

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA

INGENIERÍA EN SISTEMAS COMPUTACIONALES

INTELIGENCIA ARTIFICIAL

APLICACIÓN DE RECONOCIMIENTO DE FORMAS MEDIANTE VISIÓN ARTIFICIAL

REPORTE

FACILITADOR: IDARH C. MATADAMAS ORTIZ

ALUMNO: OMAR EDWING VÁSQUEZ ALONSO

NO. CONTROL: 12161417

OAXACA DE JUÁREZ, OAXACA A 5 DE JUNIO DE 2017

INTRODUCCIÓN

Las aplicaciones de la visión artificial están presentes en la vida cotidiana desde hace algunos años, primero en el contexto militar, luego en el industrial y empresarial, mas recientemente se ha extendido hacia el ámbito doméstico, con un sin número de usos de los mas diversos.

La visión artificial es una de las ramas de la inteligencia artificial, que se vale de diversos procedimientos y algoritmos para diferenciar formas, tomando en cuenta diversos parámetros. Por lo tanto es importante comprender como funcionan estas aplicaciones y los procesos que siguen desde la captura de la imagen hasta la clasificación de la misma.

Por esta razón, como parte de las actividades contempladas en el programa de inteligencia artificial, se ha construido una aplicación de visión artificial que realiza un reconocimiento y clasificación básico de formas geométricas. Con la finalidad de que en un segundo momento pueda implementarse como parte de una aplicación con fines didácticos.

JUSTIFICACIÓN

Para la realización de esta aplicación se utilizan los conocimientos adquiridos en la asignatura de inteligencia artificial, ya que se sigue el procedimiento convencional de las aplicaciones de visión artificial, el cual consiste en tratar la imagen como una matriz, que se convierte a escala de grises y posteriormente, después de haber fijado un umbral, se binariza y se aplican algunos algoritmos para determinar el área y perímetro de la misma.

Una vez llegado a ese punto se calcula el factor de compacidad y se procede a entrenar un clasificador, en este caso se utilizó un clasificador por proximidad, también llamado clasificador euclidiano.

Es este paso final el que permite la discriminación por parte de la aplicación, para diferenciar entre una figura y otra, de este proceso depende la efectividad de la misma.

DESCRIPCIÓN DE LA APLICACIÓN

PLATAFORMA

Se decidió utilizar android, a partir del api 25 (versión 5.0), para facilitar el acceso a la cámara y la movilidad de la aplicación. La versión se eligió por que actualmente da cobertura al 90 % de los dispositivos en funcionamiento.

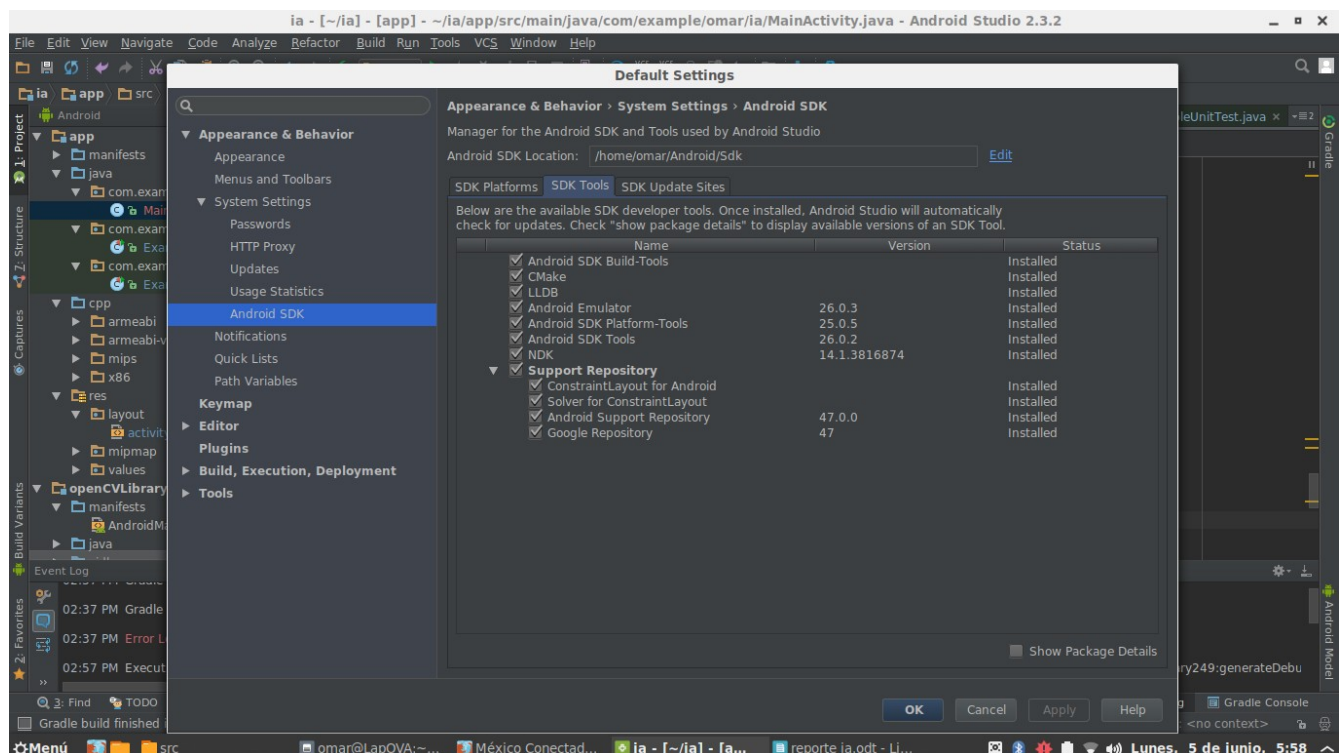
LENGUAJE DE PROGRAMACIÓN, HERRAMIENTAS DE COMPILACIÓN Y APIS UTILIZADAS

Debido a la plataforma utilizada, se utilizó java como lenguaje de programación. Aunque también se ocupó la librería OpenCV para la recuperación de la imagen. A pesar de la potencia de esta librería, se decidió implementar el resto del proceso a mano para una mejor comprensión del funcionamiento.

Se hizo uso de NDK para la compilación de código nativo de C/C++ sobre todo para hacer uso de la biblioteca OpenCV.

También se hizo necesario incorporar el paquete LLDB y Cmake para poder construir la aplicación.

Todos estos módulos se acoplaron para trabajar desde el IDE Android Studio.

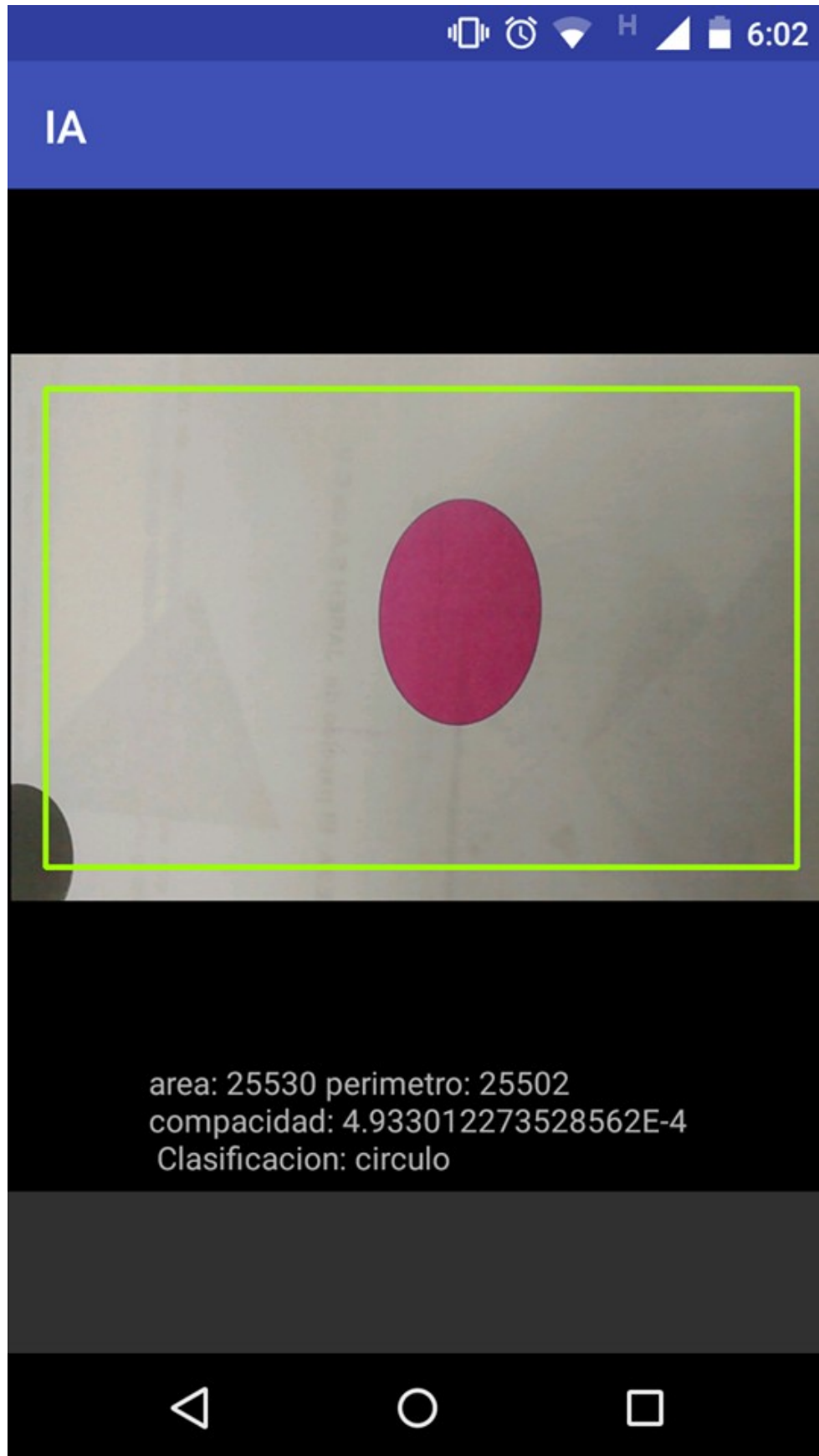


PROCESAMIENTO

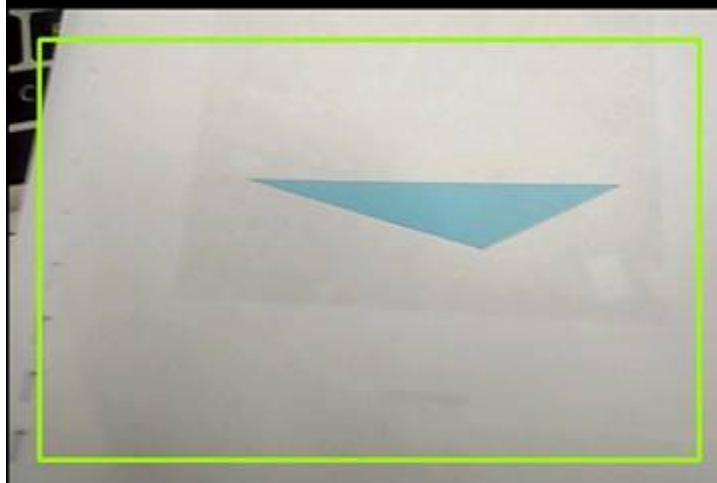
El procedimiento fue el siguiente:

- 1.- Recibir un flujo de video mediante la librería OpenCV, a partir de ahí se recuperarán imágenes en cuatro canales (RGBA).
- 2.- Se guardan los datos RGBA en una matriz y se separan, conservando únicamente RGB.
- 3.- Una vez que se tienen los datos RGB, se promedian para obtener una imagen en escala de grises.
- 4.- Se determina un umbral que permitirá considerar como blanco o negro aquello que se encuentre por debajo o por encima de dicho umbral. De esta forma se obtiene una imagen en blanco y negro, representada por una matriz binaria de ceros y unos.
- 5.- Una vez que se tiene la matriz binaria se calcula el área de la figura mediante el conteo de los píxeles puestos a uno.
- 6.- Para el cálculo del perímetro se verifica si en la vecindad de cada píxel existe al menos un píxel puesto a cero, de ser así se considera perímetro y se agrega.
- 7.- Una vez obtenidos el área y el perímetro, se calcula el factor de compacidad por la fórmula de elongación: $\text{Compacidad} = 4\pi \cdot A / P^2$
- 8.- Se realizan pruebas, la mayor cantidad posible, con muestras de las diferentes clases de imágenes a reconocer, se recuperan los valores de compacidad de cada prueba y se promedian para obtener un valor de referencia con el cual se compararán las muestras que se tomen posteriormente.
- 9.- Una vez obtenidos los valores de referencia, se asocia cada uno de ellos con una categoría o clase dentro de las que fueron definidas.
- 10.- Ya en funcionamiento, la aplicación repetirá este procedimiento, desde los pasos 1 al 7, comparando el factor de compacidad con los valores de referencia establecidos. Optando por aquel que presente la menor diferencia.

Capturas:



IA



area: 1592 perimetro: 1592
compacidad: 0.007893448878366315
Clasificacion: triangulo

CONCLUSIONES:

Tomando en cuenta únicamente un factor: la compacidad, y clasificando dos o tres tipos de figuras, se pueden obtener resultados aceptables, pero se podría incrementar la complejidad de la aplicación tomando en cuenta otros factores, como el color, para poder distinguir una mayor cantidad de clases. Además, mientras más efectivas sean las muestras utilizadas para entrenar el clasificador, mejores serán los resultados y el rendimiento de la aplicación.

Otro factor en el que se podrían realizar mejoras es la forma de determinar el umbral, puesto que en este caso fue de forma estática, por lo que no se adecuaba a las condiciones de luminosidad al momento de recibir el flujo de video. Podría utilizarse el algoritmo de otsu, para determinar dinámicamente el umbral óptimo para cada condición y de esta forma obtener mejores resultados.