

Reconocimiento de bordes con OpenCV

Jonnathan Henry Campoberde Avila, Estudiante.
Carrera de Ingeniería en Sistemas, Facultad de Ingeniería
Universidad de Cuenca, Cuenca, Ecuador
jonnathan.campoberde@ucuenca.edu.ec

Resumen- En el presente documento se encuentra realizada la práctica respectiva al reconocimiento de borde en una imagen y video, en la misma se realiza un tratamiento a imágenes obteniendo los bordes para poder contar los objetos que se encuentran en la imagen. Para el análisis de los videos se tiene com primera parte la detección de los bordes de los objetos que contiene el video, dicho análisis es posteriormente utilizado para reconocer el movimiento de los objetos en un video. Por último se muestra los resultados obtenidos al realizar la práctica respectiva.

Palabras Clave: *OpenCV, video, imagen.*

I. INTRODUCCIÓN

OpenCV es una opción libre para realizar un análisis de imágenes y aprendizaje automático de las mismas. OpenCV es una biblioteca libre para la visión artificial, la misma es una iniciativa de la compañía Intel. La librería se caracteriza por ser una que se ha utilizado en una infinidad de aplicaciones, esto es debido a que con ella se puede capturar imágenes en tiempo real, importacion y exportacion de archivos de video, tratamiento de imágenes y la detección de objetos. Actualmente OpenCV tiene una implementación de 2500 algoritmos, la misma que está disponible de forma gratuita, además esta tiene interfaces las múltiples lenguajes entre ellos los. más populares como son Python, Java y C++. Al ser esta open source esta tiene una gran aceptación por toda la comunidad de desarrolladores, siendo esta una buena opción para crear un proyecto ya que la misma es gratuita para fines comerciales y académicos.[1][2]

II. OBJETIVOS

● Objetivo General

- Realizar un reconocimiento de los bordes de imágenes usando la librería OpenCV en el tratamiento de imágenes y videos.

● Objetivos Específicos

- Emplear OpenCV para el análisis de las imágenes y videos.
- Utilizar el lenguaje de programación python para utilizar la librería OpenCV.

III. METODOLOGÍA

A. Análisis de imágenes para contar objetos

Cargando una imagen para el análisis de sus bordes. En primer lugar es importante importar las librerías que serán necesarias para poder realizar tal proceso, las mismas que son la librería de OpenCV, numpy y matplotlib. Una vez realizado dicho proceso se procede a realizar la lectura de la imagen para cargarla en memoria y poder proceder a trabajar con dicho archivo.

Detección de bordes un proceso necesario para poder contar los objetos. El primer paso para poder realizar este procesos es transformar la imagen que deseamos utilizar a una escala de grises, para realizar tal proceso se usa la función “cvt.Color” obteniendo la imagen en escala de grises, en la figura 1 se muestra la imagen usada en la escala de grises. Una vez realizado el procesamiento previo se una función de OpencCV llamada “Canny”, para la cual se necesita dos valores umbrales que son pasados como parámetros, los mismos que son minVal y maxVal. De este modo, todos los bordes que sean superiores al valor especificado de maxVal se dibujarán. Los bordes que estén dentro de los rangos de maxVal y minVal se dibujarán solo si se encuentran conectados a los que superan el valor de maxVal. Por el contrario todos los bordes que se encuentren por debajo del valor de minVal no serán dibujados, esto se encuentra mejor visualizado en la figura 2, la misma que refleja lo anteriormente mencionado y fue obtenida de la documentación de OpenCV. Para en este caso se desea tener la mayor cantidad de borde por los que los valores son de minVal igual a 20 y en maxVal igual a 150. Una vez realizado dicho proceso se puede obtener los bordes que tienen los objetos de las imágenes como se muestra en el figura 3. [3]

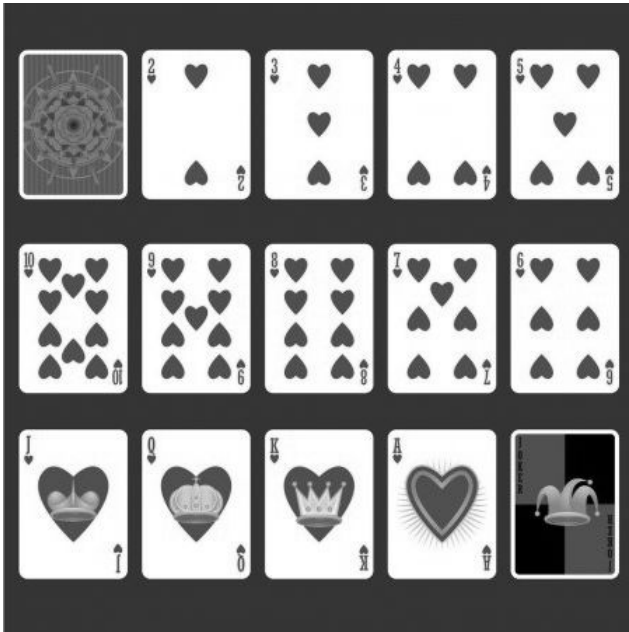


Fig. 1. Imagen usada para el procesamiento en escala de grises.

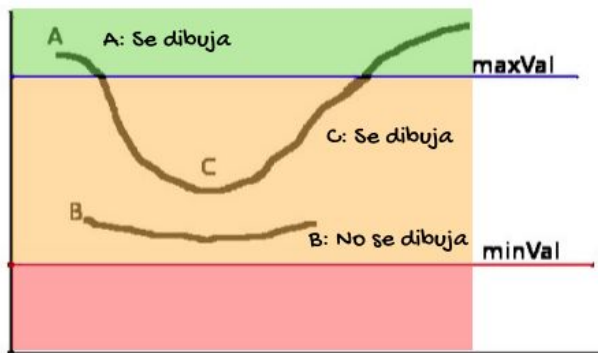


Fig. 2. Ejemplo de bordes que se dibujarán o no, según los valores de minVal y maxVal.

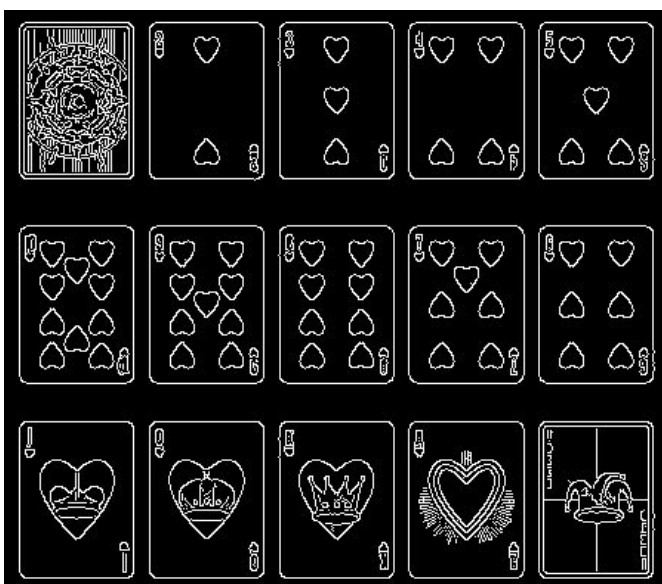


Fig. 3. Imagen procesada para poder obtener los bordes de los objetos de la imagen

Encontrar y dibujar los contornos de la imagen binaria. Después de obtener la imagen binaria que es la obtenida por la detección de los bordes, se puede aplicar la función “findContours” para poder encontrar los contornos de esta manera se pueden encontrar los contornos de las cartas de la imagen, una vez obtenidos los contornos se procede a dibujar los mismos con la función “drawContours”, gracias esta función se puede dibujar los contornos que son encontrados, en este caso se seleccionó como 2 pixeles en la imagen. Con estos pasos ya se pueden obtener los contornos dibujados en la imagen. [3]

B. Análisis de los bordes de videos

Cargando el video de un archivo para realizar el análisis de imágenes. Para poder realizar el análisis de imágenes se usa las librerías de OpenCV y NumPy, luego de cargar las librerías se procede a cargar el video que será utilizado para realizar el análisis. Después que se ha realizado la selección del video por medio de la función “read”, se procede a cargar el mismo usando otra función llamada “VideoCapture”. Una vez realizado se tiene el archivo que será utilizado para todos los otros procesos.

Características para definir los bordes en una imagen. Los bordes en las imágenes digitales se las puede definir como las transiciones que se dan entre dos regiones de niveles de gris significativamente distintos, los mismos proporcionan una valiosa información sobre las fronteras de los objetos los cuales se pueden utilizar para segmentar las imágenes, el reconocimiento de imágenes. Esta característica es de mucha importancia para poder detectar los bordes de una imagen.

La mayoría de técnicas para la detección de bordes emplean operadores locales basados en la primer y segunda derivada. Las herramienta matemática como los es la derivada proporciona información sumamente útil para la detección de los bordes, en la señal continua de una imagen se puede notar fácilmente como cambia la intensidad del píxel cambia de una manera significativa. Para ello se puede usar la primera derivada para poder saber los cambios de las intensidades de los píxeles donde el degradado es mayor que sus vecinos. Con la primer derivada se hallan los puntos donde cambia la intensidad pero con la segunda derivada se puede utilizar para calcular los bordes, debido a que las imágenes son en 2D el laplaciano seria tambien en dos dimensiones, en OpenCV se lo utiliza mediante la función “Laplacian”. De esta manera la herramienta matemática como es la derivada es de gran ayuda para poder detectar los bordes de las imágenes. [4]

El operador Sobel para detectar los bordes en las imágenes. El Operador Sobel es un operador de diferenciación discreta, el mismo calcula una aproximación del degradado de una función de intensidad de imagen. Para calcular los bordes el operador Sobel calcula los incrementos horizontales y verticales, por lo cual en cada punto de la

imagen se calcula una aproximación del degradado en ese punto combinando ambos resultados anteriores. De esta manera se puede calcular cuales son los bordes de un objeto en una imagen. [4]

C. Detección de movimiento en objetos de video

La detección de movimiento en la visión artificial. Existen muchas aplicaciones que están usando visión artificial, la cual principalmente se usa para la detección de los objetos, esto es debido a los avances que se ha tenido en la misma. El análisis y reconocimiento de movimientos a partir de sistemas basados en conocimiento(KBS), incluyen procesos que ayudan a la reducción del ruido, los mismos que son utilizados para poder resaltar las características y eliminar los posibles desenfoques que se puedan dar en las imágenes de un video. De esta manera cada vez se usan dichas herramientas de inteligencia artificial para la detección de los objetos en las diferentes aplicaciones de los campos de la ciencia. [5]

La sustracción de fondo una técnica muy usada para la detección de objetos. La sustracción de fondos consiste en tomar una imagen de alguna escena en movimiento para luego restar los sucesivos fotogramas que se tienen en el video. A la imagen que no se mueve es decir permanece estática se llama fondo o segundo plano, por lo que lo que deseamos analizar sería el primer plano, el resultado de realizar tal proceso en una imagen donde se detecta el movimiento. Esta técnica es muy sencilla debido a que es bastante sensible a los cambios de iluminación como las sombras o los cambios producidos por la luz natural.[6]

IV. RESULTADOS OBTENIDOS

A. Análisis de imágenes para contar objetos

Luego de aplicar el proceso menciona en la metodología se puede el número de objetos presentes en la imagen. Para poder enumerar a cada uno de los objetos que son encontrados en la imagen, en este caso cartas podemos hacerlo por medio de una variable, en este caso se llama “conts” que se obtiene de la función “findContours”. Después de realizar los pasos anteriores podemos imprimir la imagen resultante con los datos de los bordes y el número de objetos que han sido detectados en la imagen como se muestra en la figura 4.

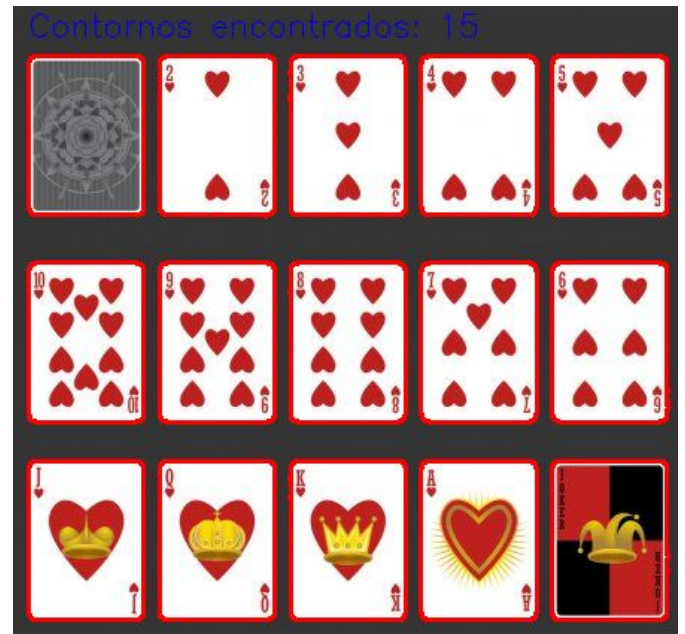


Fig. 4. Resultado final del procesamiento de la imagen.

B. Análisis de los bordes de videos

Comparación de los algoritmos de Sobel y Laplacian para calcular los bordes en videos. En la biblioteca OpenCV se tiene integrado las dos funciones por lo que no se tiene que desarrollar el algoritmo para realizar tal proceso, simplemente es invocar las funciones pertinentes para realizar el proceso deseado. En la figura 6 se muestra el algoritmo del operador de Sobel a una escala de rojo y azul para que se pueda lograr una mejor distinción de los mismos. En la figura 7 se muestra la implementación de la función “Laplacian”, esta función también está en la escala de rojo y azul para lograr su distinción de los bordes. Los dos métodos permiten encontrar los bordes en los videos, pero los resultados en la comparación muestra que el algoritmo Laplacian es mejor al mostrar todos los bordes, por lo que se puede considerar que dicho algoritmo es mejor que el algoritmo Sobel.

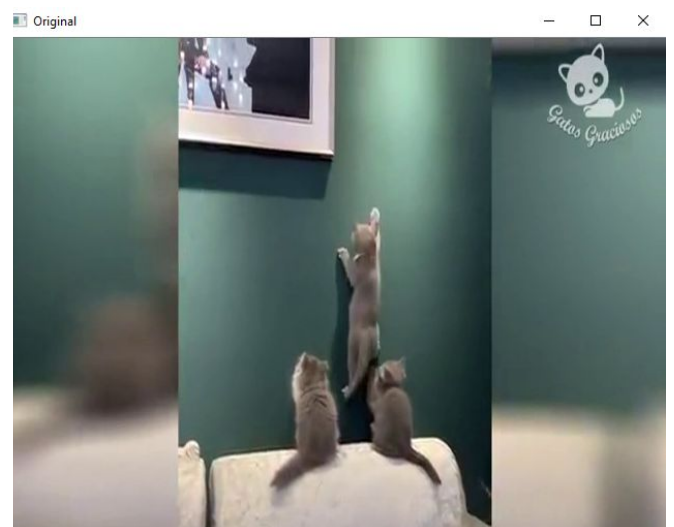


Fig. 5. Imagen original del video procesado.

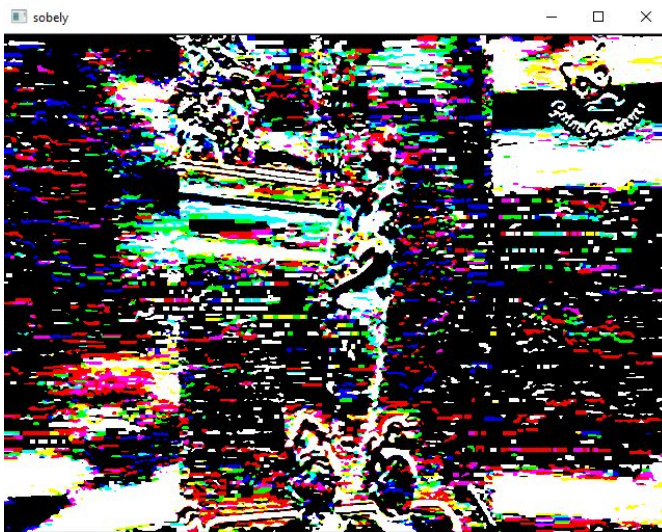


Fig. 6. Imagen del video procesado usando el algoritmo Sobel para obtener los bordes de los objetos de la imagen

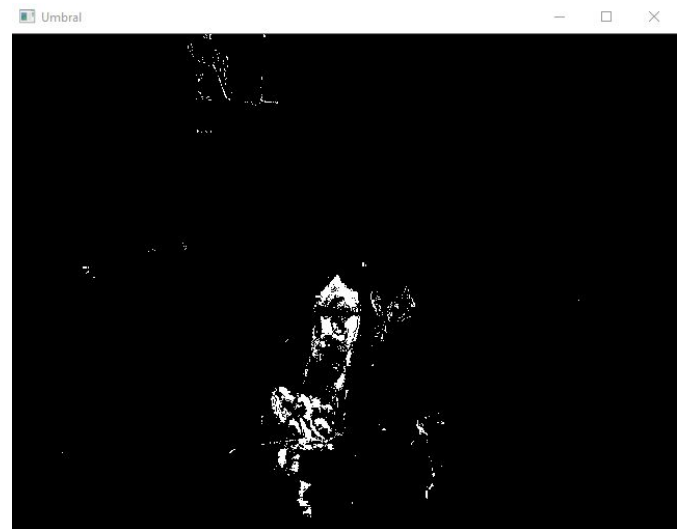


Fig. 8. Imagen del video procesado obteniendo el umbral del mismo.



Fig. 7. Imagen del video procesado usando el algoritmo Laplacian para obtener los bordes de los objetos de la imagen

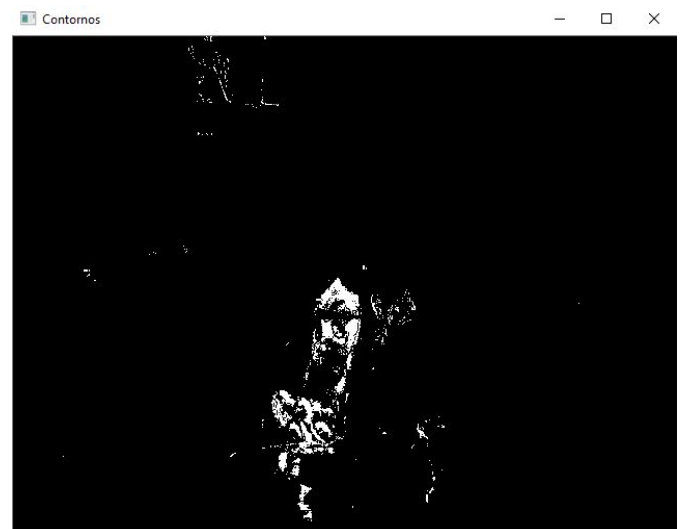


Fig. 9. Imagen del video procesado obteniendo los contornos de los objetos en movimiento.

C. Detección de movimiento en objetos de video

Detección de movimiento en los objetos. Una de las primeras fases para poder realizar la detección de los objetos es convertir los mismos a escala de grises y la eliminación de ruido, después de realizar tal proceso es la sustracción entre el primer y segundo plano. Al finalizar el proceso anterior se aplica el umbral a la imagen resultado de la resta del primer plano con el segundo como se muestra en la figura 8, para finalizar con la detección de los contornos como se muestra en la figura 9. El proceso anteriormente mencionado es muy común en la visión artificial por computadora. Usando la biblioteca OpenCV este proceso es sumamente sencillo y el resultado final de dicho proceso se muestra en la figura 10.

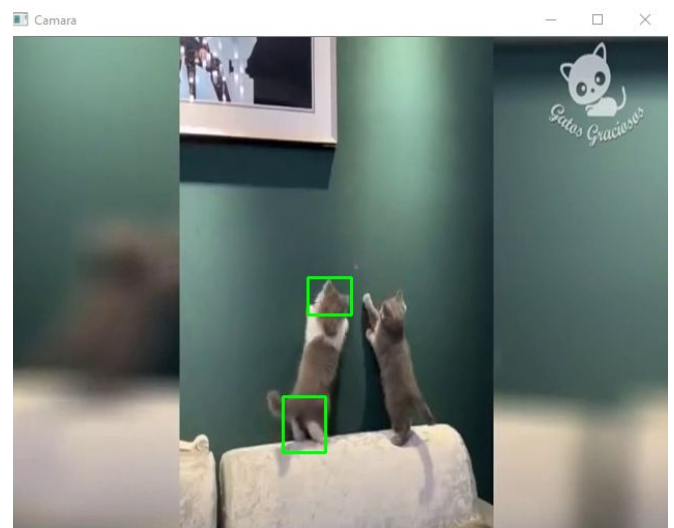


Fig. 10. Imagen del video procesado para obtener los cuadros donde se mueve los objetos.

V. CONCLUSIONES

La librería OpenCV es de gran utilidad para el tratamiento de imágenes. La librería OpenCV se caracteriza por tener una gran cantidad de funciones por lo que esto facilita el uso el análisis de las imágenes y videos de una manera sencilla.

La librería contiene funciones con algoritmos que son de gran utilidad los mismos que al ser implementados en ya no es necesario volverlos a programarlos, esto ayuda mucho al programador debido a que el solo utiliza los algoritmos preestablecidos y esto permite al programador centrarse más en la lógica del proyecto..

VI. REFERENCIAS

- [1] Robologs, «Robologs,» [En línea]. Available: <https://robologs.net/tutoriales/tutoriales-opencv/>. [Último acceso: 31 Marzo 2020].
- [2] R. Marín, «Revista Digital,» 12 Febrero 2020. [En línea]. Available: <https://revistadigital.inesem.es/informatica-y-tics/opencv/>. [Último acceso: 31 Marzo 2020].
- [3] G. Solano, «Omes,» [En línea]. Available: <https://omes-va.com/contando-objetos-aplicando-deteccion-de-borde-s-con-canny-en-python-opencv/>. [Último acceso: 7 Abril 2020].
- [4] U. Sinha, «AI Shack,» [En línea]. Available: <https://www.aishack.in/tutorials/sobel-laplacian-edge-detectors/>. [Último acceso: 7 Abril 2020].
- [5] J. F. A. D. John J. Sanabria S, «Detección y análisis de movimiento usando visión artificial,» 2011. [En línea]. Available: <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/1513>. [Último acceso: 7 Abril 2020].
- [6] L. d. V. Hernández, «Programación fácil,» [En línea]. Available: <https://programarfácil.com/blog/vision-artificial/deteccion-de-movimiento-con-opencv-python/>. [Último acceso: 7 Abril 2020].