

Procesamiento de imágenes en un dispositivo móvil

Aranda Luciana, Carnovale Pablo, Perez Luis, Pichetti Gonzalo

¹Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina
aranda.luciana.f@gmail.com, pecarnovale@yahoo.com.ar, luisrperez2006@gmail.com,
gpichetti@gmail.com

Resumen. Esta investigación se basa en las implicaciones que puede tener para el dispositivo móvil el procesar imágenes dentro de la aplicación del Vaso Inteligente para la cual el usuario debería acceder a la cámara de su dispositivo móvil, capturar la imagen de una bebida y automáticamente conectarse a la API de Google mediante la cual se traera información relacionada a la bebida.

Palabras claves: <<Android, GPU, OpenCV, ORB, SURF, SIFT>>.

1 Introducción

A partir del acelerado avance de la tecnología, en el mercado se pueden encontrar diversos dispositivos los cuales cuentan con cámaras de última generación, mayor velocidad de internet y mayor poder de procesamiento.

Sin embargo a pesar de estos avances, los dispositivos tienen recursos limitados en cuanto a CPU, RAM y batería, con lo cual es importante tener en cuenta el uso eficiente de los recursos, sobre todo en un caso como el propuesto que conlleva un procesamiento importante a la hora de reconocer imágenes.

El propósito principal de la investigación es conocer la forma en que se puede paralelizar el trabajo y cual es la mejor forma de realizar esta tarea para afectar lo menos posible los recursos del dispositivo móvil al procesar imágenes.

Existen aplicaciones capaces de buscar objetos a través de la cámara como Google Goggles sin embargo una desventaja que posee es que no siempre retorna lo esperado. Otro ejemplo es la aplicación Similfy que permite, a hombres y mujeres, encontrar y comprar moda a partir de una foto.

Estos ejemplos son aplicaciones exclusivas para la búsqueda a través de imágenes, nuestra propuesta es una funcionalidad extra dentro de nuestra aplicación del Vaso Inteligente, con lo cual no será necesario que el usuario deba utilizar otra aplicación para realizar la búsqueda. Además, con nuestra aplicación se reducirá el espectro de búsqueda solo a bebidas para de esta forma poder obtener información relevante para el usuario.

2 Desarrollo

Para el procesamiento de imágenes se pueden utilizar diversos algoritmos que permiten detectar diferentes características. La forma en la que se procese tendrá diferentes consecuencias en la administración de recursos.

Para obtener la imagen a procesar en la aplicación propuesta, en primer lugar, se tomará una imagen de entrada desde la cámara, y a esta se le aplicará el algoritmo más óptimo para nuestras necesidades.

Debido a que la API de Google no permite búsqueda por imágenes, el usuario deberá introducir una palabra descriptiva, por ejemplo “Vodka”, “Cerveza”, “Vino” la cual acompañará a la imagen para de esta forma poder realizar la búsqueda en la API de Google.

El procesamiento de imágenes no aumenta la información que se puede extraer de ellas, solo realza ciertas características. Para procesar la imagen efectivamente se debe considerar el proceso de formación y las características de interés de la imagen.

Para la realización del Proyecto es necesario contar con las bibliotecas de visión artificial ya que brindan la posibilidad de realizar el análisis y clasificación de las imágenes. En este caso se propone utilizar OpenCV, que incluye soporte específico para Android.

OpenCV está diseñado para la eficiencia computacional con un fuerte enfoque en aplicaciones en tiempo real.

Existen varios algoritmos para el reconocimiento de imágenes, para este caso analizaremos los siguientes: SIFT, SURF y ORB.

El algoritmo SIFT transforma una imagen a un vector característico invariante a escalados, traslaciones, rotaciones y parcialmente invariante a cambios de iluminación y afinidades.

El algoritmo SURF es otro algoritmo de detección de características de imágenes. Es invariante a rotaciones, cambios de escala y cambios en la iluminación de las mismas. SURF propone una caracterización de las imágenes de manera más ágil pero menos robusta que SIFT.

ORB es el algoritmo más eficiente para procesar imágenes, utiliza los algoritmos FAST y BRIEF. Otra característica es que es gratis mientras que SIFT y SURF están patentados.

Al investigar las distintas características de los algoritmos se encontró que el tiempo de cómputo empleado por el descriptor SIFT es mayor que el tiempo empleado por el descriptor SURF y ORB. Esto conduce a determinar que el descriptor SIFT no es una buena opción para aplicaciones que realicen reconocimiento en tiempo real.

El descriptor ORB es una mejor alternativa para aplicaciones de tiempo real.

En cuanto al consumo de batería, el porcentaje de batería utilizado por cada descriptor para procesar una imagen es proporcional al tiempo de ejecución analizado anteriormente.

Con respecto a la cantidad de memoria que necesitan los descriptores para almacenar sus vectores de características, sabemos que tanto SIFT como ORB son los que menos uso de memoria tienen y que el vector de características generado por el descriptor SURF necesita un promedio de casi 8 veces el peso del vector generado por ORB.

3 Explicación del algoritmo.

El siguiente código ejemplifica el uso de ORB:

```
import numpy as np
import cv2
from matplotlib import pyplot as plt

img = cv2.imread('ojo.jpg',0)

# Inicializar detector STAR
orb = cv2.ORB_create()

# Encontrar puntos clave con ORB
kp = orb.detect(img,None)

# Computar los descriptores con ORB
kp, des = orb.compute(img, kp)

# Dibujar sólo los puntos clave de localizacion
img2 = cv2.drawKeypoints(img,kp,img, flags=0)
plt.imshow(img2),plt.show()
```

4 Pruebas que pueden realizarse

Las pruebas que podrían realizarse para verificar el correcto funcionamiento implican la captura de la imagen de una botella de cualquier bebida de la cual pueda obtenerse el nombre, a partir de esto se debería reconocer la imagen y obtener información relevante para el usuario.

5 Conclusiones

A modo de resumen, en esta investigación se propuso principalmente conocer las consecuencias de incorporar el procesamiento de imágenes a nuestra aplicación del Vaso Inteligente incorporando la posibilidad de encontrar información de la bebida que el usuario requiera a partir de la cámara del dispositivo móvil. Para lo cual es imprescindible utilizar la tecnología GPU.

Hemos comprobado que comparado con los descriptores SIFT y SURF, el descriptor ORB es el que tiene mejor efectividad en cuanto al uso de memoria tiempo de procesamiento y consume de batería.

A partir de esto creemos que se debe utilizar el descriptor ORB si se quiere optimizar el uso de recursos del dispositivo móvil.

Para futuros trabajos, se debe seguir investigando acerca de la optimización de recursos y también, sería una gran mejora incorporar a la búsqueda de información de la aplicación, una reducción del espectro de búsqueda a través del procesamiento de imágenes. Es sabido que muchas veces al utilizar estas búsquedas se obtiene información irrelevante respecto de las intenciones del usuario.

6 Referencias

1. Yu-Doo Kim, Jin-Tae Park, Il-Young Moon and Chang-Heon Oh: Performance Analysis of ORB Image Matching Based on Android. International Journal of Software Engineering and Its Applications. Vol.8, No.3 (2014)
2. Batuhan Hangün, Önder Eyecioglu: Performance Comparison Between OpenCV Built in CPU and GPU Functions on Image Processing Operations. International Journal of Engineering Science and Application. Hangun and Eyecioglu, Vol.1, No.2, 2017
3. K. B. Neelima, Dr. T. Saravanan: Image Detection and Count Using Open Computer Vision (Opencv). K. B. Neelima Int. Journal of Engineering Research and Applications, September 2014.
4. Miguel Nauñay Ilbay, Luis Tipantuña Córdova, Geovanny Raura, Tatiana Gualotuña: Análisis de eficiencia en algoritmos de reconocimiento de imágenes digitales aplicables a dispositivos móviles bajo la plataforma Android. (2013)