**Examen**

Planteamiento del problema

1. Escoger al menos 3 fotos/imágenes/objetos favoritos que tengan en casa.
2. Utilizar la biblioteca OpenCV para implementar un programa que mediante una señal de video o que, al presentarle una nueva fotografía del objeto, sea capaz de identificar los objetos elegidos en el paso anterior, utilizando funciones de descriptores que ya contiene dicha biblioteca. Este programa deberá usar los siguientes descriptores:
3. ORB
4. SIFT
5. HOG
6. Comparar el desempeño de los 3 descriptores al identificar el objeto

Descripción de la solución

1. **Carga de archivos**

El primer paso en la implementación de la solución fue hacer la carga de las imágenes. Para esto cree una carpeta dentro del proyecto “ImagesQ” en donde se encuentran las tres imágenes de los objetos a clasificar. Guarde el path en la variable *path*, después guarde lo que había en ese path en la variable de *myList*.

Texto

Descripción generada automáticamente

La variable de *images* es para guardar el nombre completo de los archivos, y *classNames* es únicamente el nombre del archivo sin el tipo de archivo que sea.

Texto

Descripción generada automáticamente

Posteriormente, en un ciclo guardamos cada una de las imágenes, en la variable *images* y su nombre en *classNames*.

1. **ORB**

El primer paso es crear el descriptor ORB, esto se hace simplemente en una línea de código.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Utilizando la función *findDes* se usará el descriptor ORB para hallar los *features* de las imágenes, para esto recibe como parámetro el arreglo imágenes y dentro de un ciclo, se hace uso de la función de ORB *detectAndCompute* que encontrara los *keypoints* de cada imagen y las apilara en una lista para retornarla al terminar el ciclo.

Texto

Descripción generada automáticamente

El funcionamiento de *detectAndCompute,* detecta puntos clave y calcula los descriptores de acuerdo con el detector que se eligió.

Parámetros:

* Img, Matriz o imagen de donde se extraerán los descriptores

Salidas:

* Key Points: Los puntos clave detectados.
* Descriptors: Descriptores calculados. Salida de vectores concatenados de descriptores.

La siguiente función es *findIdORB* esta es para encontrar la imagen, que se le parezca más o tenga mas *keypoints* similares. Si el tamaño de la lista de *matches* es mayor que un parámetro de control llamado *thres* se retorna el índice de la imagen dentro del arreglo *matchList*.

Texto

Descripción generada automáticamente

1. **SIFT**

Se utilizará una función similar a FindDes, solo que en cambio de usar el descriptor de ORB, usamos el de SIFT, pero hacen lo mismo.

Texto

Descripción generada automáticamente

La siguiente función es similar a la de findIdORB, solo cambia que el descriptor es el de SIFT, pero hacen exactamente lo mismo de encontrar los keypoints y clasificar las imágenes.

1. **HOG**

Se utilizará otra función para hacer uso del descriptor de HOG, la función HOGDescriptor recibe como parámetros:

* \_winSize establece el winSize con el valor dado.
* \_blockSize establece el tamaño del bloque con el valor dado.
* \_blockStride establece el valor de blockStride.
* \_cellSize establece el tamaño de la celda con el valor dado.
* \_nbins establece nbins con el valor dado.
* \_derivAperture establece derivAperture con el valor dado.
* \_winSigma establece el winSigma con el valor dado.
* \_histogramNormType establece el tipo de norma del histograma con el valor dado.
* L2HysThreshold establece el valor de L2HysThreshold.
* \_gammaCorrection establece gammaCorrection con el valor dado.
* \_nlevels establece nlevels con el valor dado.
* \_signedGradient establece el valor de signedGradient.

Texto

Descripción generada automáticamente

Y por último esta la función findIdHOG, que encuentra la imagen que se parezca a la que se muestra en cámara, con la misma función de HOGDescriptor.

Descripción de los resultados

Las imágenes utilizados fueron las siguientes:

Un dibujo de una persona

Descripción generada automáticamente con confianza mediaTexto

Descripción generada automáticamente con confianza media

Cuadro de mi casa

Tarjeta universidad

Citybus

Citybus: buenos resultados con SIFT y ORB. Ningún resultado con HOG.

Tarjeta universidad: buenos resultados con SIFT, a veces no la reconocía ORB cuando la rotaba. Ningún resultado con HOG.

Cuadro de mi casa: buenos resultados con SIFT y ORB. Ningún resultado con HOG.

Para más información consultar la carpeta de anexos.

Discusión

Los descriptores que mejores resultados me dieron fueron el SIFT y ORB, con ciertas diferencias ya que el ORB tarda en encontrar el objeto que estoy mostrando y a veces se confunde cuando le muestro otro objeto similar pero el que más falsos positivos me dio fue el SIFT que incluso sin tener los objetos frente a la cámara me decía que estaba ahí.

El descriptor de HOG no me dio ningún resultado al usarlo, se queda congelada la pantalla y no sucede nada.

Conclusión

Me pareció interesante la forma de implementar descriptores y me entretuve bastante al intentar que este reconociera el objeto. Como se pudo notar en la discusión SIFT y ORB son bastante similares con sus ventajas respectivas, en lo personal me decantaría por ORB como el mejor. Ya que la implementación de HOG es cuanto menos molesta y al ser tan diferente de los otros, no hay tanta información en internet.