Segundo Examen Parcial

Visión Computacional

Quistian Navarro Juan Luis

A341807@alumnos.uaslp.mx

1. **Planteamiento del problema**

   El objetivo de este proyecto es desarrollar un identificador de objetos utilizando descriptores de imágenes disponibles en la biblioteca OpenCV. Los objetos seleccionados para esta tarea son tres artículos cotidianos que se pueden encontrar en casa. Se busca identificar estos objetos en una secuencia de video o en imágenes presentadas.

2. **Descripción de la solución**

   **a. Descriptores Seleccionados**

   1. **BRISK (Binary Robust Invariant Scalable Keypoints)**: Este descriptor es rápido y efectivo para detectar y describir características clave en imágenes. Es particularmente útil en situaciones donde se requiere una alta velocidad de procesamiento, como en aplicaciones de video en tiempo real.
   2. **SURF (Speeded Up Robust Features)**: Un descriptor que proporciona invariancia ante escalas y rotaciones, además de ser robusto frente a cambios en la iluminación. Aunque es más lento que BRISK, ofrece descripciones de características más ricas.
   3. **HOG (Histogram of Oriented Gradients)**: Este método es comúnmente utilizado para la detección de objetos y se basa en la distribución de gradientes en una imagen. HOG es especialmente útil para identificar patrones y formas, como las de las personas o vehículos.

      Comparado con BRISK Y SURF. Hog no regresa keypoints. En lugar de eso, examina la **estructura global** de la imagen, calculando los gradientes en pequeñas celdas y acumulando la orientación de los bordes en un histograma. Esto lo convierte en un descriptor excelente para **detectar formas y patrones globales**, como la forma de un cuerpo humano o un automóvil. Sin embargo, al no tener puntos clave específicos, **no es posible comparar o hacer coincidencias entre regiones locales** de dos imágenes (lo que se llama *matching* en BRISK y SURF).

   **b. Dataset**

   Como objetos, decidí tomar 7 en lugar de 3, para ver como se comportaban los descriptores con diferentes objetos de distintas texturas y colores:

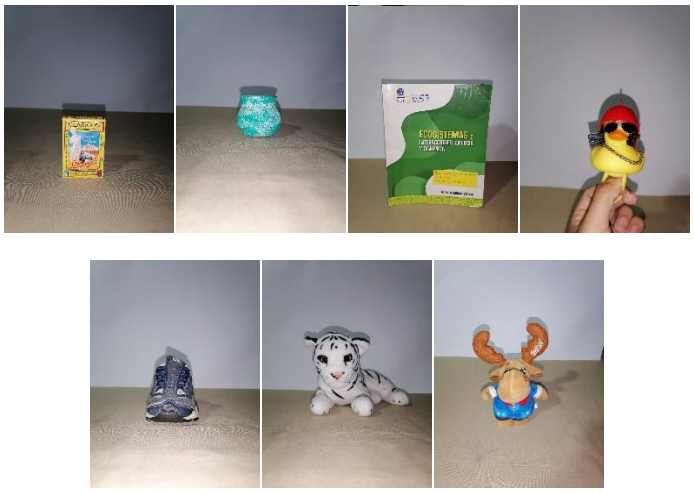

*Ilustración 1 Dataset (Cerillos, Jarro, Libro, Pato, Tenis, Tigre, Venado). De cada clase se tomaron 3 imágenes en diferentes poses.*

**c. Implementación**

La implementación se realizó utilizando Python y la biblioteca OpenCV. A continuación, se describen las funciones principales y su propósito:

**load_images(global_path, resize=None)**: Carga imágenes de las clases seleccionadas desde una ruta especificada y, si es necesario, las redimensiona.

**load_descriptor(descriptor)**: Carga el descriptor especificado. Si se elige HOG, se configura según los parámetros necesarios.

**find_descriptors(images_by_class, descriptor, descriptor_name)**: Esta función calcula los descriptores para las imágenes cargadas.

**train_svm(hog_features, labels)**: Entrena un modelo SVM (Máquina de Vectores de Soporte) utilizando las características HOG extraídas.
**augment_image(img):** Hacemos aumentación de datos.
**find_ID(img, desList, descriptor, class_names, threshold, svm_model, scaler, descriptor_name)**: Identifica el objeto en la imagen de entrada utilizando el descriptor seleccionado y devuelve el nombre de la clase identificada. Ya que se tienen varias imágenes por clase, lo que se hace es que, para cada conjunto de descriptores de una imagen de la clase, se calcula el número de "buenos" matches usando el **ratio test**.
La variable class_match_counts acumula la cantidad de buenos matches para cada imagen dentro de la clase, y de aquí se toma el máximo valor de coincidencias de "buenos" matches para cada clase.

Este enfoque significa que, para cada clase, se selecciona el número máximo de buenos matches que se logró con alguna de sus imágenes.

¿Por qué se hizo así? Bueno porque note que para las imágenes que son objetos un tanto lisos o sin tanta textura se obtenían pocos matches y colocándolos en varias poses logre que pasara por ejemplo en la clase pato de 3 matches a 6.

**d. Ejecución del Código**

El código se ejecuta tomando imágenes de la cámara en tiempo real, donde el objeto es identificado y se muestra el nombre del objeto en la pantalla. Se utilizan funciones de OpenCV para capturar el video y procesar cada fotograma.

## 3. Descripción de los resultados

**Capturas de los resultados.**

**---Brisk**


*Ilustración 2 Resultados para Brisk.*

**--SURF**


*Ilustración 3 Resultados para SURF.*

**--HOG**

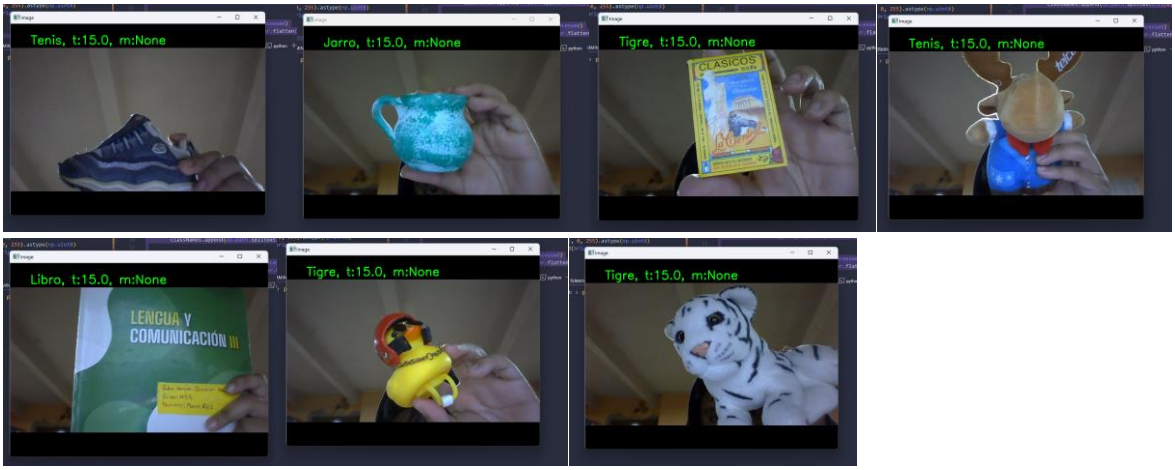


*Ilustración 4 Resultados para HOG.*

**T =** Threshold, el numero de matches para considerar que se detectó el objeto.

**M =** Matches, el numero total de matches detectado.

| Desc. | Cerrillos | Jarro | Libro | Pato | Tenis | Tigre | Venado | Promedio |
|---|---|---|---|---|---|---|---|---|
| **Brisk** | T = 4<br>M = 45 | T = 4<br>M = 19 | T = 4<br>M = 12 | T = 4<br>M = 6 | T = 4.0<br>M 13 | T=4<br>M = 8 | T= 4<br>M = 10 | T = 4<br>M = 16.14 |
| **Surf** | T= 25<br>M = 178 | T= 15<br>M = 25 | T=25<br>M =77 | T=25<br>M =16 | T=25<br>M =31 | T=25<br>M =26 | T=25<br>M =27 | T = 25<br>M = 54.28 |
| **HOG** | T=<br>M = | T=<br>M = | T=<br>M = | T=<br>M = | T=<br>M = | T=<br>M = | T=<br>M = | 0 |

**Para Hog**

implementamos svm usando las características como datos para clasificación.
El SVM obtuvo un 0.92 de precisión. Tuvimos que hacer aumentación de datos para lograrlo.
Pero como se puede observa en las ilustraciones 4, no logró detectar varias de las clases.

4. **Discusión**

   Los resultados indican que el descriptor SURF mostró el mejor desempeño general en la identificación de los objetos, seguido de BRISK y HOG. Esto puede deberse a la rapidez y eficacia de SURF para detectar características clave en entornos cambiantes y con distintas iluminaciones.

5. **Conclusión**

   El proceso de desarrollar un identificador de objetos utilizando descriptores de OpenCV ha sido muy educativo. A través de la implementación y comparación de BRISK, SURF y HOG, se ha adquirido una comprensión más profunda de cómo funcionan los descriptores y su aplicabilidad en problemas de visión computacional. OpenCV ha demostrado ser una herramienta poderosa y versátil para abordar problemas en este campo.