 'im.png'))
```

### Informe:

A lo largo de este miniproyecto irán construyendo las secciones más importantes de un artículo científico completo formato CVPR las cuáles son el abordaje (“Approach”), los experimentos y la discusión. Teniendo en cuenta que no deberán incluir las demás secciones como abstract, introducción y estado del arte, la **longitud máxima de la entrega final será de 6 páginas**, excluyendo referencias y anexos. Se pueden incluir imágenes y/o tablas en la sección de Anexos pero los resultados principales (solicitados) deben ser parte del cuerpo del informe. Todas las entregas deben realizarse en formato  $\LaTeX$  o Word, pueden obtener una plantilla del formato en el siguiente [vínculo](#).

### Bonos:

Cada pareja podrá obtener puntos adicionales por cumplir la siguiente característica:

1. Desarrollar el informe en  $\LaTeX$ . Aquellas personas que lo desarrollen en  $\LaTeX$ , deben escribir al final del informe **”Realizado en  $\LaTeX$ ”**. De lo contrario no se contará el bono. Para poder escribir el logo utilice el comando “`\LaTeX`” en su editor de Latex.

Estos puntos se asignarán de acuerdo al criterio de los profesores y los grupos que intenten reproducir la frase en un informe realizado en Word obtendrán cero (0) en la nota de dicha entrega.

## 5. Definición del problema

La tarea de clasificación ha tenido protagonismo en la visión por computador a lo largo de los años, incluso siendo de utilidad para resolver las otras dos principales tareas de visión por computador; en el caso de detección al clasificar ventanas deslizantes, y en el caso de segmentación clasificando superpíxeles. Por lo anterior, la clasificación es una tarea relevante y de gran importancia en la visión por computador; por lo cual, han existido una gran cantidad de problemas y bases de datos enfocados en resolver esta tarea. Debido a esto, en este ultimo miniproyecto trabajarán en la tarea de clasificación.

En este miniproyecto la tarea de clasificación que se trabajará será enfocada en cinco variedades diferentes de flores, estas son: la *Aechmea*, la rosa, el cartucho, el girasol y la orquídea. La base de datos que se utilizará la pueden encontrar en la carpeta `data_mp3` y en la Figura 1 se encuentran ejemplos de estas clases de interés .

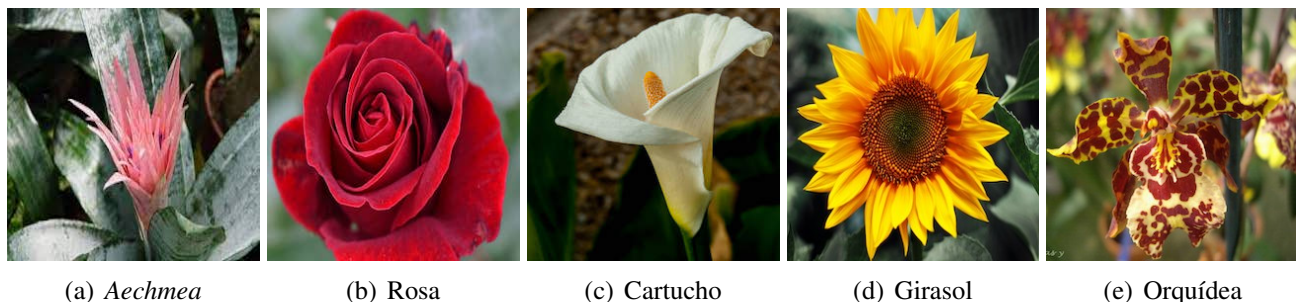


Figura 1: Ejemplos de las clases que se trabajarán en el proyecto.

Como ya saben, a lo largo de este miniproyecto realizarán las secciones de abordaje, experimentos y discusión de un artículo científico CVPR. Este estará dividido en tres entregas diferentes. En la primera y la segunda implementarán una línea de base utilizando el algoritmo de clasificación no-supervisada de k-means junto con los descriptores de color (primera entrega), de textura (segunda entrega) y de forma (segunda entrega). Realizando la respectiva experimentación de este método y descriptores definirán su método final de baseline. Por último, en su tercera entrega, utilizarán los mismos tres descriptores previamente mencionados, entrenando como mínimo, dos clasificadores supervisados: Support Vector Machines (SVM) y Random Forests (RF), para así poder establecer su método final.

**Nota Importante:** Tengan presente que al momento de entregar sus códigos en cada una de las entregas, deberán dejar comentadas las secciones de código correspondientes a la experimentación. Además, en comentarios también deben dejar claro dónde se realiza la validación y dónde la evaluación. De esta forma, cuando nosotros ejecutemos sus códigos estos deberán importar el (o uno de los modelos entrenados que se les solicitan según la entrega) y ejecutar la validación y/o la evaluación.

## 6. Entrega 1 → k-means + color

**Nota Importante:** Tengan presente que adicional a lo solicitado en la sección 2, en este caso deberán incluir:

- El archivo con el mejor modelo de k-means entrenado (un archivo `.mat`, `.npy`, `.npz`, `.pkl`, *etc.*): `final_kmeans_model_Código1_Código2.npy`, *etc.*

### 6.1. Procedimiento

A no ser que se trabaje con Deep Learning, para hacer un trabajo de clasificación, es necesario tener 2 elementos principales: un descriptor y un clasificador. El primero, tal como su nombre lo indica, es el que nos va a permitir describir la imagen, es decir expresar sus características de forma cuantitativa para poder tomar decisiones sobre esta. El segundo, cuyo nombre también nos da una pista de su utilidad, sirve para tomar las características que nos da el descriptor y poder asignar clases a las imágenes basándonos en estas. En esta primera entrega, utilizarán descriptores basados en color y un clasificador no supervisado basado en clustering conocido como k-means.

#### 6.1.1. Descriptor

Lo primero que se debe realizar, como ya se dijo, es el descriptor mediante el cual vamos a representar una imagen en un vector. En esta entrega se trabajará únicamente con descriptores basados en color. Estos, se basan en representar la imagen según los diferentes colores que la compongan mediante el uso de histogramas, ya que pretende caracterizar la frecuencia de cada una de las intensidades de los colores.

Este descriptor de histogramas de color será el primero que implementarán. La función debe tener la capacidad de calcular tanto histogramas de color conjuntos como histogramas de color concatenados los cuáles pueden obtenerlos utilizando las funciones `CatColorHistogram` y `JointColorHistogram` que se encuentran en el archivo `utils.py` de la carpeta `data_mp3`. Además, debe poder utilizar diferentes canales de color como `rgb`, `hsv` y `lab`. Un punto importante que deben tener en cuenta es que usarán esta función no solo para calcular los descriptores, sino para almacenarlos. La razón de esto es que los descriptores en algunos casos pueden tardar un tiempo considerable en ser calculados. Debido a lo anterior, sería computacionalmente costoso y tomaría tiempo en exceso hacer el cálculo de los descriptores en cada una de las entregas. El archivo que deben almacenar debe contener dos listas las cuáles tengan posiciones correspondientes entre ellas, una con la información de los descriptores (uno por imagen) y la otra con la anotación correspondiente. La función debe tener la siguiente estructura:

#### Sintaxis:

```
color_Código1_Código2(images, labels, route, type, space_bins, color_space)
```

#### Parámetros de entrada:

- `images`: Lista con las imágenes (o rutas de las imágenes) a la cuales se les calculará el descriptor.

- `labels`: Lista con las etiquetas de las imágenes pasadas por parámetro.
- `route`: Ruta donde serán guardados los descriptores junto con las etiquetas.
- `type`: Parámetro que indica el tipo de histograma de color que se calculará, puede ser “joint” o “concat”.
- `space_bins`: Número entero que indica el numero de bins que existirá por cada canal de color.
- `color_space`: Parámetro que indica el espacio de color que se usará, puede ser “rgb”, “hsv”, o “lab”.

**Respuesta:**

- `features`: Arreglo con los descriptores de las imágenes (el mismo que se almacenará en `route`).

**6.1.2. Clasificación**

Ahora, teniendo su descriptor, es necesario que en base a este entrenen su primer clasificador. En este caso haremos uso del clasificador no supervisado denominado k-means. Debido a que k-means engloba los datos en el número de clusters que ustedes determinen, al ser no supervisado no es posible ingresarle una etiqueta de forma automática. Por lo anterior, deberán entrenar su modelo de k-means y después de esto encontrar una correspondencia entre cada cluster y la etiqueta que corresponda a dicho cluster.

Este procedimiento lo harán en la clase `kmeans_classifier` que pueden encontrar en el documento `utils.py` de la carpeta `data_mp3`. La razón por la que en esta ocasión se realizará el procedimiento utilizando una clase y no funciones como se había realizado hasta el momento es por que debemos tener un modelo que permita mantener memoria de los clusters con los que se entrenó. Esta clase tiene 2 funciones principales dentro de las cuales deben desarrollar su implementación:

**fit:** La primera función que deben realizar debe ser capaz de entrenar su modelo de clasificación por k-means. Esta no solo debe entrenar el modelo de k-means que es un atributo de la clase, sino que debe ser capaz de hacer una asociación de cada cluster con una de las clases. Por lo anterior, deben modificar el atributo llamado `correspondance_dictionary` para especificar a que clase pertenece cada uno de los clusters; actualmente este se inicializa por defecto asociando cada uno de los clusters(las llaves) a la clase con valor `-1`.

**predict:** Esta debe permitir hacer la predicción a cada nuevo dato ingresado por parámetro. Para esto, deben usar tanto el modelo de clustering como el diccionario de correspondencia para así poder asignarle una única clase como predicción.

### 6.1.3. Experimentación

Para finalizar la metodología deben realizar la fase de experimentación de su descriptor, por lo cual deben usar su clase `kmeans_classifier` para clasificar sus imágenes usando el color como descriptor. Para esto deben dejar el numero de clusters como el que viene por defecto (20) al momento de inicializar su clasificador. Deben realizar al menos los siguientes experimentos:

**Color:** Grid 2x3, variando el tipo de histograma que se calcula y el espacio de color respectivamente.

## 6.2. Artículo

Una vez realizado el procedimiento es fundamental que empiecen a escribir el artículo que irán construyendo a lo largo de todas las entregas.

### 6.2.1. Abordaje

Ahora que ya realizaron una parte del procedimiento, deben incluirlo en su artículo como parte de su metodología experimental.

#### Descriptores

Inicialmente explicarán los descriptores, estos no solo los usarán para su método de baseline, sino para el método final. Para esta entrega, describan el descriptor de color, además de los parámetros que variaron y su propósito. Para esta descripción pueden basarse en las siguientes preguntas:

- ¿En qué consistía su descriptor?
- ¿Qué pretendía mostrar?
- ¿Qué características de la imagen usa?
- ¿Por qué este descriptor sería de utilidad?

#### Baseline

Paso siguiente, deben describir su método de k-means, el cual será su baseline con el que deberán compararse al tener su metodología experimental. Para esto describan brevemente el procedimiento que usaron para poder asignar etiquetas basados en clusterización. No incluyan información sobre la librería o la función específica, basta con el método, parámetros y describir su razonamiento. Pueden guiarse de las siguientes preguntas:

- ¿En qué consiste su método?
- ¿De qué elementos y procedimientos se compone?
- ¿Qué ventajas tiene?
- ¿Por qué este clasificador sería de utilidad?



### 6.2.2. Experimentos

#### Método propuesto

Finalmente, ya cuentan con los primeros experimentos así como sus primeros resultados, los cuáles deberán ser agregados a su artículo en la sección de experimentos. Primero hagan la descripción de los experimentos que realizaron para los descriptores de color. Luego, incluyan los resultados de la experimentación en una única tabla y señalen en negrilla claramente cuál fue el mejor resultado. Además, deben hacer una breve descripción de sus resultados. Para lo anterior, pueden usar las siguientes preguntas como guía para armar su descripción.

- ¿Cuál fue su mejor método?
- ¿Hubo algún parámetro que influyera mas? (i.e. el histograma conjunto tuvo mejor desempeño en términos generales que el concatenado.)
- ¿Qué combinación no funcionó de forma correcta?
- ¿Existía alguna tendencia?

### 6.3. BONO

Esta primera entrega tiene un bono asociado. Para obtener este, usted debe tomar de base la clase `kmeans_classifier` y modificar esta nueva clase para realizar una con un procedimiento similar. Esta nueva clase debe ser llamada `MeanShift_classifier`, en la cual deben cambiar el método de clusterización a **Mean Shift**. Además de esta implementación, debe hacer la descripción de esta nueva metodología en su sección de **Baseline** y reportar los resultados obtenidos implementando el mejor descriptor de color (seleccionado con los resultados de k-means) en su sección de **Experimentos**.

## 7. Entrega 2 → k-means + color/textura/forma

**Nota Importante:** Tengan presente que adicional a lo solicitado en la sección 2, en este caso deberán incluir:

- Un archivo con el nombre *texton\_dictionary\_Código1\_Código2.mat*, el cuál debe contener el diccionario de textones calculado por ustedes que obtuvo los mejores resultados.
- El archivo con el mejor modelo de k-means entrenado (un archivo *.mat*, *.npy*, *.npz*, *.pkl*, *etc.*): *final\_kmeans\_model\_Código1\_Código2.npy*, *etc.*

### 7.1. Procedimiento

En esta segunda entrega se seguirá trabajando en la metodología experimental, esta vez introduciendo y experimentando con 2 nuevos descriptores. Además de esto realizarán una experimentación sobre el método de clasificación de su baseline para así poder definir el método final de baseline compuesto por un descriptor y un clasificador con los parámetros más adecuados.

#### 7.1.1. Descriptor basado en textura

Una de las formas en las que se puede hacer una descripción de la textura de una imagen es mediante el histograma de textones, el cual es una forma de cuantificar la presencia de las textura existentes. Para esto, deben implementar una función que calcule dicho histograma utilizando un diccionario de textones. Por lo anterior, implementarán tres funciones auxiliares que les permitirán obtener sus descriptores de textura. Estas funciones deben realizarlas para poder calcular la respuesta a los filtros, calcular el diccionario de textones y calcular el histograma de textones.

**1) `calculate_filter_response_Código1_Código2`:** Esta primera función tiene como propósito hacer el cálculo de las respuestas a los filtros. Como podrán recordar de lo visto en la magistral, el primer paso para obtener sus textones, así como el mapa de textones de una imagen, es calcular la respuesta de la imagen en escala de grises a cada uno de los filtros que se encuentren en el banco. Finalmente, deben almacenar la respuesta a los filtros de cada uno de los píxeles de la imagen.

**2) `calculate_texton_dictionary_Código1_Código2`:** Esta segunda función tiene como propósito construir un diccionario de textones. Esta deberá recibir por parámetro el arreglo de píxeles descritos por sus respuestas a los filtros (en las imágenes de entrenamiento). Luego, haciendo uso de un algoritmo de clusterización (comúnmente k-means), agruparlos en lo que serán “tipos de textura”. Por último, los centroides de los clusters de textura generados, los cuales serán nuestros textones, deberán ser retornados.

**3) `calculate_texton_histogram_Código1_Código2`:** Esta última función generará el mapa de textones de una imagen, en base al cual obtendrán como respuesta el histograma de textones asociado. Esta deberá recibir por parámetro el arreglo de píxeles de la imagen a describir (la respuestas a los filtros) y asociar cada píxel a la etiqueta del centroide (textón) del cluster correspondiente, para finalmente construir y retornar el histograma de textones.

Como ya fue mencionado, haciendo uso de las funciones previamente mencionadas y el/los diccionarios de textones que han generado y almacenado, deberán construir una función que genere el descriptor de textura y debe tener la siguiente estructura:

```
texture_Código1_Código2(images, labels, route, textons)
```

**Parámetros de entrada:**

- `images`: Imágenes de la cuales se calculará el descriptor.
- `labels`: Etiquetas de las imágenes.
- `route`: Ruta donde serán guardados los descriptores junto con las etiquetas.
- `textons`: Diccionario de textones o ruta al archivo que lo contiene.

**Respuesta:**

- `features`: Arreglo con los descriptores de las imágenes (el mismo que se almacenará en `route`).

### 7.1.2. Descriptor basado en forma

Finalmente, el último descriptor que utilizaremos en este miniproyecto será basado en forma, para esto usarán el **Histograma de gradientes orientados (HOG)**. Este descriptor es especialmente importante ya que es el único descriptor de los que utilizaremos que tiene en cuenta la posición espacial. Para esto, deberán implementar una función que retorne el histograma de gradientes orientados. En esta, deben poder variar por parámetro 2 parámetros que ustedes establezcan para experimentar. Al igual que las funciones de los otros descriptores, esta función no solo debe retornar los descriptores de las imágenes sino que debe poder almacenarlos en un archivo local. Esta función debe tener la siguiente estructura:

```
shape_Código1_Código2(images, labels, route, param1, param2)
```

**Parámetros de entrada:**

- `images`: Imágenes de la cuales se calculará el descriptor.
- `labels`: Etiquetas de las imágenes.
- `route`: Ruta donde serán guardados los descriptores junto con las etiquetas.
- `param1`: Parámetro 1 que variarán para calcular HOG.
- `param2`: Parámetro 2 que variarán para calcular HOG.

**Respuesta:**

- `features`: Arreglo con los descriptores de las imágenes (el mismo que se almacenará en `route`).

### 7.1.3. Experimentación de descriptores

Utilizarán el mismo método de clasificación de *baseline* de la entrega 1 para evaluar la calidad de los descriptores y experimentar con ellos. Deben realizar al menos los siguientes experimentos:

**Textura:** Deben realizar la experimentación variando el número de textones que hay en el diccionario (es decir el numero de bins del histograma). Deben construir por lo menos 4 diccionarios diferentes.

**Hint:** La respuesta a los filtros solo deberían calcularla una vez, guarden esto en archivos y calculen los 4 diccionarios e histogramas usando estos.

**Forma:** Deben realizar la experimentación en grid 2x2 escogiendo 2 parámetros de la función para calcular HOG y 2 valores para cada uno de dichos parámetros.

### 7.1.4. Experimentación de clasificador

Ahora que ya tienen los mejores parámetros para cada uno de los descriptores, deben establecer cuál de los tres (color, textura y forma) les produjo mejores resultados. Este deberán usarlo para así poder encontrar su mejor clasificador de *baseline*.

**k-means:** La experimentación aquí deben realizarla cambiando el número de clusters. Para esto escoja 4 valores para este parámetro.

### 7.1.5. Test

Finalmente, tienen todo lo necesario para poder determinar su mejor método de *baseline* que consta de descriptor/clasificador. Después de haber establecido el modelo de k-means que les arrojó mejores resultados con su mejor descriptor en la etapa de validación, deben medir el desempeño de este en el set de test.

## 7.2. Artículo

Una vez realizada la parte práctica, deben incluir en su artículo todo lo que realizaron y obtuvieron para así poder construirlo progresivamente.

### 7.2.1. Abordaje

#### Descriptores

Ahora, deben incluir el procedimiento que acaban de realizar en su artículo como parte de su abordaje, siguiendo las pautas de la entrega pasada. Para esto describan los nuevos descriptores que usaron, además de los parámetros que variaron, su motivación, el por qué de los valores y su propósito. Pueden guiarse con las siguientes preguntas:

- ¿En qué consistían sus descriptores? ¿Qué pretendían mostrar?
- ¿Qué características de la imagen usan sus descriptores?
- ¿Por qué estos descriptores serían de utilidad?

### 7.2.2. Experimentos

#### Descriptores

Ya cuentan con resultados experimentales sobre los dos nuevos descriptores que acaban de implementar. Inclúyanlos en una tabla y señalen claramente cual fue el mejor resultado de cada uno de los descriptores. Además, deben hacer una breve descripción de sus resultados. Pueden usar las siguientes preguntas para guiar su descripción:

- ¿Cuál fue su mejor método?
- ¿Hubo algún parámetro que influyera mas?
- ¿Qué experimento no funcionó de forma esperada? ¿Por qué?
- ¿Existía alguna tendencia en los experimentos realizados?

#### Baseline

Registren los resultados de sus experimentos del clasificador de *baseline* en su artículo. Incluyan los resultados en una única tabla y resalten en negrilla su mejor resultado, además realice una breve descripción de estos resultados. Pueden usar las siguientes preguntas para guiar su descripción:

- ¿Qué numero de clusters les permitió tener un mejor desempeño?
- ¿Existe alguna tendencia en los resultados de sus experimentos? (i.e. menos clusters permitían un mejor desempeño)

### 7.2.3. Discusión

#### Descriptores

Incluyan un análisis de lo obtenido en sus experimentos. Para esto analicen las implicaciones que tienen las métricas obtenidas en los resultados así como los resultados cualitativos, teoricen sobre las razones detrás de estos resultados y cómo propondrían mejorar estas falencias.

#### Baseline

Además, deben discutir sobre el método final que obtuvieron para el *baseline* y los resultados de la experimentación que los condujeron a esta decisión, analicen las métricas y resultados cualitativos que lograron obtener.

Para la discusión pueden utilizar las siguientes preguntas como guía:

- ¿Por qué creen que el mejor experimento dio así?
- Si existe alguna tendencia ¿qué podría significar esta?
- ¿Se debe continuar con el descriptor de mejor desempeño?
- ¿Cuál de los descriptores cree usted que obtendrá mejores resultados con los clasificadores supervisados? ¿Por qué?
- ¿Cree usted que su método de *baseline* es óptimo? ¿Que ventajas/dificultades tiene?
- ¿Qué características comunes tienen aquellas imágenes mal clasificadas?

### 7.3. BONO

En esta entrega, el bono consistirá en un procedimiento similar al de la primera. Para obtener el bono ustedes deberán usar la clase `MeanShift_classifier` (misma de la entrega pasada) para hacer una experimentación similar a la que hicieron con su método de k-means, variando un parámetro propio de este descriptor. Además, deben incluir esto y sus resultados en las secciones correspondientes de su artículo.

## 8. Entrega 3 → SVM/RF + color/textura/forma

**Nota Importante:** Tengan presente que adicional a lo solicitado en la sección 2, en este caso deberán incluir:

- Un archivo con el mejor modelo de cada método entrenado (un archivo .mat, .npy, .npz, .pkl, etc.): *X\_final\_model\_Código1\_Código2.npy*, etc. Donde X hace referencia a 'SVM', 'RF' y 'MLP' según sea el caso.

### 8.1. Procedimiento

Habiendo implementado y experimentado con un método de *baseline* y los descriptores a utilizar en este miniproyecto, es momento de construir su método final. Esto lo harán experimentando con clasificadores supervisados: Support Vector Machine (SVM), Random Forest (RF) y Perceptrón Multicapa (opcionalmente, ver bono en la sección 8.3). Tengan presente que por facilidad utilizarán las implementaciones disponibles en la librería *sklearn*.

#### 8.1.1. Experimentación (validación)

Para esta sección utilizarán las implementaciones de las *máquinas de soporte vectorial* y de los *árboles de decisión*. Investiguen el propósito de cada uno de los parámetros que se tienen dentro de la implementación de cada método.

#### Descriptores

En esta sección realizarán la fase de experimentación de los descriptores implementados con respecto a los dos métodos supervisados que utilizarán. Por lo cual, deben evaluar cada uno de los clasificadores con cada uno de los 3 descriptores. Tengan presente que los descriptores a utilizar (uno para color, uno para forma y uno para textura) deberán ser los que demostraron un mejor desempeño en la etapa de experimentación de las entregas anteriores. Además, en esta etapa harán uso de sus clasificadores tal y como vienen por defecto (sin variar los parámetros originales de cada implementación).

#### Métodos supervisados (SVM y RF)

Después de haber seleccionado el descriptor que obtuvo un mejor desempeño para cada uno de los clasificadores supervisados, experimentarán con los parámetros de estos métodos (realizarán un “fine tuning”). Para cada uno escojan al menos 2 de los parámetros y varíenlos a por lo menos 2 valores distintos para cada parámetro (como mínimo obtendrán un grid 2x2 para la experimentación de cada método).

#### 8.1.2. Test

En este punto, tienen todo lo necesario para poder determinar su mejor combinación entre descriptor y clasificador basados en la experimentación realizada. Finalmente, usando el método con mejor desempeño deberán ponerlo a prueba usando su carpeta (*fold*) de test.

## 8.2. Artículo

### 8.2.1. Abordaje

#### Métodos Propuestos

Ahora que ya realizaron todo su procedimiento, deben incluirlo en su artículo. Para esto describan los clasificadores que usaron, los parámetros importantes, su motivación y propósito. Pueden guiarse de las siguientes preguntas:

- ¿En qué consisten sus métodos?
- ¿Qué ventajas tienen sobre el *baseline* y entre ellos?
- ¿Por qué estos clasificadores serían de utilidad?
- ¿Qué diferencias de aplicación y utilidad tienen los métodos propuestos?

### 8.2.2. Experimentos

#### Métodos Propuestos

En este punto ya cuentan con todos sus resultados de experimentación y evaluación. Debido a esto deben agregarlos a su artículo. Incluyan los resultados cuantitativos en tablas (idealmente dos, máximo tres) y resalten en negrilla cuál fue el mejor resultado de la experimentación. Además, incluyan los resultados cualitativos que consideren pertinentes en forma de subplots. Como ya saben, en esta sección deberán hacer una breve descripción de sus resultados. Pueden usar las siguientes preguntas como guía para construir su descripción:

- ¿Cuál fue el método que obtuvo un mejor desempeño? ¿Cuál fue el de peor desempeño?
- ¿Hubo algún parámetro que influyera mas en los resultados experimentales de cada método?
- ¿Hubo algún clasificador que tuviera mejor desempeño que el otro?
- ¿Existe algún descriptor que no obtuvo un buen desempeño al usarlo con estos clasificadores?

### 8.2.3. Discusión

Como ya saben, en esta sección deben incluir un análisis detallado de los resultados de su experimentación y evaluación así como de la descripción realizada por ustedes. Para esto analicen las implicaciones que tienen las métricas obtenidas en los resultados, teoricen sobre las razones detrás de estos resultados y contrasten sus resultados cualitativos con los cuantitativos. Pueden utilizar las siguientes preguntas como guía para iniciar su discusión:

- ¿Por qué creen que el mejor experimento de cada clasificador dio así?
- En el caso en que uno de los métodos siempre se desempeñe mejor ¿qué podría significar esto?
- ¿Las métricas de validación y test fueron congruentes?
- ¿Los resultados cualitativos y cuantitativos son congruentes?



- ¿Cree que puede realizar un procedimiento más efectivo? ¿Qué modificaciones realizaría?

Finalmente, deben concluir sobre lo que obtuvieron a lo largo de las tres entregas del miniproyecto. En este caso también deben contrastar los resultados obtenidos con su método de *baseline* en los sets tanto de validación como de test. Pueden guiarse de las siguientes preguntas:

- ¿Qué tanta mejoría se tuvo respecto al *baseline*?
- ¿Se logró el objetivo que tenían propuesto?
- ¿Qué mejoras se podrían realizar?
- ¿Hay espacio para mejoras en esta base de datos?

## 8.3. BONOS

### 8.3.1. Perceptron multicapa

El primer bono que se puede realizar en esta entrega es respecto a hacer la inclusión de un clasificador extra. Este nuevo descriptor que deben incluir es el **perceptron multicapa**. Para este, deberán correr los experimentos que realizaron para los otros clasificadores, es decir que deberán probar los diferentes descriptores y luego determinar 2 parámetros que deberán variar para hacer como mínimo la experimentación en grid 2x2. Para obtener el bono completo, deben incluir la información del clasificador, de la experimentación y discusión en las respectivas secciones de su artículo. Dado el caso que este sea su mejor método en validación, deberán seleccionarlo como método final y por ende realizar la evaluación en test.

### 8.3.2. Descriptores mixtos

El segundo bono que pueden realizar en esta entrega es sobre la creación de descriptores mixtos. Estos descriptores pueden incluir información que no solo incluya color, textura, o forma, sino que puede incluir información de dos o tres de estos en su interior. Para esto ustedes deben hacer las 3 combinaciones de descriptores (color+forma, color+textura y textura+forma) y la combinación de los 3. Esto deben realizarlo con su mejor clasificador (SVM, RF o MLP, en caso de haberlo implementado). Dado el caso que este descriptor sea parte de su mejor método en validación, deberán seleccionarlo como parte del método final y por ende realizar la evaluación en test. Como siempre, para obtener el bono deben incluir lo realizado en las respectivas secciones de su artículo.

### 8.3.3. Experimentación (aún más) exhaustiva

Basados en toda la experimentación que ya realizaron, el último bono que pueden obtener en esta entrega será realizando experimentos extra. Usando la combinación clasificador/descriptor con la que obtuvieron mejor desempeño deberán hacer una experimentación más exhaustiva variando aún más los parámetros tanto del descriptor como del clasificador. Para obtener el bono completo deberán realizar al menos 8 experimentos, adicionales a los solicitados inicialmente, para los cuáles sus decisiones de metodología experimental esté sustentada. Además, como siempre, deben incluir lo realizado en las respectivas secciones de su artículo.