

**Recepción:** 31 de julio de 2017

**Aceptación:** 18 de septiembre de 2017

**Publicación:** 29 de septiembre de 2017

# PROCESAMIENTO DE IMÁGENES DE PLANTAS ORNAMENTALES MULTI-ESCALA PARA CALCULAR SU CRECIMIENTO

## MULTI-SCALE ORNAMENTAL PLANTS IMAGE PROCESSING FOR GROWTH CALCULATION

Carlos Alberto Cásares Farías<sup>1</sup>

Nicandro Farías Mendoza<sup>2</sup>

Noel García Díaz<sup>2</sup>

Azael García Rebolledo<sup>2</sup>

1. Estudiante. Instituto Tecnológico de Colima. División de Estudios de Posgrado e Investigación. Maestría en Sistemas Computacionales. Colima, Colima (México). E-mail: [g1546006@itcolima.edu.mx](mailto:g1546006@itcolima.edu.mx)
2. Profesor. Instituto Tecnológico de Colima. División de Estudios de Posgrado e Investigación. Colima, Colima (México). E-mail: [nfarias@itcolima.edu.mx](mailto:nfarias@itcolima.edu.mx), [ngarcia@itcolima.edu.mx](mailto:ngarcia@itcolima.edu.mx), [azael.garcia@itcolima.edu.mx](mailto:azael.garcia@itcolima.edu.mx)

### Citación sugerida:

Cásares Farías, C.A., Farías Mendoza, N., García Díaz, N. y García Rebolledo, A. (2017). Procesamiento de imágenes de plantas ornamentales multi-escala para calcular su crecimiento. *3C TIC: Cuadernos de desarrollo aplicados a las TIC*, 6(3), 10-25. DOI: <<http://dx.doi.org/10.17993/3ctic.2017.57.10-25/>>.

## RESUMEN

La Agricultura de Precisión (AP), ha sido aplicada por los agricultores representando una herramienta de apoyo para aumentar la productividad en viveros de plantas ornamentales. En este artículo se desarrolla un algoritmo para calcular el crecimiento de plantas ornamentales, formulando una solución innovadora en el ámbito de agricultura de precisión, procesando imágenes de las plantas ornamentales escalando sus dimensiones para calcular su crecimiento. En el desarrollo de este trabajo se utilizaron tecnologías emergentes, como el algoritmo de reducción por multi-escala para procesar imágenes, el lenguaje de programación Python y la biblioteca de visión artificial OpenCV. Se aplicó el algoritmo multi-escala a muestras gráficas con diferentes proporciones de la misma imagen obteniendo cálculos del crecimiento de las plantas. Con base a los resultados obtenidos consideramos que el algoritmo desarrollado puede extenderse a otras áreas del conocimiento.

## ABSTRACT

Precision Agriculture (PA) has been applied by the farmers representing a support tool to increase productivity in ornamental plant nurseries. This article develops an algorithm to calculate the growth of ornamental plants, formulating an innovative solution in the field of precision agriculture, processing images of ornamental plants scaling the dimensions to calculate their growth. In the development of this work, emergent technologies were used, like the algorithm a reduction by multi-scale to process images, the programming language Python and the library of artificial vision OpenCV. The multi-scale algorithm was applied to graphic samples with different proportions of the same image getting calculations of plant growth. Based on the results obtained, we consider that the developed algorithm can be extended to other areas of knowledge.

## PALABRAS CLAVE

Reducción Multi-Escala, Algoritmo de Igualación, Medición del Crecimiento, Viveros, Plantas Ornamentales.

## KEYWORDS

Multi-Scale Reduction, Matching Algorithm, Growth Measurement, Green House, Ornamentals Plants.

## 1. INTRODUCCIÓN

La agricultura de precisión ha sido aplicada por los agricultores desde los primeros tiempos de la agricultura. Los agricultores de subsistencia trabajaron en pequeñas parcelas de tierra, las características de las cuales servían bien. Se dividieron sus tierras en áreas más pequeñas para producir cultivos donde las condiciones eran más adecuadas. Para los agricultores, una precisión suficiente fue garantizar los alimentos para subsistencia de la familia (Oliver, 2010).

La productividad de un cultivo en términos biológicos comienza a definirse desde el inicio del ciclo de producción comercial y es afectada por una multiplicidad de factores, algunos de ellos propios del genotipo, otros del ambiente y otros de las condiciones de manejo (Di Benedetto & Tognetti, 2016).

El actualmente denominado "análisis clásico" es un análisis a nivel de una planta aislada que en su forma más simple permite estimar la medida de fijación de carbono por unidad de área foliar y la proporción del carbono disponible que se emplea en la producción, los cuales contribuyen a la acumulación de biomasa de la planta entera. Este tipo de análisis es particularmente apropiado para plantas que se encuentran en una fase exponencial de crecimiento, por esta razón es muy usado en estudios ecológicos en los que se evalúa la competencia temprana entre plantas (Di Benedetto & Tognetti, 2016).

El presente trabajo de investigación está soportado por los siguientes estudios relacionados publicados previamente.

En el trabajo de investigación desarrollado por (Narumol Chumuang, 2016) se propuso el algoritmo de segmentación de imagen para clasificar sus elementos y calcular la superficie de la hoja con una técnica de segmentación de umbral para separar la superficie buena de la imagen de la hoja, utilizando un umbral constante en el modelo de color gris y calcular el grado de color verde en los modelos de valores de saturación de matices (HSV). Además, en este documento simulamos la estimación del área foliar con análisis de regresión lineal por pixel en la superficie de la hoja.

En (Lou Wei, 2010) se desarrolló un algoritmo para la estimación del crecimiento del tomate denominado algoritmo de optimización de enjambre de partículas (PSO), el cual representa una solución estocástica global que ha demostrado ser un buen método de optimización. Este algoritmo da seguimiento a operadores de búsqueda de acuerdo con su propia velocidad, la longitud del tomate de la etapa reproductiva, controles por el genotipo de las características de las especies y factores ambientales en el proceso de desarrollo del tomate como la temperatura y la luz. Con el fin de validar los parámetros se llevó a cabo un estudio simulado para la estimación que hace uso del modelo no lineal del algoritmo PSO, cuyo objetivo es determinar los parámetros del modelo de crecimiento y desarrollo del tomate, observando que los valores reales del modelo tienen un mayor grado de adaptación. El experimento fue implementado en invernaderos cuya temperatura y longitud de día pueden

ser controlados para obtener los valores medidos reales y para estimar los parámetros del modelo utilizando PSO.

Finalmente, en el trabajo desarrollado por Padrón-Pereira, (2013) se obtuvieron métodos para estimar la medida del cambio de diámetro en un fruto siendo de gran interés la investigación fisiológica de un árbol integral o de un solo fruto, ya que puede esto proporcionar información útil para el manejo del huerto o cultivo. Se planteó que el uso de imágenes digitales y software con capacidad de medición permite elaborar curvas de crecimiento para el estudio de patrones en frutos de mandarina, además de utilizar imágenes digitales para medir el diámetro ecuatorial en frutos de mandarina durante el crecimiento con la finalidad de aportar un procedimiento alternativo para el estudio de patrones de crecimiento.

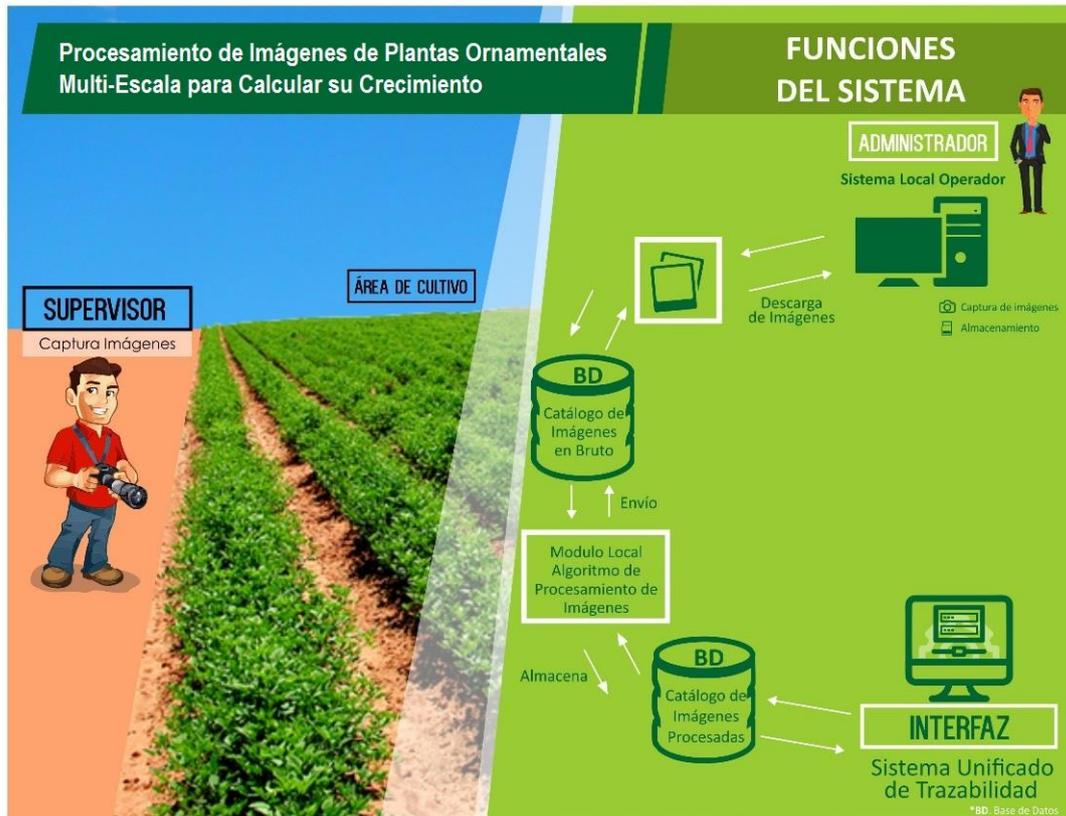
Con base a los estudios previos se propone el presente trabajo de investigación debido a que ninguno propone enfoques en la reducción de tiempo de muestreo de parámetros físicos para confirmar el desarrollo productivo de las plantas ornamentales, punto relevante en este trabajo de investigación.

La arquitectura desarrollada hace uso de tecnologías de información y su diseño se puede observar en la Figura 1, donde se observan los módulos que realizan las funciones siguientes:

- Captura de imágenes de plantas ornamentales para su procesamiento con un algoritmo de reducción a multi-escalas.
- Almacenar la información para alimentar un catálogo de plantas ornamentales.
- Simplificar el proceso productivo, con un método ágil de observación del crecimiento de las plantas ornamentales.

El proceso se inicia con el ciclo de diseño y desarrollo del trabajo, siendo necesario poner énfasis en las siguientes fases:

- El análisis de los requerimientos del sistema.
- El desarrollo de los algoritmos, métodos y técnicas que serán la base para las interfaces.
- La implementación de la aplicación gráfica que realice el procesamiento de imágenes de plantas ornamentales multi-escala para la estimación del crecimiento. Haciendo uso de la biblioteca de visión artificial OpenCV y el lenguaje de programación Python se hace la construcción del mecanismo que compara imágenes empatándolas buscando tener una igualación de los bordes de las imágenes para detectar cambios.
- El desarrollo de pruebas.
- La implantación del sistema.



**Figura 1.** Procesamiento de Imágenes de Plantas Ornamentales Multi-Escala para Calcular su Crecimiento (BigPicture). **Fuente:** Elaboración propia.

## 2. METODOLOGÍA

La investigación documental necesaria fue ampliamente estudiada sobre temas de investigaciones relacionados a técnicas, métodos y herramientas tecnológicas usadas en invernaderos y en la digitalización de imágenes, encontrando trabajos muy significativos con amplia similitud sobre los objetivos de esta investigación. Se detectaron así áreas de oportunidad que nadie ha considerado con el desarrollo de un método para calcular el crecimiento de las plantas ornamentales.

En el desarrollo del algoritmo para el cálculo del crecimiento de las plantas ornamentales se usó como base el algoritmo de detección de bordes (Mordvintsev, A. & K. Abid, 2013). Este algoritmo fue modificado incluyendo el análisis de histograma de colores con el objetivo de conocer la composición de píxeles de colores de la imagen. Posteriormente, se procesaron las imágenes multi-escala para obtener una mayor precisión en el cómputo del crecimiento de las plantas.

En la Tabla 1. se detallan las actividades desarrolladas para la implementación del algoritmo propuesto.

**Tabla 1.** Procedimiento para el Desarrollo de la Investigación.

ACTIVIDAD
<p><b>Construcción del Sistema para Reconocimiento de Patrones en Plantas Ornamentales:</b></p> <ol style="list-style-type: none"> <li>1. Efectuar la instalación del lenguaje de programación Python.</li> <li>2. Efectuar la instalación de la librería OpenCV, así como de sus componentes para procesamiento de imágenes como Numpy, CV2, Sys, liburl, entre otros.</li> <li>3. Diseño de un diagrama de flujo de las fases del algoritmo para la estimación del crecimiento implementado.</li> <li>4. Desarrollo de una aplicación compuesta por algoritmos de igualación que procesan imágenes localmente en un equipo de alto rendimiento desde el lenguaje de programación Python.</li> <li>5. Presentación de resultados.</li> </ol>

### 3. DESARROLLO

El desarrollo del algoritmo para el cálculo del crecimiento de las plantas ornamentales requiere del uso de una computadora con holgadas características de hardware para procesamiento y manejo de gráficos, un lenguaje estructurado de programación y una cámara digital para la captura de imágenes a distancias considerables, tal como se describe en la Tabla 2.

**Tabla 2.** Especificaciones de hardware y Software para el Desarrollo del Proyecto.

HERRAMIENTAS DE DESARROLLO	
<b>HARDWARE</b>	
1	Equipo de Cómputo: <ul style="list-style-type: none"> <li>• Procesador Intel Core i5 sexta generación.</li> <li>• 8 GB de Memoria RAM.</li> <li>• 240 GB de Almacenamiento HDD SATA o SSD SATA.</li> </ul>
1	Cámara de Video de 10 Megapíxeles.
<b>SOFTWARE</b>	
<b>Windows 8.1 (x32 bits/x64bits)</b>	Windows es el Sistema Operativo que ejecuta el software para computadoras compatibles con arquitecturas IBM PC. Windows es en esencia el programa que ejecuta su computadora. Windows muestra las aplicaciones (software) en su dispositivo. También almacena todos sus archivos: documentos, videos, imágenes, música y otros. Permite a usted para localizar y luego haga clic en ellos para abrirlos. También proporciona otros aspectos de la ejecución de su computadora, incluyendo el registro (con o sin una contraseña), cerrando, y así sucesivamente. Windows 8 fue un rediseño radical de la interfaz de Windows. (Wilson, 2015)
<b>Lenguaje de Programación Python para Windows</b>	Python es un lenguaje de programación orientado a objetos clara y potente (Anuskiewicz, 2000). Algunas de las características notables de Python: <ul style="list-style-type: none"> <li>• Utiliza una sintaxis elegante, por lo que los programas se escriben más fácil de leer.</li> <li>• Viene con una gran biblioteca estándar que soporta muchas de las tareas de programación comunes tales como la conexión a los servidores web, la búsqueda</li> </ul>

de texto con expresiones regulares, leer y modificar archivos.

- El modo interactivo de Python hace que sea fácil de probar pequeños fragmentos de código. También hay un entorno de desarrollo paquete llamado IDLE.
- Se puede ampliar fácilmente añadiendo nuevos módulos implementados en un lenguaje compilado como C o C++.
- Unas variedades de tipos de datos básicos están disponibles: números (punto flotante, complejos y largos enteros de longitud ilimitada), cuerdas (ASCII y Unicode), listas y diccionarios.
- Python es compatible con la programación orientada a objetos con clases y herencia múltiple.
- El lenguaje es compatible con la crianza y la captura de excepciones, lo que resulta en el manejo de errores más limpio.
- Python contiene características avanzadas de programación tales como generadores y las listas por comprensión.

**Librerías** OpenCV (Open Source Computer Vision Library) es una fuente abierta de visión por ordenador y la biblioteca de software de aprendizaje de máquina. OpenCV fue construido para proporcionar una infraestructura común para aplicaciones de visión artificial y acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto de licencia BSD, OpenCV hace que sea fácil para las empresas a utilizar y modificar el código. La biblioteca cuenta con más de 2,500 algoritmos optimizados, que incluye un amplio conjunto de clásico y de visión artificial y algoritmos de aprendizaje automático con tecnología de última generación. Estos algoritmos se pueden utilizar para detectar y reconocer las caras, identificar objetos, clasificar las acciones humanas en los vídeos, los movimientos de cámara pista, moviendo el seguimiento de objetos, extraer modelos 3D de objetos, producen nubes de puntos 3D a partir de cámaras estéreo, fusionar imágenes juntos para producir una alta resolución la imagen de una escena entera, encontrar imágenes similares de una base de datos de imágenes, eliminar los ojos rojos de las imágenes tomadas con flash, seguir los movimientos de los ojos, reconocer el paisaje y establecer marcadores para revestirlo de realidad aumentada, etc. (OpenCV Team, 1999)

### 3.1. INSTALACIÓN DEL LENGUAJE DE PROGRAMACIÓN PYTHON

La instalación del lenguaje de programación Python fue realizada con ayuda de la documentación del sitio oficial. Fue necesario descargar un paquete instalador para el sistema operativo Windows 8.1 que no requiere de configuración adicional (The Python Software, 2015). La versión usada para este desarrollo fue la 2.7.10 que muestra un desempeño muy estable.

### 3.2. INSTALACIÓN DE LA LIBRERÍA OPENCV

Este potente complemento para el procesamiento de imágenes es compatible con el lenguaje de programación Python. Para manejar los más de 2,500 algoritmos de visión que incorpora el paquete es necesario obtener la versión estable desde el sitio oficial, en este

caso fue usada la v.3.0.0. La guía completa sobre la instalación se encuentra en los documentos de OpenCV (Mordvintsev, A. & K. Abid, 2013).

### 3.3. DESARROLLO DE LA APLICACIÓN

La aplicación desarrollada con el lenguaje de programación Python incorpora código de librerías OpenCV específicas para el cálculo matemático y manejo de matrices multidimensionales de datos. *Numpy*, es usada en el procesamiento numérico; *imutils*, utilizada para algunas funciones de procesamiento de imágenes; así como el método *cv2*, usado para la manipulación de imágenes y obtención de salidas o vistas de usuario. La aplicación obtiene parámetros en valores de píxeles RGB (Red, Green, Blue o 3 canales) con la ayuda del comando *template.shape[:2]*, generando un histograma que agrupa valores entre los rangos de 0 a 255 por píxel, siendo organizados en una matriz de 2 dimensiones. Estos datos son una representación numérica de la imagen modelo, de estos mismos valores se obtiene un promedio general con referencia a la agrupación de píxeles en los 3 canales Red, Green y Blue. Este procedimiento de obtención de parámetros es aplicado con cada una de las copias o planillas a las cuales se les hace una reducción de escala.

La reducción a Multi-Escalas en las copias de la imagen para las “planillas” se realiza de la siguiente manera. La imagen es clonada obteniendo una planilla digital que será usada para realizar reducciones en las dimensiones de la altura y ancho. En el cuadro 1 se observa la obtención de a escala de reducción que es calculada en las líneas 19 y 20 del código fuente en el algoritmo que a continuación se muestra.

La igualación entre la imagen modelo y la planilla es llevada a cabo con el comando *cv2.matchTemplate* que maneja un coeficiente de aproximación de bordes. Este proceso genera valores en píxeles que se usan como coordenadas de proximidad entre ambas imágenes, de tal manera que estos puntos ( $x_1, x_2$ ) y ( $y_1, y_2$ ) son proyectados para su visualización gráfica marcando un recuadro de color rojo sobre el área de la planilla a escala procesada.

**Fórmula 1.** Código Fuente del algoritmo de igualación por reducción de Multi-Escala.

```

1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 #Fecha. 2017-02-13
4 for imagePath in glob.glob(args["imagenes"] + "/*.jpg"):
5     image = cv2.imread(imagePath)
6     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
7     found = None
8     cantidad=cantidad + 1
9     print "=====
10    print "Numero de Imagen Elegida para igualación[" , cantidad, "]"
11    print "=====
12    contador=0
13    for scale in np.linspace(0.2, 1.0, 20)[::-1]:
14        resized = imutils.resize(gray, width = int(gray.shape[1] * scale))
15        r = gray.shape[1] / float(resized.shape[1])
16        if resized.shape[0] < tH or resized.shape[1] < tW:
17            break
18        edged = cv2.Canny(resized, 50, 200)
19        result = cv2.matchTemplate(edged, template, cv2.TM_CCOEFF)
20        (_, maxVal, _, maxLoc) = cv2.minMaxLoc(result)
21        if args.get("visualize", False):
22            clone = np.dstack([edged, edged, edged])
23            cv2.rectangle(clone, (maxLoc[0], maxLoc[1]),
24                (maxLoc[0] + tW, maxLoc[1] + tH), (0, 0, 255), 2)
25            ver_h, ver_w, ver_ch = clone.shape
26            contador=contador + 1
27            print "Reduccion[" , contador, "]" , ver_h, ver_w, ver_ch
28            cv2.imshow("Visualize", clone)
    
```

```

29         cv2.waitKey(0)
30         if found is None or maxVal > found[0]:
31             found = (maxVal, maxLoc, r)
32         (_, maxLoc, r) = found
33         (startX, startY) = (int(maxLoc[0] * r), int(maxLoc[1] * r))
34         (endX, endY) = (int((maxLoc[0] + tW) * r), int((maxLoc[1] + tH) * r))
35         pixelClone1=startY
36         pixelClone2=startX
37         aPromR=0, aPromG=0, aPromB=0, tPromR=0, tPromG=0, tPromB=0
38         float(tPromR), float(tPromG), float(tPromB), float(aPromR), float(aPromG), float(aPromB)
39         for pixelClone1 in range(endY):
40             sumR=0, sumG=0, sumB=0, promR=0, promG=0, promB=0
41             float(sumR), float(sumG), float(sumB), float(promR), float(promG), float(promB)
42             for pixelClone2 in range(endX):
43                 pixelTemplateClone = image[pixelClone1,pixelClone2]
44                 pxR=int(pixelTemplateClone[0:1])
45                 pxG=int(pixelTemplateClone[1:2])
46                 pxB=int(pixelTemplateClone[2:3])
47                 sumR=sumR+pxR
48                 sumG=sumG+pxG
49                 sumB=sumB+pxB
50         cv2.rectangle(image, (startX, startY), (endX, endY), (0, 0, 255), 2)
51         cv2.imshow("Image", image)
52         cv2.waitKey(0)
    
```

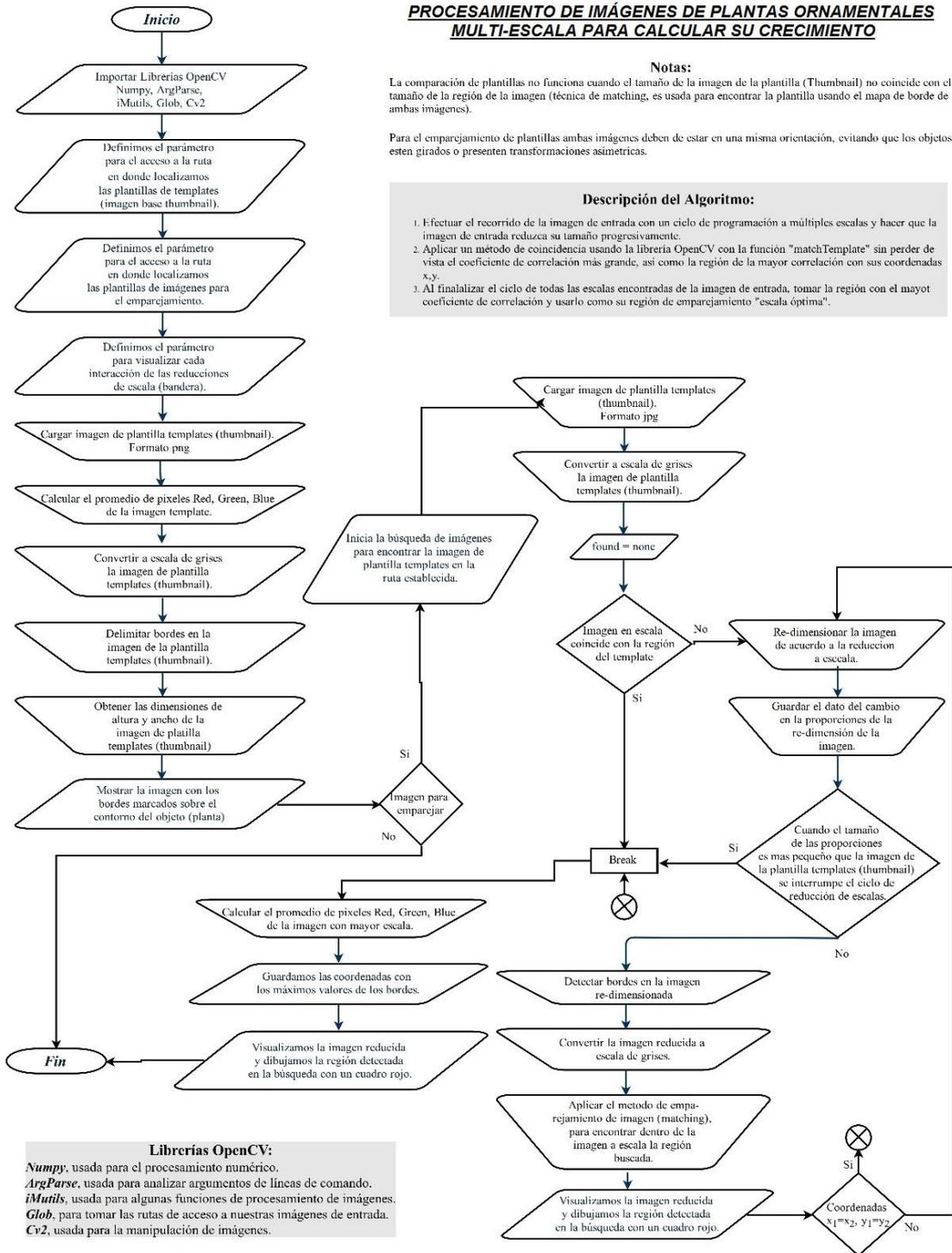
Fuente: elaboración propia.

### 3.4. DISEÑO DEL DIAGRAMA DE FLUJO AL ALGORITMO PARA LA ESTIMACIÓN DEL CRECIMIENTO

La fusión de todos estos elementos nos ha permitido diseñar un algoritmo que nos muestra el proceso que logra manipular imágenes digitales de plantas ornamentales estimando cambios en los histogramas de colores por medio de técnicas de igualación con múltiples escalas en imágenes. El funcionamiento de este método inicia con la obtención de un modelo, es decir, una imagen que será el objetivo a alcanzar la igualación. Esta imagen debe ser limpia, sin fondo u objetos ajenos, conservando únicamente la fisonomía de la planta, en este caso una planta ornamental de la especie “Flor del Desierto”.

El archivo digital de este modelo debe tener una extensión “.png” que admite la compresión sin pérdida, información de transparencia, y una gama de profundidades de color (Roelofs, 2003). El resto de las imágenes o plantillas de búsqueda no requieren de limpieza, son usadas de la manera en la cual fueron capturadas en formato de imagen digital “.jpg o .jpeg” que es capaz de desplegar millones de colores y maneja perfectamente la mezcla compleja de matices con una capacidad muy grande de compresión (Voutssás Márquez, 2006).

**PROCESAMIENTO DE IMÁGENES DE PLANTAS ORNAMENTALES MULTI-ESCALA PARA CALCULAR SU CRECIMIENTO**



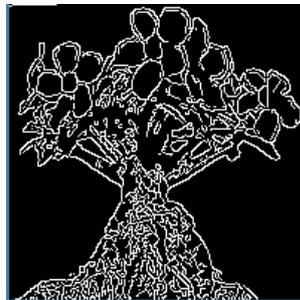
**Figura 2.** Diagrama de Flujo para Procesamiento de Imágenes de Plantas Ornamentales Multi-Escala para Calcular su Crecimiento. **Fuente:** elaboración propia.

El método de estimación de crecimiento esta compuesto por un bucle que de manera gradual obtiene copias o plantillas de la imagen modelo que se iran reduciendo selectivamente en un número finito de ocasiones bajo la condición de no sobrepasar las reducciones del tamaño en altura y ancho con referencia a la imagen modelo. El algoritmo transforma las imágenes a escalas de grises reduciendo la gama de colores a lo más

elemental, de ello se obtienen objetos puros, señalados por bordes que nos representan la fisonomía de las plantas ornamentales analizadas. En este punto las condiciones permiten el uso del algoritmo de identificación de bordes, así como la igualación que trabaja con la librería OpenCV de codificación cv2 para la manipulación de imágenes importadas.

## 4. RESULTADOS

La aplicación implementada con el algoritmo de igualación por reducción de Multi-Escala inicialmente trabaja con la preparación de la imagen modelo, realizando una transformación a escala de grises, obteniendo el contorno o fisonomía representada por píxeles en tonos claros, tal como se muestra en la Figura3.



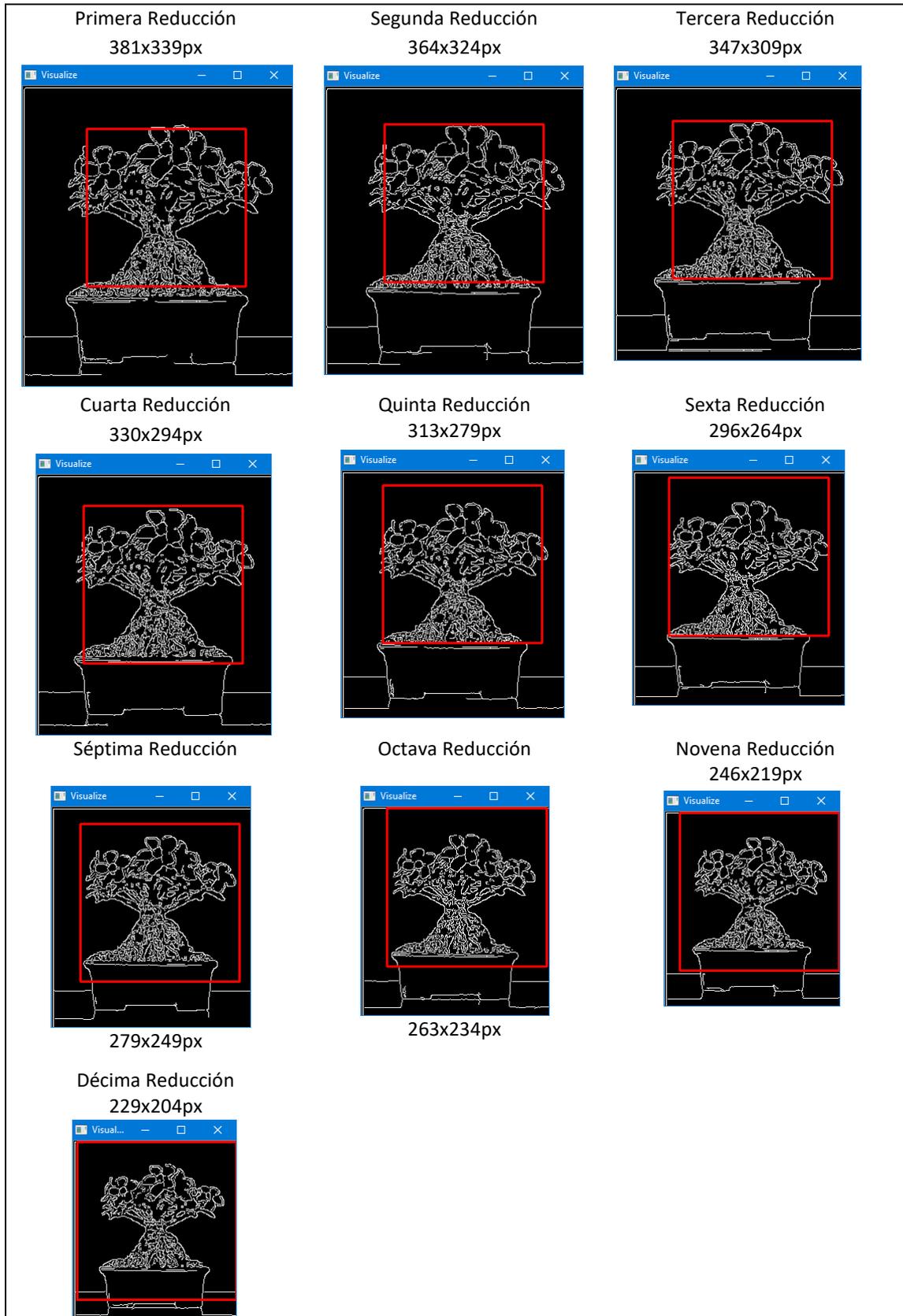
**Figura 3.** Base Modelo (Modelo) 200x200px.  
 Fuente: elaboración propia.

Posteriormente, se obtienen parámetros en valores de píxeles RGB (Red, Green, Blue o 3 canales), se suman los valores de la matriz de 2 dimensiones y se les promedia. Esto se observan en la Tabla 3.

**Tabla 3.** Concentración de Valores de los tres canales RGB para la imagen modelo.

DESCRIPCIÓN IMAGEN	PROMEDIO DE PIXELES EN IMAGEN (RED)	PROMEDIO DE PIXELES EN IMAGEN (GREEN)	PROMEDIO DE PIXELES EN IMAGEN (BLUE)
Base Modelo = 200 x 200px	180.41	186.715	191.435

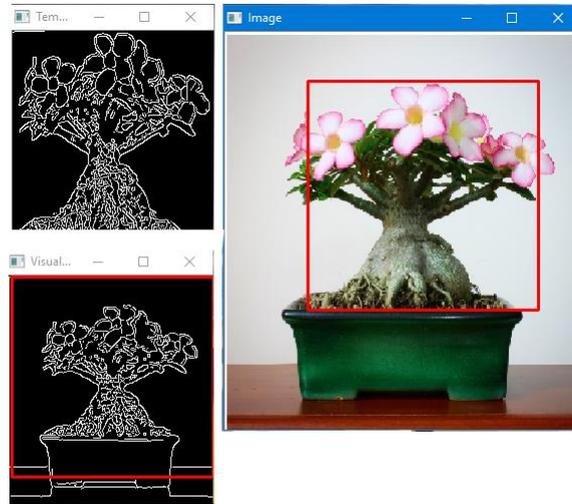
A continuación, con el algoritmo de igualación por reducción de Multi-Escala se obtienen copias de la imagen o “planillas” con dimensiones reducidas, de las cuales se busca obtener reducciones del tamaño en altura y ancho que sean aproximadas para evaluar la igualación, el resultado de este proceso se visualiza gráficamente con un recuadro en color rojo. El desarrollo del proceso de reducción del algoritmo de igualación por reducción de Multi-Escala para calcular el crecimiento de las plantas ornamentales se observa en la Figura 4. El momento en que se finaliza es cuando la reducción hecha excede las dimensiones de altura y ancho de la imagen modelo.



**Figura 4.** Reducciones realizadas por el algoritmo de igualación por reducción de Multi-Escala.

**Fuente:** elaboración propia.

El algoritmo de igualación por reducción de Multi-Escala ofrece resultados positivos y culmina en el momento que la aplicación detecta que la reducción tiene las mismas dimensiones en altura y ancho que la imagen modelo. En este punto se muestran gráficamente los resultados en la imagen de plantilla sin reducción a colores, delimitando con un recuadro rojo el área en donde se presenta una coincidencia positiva de igualación.



**Figura 5.** Obtención positiva de la Igualación.  
**Fuente:** elaboración propia.

El algoritmo de igualación por reducción de Multi-Escala obtiene valores sobre las características de la imagen que son representados en pixeles. A manera de registro se obtienen los valores de cada una de las reducciones generadas. Se observa que la reducción de altura y la reducción de ancho efectuada por el algoritmo de multi-escala es en valores promedio y constantes, tal como se observa en la Tabla 4.

**Tabla 4.** Valores obtenidos del algoritmo de igualación por reducción de Multi-Escala.

NOMBRE PLANTA	IMAGEN	NÚMERO DE REDUCCIÓN (R)	ALTURA (H)	ANCHO (W)	REDUCCIÓN DE ALTURA (PX)	REDUCCIÓN DE ANCHO (PX)	CANAL
Flor de Desierto 398 x 354 px	1	1	398	354	0	0	3
	1	2	381	339	17	15	3
	1	3	364	324	17	15	3
	1	4	347	309	17	15	3
	1	5	330	294	17	15	3
	1	6	313	279	17	15	3
	1	7	296	264	17	15	3
	1	8	279	249	17	15	3
	1	9	263	234	16	15	3
	1	10	246	219	17	15	3
	1	11	229	204	17	15	3

De la imagen usada como plantilla para las reducciones se puede comprobar que el número de píxeles requeridos para llegar al tamaño de la imagen modelo fue de 169 píxeles en la altura y de 150 píxeles en el ancho, tal como se observa en la Tabla 5.

**Tabla 5.** Simetría en los valores procesados en el algoritmo de igualación por reducción de Multi-Escala.

DIFERENCIA ALTURA (PX)	SUMA DE LA REDUCCIÓN DE ALTURA (PX)
H1 - H11 = 169	H1 - H11 = 169
DIFERENCIA ANCHO (PX)	SUMA DE LA REDUCCIÓN DE ANCHO (PX)
W1 - W11 = 150	W1 - W11 = 150

Al finalizar el procesamiento del algoritmo de igualación por reducción de Multi-Escala se obtienen parámetros en valores de píxeles RGB (Red, Green, Blue o 3 canales) de ambas imágenes. Estos son promediados para obtener un porcentaje que nos indica cambios entre ambas imágenes, estimando así un crecimiento en la planta, tal como se observa en la Tabla 6.

**Tabla 6.** La igualación de la concentración de Valores de los tres canales RGB para las imágenes modelo – original.

DESCRIPCIÓN IMAGEN	PROMEDIO DE PÍXELES EN IMAGEN (RED)	PROMEDIO DE PÍXELES EN IMAGEN (GREEN)	PROMEDIO DE PÍXELES EN IMAGEN (BLUE)
Base Modelo = 200 x 200px	180.41	186.715	191.435
Imagen Original = 398 x 354 px	181.7841727	186.1726619	187.9388489
Diferencia	1.3741727	-0.5423381	-3.4961511
Porcentaje de Píxeles	1.32%	-3.43%	-0.54%

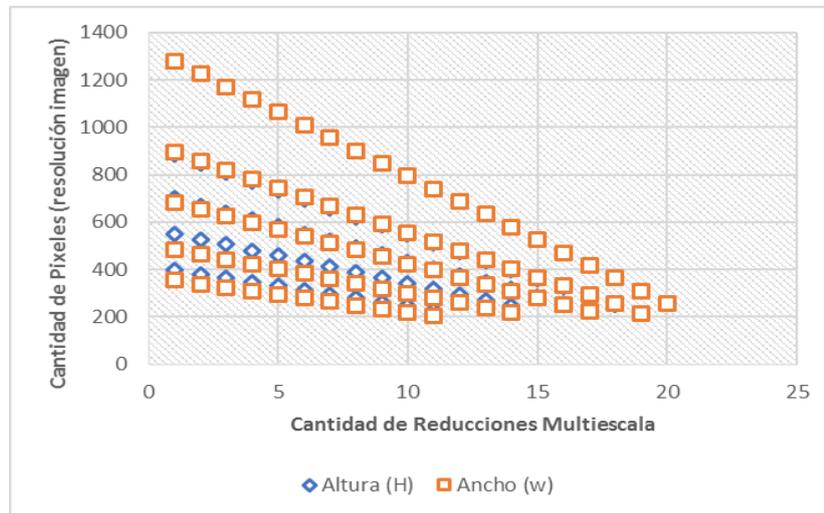
La prueba implicó el procesar la misma la misma variedad de planta en otros tamaños con dimensiones mayores obtenidas de la misma planta durante intervalos de 3 semanas. De estas se obtuvieron promedios de los tres canales de color similares, tal como se observa en la Tabla 7.

**Tabla 7.** Evaluación de otras capturas con algoritmo de igualación con la concentración de Valores de los tres canales RGB para las imágenes modelo – original en varias dimensiones.

DESCRIPCIÓN IMAGEN	PROMEDIO DE PÍXELES EN IMAGEN (RED)	PROMEDIO DE PÍXELES EN IMAGEN (GREEN)	PROMEDIO DE PÍXELES EN IMAGEN (BLUE)
Base Modelo = 200 x 200px	180.41	186.715	191.435
Imagen Original = 398 x 354 px	181.7841727	186.1726619	187.9388489
Imagen Original = 552 x 484 px	181.4360313	185.9033943	187.6396867
Imagen Original = 700 x 684 px	177.8721649	183.0948454	185.0123711
Imagen Original = 884 x 896 px	178.8879599	184.0250836	185.9464883
Imagen Original = 1280 x 1280 px	179.6975666	184.6940904	186.5816918

En la Figura 6 se observa que los valores obtenidos sobre las dimensiones de los tres canales siguen una tendencia lineal hacia el punto de igualación, donde las dimensiones de altura y ancho de las plantillas con reducciones son iguales a las del modelo.

**Figura 6.** Datos de las Dimensiones en Reducción de Imagen a Escala.



## 5. CONCLUSIONES Y RECOMENDACIONES

El algoritmo desarrollado se aplica en el crecimiento de las plantas comparando las condiciones para la producción de plantas, haciendo un análisis entre los nutrientes aplicados, la irrigación y parámetros físicos como: la luminosidad, conductividad, temperatura, humedad, pH y otros factores ambientales que intervienen en el crecimiento de las plantas ornamentales que nos lleve a un ciclo de mejora continua con la acertada toma de decisiones que incrementen la productividad y competitividad de los viveristas.

De este trabajo se concluye que el algoritmo de igualación de imágenes multi-escala es una herramienta que determina las similitudes entre dos imágenes haciendo uso de hardware con capacidad de alto procesamiento y lenguajes de programación de fácil acceso. Con su uso, es posible determinar el crecimiento de una planta al analizar los píxeles que la conforman; por otra parte, se destaca que la aplicación del algoritmo puede ser migrada al análisis del crecimiento de otros objetos. El algoritmo desarrollado puede ser calibrado con una mayor aproximación para dar mejores resultados.

Como trabajo futuro se recomienda implementar esta tecnología en dispositivos con autonomía de procesamiento en una Raspberry Pi para buscar efectuar la tarea del cálculo de crecimiento en tiempo real. También implementar una regla virtual para calcular el crecimiento de las imágenes de las plantas ornamentales con una representación gráfica en escala métrica.

## 6. REFERENCIAS BIBLIOGRÁFICAS

- Anuskiewicz, N. (2000). <https://wiki.python.org/moin/BeginnersGuide/>. Recuperado el 2016, de <https://wiki.python.org/moin/BeginnersGuide/>
- Di Benedetto, A., & Tognetti, J. (2016). Técnicas de análisis de crecimiento de plantas: su aplicación a cultivos intensivos. RIA. Revista de investigaciones agropecuarias, 259. ISSN 1669-2314.
- Lou Wei, H. J. (2010). The Research of Parameter Estimation of Tomato Growth and Development Model based on PSO Algorithm. IEEE 331-334.
- Mordvintsev, A. & K. Abid (2013). OpenCV-Python Tutorials. Obtenido de [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_tutorials.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html).
- Mordvintsev, A. & K. Abid (2013). OpenCV-Python Tutorials. Obtenido de [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_table\\_of\\_contents\\_imgproc/py\\_table\\_of\\_contents\\_imgproc.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_table_of_contents_imgproc/py_table_of_contents_imgproc.html).
- Narumol Chumuang, S. T. (2016). Algorithm Design in Leaf Surface Separation by Degree in HSV Color Model and Estimation of Leaf Area by Linear Regression. DOI 10.1109/SITIS.2016.104.
- Oliver, M. (2010). Geostatistical Applications for Precision Agriculture. (M. Oliver, Ed.) Geostatistical Applications for Precision Agriculture, 337. DOI 10.1007/978-90-481-9133-8.
- OpenCV, T. (01 de 01 de 1999). OpenCV. Recuperado el 2016, de <http://opencv.org/>.
- Padrón-Pereira, C. A. (2013). UTILIZACIÓN DE IMÁGENES DIGITALES PARA MEDICIÓN DEL DIÁMETRO DE FRUTOS DE MANDARINA (*Citrus reticulata*) EN CRECIMIENTO. Revista Ciencia y Tecnología, 2. ISSN 1390-4043.
- Roelofs, G. (2003). PNG The Definitive Guide (Segunda Edición ed.). Sebastopol, California, USA: O'Reilly and Associates.
- The Python Software, F. (23 de Mayo de 2015). Python.org. Obtenido de <https://www.python.org/downloads/>.
- Voutssás Márquez, J. (2006). Bibliotecas y Publicaciones Digitales (Primera Edición ed.). México, Distrito Federal, México: UNAM.
- Wilson, K. (2015). Everyday Computing with Windows 8.1. (S. Anglin, Ed.) New York, USA: Apress. DOI 10.1007/978-1-4842-0805-2.