

Clasificación de Piñas mediante la Vision por computadora

Saul Romero Prado

Abstract—Este documento un análisis sobre la clasificación de frutas, mas en concreto de las piñas, tratando de encontrar una forma de clasificarlas usando la visión por computadora o visión artificial. Desde hace años la forma tradicional de clasificar las frutas ha sido de una manera manual, donde empleados realizan esta tarea, aquí se tratará de automatizar ese proceso es por eso por lo que aquí se estudia el cómo se podrían clasificar las piñas de una cosecha sin la necesidad del trabajo humano. En este documento se presenta el proceso realizado durante varias semanas para poder lograr la clasificación, aquí se presentan, cuales fueron los mayores retos que se enfrentaron en el proceso y como se solucionaron, así también que se logro y como es importante eso que se logró.

Index Terms—Visión por computadora, Inteligencia Artificial, Clasificación, Piña.



1 INTRODUCCIÓN

EN los últimos años, los sistemas de visión por computadora o visión artificial han venido en amplio crecimiento. Los sistemas de detección de frutas se han desarrollado principalmente para la recolección robótica de frutas. Sin embargo, esta tecnología puede adaptarse fácilmente a otras aplicaciones, como la supervisión del rendimiento en el árbol, la supervisión del estado de salud de los cultivos, la detección de enfermedades, la detección de la madurez y otras operaciones que requieren la visión como sensor. Algo que tienen en común todos los sistemas de recolección o reconocimiento de frutas es que se ocupan técnicas muy similares para detectar cada objeto, sin importar que tipo sea ese, ya de ahí cada sistema se enfoca en lo que quiere hacer con la información obtenida, esta información, puede ser el color, la forma o el tamaño, etc. En el caso que se abarcara en este artículo será la calificación de una fruta en concreto, la piña, una fruta que siempre ha presentado problemas a la visión artificial por su textura tan peculiar

además de su corona y otros aspectos que la hacen única pero complicada, aquí se clasificaran piñas en diferentes categorías, se analizara su color y tamaño, para determinar la condición de esta en la cosecha.

Como ya se menciono antes, para lograr esto de ocuparan diferentes técnicas para el análisis de frutas y verduras, tratando de calcular el entorno en el que se puede aplicar esto para así poder emularlo en las pruebas y tener una serie de resultados mas cercanos al entorno real. Así también planteamos las condiciones ideales que deben presentarse para que la detección sea más precisa y exacta.

En este trabajo se presenta una detección de piñas utilizando una función basada en un par características. Una simple característica no sirve para hacer las cuatro clasificaciones de piña que se tienen planeadas hacer. Todas las características ocupadas son medidas contra un umbral el cual nos ayuda a hacer la clasificación final. Gracias ocupar ese par de características de la piña la clasificación puede llevar de manera correcta siempre y cuando se respeten las condiciones ideales planteadas. El documento reporta desde trabajos similares previos a este hasta, pasando por todo el proceso y quedándonos con las conclusiones y los

- *Saul Romero que estudia Ingeniería Informática en la Facultad de Ingeniería Eléctrica y Electrónica de la Universidad Veracruzana, Boca del Río, Veracruz.
E-mail: rpsaul18@outlook.com*

resultados y que viene después de esto.

2 ESTADO DEL ARTE

Recientemente se han publicado gran variedad de artículos que hablan sobre la clasificación de frutas y verduras, en cada uno existen diferentes enfoques, cada uno trata de adaptar a su necesidad principal las formas, pero entre todos esos existen algunos los cuales comparten metodologías o presentan maneras interesantes de atacar la clasificación por medio de un algoritmo de inteligencia artificial ocupando la visión por computadora.

En artículo de 2017 titulado "Fruit image classifier based on artificial intelligence" se nos plantea una forma simple para representar una imagen y trabajar con ella de forma más factible. La técnica o soluciones básicamente ver la imagen como una matriz, esto se facilita con herramientas como opencv las cuales hacen ese trabajo por nosotros, que nos regresa una matriz la cual en cada posición $[i, j]$ nos regresa un arreglo con los tres valores de los canales BGR.

La forma anterior de leer o representar una imagen, no es la única, sin embargo, es la más utilizada actualmente, y es muy efectiva para poder trabajar no solo la clasificación de frutas si no para trabajar cualquier problema con visión artificial. Es por eso por lo que en la mayoría por no decir todos los artículos se enfocan en el trabajo sobre esta representación de las imágenes. En Fig. 1 se puede apreciar como se representa una imagen en su forma matricial, en la imagen se ve un solo numero, para BGR, en una tupla de tres valores, una por canal, pero en esencia es lo mismo, es el color y tono de un píxel representado por tres números.

En un artículo publicado por Jyoti A Kodagali en 2012 se presenta los desarrollos y la aplicación del análisis de imágenes y del sistema de visión por computadora en un sistema de detección automática de frutas. El objetivo de ese artículo es proporcionar una introducción en profundidad del sistema de visión artificial, sus componentes y los trabajos recientes sobre un sistema de clasificación automática de la fruta.

En un artículo bastante mas reciente publicado por M. Shamim Hossain en 2018 propone un método eficiente para la clasificación de frutas utilizando Deep Learning. El método está basado en un par de arquitecturas de Deep Learning diferentes. La primera es un modelo de luz propuesto de seis capas de redes neuronales convolucionales, mientras que la segunda es un modelo de Deep Learning pre entrenado de geometría visual de grupo.

Un artículo que del 2015 publicado por Bandit Suksawat en la página de la IEE que habla de la clasificación de la calidad de la piña mediante el procesamiento de imágenes nos presenta una herramienta para la clasificación de la calidad de las piñas de acuerdo con el peso y el tamaño estándar de la norma de alimentos agrícolas tailandesa. La herramienta desarrollada consta de componentes de hardware y un programa de software de clasificación. Esta investigación es bastante completa, la herramienta que desarrollan es bastante similar a la que se presenta en este documento, acá la clasificación es utilizando algunas características diferentes, empezando por que las normas base son tailandesas, eso hace que el acatar las normas de la región piñera de Veracruz se deba de enfocar de manera diferente.

La mayoría de los artículos sobre clasificación de frutas tratan del clasificar las frutas por que frutas son, sin embargo, muchas de las técnicas que usan para el análisis de sus imágenes pueden ser usadas para nuestra clasificación ya que la forma en como se analizan a imágenes sin importar la clasificación puede ser las mismas solo que deben ser enfocadas al objetivo planteado.

El artículo de Bandit Suksawat es el que en teoría se parece mas el objetivo de este documento, a pesar de que su clasificación es por algunas otras características al final también se clasifican la calidad o el estado de las piñas, sin embargo su método de Deep Learning si bien es más moderno y posiblemente mas efectivo, se considera que no es necesario para nuestro objetivo ya que para tratar de igualar esas condiciones y mejorar la capacidad de nuestro programa se establecen unas condiciones ide-

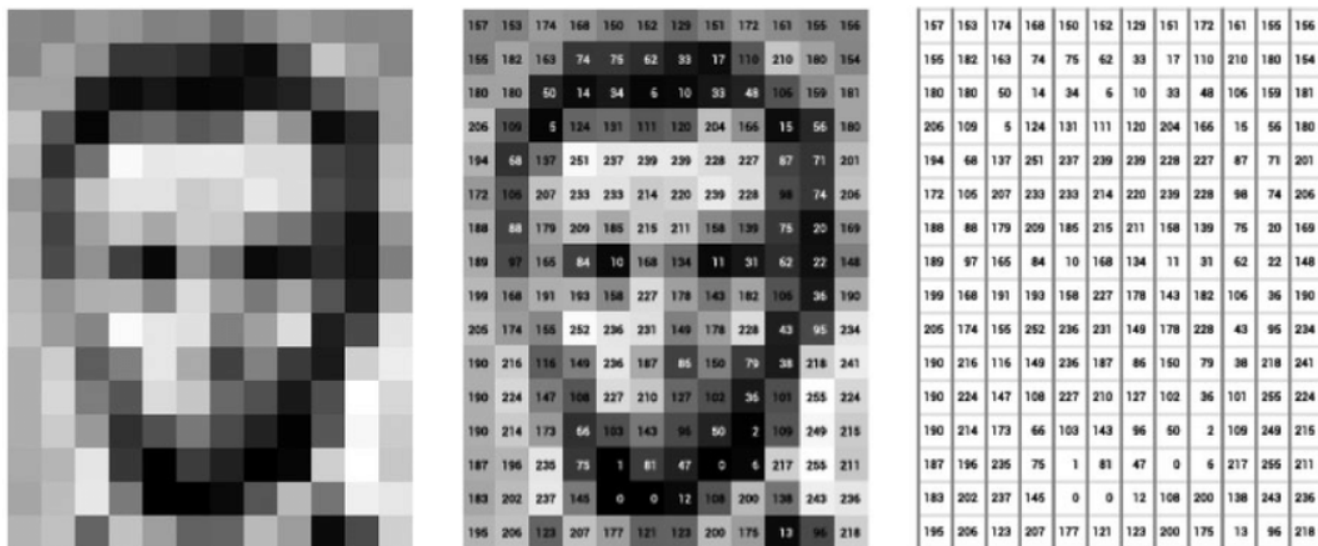


Fig. 1. Representación matricial de una imagen.

ales las cuales disminuyen el margen de error y hacen que métodos mas antiguos funcionen de manera perfecta.

Al final la clasificación por visión artificial es algo que esta muy presente hoy en día y que aplicarlo en una zona bastante rural pero gran productora de piña seria un gran paso en el desarrollo de los productores.

3 DISEÑO E IMPLEMENTACIÓN

Para poder implementar esta herramienta fue necesario el lenguaje de programación Python y la librería de visión artificial OpenCV, la cual es una de las mas famosas en esa rama ya que cuenta con una muy importante serie de funciones las cuales nos facilitan mucho el trabajo de análisis de la imagen y recientemente han implementado incluso una red neuronal, sin embargo, esta no será necesaria en esta ocasión. OpenCV es la librería mas importante usada pero también se ocupará una librería que funciona muy bien en combinación con OpenCV como lo es Numpy, la cual es un excelente complemento para trabajar con OpneCV. Por último, pero no menos importante también se importa el modulo ndimage de la librería scipy.

3.1 Detectar Piña

Evidentemente antes de todo lo primero que se debe hacer es obtener la imagen en su

representación matricial, eso se realiza con la función `imread` que nos da la librería `opencv`.

Ahora si ya con la imagen obtenida en lo que nos enfocamos fue en la mera detección de la piña tratando de eliminar el mínimo ruido que puede existir, pasando la imagen por una serie de filtros los cuales transforman la imagen lo necesario para que su análisis sea mas fácil.

Para empezar con análisis se pasa la imagen por un filtro el cual reduzca el ruido de la imagen. Existen varios métodos para eliminar el ruido de la imagen, `opencv` cuenta con dos muy famosos como lo son `MedianBlur` y `GaussianBlur`. Se hicieron varias pruebas, trataron de combinar ambos, mas sin embargo, los mejores resultados fueron obtenidos usando solo el `GaussianBlur` con una matriz kernel de `9x9` y con un valor de sigma de `0`.

```
gauss = cv2.GaussianBlur(image, (9, 9), 0)
```

Es importante realizar el paso anterior ya que después pasamos a usar una función la cual trabaja mejor con una imagen suavizada, estamos hablando de el proceso Canny el cual es un algoritmo desarrollado por John F. Canny en 1986 que utiliza un algoritmo de múltiples etapas para detectar una amplia gama de bordes en imágenes.

Nuevamente `opencv` nos provee una función útil para eso, pero para usar esta, previamente

```

18 gray = cv2.cvtColor(gauss, cv2.COLOR_BGR2GRAY)
19 canny = cv2.Canny(gray, 25, 110)

```

Fig. 2. Proceso Canny.

debemos convertir la imagen a escala de grises para que la función trabaje correctamente.

Canny hace bien su trabajo, sin embargo la textura de la piña es el primer problema con el que nos topamos, ya que debido a su porosidad los bordes que Canny detecta tienden a no cerrar, por ende no se puede capturar el área de la piña.

Para solucionar esto, se engordan los bordes mediante una función llamada Dilate, la cual agrandara los bordes para que se forme un solo objeto, sin embargo a veces quedan algunos huecos los cuales se llenan con una función de `spcipy.ndimage` la cual rellena lo huecos dentro de la figura.

```

22 ob = canny.copy()
23 fat_border = cv2.dilate(ob, numpy.ones((7, 7), numpy.uint8), iterations=2)
24
25 mask_fill = ndi.binary_fill_holes(fat_border).astype("uint8")

```

Fig. 3. Proceso de dilatación y sellado de bordes en código.

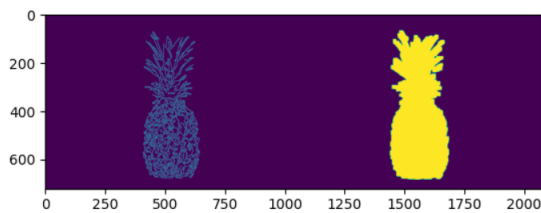


Fig. 4. Imagen antes y después del proceso de dilatación y sellado de bordes.

En Fig. 4 se puede apreciar la diferencia que tiene la imagen antes de y después de engordar los bordes y rellenar los huecos, podemos ver como solucionamos el problema de la porosidad de la piña, ya con esto tenemos en una imagen binaria la detección de la piña en cuerpo completo.

3.2 Eliminar la corona

La corona de la piña es otro de los problemas que se dan al querer procesar la imagen

esta, ya que su tamaño es muy variante y en ocasiones puede ser superior a el de la piña en si, lo cual genera que al momento de analizar sus características pueda haber afectaciones por parte de esta. A los productores solo les importa una cosa y eso es el cuerpo de la piña. Como se puede observar en la Fig. 4, la mascara binaria que obtenemos toma en cuenta la corona ya que al estar pegada al cuerpo la detecta como el mismo objeto.

El hecho de que este detectado como un mismo objeto en un problema bastante grande, ya que hay que enseñarle a la computadora a encontrar el punto de separación del cuerpo con la corona.

Analizando la imagen, se puede notar que ese punto de cambio es bastante mas angosto que las demás áreas de la piña, por ende nos enfocamos en encontrar las coordenadas de ese cuello.

Para encontrar las coordenadas primero vamos a ocupar una función la cual nos encuentra los contornos de la imagen y nos regresa las coordenadas de cada punto que conforma el contorno. Esta función se aplica sobre la mascara binaria, para que le sea fácil encontrar los contornos a rededor de la imagen ya que solo tiene 2 valores, 0 o 1.

Los recibimos finalmente son coordenadas de los puntos que rodean la piña, pero por si solos estos no nos dicen nada, aun debemos encontrar las dos coordenadas que forman el cuello.

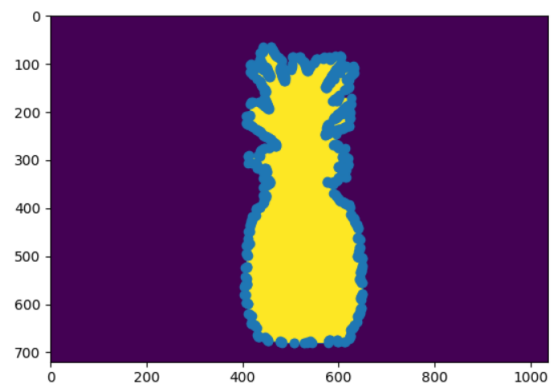


Fig. 5. Coordenadas de los contornos de la piña.

Para encontrar ese par de coordenadas va-

mos a necesitar seccionar la imagen, primero solo nos vamos a quedar con el área que ocupa a piña completa, recortamos la imagen a solo esa área y entonces ahí se aplica la segmentación. La segmentación consiste en lo siguiente:

- 1) Lo primero que vamos a hacer es quitar un porcentaje de la parte superior e inferior de la imagen. Esto para descartar los puntos de arriba y abajo que evidentemente no son el cuello y causaran problemas a futuro.
- 2) Después procederemos a dividir los puntos que nos quedan aun en dos partes, los de la derecha y los de la izquierda.
- 3) Ya con los puntos de la derecha e izquierda separados, vamos a ordenar ambos grupos de puntos.
- 4) Ya con los grupos de puntos ordenados vamos a saltarnos algunos ya que muchos están demasiado juntos y eso nos da muchos puntos y mas procesamiento. Al final se obtienen unos puntos con una separación mas considerable.
- 5) Cuando ya tenemos los puntos entonces empezamos a comparar las distancias entre los de la izquierda y los de la derecha.
- 6) El punto es evaluar y quedarnos por los pares que están mas cerca entre si.
- 7) Ya que tenemos los pares que están mas cerca entre si, nos vamos a quedar con el par que este mas hacia abajo en el eje de las "y".

No escoge al par mas cerca entre si por una razón, como se puede observar en la Fig.6 hay partes en la corona que pueden ser las mas cercanas entre si, entonces, por eso se escoge a un grupo y se selecciona la mas cercana al sur, porque como se ve, la parte del cuerpo de la piña es bastante ancha, entonces nunca entrara en el grupo de las mas cercanas, por tanto la mas al sur, es la del cuello.

Ya con esas coordenada podemos proceder a cortar la imagen tomando los valores en el eje "y".

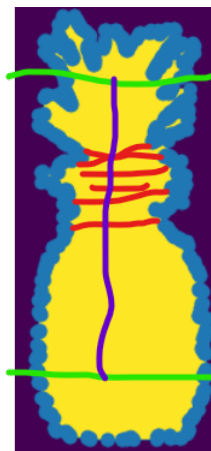


Fig. 6. Representación de la segmentación en la imagen.

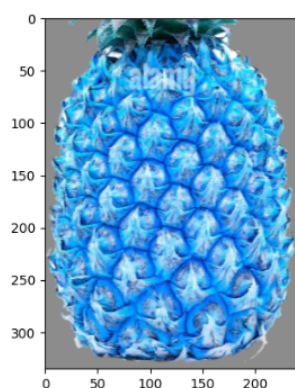


Fig. 7. Resultado de la piña después de la segmentación y la eliminación de su corona

En la Fig. 7 se puede apreciar el resultado después de ocupar las coordenadas obtenidas y se puede ver como se elimina casi por completo la corona de la piña.

Para estar seguro de tener el área mas precisa, se vuelven a calcular los contornos, ahora solo con el fragmento del cuerpo, para así poder recortar de la imagen el rectángulo mínimo que cubra la piña.

3.3 Obtener características

Hasta este punto ya contamos con la piña detectada y también logramos separar el cuerpo de la corona de la piña. Todo el trabajo hecho anteriormente tiene como objetivo facilitar la obtención de características. Así que ya con todo lo conseguido anteriormente podemos empezar a obtenerlas.

3.3.1 Color o Tono

La primera característica sobre la que se trabajó fue el tono o color, para poder diferenciar entre las verdes y las maduras. Para sacar las características del color se realizó lo siguiente:

- 1) Lo primero que se hace es utilizar la máscara en contra de imagen real, para poder pintar de color negro el fondo y que este se pueda distinguir.
- 2) La corporación es simple, por cada píxel negro de la máscara se vuelve negro el de la imagen original.
- 3) Ahora que la imagen ya distingue su fondo, calculamos con ayuda de `opencv` el histograma por canal de misma, obteniendo al final, tres histogramas.
- 4) Antes de seguir es importante que los histogramas obtenidos se normalicen para que al comparar sean a una escala de 0 a 1.
- 5) Ahora se calcula el promedio en Rojo, Verde y Azul.
- 6) Por último se calcula la desviación estándar del promedio de los tres tonos y ese es el valor de nuestra característica.

3.3.2 Tamaño

La segunda característica a obtener, esta mucho más simple, es el tamaño. El proceso que se sigue para obtener esta es tan simple como ocupar los contornos que cubren solo el cuerpo de la piña y calcular el área que estos abarcan. Aunque es simple hay varios factores que se deben cumplir para que esta funcione, eh ahí el porque de las condiciones ideales, ya que no debe haber variación de píxeles y todas las piñas deben ser capturadas por la misma cámara.

Otro punto importante es que solo se calcule el área del cuerpo, por lo que aquí ayuda mucho el proceso previo a la obtención de esta característica. Al final solo regresa el valor del área.

3.4 Procesar Características

Una vez que logramos obtener las características que nos representan las imágenes estas deben ser procesadas para su clasificación.

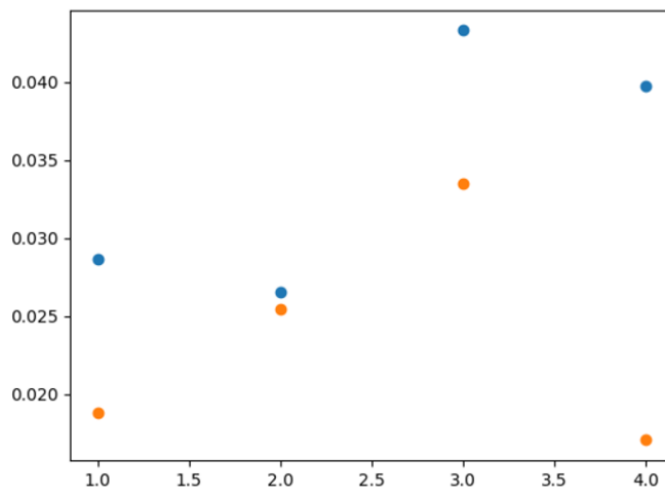


Fig. 8. Resultados de las pruebas por el canal Azul.

Aquí es donde se implementa un herramienta que trata de emular el funcionamiento real.

Se trata de una serie de imágenes las cuales fueron modificadas para emular el entorno, entonces cada imagen se procesa individualmente y los resultados obtenidos son comparados contra un umbral calculado previamente el cual determina cual es tu condición, en color y área, es entonces cuando la clasificación queda hecha y se muestra por pantalla la piña encerrada con un color dependiendo su clasificación.

4 RESULTADOS

El proceso de la obtención de las características es un proceso que parece simple, pero para llegar a obtenerlo se tuvieron que hacer muchas pruebas en las cuales hubo varios factores que complicaron el mismo.

Después de numerosos intentos y combinaciones con las diferentes herramientas que se tienen a la mano se llegó a un proceso el cual trabaja de muy buena manera.

Una vez que el proceso estaba listo había que probar el funcionamiento de este. Inicialmente no se iba analizar los tres canales, se hicieron pruebas usando cada canal por individual y parecía que la media del canal azul era suficiente para poder clasificar por color las piñas, sin embargo al agregar más piñas no se lograba encontrar una diferencia clara entre las verdes y maduras.

Como se puede observar en la Fig. 8 a pesar de que alguna si se paran entre si, algunas otras se mezclan de manera en que resulta imposible separarlas.

Asi como se muestran los resultados por el canal azul, se hicieron pruebas en los otros canales, pero estos entregaron resultados aun peor. Se pensaba que el canal verde podria ser la respuesta, sin embargo la obscuridad de algunas piñas causaba ruido y los resultados por ese canal no fueron los esperados. Es por eso que se siguieron haciendo pruebas hasta encontrar la forma mas precisa, no solo en el color si no en el área también.

4.1 Color

Ahora que ya se explicaron los problemas y las fases erróneas en el proceso de encontrar una forma de identificar la característica que nos dividiera las piñas por color, se presentan los resultados del método final para clasificar las piñas por el tono.

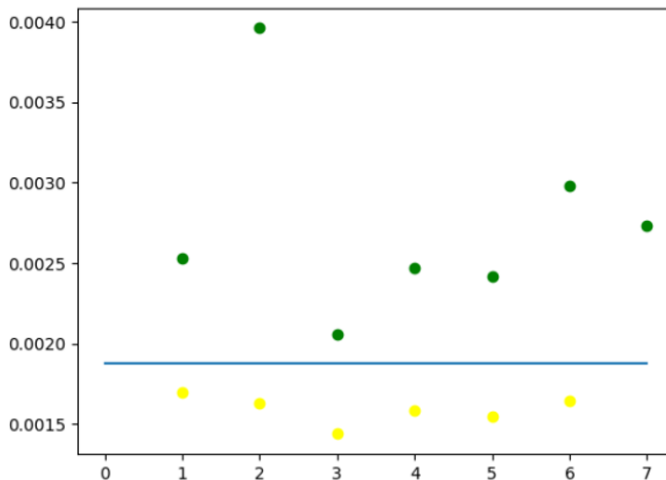


Fig. 9. Gráfica de resultados de la separación de las piñas verdes y maduras.

Para ver si la clasificación se estaba llevando correctamente, se analizaron por separado las imágenes de piñas verdes y maduras sin importar el tamaño. La Fig. 9 nos muestra una gráfica con los resultados de las piñas verdes y maduras, los puntos verdes representan las piñas verdes y como se puede observar estos tienen una desviación estándar mas alta a

las piñas amarillas, se puede ver también que las piñas amarillas están mas cerca entre si, y varían menos los resultados de unas y otras.

La apreciación visual es muy buena pero se necesitaba un valor real para la separación de esta, para eso se calculo un umbral el cual no es mas que el elemento mas que el promedio del valor mas bajo de los verdes con el mas bajo de los amarillos. La linea azul que se aprecia en Fig. 9 se encuentra posicionada en el valor obtenido del umbral el cual es **0.0018790620379149914**.

4.2 Área

El área es algo que siempre ha causado dificultades en la visión artificial, ya que la perspectiva, la distancia y el ángulo afectan mucho en el intento de calcular el tamaño de algo, sin embargo, en esta ocasión no fue así, ya que en el planteamiento del problema presentado se establece que se esperan las condiciones ideales para el análisis de las piñas, proponiendo que estas deben ser analizadas por la misma cámara y a la misma distancia. Para las pruebas se trato de emular el entorno usando una imagen que simula un fondo y pegando imágenes de piñas grandes y pequeñas a la misma escala.

Ya con el entorno emulado fue tan fácil como calcular el área de esto para saber que tan grande eran.

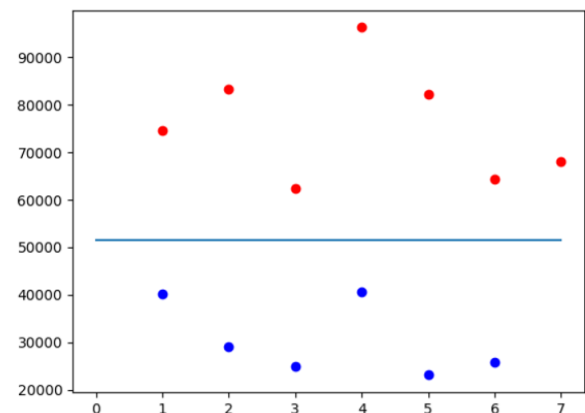


Fig. 10. Gráfica de resultados de la separación de las piñas grandes y pequeñas.

Como se puede ver en la Fig. 10 los resultados son bastante claros, al tener las condiciones ideales el calculo del área es bastante simple y


```

D:\Codigos\ProyectosPY\ProyectoTIA_Saul\venv\Script
=====Results=====

Mature - <----- 0 -----> + Green
Small - <----- 0 -----> + Big

FileName | Mature Proximity | Area Proximity |
MG1.png: | -0.00024587614461779594 | 22967.25 |
MG2.png: | -0.0002936733653768897 | 31817.25 |
MG3.png: | -0.0003288635052740574 | 10838.75 |
MP1.png: | -0.00018100801389664412 | -11422.25 |
MP2.png: | -0.0004401163896545768 | -22319.75 |
MP3.png: | -0.00023653160315006971 | -26501.25 |
VG1.png: | 0.0020816712640225887 | 44701.25 |
VG2.png: | 0.0005907637532800436 | 30665.25 |
VG3.png: | 0.0005396835040301085 | 12912.75 |
VG4.jpg: | 0.0008511838968843222 | 16511.75 |
VP1.png: | 0.0006547169759869576 | -10838.75 |
VP2.png: | 0.0001810078974813223 | -28384.25 |
VP3.jpg: | 0.0011042892001569271 | -25612.75 |

Process finished with exit code 0

```

Fig. 11. Resultados de efectividad de calculo.

eso se puede ver en los resultados los cuales son bastante claros, siendo los puntos rojos las piñas grandes y los puntos azules las pequeñas. Nuevamente para hacer la decisión se calculo un umbral de la misma forma que en el color, obteniendo como resultado **51540.75** el cual es la posición de la línea azul en el eje de las "y" y justo se puede ver como esta divide bien las piñas.

4.3 Efectividad

Antes de presentar los resultados de la herramienta final, es importante demostrar que tan efectivo puede ser el programa. Al no ocupar redes neuronales, no se ocupan archivos de entrenamiento, por lo cual no hay forma de sacar una matriz de confusión u otro reporte conocido para mostrar la efectividad. Al usar solo un umbral se decidió que la forma de representar la efectividad fuera mas simple. En la Fig. 11 tenemos una especie de tabla que intenta representar la efectividad de las funciones creadas. Lo que se realizo para medir

la efectividad fue leer las imágenes y calcular con las funciones su valor para la característica del tono y para la del área, a esos resultados se les calcula la diferencia que tienen sobre el umbral que los separa de la otra clasificación. Cada archivo lleva en su nombre que tipo es, esto evidentemente se hizo manualmente, por ejemplo el primero se llama MG1, eso significa que es madura y grande.

Ahora ¿como es que sabemos que tan exacto es?, bueno, pues empezaremos por la diferencia de color, como se ve en la parte superior si la diferencia en negativa entonces es madura y mientras mas bajo sea el valor, mejor representante de las maduras es, caso contrario con las verdes, las cuales tienen diferencia positiva y mientras mas alto el valor, mejor representan una piña verde.

En en apartado de grande y pequeña es mas de lo mismo, la parte positiva es para las grandes y la negativa para las chicas. En ambos casos mientras mas cerca un resultado esta del cero, son piñas que podrían ser confundidas, ya que no tienen las características tan claras, o mejor dicho están en un punto intermedio. Como podemos ver en los resultados de la Fig. 11 la primera dice MG1, lo cual significa que es grande y madura, esto se rectifica en los resultado, sonde vemos que su aproximado de madurez negativo lo cual nos indica que la desviación estándar de la media de los histogramas de los tres canales es baja y del otro lado sale positivo, lo cual nos dice que el área que ocupa en el entorno es lo suficiente para ser considerada grande. Y así vemos como todas corresponden a lo que su nombre demanda, algunas mas exactas y otras mas cercanas al 0 pero eh ahí el porque de analizar varias, para poder encontrar un umbral que las separe bien.

4.4 Herramienta Final

Por ultimo tenemos la herramienta final, esto el lo mas cercano a el uso de nuestro programa en un entorno real, justo aquí fue donde se puso a prueba todo lo antes analizado, después de error, pruebas y todo lo que conlleva el proyecto, este es el resultado, una herramienta la cual recibe imágenes y las muestra clasificadas por

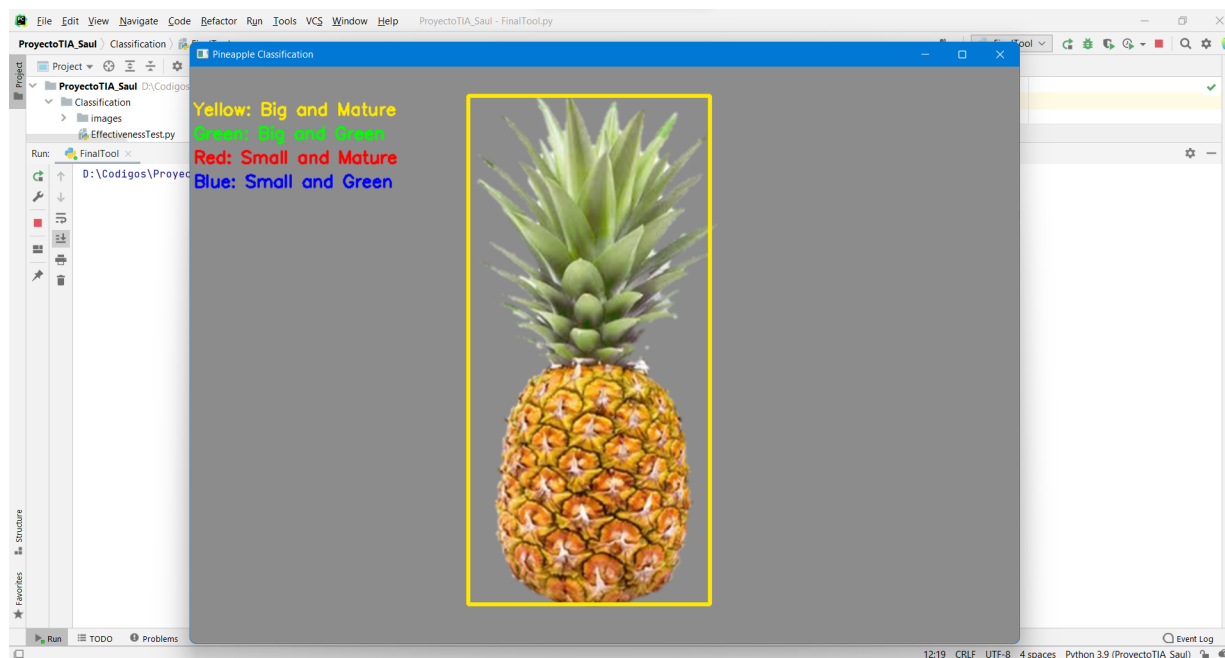


Fig. 12. Demostración de como se ve la herramienta final.

pantalla. Justo en la Fig. 12 se puede apreciar como la herramienta encierra de amarillo una piña grande y madura ya que ese es el color que se le designa a ese conjunto, y así como clasifico esa lo hace con las demás piñas. Obviamente que esto es una simulación, en un entorno real no solo se necesitara que se muestre la piña clasificada, si no que también nuestro programa deberá enviar esas señales a algún aparato mecánico que haga la clasificación con la respuesta enviada por nosotros.

Y es así como se obtuvieron resultados positivos en relación a lo esperado, ya que el programa clasifica las piñas de manera satisfactoria, siempre y cuando este en las condiciones ideales.

5 CONCLUSIONES Y TRABAJO FUTURO

El clasificador de imágenes de piñas se plantea principalmente en la extracción de características de la imagen usando la visión por computadora o visión artificial, sacando dos características que nos permiten clasificar estas en 4 grupos. Para estos se ayuda del lenguaje Python y algunas librerías de visión artificial que son de mucha ayuda.

En general este sistema aplicado aquí es uno el cual utiliza tecnología simple, las características de lo que se busco resolver permitieron establecer condiciones ideales que finalmente ayudaron a poder resolver el problema de una manera mas simple, sin embargo, simple y fácil son cosas muy distintas, ya que aun así se tuvieron que resolver ciertos problemas que se presentaron a la hora de analizar la imagen.

Durante la realización de esta herramienta se aprendieron muchas cosas, como algo que parece tan simple puede causar problemas, tal así el caso de la porosidad de la piña, el cual es fue un problema para analizar y también la corona, ya que esta por su tamaño puede afectar el análisis. La piña es una de las frutas más difíciles de analizar por su composición, aun así ningún problema presentado fue imposible de resolver ya que hay muchas herramientas de visión artificial que nos ayudan a resolver estos.

En cuanto a **que viene después**, es importante recalcar, que de momento el programa se queda en una fase de prueba hipotética, se espera que en un futuro se pueda llevar a un entorno real, y así poder adaptarla a las condi-

ciones ya que muy probablemente al probar en un entorno real necesitara ajustes en el umbral u algunos otros valores.

O también seria interesante probar en un entorno real porque pueden surgir nuevas ideas para mejorar esta herramienta y que sea aun mas efectiva.

REFERENCES

- [1] M. Shamim Hossain, Muneer Al-Hammadi, Ghulam Muhammad *Automatic Fruit Classification Using Deep Learning for Industrial Applications.* King Saud University, Riyadh, Saudi Arabia, 2019.
- [2] Bandit Suksawat, Preecha Komkum *Pineapple quality grading using image processing and fuzzy logic based on Thai Agriculture Standards.* Singapore, 2015.
- [3] Diego Heras *Fruit image classifier based on artificial intelligence.* Universidad Católica de Cuenca Cuenca, Ecuador, 2017.
- [4] R. Shantha Selva Kumari, V. Gomathy *Fruit Classification using Statistical Features in SVM Classifier.* Chennai, India, 2018.