



UNA UNIVERSIDAD CON ALMA DE GUERRERO

UTILIZACIÓN DE TÉCNICAS DE VISIÓN ARTIFICIAL PARA LA DETECCIÓN AUTOMÁTICA DE DEFECTOS EXTERNOS DEL MANGO

Tesis para optar por el Título de Ingeniero
de Sistemas, que presenta los bachilleres.

AUTORES

Erick David Lucero Carrillo
Heiner Anibal Saldaña Saldaña

ASESOR

Ing. Heber Ivan Mejía Cabrera

CHICLAYO - PERÚ 2016



FORMATO DE PÁGINA DE PRESENTACIÓN DE LA APROBACIÓN

UTILIZACIÓN DE TÉCNICAS DE VISIÓN ARTIFICIAL PARA LA DETECCIÓN AUTOMÁTICA DE DEFECTOS EXTERNOS DEL MANGO

Aprobación del Proyecto

Mg. Esquivel Paredes Lourdes Jossefyne
Asesor metodológico

Ing. Mejía Cabrera Heber Ivan
Asesor especialista

Dr. Ramos Moscol Mario Fernando
Presidente del jurado de Tesis

Mg. Tuesta Monteza Victor Alexci
Secretario del jurado de Tesis

Ing. Mejía Cabrera Heber Ivan
Vocal del jurado de tesis

DEDICATORIA

Dedicamos la presente tesis con mucho cariño a nuestras familias, por estar siempre a nuestro lado brindándonos su ayuda incondicional y apoyo.

A nuestros amigos por estar con nosotros todo este tiempo ayudándonos a que sea posible el desarrollo de este proyecto.

AGRADECIMIENTO

A todas las personas, que en su momento, con sus consejos y apoyo nos ayudaron a seguir adelante. Y a todos los señores profesores que compartieron sus conocimientos para nuestra formación intelectual y profesional

ÍNDICE

INTRODUCCIÓN.....	12
CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN.....	13
1.1. Situación Problemática.....	14
1.2. Formulación del Problema.....	21
1.3. Delimitación de la Investigación	21
1.4. Justificación e Importancia de la Investigación	22
1.5. Limitaciones de la Investigación	22
1.6. Objetivos	22
CAPÍTULO II: MARCO TEÓRICO	24
2.1. Antecedentes de la Investigación.....	25
2.2. Estado del Arte	34
2.3. Bases teóricas Específicas	39
2.3.1. Clasificación del Mango	39
2.3.1.1. Defectos del Mango	40
2.3.2. Modelos de Color	42
2.3.2.1. Espacio de Color RGB	42
2.3.2.2. Espacios de Color HSV.....	44
2.3.2.3. Espacios TV Color – YUV, YIQ y YC_bC_y.....	47
2.3.2.3.1. Espacios de Color YUV.....	48
2.3.2.3.2. Espacios de Color YIQ	49
2.3.2.3.3. Espacios de Color YC_bC_y	49
2.3.2.4. Espacios de Color para la Impresión CMY y CMYK.....	51
2.3.3. Procesamiento de Imagen Digital	53
2.3.3.1. Segmentación	55
2.3.4. Descriptores de Imágenes	57
2.3.5. Descriptores de forma	59
2.3.5.1. Descriptores BSM (Blurred Shape Model).....	59
2.3.5.2. Square Model Shape Matrix.....	63
2.3.6. Algoritmo de Textura.....	64
2.3.6.1. Co-ocurrencia Matriz de cálculo	65
2.3.7. Clasificación Supervisada	68



2.3.7.1.	Vecino más cercano (1-NN).....	68
2.3.7.2.	Algoritmo K-NN (K-Nearest Neighbor).....	68
2.3.7.3.	Adaboost	70
2.4.	Definición de la Terminología.....	71
CAPÍTULO III: MARCO METODOLÓGICO		73
3.1.	Tipo y diseño de la investigación	73
3.2.	Población y Muestra.....	74
3.3.	Hipótesis.....	74
3.4.	Variables.....	74
3.5.	Operacionalización	75
3.6.	Métodos, Técnicas e Instrumentos de recolección de datos.....	76
3.7.	Procedimiento para la recolección de los datos	78
3.8.	Análisis estadístico e Interpretación de los datos	78
3.9.	Principios éticos.....	79
3.10.	Criterios de rigor científico.....	79
CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE RESULTADOS		80
4.1.	Descripción de resultados.....	81
4.1.1.	Evaluación de métodos de extracción de características....	82
4.1.2.	Resultados con los algoritmos evaluados de extracción de características y clasificación	86
4.1.3.	Cuadro comparativo de los algoritmos evaluados	88
4.1.4.	Defectos encontrados para la Detección Automática por descriptores.....	89
4.2.	Discusiones y resultados.....	89
CAPÍTULO V: DESARROLLO DE LA PROPUESTA DE INVESTIGACIÓN .		92
5.1.	Adquisición de Imágenes.....	94
5.2.	Pre – Procesamiento	97
5.3.	Segmentación	99
5.4.	Extracción de Características.....	103
5.5.	Clasificación	109
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES		124
6.1.	Conclusiones	125
6.2.	Recomendaciones	126



REFERENCIAS 127
ANEXOS..... 130

FIGURA

Figura 1: Valor de exportación de principales frutas frescas periodo 2000 - 2010..... 15
Figura 2: Producción de Mangos por Regiones 16
Figura 3: Representación Gráfica del espacio de color RGB..... 43
Figura 4: Mapeo de RGB al espacio HSV 47
Figura 5: Modelo unidimensional y continuo de un borde ideal 60
Figura 6: Algoritmo en pseudocódigo Blurred Shape Model 63
Figura 7: Ejemplos de texturas: bejuco, papel, frijoles, ladrillo, monedas, trenzade alambre (de arriba a abajo, de izquierda a derecha). 65
Figura 8: Muestras tomadas con defectos y sin defectos del mango. 81
Figura 9: Diseño estructural para la investigación 93
Figura 10: Instrumento empleado para la captura de las imágenes digitales. 94
Figura 11: Mango con defecto de Antracnosis 95
Figura 12: Mango con defecto de Lenticelas oscuras esparcidas..... 96
Figura 13: Mango causada por bacterias 96
Figura 14: Mango con defecto de Manchas oscuras 97
Figura 15: Utilizando Filtro GaussianBlur a la imagen del mango..... 99
Figura 16: Segmentación por Umbralización a una imagen digital 100
Figura 17: Segmentación de imágenes utilizando K-Means con distinta cantidad de clústeres (2, 3, 10 y 20 clusters respectivamente). 101
Figura 18: Aplicando el algoritmo Canny para detectar bordes 103
Figura 19: A la izquierda está la imagen original y a la derecha se ha utilizado el algoritmo BSM para detectar el contorno y defectuosidad del mango 104
Figura 20: En esta imagen segmentada se detectaron 9 defectos los cuales serán extraídos sus características de Color. 106
Figura 21: Fueron extraidos todas las características de color solo a los defectos que se encontraron de la figura 20 106



Figura 22: Extracción de características de textura utilizando descriptor GLCM correspondiente a la figura 19.....	107
Figura 23: Unidad procesadora básica Backpropagation	110
Figura 24: Red Backpropagation completamente interconectada.....	111
Figura 25: Cálculo de valores δ_J de la capa de salida	113
Figura 26: Cálculo de valores δ_J de las capas ocultas	114
Figura 27: Adquisición de Imágenes y Pre - Procesamiento.....	115
Figura 28: Extracción de características de textura utilizando GLCM.....	116
Figura 29: Extracción de Características de Color utilizando Histogramas de color	116
Figura 30: Clasificación con la Red Neuronal.....	117
Figura 31: Método de Etiquetado de pixeles conexos.....	118
Figura 32: Detección automática de defectos externos del mango.....	119
Figura 33: Recorte automático de los defectos encontrados en la imagen digitalizada.....	119
Figura 34: Detección y cuantificación automático de defectos externos del mango utilizando la técnica de etiquetado de pixeles conexos.	122
Figura 35: Método Ensamblado	122

FUENTE

Fuente 1: SUNAT elaborado por PROMPERU	15
Fuente 2: MINAG - Series Históricas de Producción Agrícola	16
Fuente 3: Elaboración Propia	86
Fuente 4: Elaboración Propia	87
Fuente 5: Elaboración Propia	88
Fuente 6: Elaboración Propia	89

GRÁFICO



Gráfico 1: Número de aciertos y errores en la clasificación de muestras utilizando Histograma de Color y segmentación Otsu. Se muestra los resultados obtenidos utilizando dos tipos de clasificadores. 82

Gráfico 2: Pruebas de efectividad del descriptor para cada tipo de mango con el uso de diversos clasificadores. 83

Gráfico 3: Número de aciertos y errores en la clasificación de las imágenes del repositorio utilizando GLCM y segmentación Otsu. Se muestra los resultados obtenidos utilizando dos tipos de clasificadores. 84

Gráfico 4: Pruebas de efectividad del descriptor de Forma para cada defecto del mango con el uso de diversos clasificadores. 85

TABLA

Tabla 1: Clasificación con la Red Neuronal 86

Tabla 2: Clasificación con KNN 87

Tabla 3: Algoritmos Evaluados 88

Tabla 4: Defectos externos del mango a Evaluar 89

Tabla 5: Defectos encontrados automáticamente por la técnica de Conexos. 123



Resumen

La evaluación del estado de los mangos se hace generalmente con la visión humana, hecho por los mismos trabajadores basados en la experiencia, lo cual puede ser susceptible a errores. Las dificultades que se presentan en el proceso de evaluación del estado de los mangos, crean la necesidad de la búsqueda de automatizar estos procesos.

La presente investigación se centra en detectar e identificar automáticamente los defectos externos del mango usando descriptores color y textura en imágenes de mangos. Luego de la adquisición de imágenes de un repositorio de datos, estas son sometidas a un proceso de segmentación utilizando el método de Otsu. A las imágenes resultantes se aplicaron los descriptores de Histogramas de color y Gray-Level co-occurrence matrix, para la extracción de características de color y textura. Finalmente, se implementaron los clasificadores 1-NN y Red neuronal, que usando las características extraídas, permiten inferir en el tipo de clasificación de mangos que presente en cada imagen analizada.

Palabras Claves: Visión Artificial, Procesamiento de Imágenes, Descriptores de Forma.

Abstract

The assessment of the state of the handles are usually made with human vision, made by employees based on experience, which can be error prone. The difficulties that the process of assessing the state of the handles are present, create the need for finding automate these processes.

This research focuses on automatically identify external defects using descriptors handle shape, color and texture of handles images. Following the acquisition of images of a database, these are subjected to a segmentation process using Otsu's method. In the resulting images descriptors Blurred Shape Model (BSM), Gray-Level, RGB, were applied to extract features of shape, color and texture. Finally, one-NN classifiers, which, using the extracted features, which allow us to infer the type of classification of mangoes present in each analyzed image is implemented.

Key Words: Computer Vision, Image Processing, Shape Descriptors.

INTRODUCCIÓN

El trabajo de investigación realizada trata sobre las técnicas de visión artificial para detectar defectos del mango. Existen investigaciones con relación a las técnicas como la que realizo Sánchez Martínez Hugo, Morales García Ana. (2009) con el nombre “Sistema de Visión Artificial para la inspección, selección y control de calidad de fresas”, que consistía si la fresa presenta defectos en la forma de su ápice aplicando para esto el método de la signatura, también la medición del diámetro máximo (cercano al cáliz) de la fresa estableciendo previamente la relación correspondiente entre pixeles y milímetros. El problema que estudia el trabajo de investigación es ¿de qué manera las técnicas de visión artificial se podrá detectar automáticamente los defectos externos del mango?. La hipótesis de trabajo de investigación es sí se utiliza las técnicas de visión artificial entonces se detectara automáticamente los defectos externos del mango, puesto que el objetivo es Detectar automáticamente defectos externos del mango utilizando técnicas de visión artificial. Los resultados se basaron con respecto a los indicadores las cuales son Precisión, Revocación y valor-F y los algoritmos de clasificación que se utilizaron fueron K-NN y Red Neuronal. De tal manera que el algoritmo que obtuvo mejor efectividad al clasificar es el algoritmo de Red Neuronal.

CAPÍTULO I: PROBLEMA DE INVESTIGACIÓN

1.1. Situación Problemática

A nivel Mundial la producción del mango, se encuentra en países importantes como la India, Indonesia, Florida, Hawai, México, Sudáfrica, Queen Island, Egipto, Israel, Brasil, Cuba, Filipinas y otros numerosos países. Probablemente la India tiene más plantaciones comerciales que el total del resto del mundo. Sin embargo, la importancia económica real del mango estriba en el tremendo consumo local que se realiza en cada villa y ciudad de las tierras bajas de los trópicos, ya que se trata de una de las plantas más fructíferas de los países tropicales. Esta especie se cultiva en todos los países de Latinoamérica, siendo México el principal país exportador del mundo. **(ASERCA, 2014)**

A nivel Nacional, Piura concentra la mayor producción de mango, con un 75 por ciento; seguida de Lambayeque con un 15 por ciento, y Ancash con 10 por ciento. De acuerdo a las variedades de mango con mayor fuerza para exportación, el mango Kent es el que representa el 95 por ciento de las exportaciones de este fruto. Es la variedad de mango que más se cosecha y más se exporta, precisó el ejecutivo Juan Carlos Rivera a la Agencia Andina **(Sunat, 2013)**.

Figura 1: Valor de exportación de principales frutas frescas periodo 2000 - 2010

PRODUCTOS	2000	2005	2006	2007	2008	2009	2010
Mango	23.30	38.40	61.70	65.90	81.80	80.20	110.70
Mandarina	6.30	12.00	14.70	16.70	24.50	23.60	35.10
Naranja	0.00	0.00	0.00	3.60	7.20	2.20	4.00
Tangelo	0.00	5.90	7.70	13.30	16.90	15.50	15.80
Palta	4.80	23.40	38.80	47.00	72.70	67.60	84.60
Plátano (banana)	6.20	17.60	26.10	31.20	45.60	51.30	55.60
Uva	6.00	35.20	50.60	60.60	83.40	135.70	180.00
TOTAL	46.60	132.50	199.60	238.30	332.10	376.10	485.80

Fuente 1: SUNAT elaborado por PROMPERU

A nivel Regional, el gerente regional de Agricultura de Lambayeque, Ricardo Velezmoro Ruiz, manifestó que en esta campaña los productores de mango necesitan mayor capacitación para poder exportar sus productos. El funcionario explicó que los productores que se ubican en las zonas de Motupe, Olmos y Chochope, zonas donde más se produce la tradicional fruta, hasta el momento no han recibido las orientaciones técnicas para el procesamiento y envasado del producto que les permita exportar en mayor porcentaje. En estas zonas se siembran un promedio de 12 mil hectáreas de cultivo de mango todos ellos con capacidades de exportación. De toda la producción, el 80% es dedicado a la exportación y el 20% es distribuido en jabas al mercado local (APEM, 2014).



Figura 2: *Producción de Mangos por Regiones*



Fuente 2: *MINAG - Series Históricas de Producción Agrícola*

Como se ve, la exportación de mangos es una fuente de ingresos importante para la economía del país, sin embargo los procesos de selección y control de calidad aún se realizan de manera manual, y es importante incorporar tecnología a este proceso y se han hecho muchos esfuerzos para lograrlo desde distintas maneras, entre ellos el procesamiento digital de imágenes pero aun no son suficientes, estos aportes son de muchos investigadores que están trabajando en esto tales como:

García Garino. C, Mirasso Aníbal, Mario A. Storti, Miguel E. Tornello (2013), en su trabajo de investigación “Ponderación de calidad en frutas usando técnicas de visión artificial para la estimación de daños”. Se presentó la clasificación de los frutos de acuerdo a requerimientos diversos, como pueden ser tamaño, calibre y color. La inspección era visual y suele ser llevada a cabo por personal

experimentado. Se trata, sin embargo, de una tarea repetitiva, propensa a errores, que acarrea perjuicios económicos, pero que pudieron ser automatizados usando técnicas de visión artificial. El problema que enfrentaron era la eliminación de brillos y sombras. Además, los resultados son sólo aproximaciones, ya que deben introducirse modificaciones necesarias de manera que el sistema no contabilice como área dañada ni el pedúnculo ni el cáliz de la fruta. Puesto que el color es uno de los atributos más importantes en las frutas y verduras, y que influye directamente en la decisión de los consumidores para aceptar o rechazar un determinado producto.

Moreno Bermúdez. A, Ballesteros Padilla. D, Sánchez Torres. G (2012), trabajo en “Un prototipo mecánico para la automatización del proceso de selección del mango tipo exportación”. Los componentes adicionales como el tiempo incorporado por el proceso de transporte permanecieron constantes independientemente de la clasificación del fruto y del número de imágenes. Sin embargo, no se consideraron características tales como el largo de la banda, la velocidad de rotación de los motores que las activan, y la longitud del clasificador final. Futuras mejoras estarán dirigidas hacia la disminución de los tiempos obtenidos. Para esto un esquema de procesamiento paralelo diseñado para el análisis de diferentes imágenes simultáneamente, o el análisis de vistas individuales mediante varios

componentes de procesamiento simultáneo como la tecnología CUDA (Compute unified device architecture), por el cual ayudo a disminuir el tiempo con componentes computacional. La optimización de las longitudes de las bandas transportadoras y del clasificador final constituye otro aspecto de diseño que permitió disminuir los tiempos del prototipo propuesto **(Moreno Bermúdez .A, 2012)**.

Sánchez Martínez Hugo, Morales García Ana. (2009), trabajo en un “Sistema de Visión Artificial para la inspección, selección y control de calidad de fresas”. Los investigadores se centraron en determinar la calidad en fresas. El sistema trabajó sobre imágenes previamente adquiridas utilizando para su captura la cámara fotográfica Canon Powershot A 430, y un sistema de iluminación de tipo dia nublado con leds como fuentes de luz. El procesamiento de estas imágenes tiene como primer paso la segmentación de la fresa de fondo para lo cual se aplicó el método de Umbralización trabajando en el canal H del modelo de color HSI. Posteriormente se realizó la extracción de características para la clasificación de la fresa donde se llega a detectar la presencia de diferentes tipos de defectos superficiales gracias a que las propiedades de color que dichas imperfecciones presentan permiten diferenciarlos del resto de la fresa. También se estableció si la fresa presenta defectos en la forma de su ápice aplicando para esto el método de la signatura. Además se

determinó el grado de maduración de la fresa separando la región correspondiente al área madura gracias a la caracterización del color propio de esta zona, y midiendo el porcentaje que representa respecto al área total del fruto. Se realizó también la medición del diámetro máximo (cercano al cáliz) de la fresa estableciendo previamente la relación correspondiente entre pixeles y milímetros (**Sánchez Martínez. H, 2009**).

Vassallo Barco M. (2015), en su trabajo de investigación **“Detección automática de deficiencias nutricionales en imágenes de hojas de café usando descriptores de forma y textura”**. El investigador del artículo se centró en la identificación de deficiencias nutricionales utilizando técnicas de reconocimiento de imágenes era una tarea complicada debido a la enorme cantidad y variedad de síntomas visibles que presentan las hojas que poseen ciertos niveles de deficiencias nutricionales a nivel foliar, además de que pueden presentarse muchas a la vez. La dificultad que acarrea es la deficiencia nutricional con la que se tuvo más problemas fue con Calcio (Ca), que en una considerable cantidad de pruebas obtuvo evaluaciones bajas de FMeasure (por debajo del 30%) (Vassallo Barco, 2015).

Vilcherrez Chavarry K. (2015), en su trabajo de investigación **“Procesamiento de Imágenes utilizando Descriptores de Color**

MPEG-7". Los descriptores que se utilizaron en este trabajo producen un alto acierto sobre la deficiencia de Nitrógeno y Fosforo. Hierro, Potasio, Magnesio y Manganese por otra parte es también la otra deficiencia con un grado de acierto promedio (regular).

El descriptor CLD produce mejor grado de acierto en deficiencias bien clasificadas, su diferencia con el descriptor DCD es de apenas un 13% por medio del algoritmo Multi Layer Perceptron y un 8% en KNN, lo cual resume que ambos descriptores son compactos para la extracción de característica de color sobre una imagen, pero en cierto grado Color Layout Descriptor, ofreció un mejor grado de acierto. La invariabilidad de colores verdes entre las deficiencias de hierro, magnesio y manganeso genera un margen de error producido por la cantidad de variaciones de verde que presentan las mencionadas deficiencias. Uno de los otros descriptores de color de MPEG-7 (Color Structure Descriptor), permite medir y manejar mejor esas variaciones de color por medio de diferentes niveles de resolución de imágenes (**Vicherrez Chavarry, 2015**).

El procesamiento de imágenes es una solución computacional que puede generar buenos resultados, uno de esos casos es segmentación por color, que se propuso utilizar esta técnica por tener variabilidad entre múltiples defectos externos del mango.

El encontrar una solución a este problema agrícola, que puede presentarse en diferentes tipos de mangos, ayudará como aporte

tecnológico al sector agrícola y de guía para otras investigaciones futuras. Para la agronomía peruana, resolver este tipo de problemas, a largo o a corto plazo ya es un gran avance para el desarrollo de ésta área y desde el punto de vista computacional, relacionar inteligencia artificial con otras áreas como la agronomía, consigue poder resolver problemas de la vida real en las que se busca expandir las N soluciones que se pueden presentar para problemas que normalmente se resuelven sin tecnología.

1.2. Formulación del Problema

¿De qué manera las técnicas de visión artificial se podrán detectar automáticamente los defectos externos del mango?

1.3. Delimitación de la Investigación

Esta investigación busca detectar los defectos externos del mango que nos enfocaremos en solo 4 defectos para estudiarlos más a fondo que son: Antracnosis, lenticelas oscuras esparcidas, manchas necróticas y manchas oscuras .El sistema podrá Detectar y clasificarlas mas no clasificara otros defectos que no se hayan estudiado. Por otro lado, los expertos en agronomía dan sus diagnósticos, pero no confirma que otros expertos de otros lugares verifiquen que sea correcto o no, el punto de vista de uno u otro siempre será distinto, por lo que siempre habrá un porcentaje de error.

1.4. Justificación e Importancia de la Investigación

Justificación Tecnológica

En este proyecto de investigación se evaluó las técnicas de visión artificial para lograr la selección de categoría del mango que aportara al mejoramiento de reconocimientos de los defectos externos del mango, para automatizar los procesos de selección agroindustrial.

Justificación Económica

La elaboración del prototipo debe pasar por etapas los cuales son: investigación, innovación y desarrollo. El presente proyecto se centra en la primera parte que es investigación sobre el problema de reconocimiento automático de defectos externos del mango. Posiblemente si procede la siguiente etapa puede generar un impacto económico.

1.5. Limitaciones de la Investigación

Es que no se probaron en ambiente de producción real.

1.6. Objetivos

Objetivo General

Detectar automáticamente defectos externos del mango utilizando técnicas de visión artificial

Objetivo Especifico

- a) Identificar técnicas de Visión Artificial.
- b) Determinar defectos externos del mango.
- c) Seleccionar algoritmos de procesamiento de imágenes.
- d) Implementar algoritmos en lenguaje de programación
- e) Evaluar los algoritmos y las técnicas de visión artificial.

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes de la Investigación

García Garino. C, Mirasso Aníbal, Mario A. Storti, Miguel E. Tornello (2013). Ponderación de calidad en frutas usando técnicas de visión artificial para la estimación de daños. (papers). Universidad Nacional de Santiago del Estero. Mendoza – Argentina. Los investigadores se centraron en el desarrollo de técnicas para inspeccionar la calidad de productos agrícolas. El sistema está dividido en 3 etapas generales: pre-procesamiento, segmentación y conteo de objetos. pre-procesamiento: Se han realizado métodos de espacios de color RGB, Umbralización global, escala de grises. El resultado de esta investigación es utilizar una tecnología de visión por computador para automatizar la inspección de la calidad de frutas y verduras frescas. El empleo de esta tecnología en las centrales hortofrutícolas para inspeccionar de manera automática la calidad de la fruta fresca y clasificarla en categorías. En el modelo propuesto, deben realizarse ajustes en la adquisición de las imágenes, de manera de reducir el tiempo empleado en el pre-procesamiento, por ejemplo en la eliminación de brillos y sombras. Además, los resultados son sólo aproximaciones, ya que deben introducirse modificaciones necesarias de manera que el sistema no contabilice como área dañada ni el pedúnculo ni el cáliz de la fruta (**Ledda I. Larchera, 2013**).

El proyecto de investigación utilizó el método de espacio de color RGB empleando para eliminar el fondo y las sombras de la Fruta (Pera), en esta investigación también utilizaremos dicho método para eliminar sombras, brillos de la Fruta (Mango) para posterior proceso la imagen esté libre de ruido.

Adolfo Moreno Bermúdez, Deimer Ballesteros Padilla, Germán Sánchez Torres. 2012. Un prototipo mecánico para la automatización del proceso de selección del mango tipo exportación (papers). Universidad de Medellín – Colombia. La investigación se centró en diseñar un prototipo mecánico para automatizar los procesos de clasificación del mango de azúcar. Este prototipo está constituido por un sistema de bandas que permite transportar y rotar el fruto para que un sistema de adquisición capture diferente caras de este. Las imágenes obtenidas se analizaron mediante técnicas computacionales que permiten obtener una aproximación tridimensional, lo que posibilita la estimación del volumen de cada fruto. Los resultados de la operación del mecanismo, y la aplicación de los métodos mostraron una precisión superior al 90%. Para la evaluación del clasificador global se realizó un análisis de indicadores de desempeño derivados de la matriz de confusión construida a partir de los resultados obtenidos de un grupo de 150 mangos previamente clasificados por parte de un experto. El método

de clasificación global presentó una exactitud promedio de 90,5%: de los 150 mangos 136 fueron clasificados correctamente (**Adolfo Moreno Bermúdez, 2012**).

Se concluyó Los componentes adicionales como el tiempo incorporado por el proceso de transporte permanecen constantes independientemente de la clasificación del fruto y del número de imágenes. Sin embargo, no se consideraron características tales como el largo de la banda, la velocidad de rotación de los motores que las activan, y la longitud del clasificador final, debido a que sí inciden en los tiempos empleados para el análisis de los frutos.

Futuras mejoras estarán dirigidas hacia la disminución de los tiempos obtenidos. Para esto un esquema de procesamiento paralelo diseñado para el análisis de diferentes imágenes simultáneamente, o el análisis de vistas individuales mediante varios componentes de procesamiento simultáneo como la tecnología CUDA (Compute unified device architecture), podrían ayudar a disminuir el tiempo del componente computacional. La optimización de las longitudes de las bandas transportadoras y del clasificador final constituye otro aspecto de diseño que permitiría disminuir los tiempos del prototipo propuesto.

Nora La Serna Palomino, Ulises Román Concha. 2007. Técnicas de Segmentación en Procesamiento Digital de Imágenes (papers). Universidad Nacional Mayor de San Marcos, Perú – Lima. En la

investigación no se detalló el problema, pero se realizó una investigación informativa en el tema de procesamiento de imágenes digitales. Los investigadores emplearon métodos como la segmentación subdivide una imagen en sus partes constituyentes u objetos, con el fin de separar las partes de interés del resto de la imagen de acuerdo al tipo del problema que se va a resolver. Fundamentalmente, se han revisado los siguientes tópicos en segmentación: Detección de bordes, Umbralización, y Basado en regiones (**La Serna Palomino .N, 2007**).

La detección de bordes es la técnica para detectar discontinuidades significativas en el nivel de gris se basan en la diferencia que experimenta una característica en dos regiones adyacentes y que indican la existencia de un borde. En esta investigación utilizaremos la técnica Canny para detectar el borde externo del mango, blurred shape model para detectar la forma del mango.

Sánchez Martínez Hugo, Morales García Ana. 2009. Sistema de Visión Artificial para la inspección, selección y control de calidad de fresas. (papers). Universidad Industrial de Santander. Bucaramanga - Colombia.

Los investigadores se centraron en determinar la calidad en fresas. El sistema trabaja sobre imágenes previamente adquiridas utilizando para su captura la cámara fotográfica Canon Powershot A 430, y un sistema

de iluminación de tipo día nublado con leds como fuentes de luz. El procesamiento de estas imágenes tiene como primer paso la segmentación de la fresa de fondo para lo cual se aplica el método de umbralización trabajando en el canal H del modelo de color HSI. Posteriormente se realiza la extracción de características para la clasificación de la fresa donde se llega a detectar la presencia de diferentes tipos de defectos superficiales gracias a que las propiedades de color que dichas imperfecciones presentan permiten diferenciarlos del resto de la fresa. También se establece si la fresa presenta defectos en la forma de su ápice aplicando para esto el método de la signatura. Además se determina el grado de maduración de la fresa separando la región correspondiente al área madura gracias a la caracterización del color propio de esta zona, y midiendo el porcentaje que representa respecto al área total del fruto. Se realiza también la medición del diámetro máximo (cercano al cáliz) de la fresa estableciendo previamente la relación correspondiente entre pixeles y milímetros **(Sánchez Martínez. H, 2009)**.

Los resultados obtenidos con el sistema implementado demuestran la eficacia del mismo presentando una efectividad mínima del 95% en el caso de la clasificación según el calibre de la fresa, 99% en la clasificación según el estado de madurez y 100% para la clasificación según la categoría de la fresa.

La técnica que utilizaron es la Umbralización consiste en separar la zona de interés del resto de la imagen. La relación de la investigación se caracteriza el color de la fresa y nuestra investigación utilizara la misma técnica para separar cierta parte de la imagen para obtener mejores resultados.

Vassallo Barco Marcelo. 2015. Detección automática de deficiencias nutricionales en imágenes de hojas de café usando descriptores de forma y textura. (papers). Universidad Señor de Sipán. Lambayeque - Perú. El investigador se centró en la identificación de deficiencias nutricionales utilizando técnicas de reconocimiento de imágenes afectando la producción y por tanto es importante identificar en forma temprana estas deficiencias. En particular las deficiencias nutricionales de Boro (B), Calcio (Ca), Hierro (Fe) y Potasio (K). Los métodos propuestos son: descriptores de forma y textura en imágenes de hojas de cafeto. Luego de la adquisición de imágenes de hojas de café, estas son sometidas a un proceso de segmentación utilizando el método de Otsu. A las imágenes resultantes se aplicaron los descriptores Blurred Shape Model (BSM) y Gray-Level Co-occurrence Matrix (GLCM) para la extracción de características de forma y textura respectivamente. Finalmente, se implementaron los clasificadores 1-NN, Naive Bayes y Neural Network que, usando las

características extraídas, los cuales permiten inferir en el tipo de deficiencia que presente en cada imagen analizada.

Se obtuvo resultados identificando deficiencias de Boro (B) y Hierro (Fe). Esto puede deberse a que los síntomas correspondientes a deficiencias de boro y de hierro son más notables que en otras deficiencias, por lo que son más fácilmente identificables. Utilizando descriptores de forma, se obtuvo mejores resultados identificando deficiencias de Boro, mientras que utilizando descriptores de textura, se obtuvo mejores resultados identificando deficiencias de Hierro. Así, también se pudo identificar deficiencias nutricionales de Potasio (K) con un grado de acierto aceptable. La deficiencia nutricional con la que se tuvo más problemas fue con Calcio (Ca), que en una considerable cantidad de pruebas obtuvo evaluaciones bajas de FMeasure (por debajo del 30%) **(Vassallo Barco, 2015)**.

La técnica que utilizaron blurred shape model (BSM) para diferenciar las hojas por cada deficiencia, mientras nuestra investigación se basará en las características de color y textura.

Vilcherrez Chavarry Kevin. 2015. Procesamiento de Imágenes utilizando Descriptores de Color MPEG-7 (Peapers). Universidad Señor de Sipán. Lambayeque - Perú. El investigador se centró en las deficiencias nutricionales en plantas de café se identifican visualmente mediante los cambios en coloración de las hojas. Los métodos que

empleo fueron: Descriptores de color del estándar MPEG-7: Dominant Color Descriptor (DCD) y Color Layout Descriptor (CLD) para identificar deficiencias de hierro, potasio, magnesio, manganeso, nitrógeno y fósforo en imágenes de hojas de café. Inicialmente se identifican cinco colores dominantes, presentes en las regiones de interés usando el descriptor DCD, el cual está basado en el algoritmo de K-means. El descriptor CLD extrae un vector de característica basado en la transformada del coseno discreto (DCT). Los valores de los descriptores fueron usados como entrada al algoritmo Multi Layer Perceptron (MLP) de WEKA, y el clasificador K-nearest neighbors (KNN). Los descriptores fueron seleccionados por su rapidez y robustez al extraer la mayor cantidad de características del color de las imágenes.

Los resultados obtenidos es que se utilizó el algoritmo Multi Layer Perceptron y el clasificador KNN para verificar la cantidad de aciertos efectivos por los descriptores CLD y DCD. Después del análisis de 180 imágenes entre muestras de hierro, potasio, magnesio, manganeso y nitrógeno; en conjunto con las muestra de prueba, se contó un total de 751 muestras usadas, durante el entrenamiento. Luego para la etapa de pruebas se utilizaron 50 muestras nuevas para ser clasificadas. Como resultados se presentan la cantidad de imágenes analizadas, así como la cantidad de aciertos y fallos, las métricas Precisión, Recall, F-Measure y los True-False Posetive Rat.

Multi Layer Perceptron (MLP) : Se utilizó el algoritmo Multi Layer Perceptron del framework WEKA, para verificar la cantidad de aciertos efectivos por los descriptores CLD y DCD después del análisis de 751 imágenes, entre muestras de hierro, potasio, magnesio, manganeso, nitrógeno y fosforo. En la implementación del descriptor Color Layout Descriptor (CLD). Se obtuvo un porcentaje de aciertos correctos de 84.667% y un 15.333% de aciertos incorrectos. Se obtuvo un coeficiente kappa de 0.79.

K-nearest neighbors (KNN): Mediante la clasificación por KNN, se usó una clasificación por un vecino. En la siguiente tabla se describe los resultados mediante el descriptor Color Layout Descriptor, se obtuvieron un porcentaje de 80% de aciertos correctos y un 20% de incorrectos. Como resultado de la siguiente tabla, se obtuvo un coeficiente kappa de 0.76. El descriptor CLD produce mejor grado de acierto en deficiencias bien clasificadas, su diferencia con el descriptor DCD es de apenas un 13% por medio del algoritmo Multi Layer Perceptron y un 8% en KNN, lo cual resume que ambos descriptores son compactos para la extracción de característica de color sobre una imagen, pero en cierto grado Color Layout Descriptor, ofrece mejor grado de acierto.

La invariabilidad de colores verdes entre las deficiencias de hierro, magnesio y manganeso genera un margen de error producido por la cantidad de variaciones de verde que presentan las mencionadas

deficiencias. Uno de los otros descriptores de color de MPEG-7 (Color Structure Descriptor), permite medir y manejar mejor esas variaciones de color por medio de diferentes niveles de resolución de imágenes. La técnica K-NN se utilizó en la clasificación de deficiencias nutricionales en plantas de café, la relación de la investigación se empleara la misma técnica para obtener resultado de las características como forma y textura (Vicherrez Chavarry, 2015).

2.2. Estado del Arte

La primera vez que se hizo uso de las técnicas de imágenes digitales fue en el 1920 en la transmisión de imágenes de fotografías periodísticas a través de cable submarino entre Londres y New York. Esto supuso que el tiempo entre emisión y recepción de las imágenes pasase de una semana (por barco) a tres horas (por cable). A partir de esto y durante algún tiempo, se desarrollaron las técnicas de codificación y reproducción de las imágenes logrando pasar en la representación de imágenes de cinco a quince tonos de gris.

1950 hasta 1960 en que aparecen los primeros computadores digitales y la necesidad de disponer de técnicas para la transmisión y procesamiento de imágenes desde satélite, cuando estas técnicas empiezan a ser desarrolladas de forma sistemática.

En 1967 MacQueen creó el algoritmo K-means es el algoritmo de clustering más conocido y utilizado ya que es de muy simple aplicación y eficaz. Sigue un procedimiento simple de clasificación de

un conjunto de objetos en un determinado número K de clústeres. El algoritmo más común utiliza una técnica de refinamiento iterativo.

En 1997 se realizaron los procesos de clasificación manual existe un riesgo relativamente elevado de error humano, ya que las decisiones tomadas por los operarios se ven afectadas por factores psicológicos tales como la fatiga o los hábitos adquiridos. Un estudio llevado a cabo con diferentes variedades de manzanas, donde personal cualificado comparó varios parámetros de forma, tamaño y color, mostró la limitada capacidad humana para reproducir la estimación de la calidad, lo que los autores definen como “inconsistencia”.

En el año 2002 un estudio llevado por John B. Njoroge. Kazunori Ninomiya. Naoshi Kondo y Hideki Toita del Departamento de Desarrollo Tecnológico de Japón utiliza técnicas de procesado de imagen digital para mejorar el transporte y clasificación de la fruta. Su sistema se basa en la inspección interna y externa de la fruta ya que se analiza tanto el color del fruto como su contenido en ácidos y azúcar. Las imágenes obtenidas a partir de tres cámaras son procesadas por tres ordenadores diferentes. Asimismo disponen de un aparato de Rayos-X para detectar los defectos biológicos de la fruta y varios sensores que analizan azúcares y ácidos. La detección de defectos se hace analizando el nivel de componente azul de las imágenes tomadas,

comparándolo con las componentes sumadas R, G y B y siempre teniendo en cuenta la iluminación en las imágenes.

En el mismo año un estudio denominado Sistema inteligente de selección de fruta basado en el estudio del color con procesado de imagen” realizado por Guo Feng y Cao Qixin, del Instituto de Investigación de Robótica de la Universidad Jiao Tong de Shanghai, China. Sus estudios se han orientado hacia la segmentación de la fruta, la eliminación de ruidos y la detección de contornos. El primer paso de la técnica aplicada es segmentar la imagen aplicando una binarización (estableciendo un umbral que determina si el píxel a tratar será blanco o negro) sobre la imagen convertida a espacio de color OHTA (de RGB a OHTA). El cambio de RGB a OHTA es lineal, cosa que no sucede con las conversiones a HSV, HS, etc. Se logró encontrar un conjunto de características ortogonales de color que se llamaron a partir de entonces “espacio de color Ohta”. Este espacio de color casi no tiene coste computacional y su resultado es bastante satisfactorio. Después se aplica un algoritmo especial que elimina ruido en la imagen binaria para después aplicar un algoritmo de detección de contornos y así poder segmentar la imagen de manera robusta.

El siguiente estudio de Desarrollo de una máquina con sistema de visión en tiempo real para un robot recolector de manzanas es realizado por D.M Bulanon, T. Kataoka, H. Ukamoto y S. Hata de la

Escuela de Graduación en Agricultura, de la Universidad de Hokkaido en Sapporo, Japón, es muy completo puesto que el objetivo final es la creación de un robot que recoja las manzanas directamente del árbol. La parte de procesamiento de imagen que incluye este estudio incorpora explicaciones sobre cómo se distingue el fruto del resto del árbol; es decir, cómo decide el aparato aquello que debe recolectar. Inicialmente hace falta decir que el robot incorpora un sistema de visión que primero adquiere la imagen, gradúa la luminosidad y realiza un análisis y procesamiento digital de la imagen. Para la realización de la segmentación se utiliza una técnica denominada “decisión - acercamiento teórico” que consiste en una función que discrimina valores o parámetros según el reconocimiento de un cierto patrón: esta técnica actuará de una u otra manera en función de lo que detecta. Tras segmentar la fruta se procede a aplicar un filtro paso bajo para eliminar el ruido de la imagen. El robot no detecta defectos, pero sí es capaz de discernir entre fruta y resto de árbol: follaje, ramas, etc.

En el 2013 Los investigadores Alejandro Díaz Sotolongo, Danays Barbara Costa Alonso. Con el tema combinación de técnicas de procesamiento digital de imágenes e inteligencia artificial para el análisis de imágenes de mamografía. El resultado de esta investigación es el desarrollo de un algoritmo para la detección/clasificación del cáncer de mama en sus etapas iniciales a partir de imágenes de

mamografías, alcanzando resultados satisfactorios. Se demostró que con la correcta combinación de características de forma y textura, se obtienen resultados relevantes en la detección/clasificación de las diferentes lesiones patológicas presentes en las imágenes de mamografías **(Alejandro Díaz Sotolongo, 2013)**.

Colombia. Según los autores Adolfo Moreno Bermúdez; Deimer Ballesteros Padilla; Germán Sánchez Torres en su revista titulada “un prototipo mecánico para la automatización del proceso de selección del mango tipo exportación”, presenta que la industria de la exportación de frutas tropicales, en particular la del mango de azúcar, se caracteriza por el tratamiento manual en cada una de las etapas previas a su comercialización, particularmente, la aplicación de los procedimientos de clasificación acordes con los criterios estandarizados nacionales de calidad (NTC-5139). En este artículo de investigación se presenta el diseño de un prototipo mecánico para automatizar el proceso de clasificación del mango de azúcar (*Mangifera Indica L*) tipo exportación. Este prototipo está constituido por un sistema de bandas que permite transportar y rotar el fruto para que un sistema de adquisición capture diferente caras de este (Bermúdez, Padilla, & Torres, 2013). Las imágenes obtenidas se analizaron varios componentes de procesamiento simultáneo como la tecnología CUDA (Compute unified device architecture), por lo tanto ayudó a disminuir el tiempo del componente computacional, que permiten obtener una aproximación

tridimensional, lo que posibilita la estimación del volumen de cada fruto. Los resultados de la operación del mecanismo, y la aplicación de los métodos mostraron una precisión superior al 90%.

2.3. Bases teóricas Específicas

2.3.1. Clasificación del Mango

Categoría Extra

Debería estar exentos de materias extrañas visibles, daños por quemaduras de sol y de quemaduras producidas por látex. Podrán permitirse defectos superficiales muy leves, siempre y cuando no afecten el aspecto general del producto, su calidad y estado de conservación y presentación en el envase.

Categoría I

Debería estar exentos de materias extrañas visibles, daños por quemaduras de sol y de quemaduras producidas por látex. Podrán permitirse, sin embargo los siguientes defectos leves, siempre y cuando no afectan al aspecto general del producto, su calidad y estado de conservación y presentación en el envase:

- a) Defectos leves de forma

b) Defectos leves en la cascara como rozaduras, costras y cicatrices, que en su conjunto no excedan de 1 cm² para todos los calibres de mango. El área se verifica mediante escala milimétrica.

Categoría II

Podrán permitirse los siguientes defectos, siempre y cuando los mangos conserven sus características esenciales en lo que respecta a su calidad, estado de conservación y a su presentación en el envase:

- a) Defectos leves de forma.
- b) Defectos leves en la cascara como rozaduras, costras y cicatrices, que en su conjunto no excedan de 3 cm² para todos los calibres de mango. El área se verifica mediante escala milimétrica.

Se permite la presencia de lenticelas rojizas suberizadas esparcidas, así como el amarillamiento de las variedades de color verde, debido a una exposición directa a la luz solar, pero sin que exceda el 10 % de la superficie, siempre y cuando no existan señales de necrosis (blanco amarillento) (INDECOPI, 2002).

2.3.1.1. Defectos del Mango

A reserva de las disposiciones especiales y las tolerancias permitidas para cada categoría, los mangos deberán:

- a) Estar enteros;
- b) Ser de consistencia firme;

- c) Tener aspecto fresco;
- d) Estar sanos; excluyendo los productos afectados por pudrición o deterioro que impidan su consumo;
- e) Estar limpios y prácticamente exentos de materias extrañas visibles;
- f) Estar prácticamente exentos de secreción de látex;
- g) Estar exentos de manchas necróticas;
- h) Estar prácticamente exentos de quemaduras producidas por látex;
- i) Estar prácticamente exentos de daños por quemaduras de sol;
- j) Estar prácticamente exentos de defectos en la cáscara producidos por rozaduras;
- k) Estar exentos de magulladuras profundas;
- l) Estar exentos de daños causados por plagas;
- m) Estar exentos de daños causados por temperaturas bajas;
- n) Estar exentos de humedad externa anormal, salvo la condensación siguiente; a su remoción de una cámara frigorífica;
- o) Estar exentos de cualquier olor y/o sabor extraños;
- p) Estar suficientemente desarrollados y presentar un grado de madurez satisfactorio según la naturaleza del producto;
- q) Presentar forma característica de la variedad;
- r) Tener pedúnculo no inferior a 0,5 cm de longitud y el corte deberá ser transversal;
- s) Cumplir con las especificaciones fitosanitarias establecidas.

2.3.2. Modelos de Color

El modelo de color se define como un modelo de representación del color con respecto a los valores de intensidad.

El propósito de un modelo de color es facilitar la especificación de colores en algún formato estándar. Un modelo de color es una especificación de un modelo de coordenadas 3-D y un sub-espacio dentro de ese sistema donde cada color se representa por un punto único.

La mayoría de los modelos de color que se utilizan están orientados hacia el hardware como monitores o impresoras o aplicaciones de manipulación de color. El modelo orientado al hardware más común es el RGB (red-green-blue), el modelo CMY (cyan-magenta-yellow) para impresoras en color y el YIQ que es el estándar para la difusión de TV (Y = luminancia, I y Q son dos componentes cromáticas). Para la manipulación del color se suelen utilizar los modelos HSI (matiz, saturación e intensidad) y HSV (matiz, saturación y valor) **(Corral Martín, 2011)**

2.3.2.1. Espacio de Color RGB

El modelo RGB es el espacio de color más utilizado en muchas ramas y el que utilizan la mayoría de digitalizadores para construir una imagen

a color, por lo tanto, es importante en el campo de la inteligencia artificial (Gill & Torres, 2004).

El espacio RGB se representa como un cubo [Fig. 3] donde un color viene definido por la mezcla de valores de intensidad de tres colores primarios: rojo (R), verde (G) y azul (B). Un color viene descrito por la intersección de 3 coordenadas en el cubo, como se muestra en la imagen (Gill & Torres, 2004).

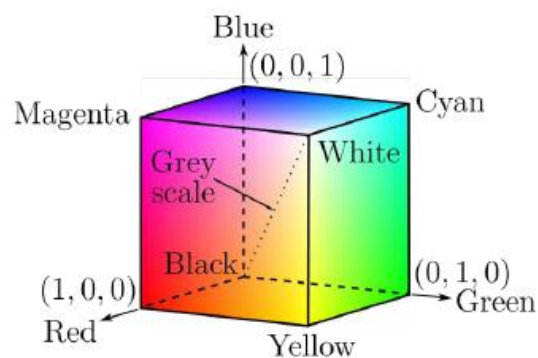


Figura 3: Representación Gráfica del espacio de color RGB

Este esquema de color es un modelo aditivo: las intensidades de los colores primarios se suman para producir otros colores. Cada punto de color contenido en los límites del cubo puede representarse como la tríada (R, G, B), donde los valores de R, G, B se asignan en el intervalo de 0 a 1.

Por ejemplo, el vértice magenta se obtiene sumando el rojo y el azul para producir la tríada (1, 0, 1) y el blanco en (1, 1, 1) es la suma de los vértices rojo, verde y azul. Las sombras de gris se representan a lo largo de la diagonal principal del cubo del origen (negro) al vértice blanco. Cada punto situado en esta diagonal tiene una contribución igual de cada color primario, de manera que una sombra de gris en medio del negro y el blanco se representan como (0.5, 0.5, 0.5) (Corral Martín, 2011).

2.3.2.2. Espacios de Color HSV

En el espacio de color HSV, los colores son especificados por los componentes de tono, saturación y valor. A menudo, como en los productos de Adobe y el API Java, el espacio HSV se llama HSB. Mientras que el acrónimo es diferente (en este caso B = brillo), denota el mismo espacio de color. El espacio de color HSV se muestra tradicionalmente.

Convertir de RGB a Color espacio, encontramos en primer lugar la saturación del RGB componentes de color $R, G, B \in [0, C_{max}]$ con C_{max} siendo el máximo Valor del componente (normalmente 255), como

$$S_{HSV} = \begin{cases} \frac{C_{rng}}{C_{high}} & \text{for } C_{high} > 0 \\ 0 & \text{de otra manera,} \end{cases}$$

y la luminancia (valor)

$$V_{HSV} = \frac{C_{high}}{C_{max}}$$

$$C_{high} = \max(R, G, B), C_{low} = \min(R, G, B), C_{rng} = C_{high} - C_{low}$$

Por último, tenemos que especificar el H_{HSV} valor de matiz. Cuando los tres componentes de color RGB tienen el mismo valor ($R = G = B$), entonces se trata de un (gris) pixel acromático. En este caso particular $C_{rng} = 0$ y por lo tanto el valor de saturación $S_{HSV} = 0$, en consecuencia, la tonalidad es indefinida. Para calcular H_{HSV} cuando $C_{rng} > 0$, primero normalizamos cada componente utilizando.

$$R = \frac{C_{high} - R}{C_{rng}}, \quad G = \frac{C_{high} - G}{C_{rng}}, \quad B = \frac{C_{high} - B}{C_{rng}},$$

Entonces, dependiendo de cuál de los tres componentes de color original tenía el valor máximo, que un cálculo una tonalidad H preliminar como:

$$H = \begin{cases} B - G \\ R - B + 2 \\ G - R + 4 \end{cases}$$

$$\text{if } R = C_{high}$$

$$\text{if } G = C_{high}$$

$$\text{if } B = C_{high}$$

Ya que el valor resultante para H se encuentra en el intervalo $[-1 \dots 5]$, se obtiene el valor final normalizando al intervalo $[0,1]$ como

$$H_{HSV} = \frac{1}{6} \begin{cases} (H + 6) & \text{for } H < 0 \\ H & \text{otra manera} \end{cases}$$

Por lo tanto, los tres componentes H_{HSV} , S_{HSV} , y V_{HSV} se encuentran dentro del intervalo $[0,1]$. El valor de matiz H_{HSV} , naturalmente, también puede ser calculado en otro intervalo de ángulo, por ejemplo en el intervalo de 0 a 360 usando

$$H_{HSV} = H_{HSV} * 360$$

Bajo esta definición, la unidad de espacio del cubo RGB se asigna a un cilindro con altura y el radio de longitud. En contraste con la representación tradicional, todos los puntos HSV dentro de todo el cilindro corresponden a coordenadas de color válidas en el espacio BRG. El mapeo de RGB al espacio HSV es no lineal, como se puede observar mediante el examen de cómo los tramos de punto negro.



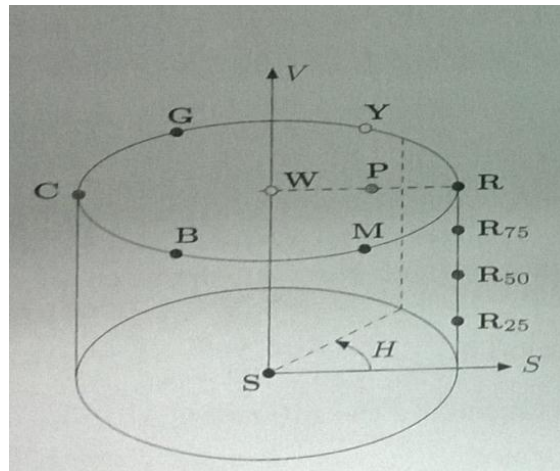


Figura 4: Mapeo de RGB al espacio HSV

2.3.2.3. Espacios TV Color – YUV, YIQ y YC_bC_r

Estos espacios de color son una parte integral de las normas que rodean la grabación, almacenamiento, transmisión y visualización de señales de televisión. YUV y YIQ son los métodos de codificación de color fundamentales para los sistemas NTSC y PAL analógicas y YC_bC_r es una parte de las normas internacionales que rigen la televisión digital. Todos estos espacios de color tienen en común la idea de separar el componente de luminancia Y a partir de dos componentes de crominancia y, en lugar de directamente codificar colores, la codificación de las diferencias de color. De esta forma, la

compatibilidad con los sistemas en blanco y negro de legado se mantiene mientras que al mismo tiempo el ancho de banda de la señal se puede optimizar mediante el uso de diferentes anchos de banda de transmisión para el brillo y los componentes de color.

2.3.2.3.1. Espacios de Color YUV

YUV es la base para la codificación de color utilizado en la televisión analógica, tanto en el Norte de América NTSC y el sistema PAL Europea. La componente de luminancia Y se calcula a partir de los componentes RGB como

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

Bajo la suposición de que los valores RGB ya han sido gamma corregida de acuerdo con el estándar de codificación de TV (NTSC = 2,2 Y PAL =2.8) para la reproducción. Los componentes UV se calculan a partir de una diferencia ponderada entre la luminancia y los componentes azules o rojas como

$$U = 0.492(B - Y) \quad y \quad V = 0.877(R - Y)$$

Y toda la transformación de RGB a YUV es:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

La transformación de YUV a RGB se encuentra invirtiendo la matriz en la Ecuación (19).



$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.000 & 0.000 & 1.140 \\ 1.000 & -0.395 & -0.581 \\ 1.000 & 2.032 & 0.000 \end{pmatrix} \cdot \begin{pmatrix} Y \\ U \\ V \end{pmatrix}$$

2.3.2.3.2. Espacios de Color YIQ

El sistema original NTSC utiliza una variante de1 YUV llama YIQ ("en fase", Q para "cuadratura"), donde al girarse y reflejarse de tal manera que tanto los Vectores de color U y V

$$\begin{pmatrix} I \\ Q \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \cos \beta & \sin \beta \\ -\sin \beta & \cos \beta \end{pmatrix} \cdot \begin{pmatrix} U \\ V \end{pmatrix}$$

Donde $\beta = 0.576(36^\circ)$. el componente Y es el mismo que en YUV. Aunque el YIQ tiene ciertas ventajas con respecto a los requisitos de ancho de banda que ha sido completamente reemplazado por YUV.

2.3.2.3.3. Espacios de Color YCbCr

El espacio de color $Y C_b C_r$ es una variante internacionalmente estandarizado de YUV que se utiliza tanto para la compresión de televisión y la imagen digital (por ejemplo, en JPGE). Los compontes croma $C_b C_r$ son (similares a U, V) valores de diferencia entre la luminancia y los componentes azul y rojo, respectivamente. En contraste con YUV, los pesos de los componentes RGB para la luminancia (Y) dependen explícitamente de los coeficientes utilizados para los valores de croma $Y C_b$ y C_r . Para pesos arbitrarios $w_B w_R$, la transformación se define como:

$$Y = w_R \cdot R + (1 - w_R - w_B) \cdot G + w_B \cdot B$$



$$C_b = \frac{0.5}{1 - w_B} \cdot (B - Y),$$

$$C_r$$

$$= \frac{0.5}{1 - w_R} \cdot (B - Y),$$

y la transformación inversa de YC_bC_r , RGB es

$$R = Y + \frac{1 - w_R}{0.5} \cdot C_r$$

$$G = Y - \frac{w_B \cdot (1 - w_B) \cdot C_b - w_R \cdot (1 - w_R) \cdot C_r}{0.5 \cdot (1 - w_B - w_R)}$$

$$B = Y + \frac{1 - w_B}{0.5} \cdot C_b$$

La recomendación UIT especifica el valor $w_R = 0.299$ y $w_B = 0.114$ ($w_G = 1 - w_B - w_R = 0.587$). El uso de estos valores, la transformación se convierte

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

y la transformación inversa se convierte

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.000 & 0.000 & 1.403 \\ 1.000 & -0.344 & -0.714 \\ 1.000 & 1.773 & 0.000 \end{pmatrix} \cdot \begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix}$$

Diferentes pesos se recomiendan en función de cómo se utiliza el espacio de color; recomienda $w_r = 0,2125$ y $w_b = 0,0721$ para ser utilizado en la producción de HDTV digital. Los valores de U, V, I, Q y w_b, C_r pueden ser tanto positivo o negativo para codificar C_b , valores de C_r a números digitales, un adecuado desplazamiento se añade



típicamente para obtener de sólo valores positivos, por ejemplo, $128 = 2^7$ en caso de componentes de 8 bits.

2.3.2.4. Espacios de Color para la Impresión CMY y CMYK

En contraste con el esquema aditivo RGB del color (y sus diversos espacios de color), la impresión en color hace uso de una luz sustractiva en esa ubicación. La impresión en color requiere un mínimo de tres colores primarios; tradicionalmente cian (C), se han utilizado magenta (M) y amarillo (Y) se han utilizado. Utilizando la mezcla de colores sustractiva sobre un fondo blanco, $C = M = Y = 0$ (sin tinta) resulta en el color blanco y $C = M = Y = 1$ (saturación completa de los tres enlaces) en el color negro. Una tinta de color cian absorberá rojo (R) más fuertemente, magenta absorbe verde (G) y amarillo absorbe el azul (B). La forma más simple del modelo CMY se define como

$$C = 1 - R,$$

$$M = 1 - G,$$

$$Y = 1 - B.$$

En la práctica, el color producido por saturar completamente las tres tintas no es físicamente un verdadero negro. Por lo tanto, los tres colores primarios de C, M, Y normalmente se complementan con una tinta de color negro (K) para aumentar la gama de colores y la cobertura (gama). En el caso más simple, la cantidad de negro es

$$K = \min(C, M, Y)$$

con el aumento de los niveles de negro, sin embargo, la intensidad de los componentes C, M, Y puede ser reducido gradualmente. Se han propuesto muchos métodos para reducir los tintes primarios y nos fijamos en tres de ellos en la siguiente.

CMY → CMYK (versión 1): En este simple variante de los valores de C, M, Y se reducen linealmente con el aumento de K y los componentes de C' modificada, M', Y', K' se definen como

$$\begin{pmatrix} C' \\ M' \\ Y' \\ K' \end{pmatrix} = \begin{pmatrix} C - K \\ M - K \\ Y - K \\ K \end{pmatrix}$$

CMY → CMYK (versión 2): La segunda variante corrige el color mediante la reducción de la C, los componentes M, Y por $S = \frac{1}{1-K}$, lo que resulta en colores más fuertes en las zonas oscuras de la imagen:

$$\begin{pmatrix} C' \\ M' \\ Y' \\ K' \end{pmatrix} = \begin{pmatrix} (C - K) \cdot s \\ (M - K) \cdot s \\ (Y - K) \cdot s \\ K \end{pmatrix}, \text{ con } S = \begin{cases} 1 & \text{para } K < 1 \\ 1 - K & \\ 1 & \text{de otra manera} \end{cases}$$

En ambas versiones, los componentes K (como se define en la ecuación (27)) se utiliza directamente sin modificación, y todos los tonos de gris (es decir, cuando R = G = B) se imprimen usando tinta K' de color negro, sin ninguna contribución de C', M', o Y'. En la práctica, las cantidades necesarias de negro K y C, M, Y dependen en gran medida del proceso de impresión y el tipo de papel usado que los trabajos de impresión se calibran de forma rutinaria individual

CMY → CMYK (versión 3): en la producción de impresión, funciones de transferencia espaciales se aplican para afinar los resultados. Componentes CMY y una función de negro generación separada FBG (K) para controlar la cantidad de negro. Estas funciones se utilizan en la forma

$$\begin{pmatrix} C' \\ M' \\ Y' \\ K' \end{pmatrix} = \begin{pmatrix} C - f_{UCR}(K) \\ M - f_{UCR}(K) \\ Y - f_{UCR}(K) \\ f_{BG}(K) \end{pmatrix}$$

donde $K = \min (C, M, Y)$ de nuevo (como se define en la ecuación (27)). Las funciones f_{UCR} y f_{BG} son por lo general no lineal, y los valores resultantes C', M', Y', K' se escalan (típicamente por medio de sujeción) en el intervalo $[0,1]$. Fue producida usando las funciones

$$f_{UCR}(K) = \begin{cases} 0 & \text{para } K < K_0 \\ K_{max} \cdot \frac{K - K_0}{1 - k_0} & \text{para } K \geq K_0 \end{cases}$$

donde $sk = 0,1$, $k_0 = 0,3$ y $k_{max} = 0,9$

con esta definición, reducir fucre los componentes CMY por 10% del valor K que afecta principalmente a las áreas oscuras de la imagen con altos valores de k.

2.3.3. Procesamiento de Imagen Digital

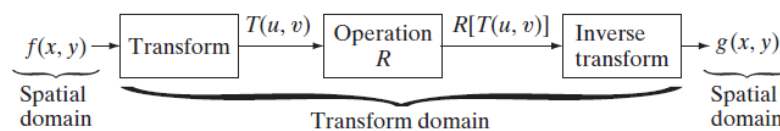
Todo el procesamiento de imágenes enfoques analizados hasta ahora operan directamente en píxeles de la imagen de entrada de; que es el



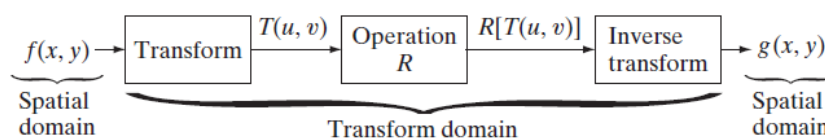
trabajo directamente en el dominio espacial. en algunos casos, las tareas de procesamiento de imágenes se formulan mejor mediante la transformación de las imágenes de entrada, llevando la tarea especificada en un dominio de la transformada, y la aplicación de la transformada inversa para volver a el dominio espacial. Una clase particularmente importante de transformaciones lineales 2-D, T denotado (u, v) , se puede expresar en la forma general.

$$T(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)r(x, y, u, v)$$

donde $f(x, y)$ es la imagen de entrada, se llama el núcleo transformación hacia adelante. Se evalúa para $u = 0,1,2 \dots, M - 1$ y $v = 0,1,2, \dots, N - 1$. Como antes, X e Y son variables espaciales, mientras que M y N



son las dimensiones de fila y de columna de f . Las variables u y v se llaman transformar variables $T(u, v)$ se denomina transformada directa de $f(x, y)$. Dada $T(u, v)$, Podemos recuperar $f(x, y)$ usando la transformada inversa de $T(u, v)$,



Para $x=0,1,2,\dots,M-1$ y $Y=0,1,2,\dots,N-1$, donde $s(x,y,u,v)$ es llamar el núcleo de transformación inversa. (Leemans, V., Magein, H., & Destain 1998).

2.3.3.1. Segmentación

En el campo de procesamiento de imágenes digitales, la segmentación es el proceso que consiste en dividir o particionar una imagen en sub regiones, componentes, partes u objetos, de manera que ayude en el análisis de la misma. La segmentación puede también ser definida como el encontrar, por medio de un algoritmo numérico, sub regiones homogéneas y sus bordes (Azuela, 2006).

El resultado de la segmentación es un número de regiones homogéneas, teniendo cada uno una etiqueta única. Una imagen se define así por un conjunto de regiones que están conectados y no superpuestos, de manera que cada píxel de la imagen adquiere una región única etiqueta que indica la región que pertenece (**Ajoy, 2005**).

Entre los métodos de segmentación de imágenes más relevantes tenemos el Método de Otsu, Fuzzy C-Means y K-Means.

a) K-Means

El nombre de K-means viene porque representa cada uno de los clusters por la media (o media ponderada) de sus puntos, es decir, por su centroide. La representación mediante centroides tiene la ventaja de que tiene un significado gráfico y estadístico inmediato. Cada clúster por tanto es caracterizado por su centro o centroide que se encuentra en el centro o el medio de los elementos que componen el cluster. Kmeans es traducido como K-medias. O un conjunto de objetos $D_n = (x_1, x_2, \dots, x_n)$, para todo el i , x_i reales y k , 1, los centros de los K clúster. El algoritmo del K-means se realiza en 4 etapas:

Etapas 1: Elegir aleatoriamente K objetos que forman así los K clúster iniciales. Para cada clúster k , el valor inicial del centro es $= x_i$, con los x_i únicos objetos de D_n pertenecientes al clúster.

Etapas 2: Reasigna los objetos del clúster. Para cada objeto x , el prototipo que se le asigna es el que es más próximo al objeto, según una medida de distancia, (habitualmente la medida euclidiana).

Etapas 3: Una vez que todos los objetos son colocados, recalculan los centros de K clúster.

Etapas 4: Repetir las etapas 2 y 3 hasta que no se hagan más reasignaciones. Aunque el algoritmo termina siempre, no se garantiza el obtener la solución óptima. En efecto, el algoritmo es muy sensible a la elección aleatoria de los K centros iniciales. Esta es la razón por la que, se utiliza el algoritmo del K-means numerosas veces sobre un mismo conjunto de datos para intentar minimizar este efecto,

sabiendo que a centros iniciales lo más espaciados posibles dan mejores resultados.

B) Segmentación por Umbralización

La Umbralización es una técnica efectiva para obtener la segmentación de imágenes donde estructuras diferentes tienen intensidades contrastantes u otras características diferenciables. Para imágenes escalares, los métodos interactivos pueden estar basados en la apreciación visual del usuario ya que la operación de Umbralización es implementable en tiempo real.

Al aplicar un umbral, T , la imagen en escala de grises, (x_i) quedará binarizada; etiquetando con '1' los píxeles correspondientes al objeto y con '0' aquellos que son del fondo **(U.N.Q., 2005)**

Su principal limitación es que en su forma más simple solo se generan dos clases y por lo tanto no se puede aplicar a imágenes multicanal. En adición, la Umbralización usualmente no toma en cuenta las características espaciales de la imagen, haciendo que sea sensible al ruido **(Coto, 2003)**.

2.3.4. Descriptores de Imágenes

Los descriptores visuales describen las características visuales de los contenidos dispuestos en imágenes o en vídeos. Describen

características elementales tales como la forma, el color, la textura o el movimiento, entre otros (**Boullosa, 2011**).

Respecto de la necesidad de describir el contenido de la información multimedia de forma objetiva y automatizada, surgen como respuesta los descriptores audio-visuales. Más concretamente en el caso de las imágenes podemos referirnos a los descriptores de imagen.

Idealmente, un descriptor visual debería poseer las siguientes propiedades

a) Simplicidad: El descriptor debería representar las características extraídas de la imagen de manera clara y sencilla para permitir una fácil interpretación de su contenido.

b) Repetibilidad: El descriptor generado a partir de una imagen debe ser independiente del momento en el que se genere.

c) Diferenciabilidad: Dada una imagen, el descriptor generado debe poseer alto grado de discriminación respecto de otras imágenes y al mismo tiempo contener información que permita establecer una relación entre imágenes similares.

d) Invarianza: Cuando existen deformaciones en la representación de dos imágenes, es deseable que los descriptores que las representan aporten la robustez necesaria para poder relacionarlas aún bajo diferentes transformaciones.

e) Eficiencia: Es deseable que los recursos consumidos para generar el descriptor sean aceptables para poder ser utilizados en aplicaciones con restricciones críticas de espacio y/o tiempo.

El algoritmo utilizado para extraer las características de las imágenes introducidas en el Sistema está basado en el descriptor BSM desarrollado por el centro de visión por Computador de la Universidad Autónoma de Barcelona.

2.3.5. Descriptores de forma

Los descriptores de forma se utilizan para describir un objeto valiéndose de las características que se pueden obtener según su forma, lo cual es posible gracias al uso de distintas técnicas de detección de bordes. Para esto generalmente, se utilizan características de bajo nivel debido a que estas pueden ser extraídas automáticamente de una imagen sin tener previamente información sobre la forma del objeto (**Nixon, 2002**).

Estas características de bajo nivel son utilizables en la extracción de características de más alto nivel.

2.3.5.1. Descriptores BSM (Blurred Shape Model)

Para detectar bordes se debe obtener la forma de la imagen a partir de la cual se calcula el BSM es necesario Obtener los bordes o contorno

de la imagen. Estos nos facilitan una valiosa información sobre las fronteras de los objetos que nos interesan reconocer del resto de la imagen.

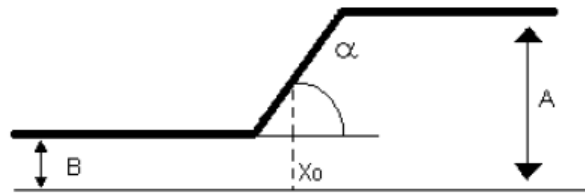


Figura 5: Modelo unidimensional y continuo de un borde ideal

La figura 5 muestra un modelo unidimensional y continuo de un borde. Este modelo representa una rampa desde un nivel de gris bajo B a uno alto A . H corresponde a la variación de la intensidad. Se calcula de la siguiente forma

$$H = A - B$$

donde α corresponde al ángulo de inclinación de la rampa α y X_0 corresponde a la coordenada horizontal X_0 donde se encuentra el punto medio de la rampa.

Un operador que proporcionara los valores de X_0 y H daría unos datos muy valiosos sobre la imagen, ya que proporcionaría la amplitud del borde, y la localizaría con exactitud dentro de la imagen.

En las imágenes reales los bordes nunca se ajustan totalmente al modelo anterior. Las causas que ocasionan este mal ajuste son diversas:

- a) Las imágenes son discretas
- b) Las imágenes están afectadas por ruido diverso
- c) El origen de los bordes puede ser muy diverso: bordes de oclusión, superficies de diferente orientación, distintas propiedades reflectantes, distinta textura, efectos de iluminación (sombras y/o reflejos), etc.

Debido a las circunstancias citadas anteriormente, el proceso de obtención de bordes de una imagen adquiere una gran complejidad.

Además existen en este proceso diferentes tipos de errores:

Error de detección: Un operador es un buen detector si la probabilidad de detectar el borde es alta cuando éste realmente existe en la imagen, y baja cuando no existe.

Error en la localización: Un operador localiza bien un borde cuando la posición que proporciona coincide con la posición real del borde de la imagen. Ambos errores están íntimamente ligados a los problemas comentados anteriormente, y sobre todo a la presencia de ruido en la adquisición de la imagen. En la práctica, la calidad de detección y localización están en conflicto.

Respuesta múltiple: Es el caso en que se detectan varios píxeles en un único borde.

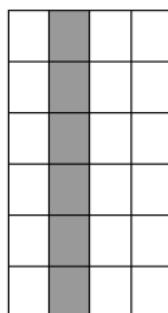


Fig. 2: Borde real

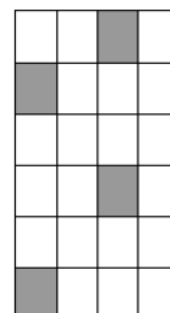


Fig. 3: Pobre detección

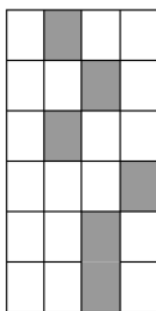


Fig. 4: Pobre localización

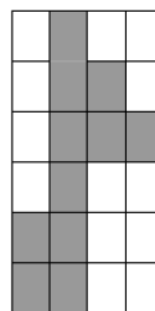


Fig. 5: Múltiple respuesta

La fig.2 muestra la posición de un borde real, como se ha comentado en el punto anterior para garantizar la extracción óptima de la forma del objeto nuestro localizador ha de proporcionarnos valores ajustados a este borde. En ocasiones, a causa de la distorsión o de la incorrecta elección del localizador los valores de los bordes obtenidos no son correctos. La fig. 3-5 son ejemplos de una obtención defectuosa de los valores del borde.

La mayoría de las técnicas utilizadas en la detección de bordes utilizan operadores locales basados en distintas aproximaciones discretas de la primera y segunda derivada de los niveles de grises de la imagen, si bien existen otras probabilidades para ello, como el empleo de

Insertar una imagen binaria I.
 Obtener la forma S contenida en I
 Dividir I en n x n subregiones de idénticas dimensiones

$$R = \{ r_1, \dots, r_{max} \}$$

Con c_i el centro de coordenadas de cada región
 Dejar $N(r_i)$ ser la región vecina de la región r_i , definida como:

$$N(r_i) = \{ r_k | r \in R, \| c_k - c_i \|^2 \leq 2 \times g^2 \}$$

donde g es el tamaño de la celda

Para cada punto $x \in S$
 Para cada $r_i \in N(r_x)$

$$d_i = d(x, r_i) = \| x - c_i \|^2$$

donde d es la distancia del punto al centroide

Fin Para
 Actualizar las probabilidades de las posiciones del vector v como:

$$v(r_i) = \frac{1}{D} \sum_{x \in N(r_i)} \frac{1}{\| x - c_i \|^2}$$

Fin Para
 Normalizar el vector v, de la siguiente forma:

$$v = \frac{v_i}{\sum_{j=1}^n v_j} \quad \forall i \in [1, \dots, n^2]$$


patrones de bordes ideales. Las librerías de procesamiento de imágenes OpenCV poseen ya algunos métodos que implementan la funcionalidad de los algoritmos comentados anteriormente. Como ejemplos de algoritmos para extraer los contornos de una imagen destacaríamos Kirsch N, Kirsch N-E , Canny, Skeleton , Sobel. El algoritmo que utiliza el sistema BSM para obtener los contornos o bordes de la imagen es el algoritmo de Canny.

Figura 6: *Algoritmo en pseudocódigo Blurred Shape Model*

2.3.5.2. Square Model Shape Matrix

Es un descriptor de forma también conocido como Grid Descriptor (Graf, 2012), tolerante a la rotación, traslación y escala de un objeto bidimensional. El descriptor muestra como salida una matriz binaria que contiene información relevante sobre la forma de un objeto de manera invariante, que puede ser utilizado con éxito como características para el reconocimiento de objetos (**Flusser, 1992**).

El algoritmo actúa de la siguiente manera (**Flusser, 1992**): para la forma S de un objeto, se construye un cuadrado centrado en el centro de gravedad G de S (centroide). El tamaño de cada lado es igual a $2L$, donde L es la distancia euclídea máxima de G a un punto M en el límite del contorno del objeto. El punto M se encuentra en el centro de un lado y GM es perpendicular a este lado. Seguidamente, se divide la imagen en regiones de $n \times n$, el cual a su vez también determina el tamaño de la matriz de salida, donde $S_{kj}, k = 1, \dots, N$, las subregiones de la matriz. Definir la matriz de forma $SM = [BKJ]$,

$$B_{kj} = \begin{cases} 1 & \leftrightarrow \mu(S_{kj} \cap S) \geq \mu(S_{kj})/2 \\ 0 & \text{de lo contrario} \end{cases}$$

Donde (F) es el área de la región plana F .

2.3.6. Algoritmo de Textura

La textura en una imagen tiene que ver mucho con la resolución. Lo que a cierta resolución son objetos claramente distinguibles, a una resolución menor se ve como cierta textura y una resolución aún menor puede parecer una región uniforme.

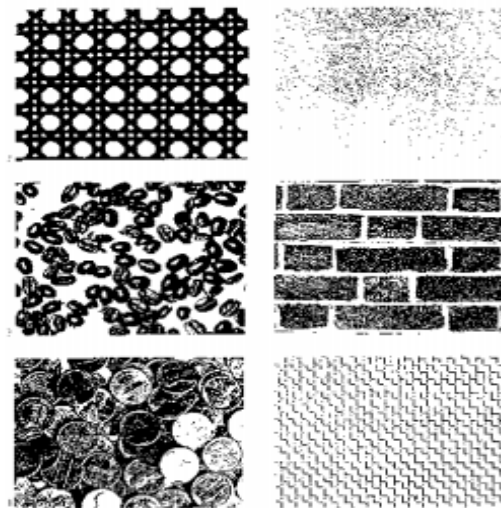
El analizar y reconocer diferentes tipos de textura es útil para el reconocimiento de ciertas clases de objetos e incluso en otros



aspectos de visión como la determinación de forma tridimensional (shape from texture).

La figura 7 muestra ejemplos de diferentes tipos de texturas. Para algunas de ellas los elementos básicos o primitivos son claramente distinguibles, como el caso de los ejemplos de las texturas de frijoles, ladrillos y monedas. Para los otros ejemplos es más difícil definir los elementos primitivos.

(Julesz,



1975).

Figura 7: Ejemplos de texturas: bejuco, papel, frijoles, ladrillo, monedas, trenza de alambre (de arriba a abajo, de izquierda a derecha).

2.3.6.1. Co-ocurrencia Matriz de cálculo

Matrices de co-ocurrencia es llevar información muy útil sobre la organización espacial de textura. Si es gruesa, su masa tiende a concentrarse alrededor de la diagonal principal, es decir, en los elementos de $C_{kl} |k - l| < t$. Si la textura es fina, los valores de la matriz de co-ocurrencia son mucho más extendido. Si textura lleva

información direccional fuerte, el co-ocurrencia matrices C_d tienden a tener su masa en la diagonal principal, para los vectores de desplazamiento d correspondiente a la dirección de la textura. Se han propuesto varios descriptores de textura para caracterizar el contenido de matriz de co-ocurrencia.

a) **probabilidad máxima**

$$P_d = \max_{k, l} C_{kl}$$

Este descriptor indica la más probable pareja de nivel de gris (f_k, f_l) , situada a una distancia d en la imagen

b) **entropía**

$$H_d = - \sum_{k=1}^N \sum_{l=1}^N C_{kl} \ln c_{kl}$$

Entropía se maximiza cuando la probabilidad conjunta $P(F_K, F_L d)$ se distribuye uniformemente

c) **Elemento-diferencia momento de orden m**

$$I_d = \sum_{k=1}^N \sum_{l=1}^N |k - l|^m C_{kl}$$

Este descriptor se minimiza cuando se concentra la masa matriz de co-ocurrencia alrededor de la diagonal principal, porque las diferencias $|k - l|^m$ son muy pequeñas. Se alcanza valores pequeños en el caso



de texturas gruesas. Este descriptor textura es equivalente a la inercia de la matriz de co-ocurrencia para $m = 2$.

La caracterización espectral de la textura de la imagen se basa ya sea en la función de auto correlación de una imagen en dos dimensiones o en su espectro de potencia. La función de auto correlación $R_{ff}(k, l)$ de una imagen $f(i, j)$ viene dada por la siguiente relación:

$$R_{jj}(k, l) = \frac{1}{(2N_1 + 1)(2N_2 + 1)} \sum_{i=-N_1}^{N_1} \sum_{j=-N_2}^{N_2} f(i, j)f(i + k, j + 1)$$

Puede calcularse tanto para retardos positivos y negativos (k, l) . Por lo general, alcanza un máximo de retraso cero $(0,0)$ y cae exponencialmente con (k, l) (positivo o negativo). La tasa de retorno es una indicación de la tosquedad textura. Que la textura es fina la tasa de retorno es grande, mientras que las pequeñas tasas de abandono fuera indican grandes elementos de textura la simetría circular de la función de auto correlación indica isotropía espacial.

Finalmente, la periodicidad en los picos de $R_{jj}(k, l)$ da una indicación de la periodicidad espacial de los patrones de textura. Se prefiere cálculo directo de la función de auto correlación para un pequeño número de retardos (k, l) utilizando la ecuación anterior. Se necesita el cálculo de $R_{jj}(k, l)$ para un gran número de retardos, la siguiente propiedad de la función de auto-correlación puede ser muy útil: (Sons, 2000).

$$R_{jj}(k, l) = IDFT[F(w_1, w_2)F^*(w_1, w_2)]$$

2.3.7. Clasificación Supervisada

En los modelos de clasificación supervisada, se parte de un conjunto de clases conocido para el entrenamiento. Estas clases deben caracterizarse en función del conjunto de variables mediante la medición de las mismas en individuos cuya pertenencia a una de las clases no presente dudas (áreas de entrenamiento). El objetivo principal de las técnicas de clasificación supervisada es obtener un modelo clasificatorio válido para permitir tratar casos futuros.

2.3.7.1. Vecino más cercano (1-NN)

La regla de clasificación por vecindad más simple es la regla de clasificación del vecino más cercano o simplemente 1-NN. Se basa en la suposición de que la clase del patrón a etiquetar, X , es la del prototipo más cercano en R , al que notaremos por X_{NN} .

$$d(X) = \omega_e \text{ si } \begin{cases} \delta(X, X_{NN}) = \min_{i=1..N} \{\delta(X, X_i)\} \\ (X_{NN, \omega}) \in R \end{cases}$$

2.3.7.2. Algoritmo K-NN (K-Nearest Neighbor)

K-NN es un método de clasificación muy simple pero a la vez muy potente. Para explicar su funcionamiento utilizaremos un ejemplo:



Supongamos que tenemos un conjunto de entrenamiento de N vectores de características. Estos vectores están agrupados en distintas clases (C_1, C_2, C_3, \dots). Supongamos también que se introduce en el sistema un nuevo vector de características. Nuestro objetivo sería determinar la clase a la que pertenece. Para determinarla el algoritmo K-NN busca los K vecinos más próximos al vector de entrada, y posteriormente realiza una valoración de las clases a las que pertenecen estos vectores vecinos. La valoración consiste en comparar las distancias existentes entre las clases. Los algoritmos utilizados en esta valoración pueden ser distintos, en nuestro sistema hemos utilizado la Distancia Euclídea. (Cuenca, 2008).

Distancia Euclídea

La distancia Euclídea es una de las métricas más utilizadas, ésta se obtiene a partir de una generalización del teorema de Pitágoras.

En un espacio tridimensional, la distancia euclídea entre 2 puntos (X_1, X_2, X_3) y (Y_1, Y_2, Y_3) viene dada por la siguiente ecuación:

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$

Más concretamente, la distancia euclidiana entre dos puntos p_1 y p_2 es:

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



2.3.7.3. Adaboost

Adaboost es un algoritmo utilizado para construir clasificadores sólidos utilizando la combinación lineal de clasificadores simples.

El primer paso consiste en generar los ejemplos, a éstos se les asigna el mismo peso ($1/m$). A medida que se genera un nuevo modelo se cambian los pesos de los nuevos ejemplos utilizados para el siguiente clasificador. El objetivo consiste en minimizar en cada iteración el error esperado. Es por ello que se asignan pesos superiores a los ejemplos mal clasificados. Con este algoritmo se pretende crear modelos que se vuelvan expertos en los datos que no pudieron ser explicados por los modelos anteriores. Después de cada interacción los pesos reflejan la medida en que las instancias han seguido mal clasificadas por los clasificadores que se tienen hasta ese momento. Se generan igual T clasificadores de muestras de ejemplos pesados. El clasificador final se construye utilizando un esquema de votación pesado que depende del trabajo de cada clasificador en su conjunto de entrenamiento. **(Cuenca, 2008).**

Tenemos: $(x_1, y_1), \dots, (x_m, y_m)$; $x_i \in X, y_i \in \{-1, 1\}$
 Inicializar los pesos $D_1(i) = 1/m$
 Para $t = 1, \dots, T$:

1. (Invocamos al clasificador débil), el cual retorna $h_t: X \rightarrow \{-1, 1\}$ con un mínimo error D_t
2. Escogemos $\alpha_t \in \mathbb{R}$,
3. Actualizamos

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

donde Z_t es el factor de normalización escogido tal que D_{t+1} es una función de distribución

Como salida obtenemos un clasificador más fuerte:

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$


2.4. Definición de la Terminología

Visión Artificial

También conocida como visión por computador o visión técnica, es un sub-campo de la inteligencia artificial. El propósito de la visión artificial es programar un computador para que "entienda" una escena o las características de una imagen.

Procesamiento de Imágenes

Es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

Algoritmo

Es un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad.



CAPÍTULO III: MARCO METODOLÓGICO

3.1. Tipo y diseño de la investigación

La presente investigación es del tipo Tecnológico y su diseño cuasi-
Experimental

3.2. Población y Muestra

Población: La población está determinada por la cantidad de 100 imágenes de mangos. En investigaciones realizadas referente a procesamiento de imágenes similares a este trabajo propuesto se consideró el repositorio denominado Mango 'Kent' que consta de 100 imágenes calibradas de mangos 'Kent' (50 mangos fotografiados por ambos lados). **(Vidal A, 2013)**

Muestra: En el presente trabajo de investigación se trabajó con una muestra no probabilística de 10 imágenes, que fue suficiente para el desarrollo de la investigación, basado en la investigación “Un prototipo mecánico para la automatización del proceso de selección del mango tipo exportación”, utilizó 5 imágenes en su trabajo de investigación **(Moreno Bermúdez .A, 2012).**

3.3. Hipótesis

Si se utiliza las técnicas de visión artificial entonces se detectara automáticamente los defectos externos del mango.

3.4. Variables

3.4.1. Variable Independiente

Técnicas de Visión Artificial

3.4.2. Variable Dependiente

Detección automática de defectos externos del mango

3.5. Operacionalización

Variable Independiente	Dimensiones	Indicadores	Técnicas e instrumentos de recolección de datos
Técnicas de	cantidad de	tiempo de	Documentación
Visión Artificial	imágenes	procesamiento	

Variable Dependiente	Dimensiones	Descripción	Indicadores	Técnicas e instrumentos de recolección de datos
-----------------------------	--------------------	--------------------	--------------------	--

Detección automática de defectos externos del mango	Precisión	Es la proporción de muestras correctamente diagnosticadas del total de muestras evaluadas.	$P = \frac{ \#Taciertos \cap \#Tmues }{ \#Tmuestras }$	Se realizara usando técnicas de Observación
	Revocación	Es la proporción de muestras correctamente diagnosticadas para cada tipo de deficiencia de calidad.	$R = \frac{ \#Taciertos \cap \#Tmues }{ \#aciertos }$	
	Valor-F	Nos permite corregir el error de la distancia, en los casos en los que la revocación (R) y la precisión (P) se compensan, dando así, una medida de precisión del algoritmo.	$F = 2 \frac{Precision * Revocacion}{Precision + Revocacion}$	

3.6. Métodos, Técnicas e Instrumentos de recolección de datos

Métodos Empíricos: extracción de las características del mango. Para ello vamos a evaluar el color y textura.

Método de la Observación: Observación de los defectos externos del mango para cuantificar el cierto porcentaje de daño, y a la vez clasificar



los mangos según la categoría basándose en la norma técnica peruana (**NTP 011.010:2002:MANGO.Mangos frescos – 2002**).

Método de la Medición: Los resultados obtenidos se validaran para verificar la calidad del mango.

Para la recolección de datos se usara las siguientes técnicas.

a) Observación

Es el registro visual de lo que ocurre en una situación real, clasificando consignando los acontecimientos pertinentes de acuerdo con algún esquema previsto y según el problema que se estudia.

b) Entrevista:

Este método se utiliza para recabar información en forma verbal, a través de preguntas que propone el analista. Sirve para analizar la realidad y estado de la situación problemática, y solamente se trabajara con un especialista que sería un ingeniero agrónomo con experiencia en exportación de mangos el cual validara los resultados obtenidos en la investigación.

c) Documentación

Se consultara material bibliográfico (Libros, Papears, etc), como también

Normas técnicas el cual rige las características del mango.

3.7. Procedimiento para la recolección de los datos

Se va a entrevistar para corroborar los datos proporcionados en el proceso, el Ing. Jorge Leyva Piedra será el entrevistado quien es especialista en este tema en el cual nos validara si es correcto el proceso que estamos siguiendo.

3.8. Análisis estadístico e Interpretación de los datos

Para la evaluación de cada algoritmo y descriptor de forma utilizado en cada prueba, se calculara las siguientes formas matemáticas:

a) Precisión

Es la porción de muestras correctamente diagnosticadas del total de muestras evaluadas

$$P = \frac{|\#Taciertos \cap \#Tmuestras|}{|\#Tmuestras|}$$

b) Revocación

Es la proporción de muestras correctamente diagnosticadas para cada tipo de deficiencia de calidad.

$$R = \frac{|\#Taciertos \cap \#Tmuestras|}{|\#aciertos|}$$

c) Valor-F

Nos permite corregir el error de la distancia, en los casos en los que la revocación (R) y la precisión (P) se compensan, dando así, una medida de precisión del algoritmo.

$$F = 2 * \frac{Precision * Revocacion}{Precision + Revocacion}$$

3.9. Principios éticos

Consentimiento informado: el tesista deberá estar de acuerdo con ser informante y conocer sus derechos y responsabilidad durante la investigación.

Confidencialidad: asegurar la protección de identidad de sus fuentes, como también de las personas que participen como informantes de la investigación.

3.10. Criterios de rigor científico

La presente propuesta de investigación se realiza siguiendo los juicios científicos establecidos, estos permiten garantizar la calidad de la propuesta de investigación. Así, seguimos la coherencia metodológica durante el desarrollo de la propuesta de la investigación, realización apropiada del muestreo de datos, los cuales son al azar para ser totalmente imparcial en el recojo de datos. Los datos recolectados se basan en la forma, color y textura del mango, los cuales son tomados como datos principales.

CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1. Descripción de resultados

Para este objetivo, se utilizará un repositorio de imágenes tanto para el entrenamiento del sistema como para la etapa de clasificación. El número de imágenes consideradas en total ha sido de 100 imágenes. Entre las cuales las muestras tomadas están con defectos y sin defectos del mango.

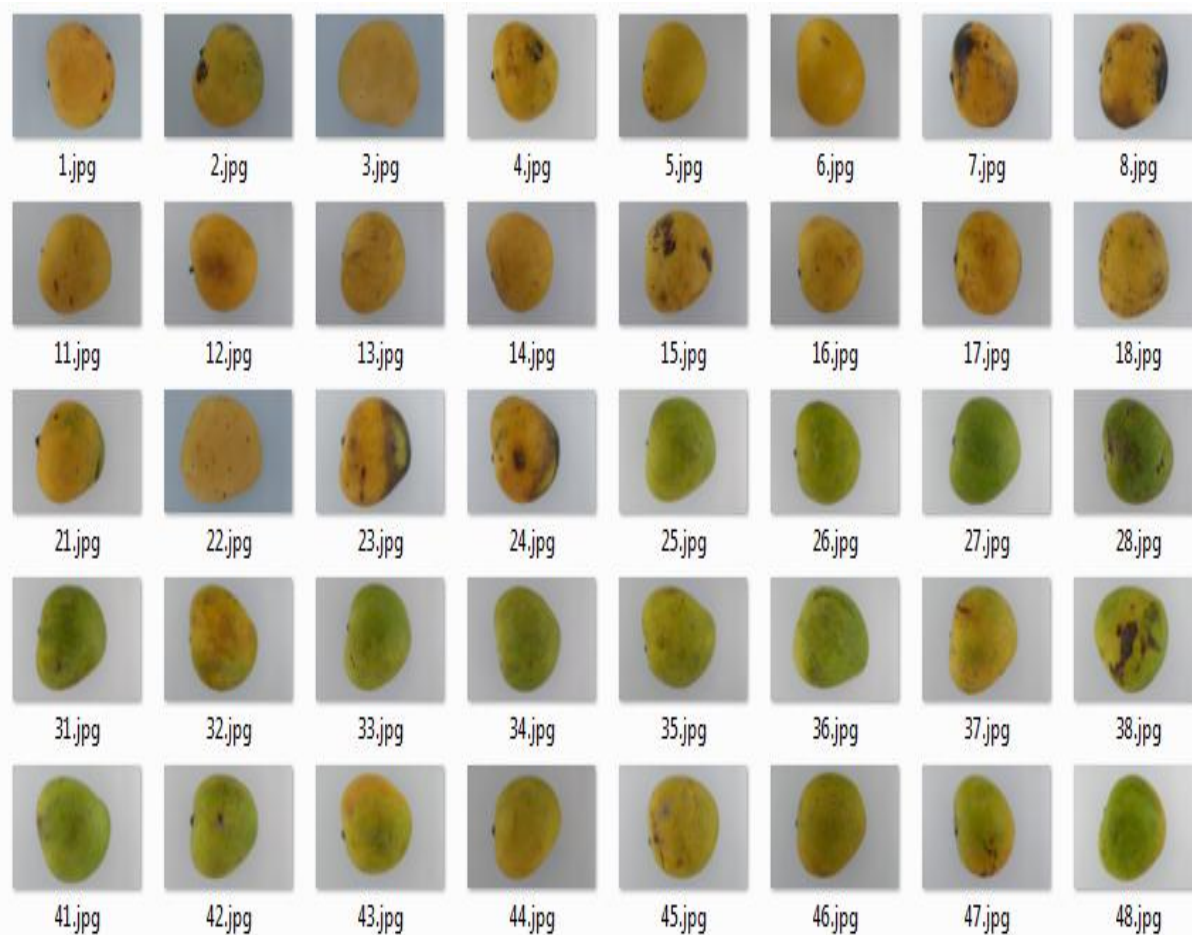
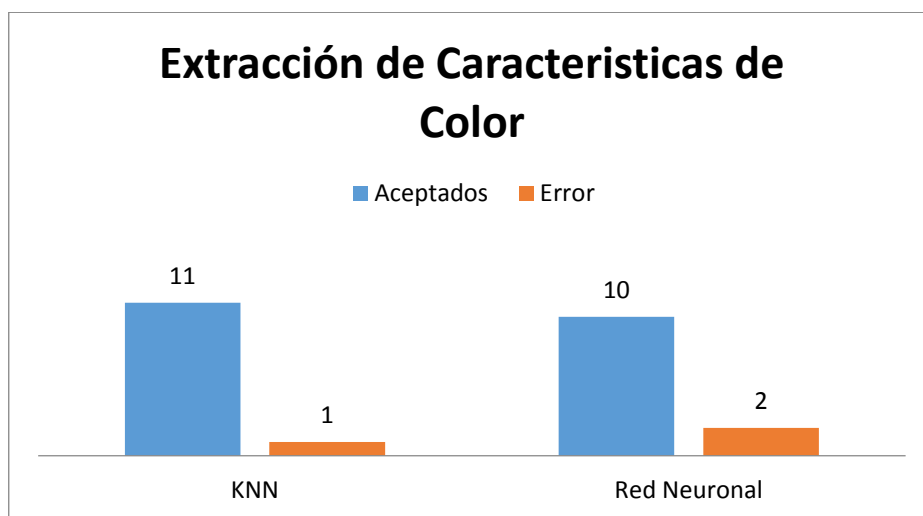


Figura 8: Muestras tomadas con defectos y sin defectos del mango.

4.1.1. Evaluación de métodos de extracción de características

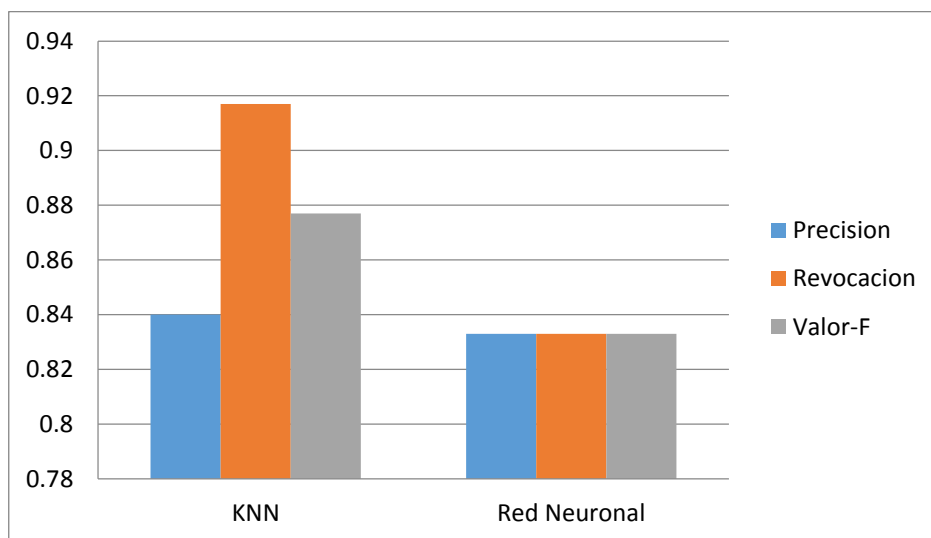
A continuación, se presentan los resultados obtenidos por los descriptores utilizados en cuanto a su efectividad. Las pruebas se realizaron utilizando 2 clasificadores que se detallan en el Capítulo 5. Los datos mostrados en cada gráfico corresponden a la cantidad de imágenes correctamente clasificadas, erróneamente clasificadas, el porcentaje de acierto global y las métricas de evaluación Precisión, Revocación y Valor-F obtenidos para cada clase.

Gráfico 1: Número de aciertos y errores en la clasificación de muestras utilizando Histograma de Color y segmentación Otsu. Se muestra los resultados obtenidos utilizando dos tipos de clasificadores.



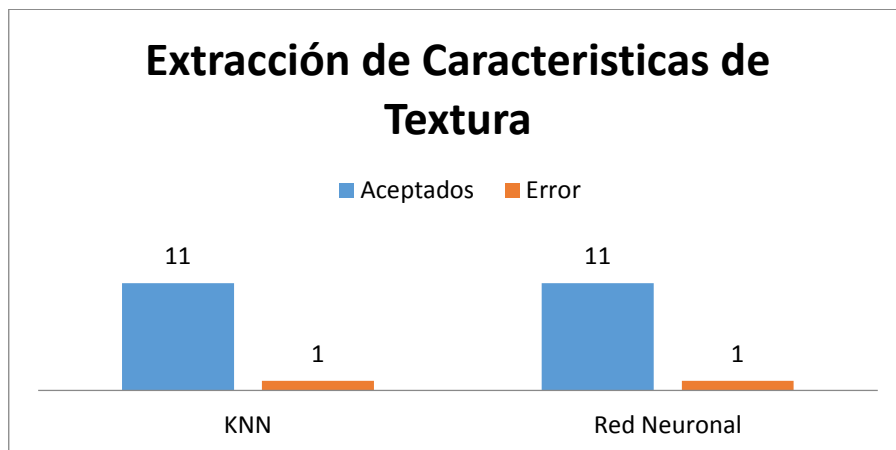
Los resultados obtenidos utilizando HC se muestran en el Gráfico 1 y gráfico 2. Con el método de segmentación Otsu y usando K-NN se obtuvo un porcentaje de acierto global del 91.66%, mientras que con la Red Neuronal se obtuvo un 83.33% de acierto global.

Gráfico 2: Pruebas de efectividad del descriptor para cada tipo de mango con el uso de diversos clasificadores.



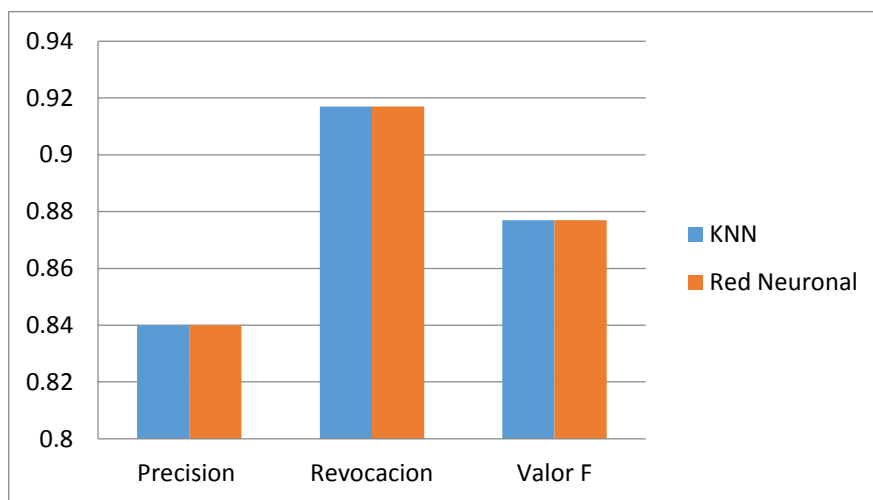
Se obtuvo los mejores valores al clasificar con K-NN, logrando una Precisión de 84%, revocación de 91% y valor-F de 87%. Usando Red Neuronal se obtuvo una Precisión de 83%, revocación de 83% y un valor-F de 83%.

Gráfico 3: Número de aciertos y errores en la clasificación de las imágenes del repositorio utilizando GLCM y segmentación Otsu. Se muestra los resultados obtenidos utilizando dos tipos de clasificadores.



Los resultados obtenidos utilizando GLCM que se muestran en el Gráfico 3 y gráfico 4. Con el método de segmentación Otsu y Utilizando KNN se obtuvo un porcentaje de acierto global del 91.66%, al igual que con la Red Neuronal se obtuvo un 91.66% de acierto global.

Gráfico 4: Pruebas de efectividad del descriptor de Forma para cada defecto del mango con el uso de diversos clasificadores.



Se obtuvo los mejores valores al clasificar con K-NN, logrando una Precisión de 84%, revocación de 91% y valor-F de 87%. Usando Red Neuronal se obtuvo una Precisión de 84%, revocación de 91% y un valor-F de 87%.

4.1.2. Resultados con los algoritmos evaluados de extracción de características y clasificación

Tabla 1: Clasificación con la Red Neuronal

RED NEURONAL		
Imágenes	Color	Textura
Img. 1	83.33%	91.66%
Img. 2	100%	66.66%
Img. 3	69.23%	76.92%
Img. 4	55.55%	77.77%
Img. 5	100%	100%
Img. 6	100%	100%
Img. 7	94.11%	94.11%
Img. 8	83.33%	66.66%
Img. 9	92.30%	84.61%
Img. 10	100%	100%
Efectividad	87.79%	85.84%

Fuente 3: Elaboración Propia



Tabla 2: *Clasificación con KNN*

KNN		
Imágenes	Color	Textura
Img. 1	91.66%	91.66%
Img. 2	100%	66.66%
Img. 3	61.55%	74.92%
Img. 4	55.55%	55.55%
Img. 5	100%	100%
Img. 6	100%	100%
Img. 7	94.11%	94.11%
Img. 8	83.33%	66.66%
Img. 9	84.11%	92.30%
Img. 10	100%	100%
Efectividad	87.03%	84.19%

Fuente 4: *Elaboración Propia*

4.1.3. Cuadro comparativo de los algoritmos evaluados

Tabla 3: *Algoritmos Evaluados*

	Tiempo	Objetivo de procesamiento	Objetivo de Extracción de Características
Canny	Su tiempo para detectar forma es eficiente que otros algoritmos relacionados a este.	El algoritmo canny se utiliza para detectar bordes.	
Otsu	El procesamiento de imágenes el algoritmo otsu es más eficiente que otros algoritmo, el tiempo de ejecución es aceptable para nuestra investigación.	El algoritmo se encarga de separar el fondo y el objeto para tener mejor precisión.	
K-means	El tiempo de procesamiento de imágenes en K-means no es favorable para nuestra investigación.	El algoritmo utiliza cantidad de Clusteres (k) cada objeto a agrupar se calcula su distancia y se asigna cada objeto al grupo.	
GLCM	Su tiempo de ejecución es favorable para nuestra investigación		El algoritmo se encarga de extraer características de Textura.
BSM			El algoritmo se encarga de extraer características de Forma, se utilizó el algoritmo BSM.

Fuente 5: *Elaboración Propia*



4.1.4. Defectos encontrados para la Detección Automática por descriptores

Tabla 4: Defectos externos del mango a Evaluar

Defectos\Descriptores	FORMA	COLOR	TEXTURA
Antracnosis		X	X
Lenticelas oscuras		X	
Manchas necróticas		X	X
Manchas oscuras		X	

Fuente 6: Elaboración Propia

El repositorio de mangos cuenta con una variedad de defectos y plagas, pero para un mejor estudio solo hemos escogido 4 defectos ya que son los que más se repiten y nos será más fácil determinar los defectos utilizando los clasificadores KNN y Red neuronal. Para ver todos los defectos encontrados de cada imagen ir a **ANEXO 1**.

4.2. Discusiones y resultados

Con respecto a los antecedentes

Discusión 1

Ponderación de calidad en frutas usando técnicas de visión artificial para la estimación de daños. (papers). En su proyecto de investigación se utilizó el método de espacio de color RGB para eliminar el fondo y sombras de la fruta (Pera). Los resultados son sólo aproximaciones, ya



que deben introducirse modificaciones necesarias de manera que el sistema no contabilice como área dañada ni el pedúnculo ni el cáliz de la fruta. En esta investigación se trabajó dos técnicas de segmentación k-measn y Otsu para quedarnos con la imagen de interés. Los resultados que se obtuvieron como mejor método es Otsu ya que utiliza un umbral por defecto para cada imagen y es más rápido de procesar las imágenes.

Discusión 2

Un prototipo mecánico para la automatización del proceso de selección del mango tipo exportación (papers).en su proyecto de investigación los resultados de la operación del mecanismo, y la aplicación de los métodos mostraron una precisión superior al 90%. En este proyecto de investigación se obtuvo una precisión de un 87.79% de efectividad para la clasificación correcta de los defectos externos del mango utilizando Redes Neuronales.

Discusión 3

Técnicas de Segmentación en Procesamiento Digital de Imágenes (papers). En este proyecto de investigación informativa se emplearon métodos como la segmentación, detección de bordes, Umbralización, y Basado en regiones. En esta investigación se trabajó métodos de

segmentación, extracción de características de color y textura de los defectos externos del mango.

Discusión 4

Sistema de Visión Artificial para la inspección, selección y control de calidad de fresas. (papers). El resultado obtenido es de 99% en la clasificación según el estado de madurez y 100% para la clasificación según la categoría de la fresa. En este proyecto de investigación se obtuvieron resultados muy cercanos a los que se obtenía al utilizar el clasificador Red Neuronal se obtuvieron mejores resultados con un porcentaje de 87.79 % de efectividad.

Discusión 5

Detección automática de deficiencias nutricionales en imágenes de hojas de café usando descriptores de forma y textura. (papers). En su proyecto de investigación se adquirieron las imágenes de hojas de café, estas son sometidas a un proceso de segmentación utilizando el método de Otsu. A las imágenes resultantes se aplicaron los descriptores Blurred Shape Model (BSM) y Gray-Level Co-occurrence Matrix (GLCM) para la extracción de características de forma y textura respectivamente. En este proyecto de investigación también se realizó la adquisición de imágenes de mangos, luego se aplica el método de segmentación Otsu. A las imágenes resultantes se aplicaron los descriptores como Gray-Level Co-occurrence Matrix (GLCM) e

histogramas de color RGB, para la extracción de características de textura y color.

CAPÍTULO V: DESARROLLO DE LA PROPUESTA DE INVESTIGACIÓN



Método 1: Dada la necesidad de identificar defectos externos del mango en imágenes digitales mediante un sistema computacional, se plantea crear una infraestructura de sistema genérica, basada en la utilizada en la mayoría de sistemas diseñados para reconocer imágenes. En la Fig. 8 se muestra el esquema del diseño que tendrá el desarrollo de los algoritmos de visión artificial para el reconocimiento de defectos externos del mango.



Figura 9: *Diseño estructural para la investigación*

Se recibe como primera entrada un conjunto de imágenes obtenidas del mango en un ambiente controlado. A continuación, las imágenes son procesadas con los descriptores presentados más adelante, con el fin de extraer las características necesarias sobre su color y textura.

Para crear la infraestructura anterior se han tenido que utilizar diversas tecnologías y herramientas. La investigación se desarrolló mediante el IDE QT utilizando el lenguaje de programación c++. Se optó por utilizar el lenguaje c++ debido a que existen investigaciones realizadas a este lenguaje en lo que es reconocimiento de imágenes digitales el cual nos facilitó implementar las técnicas de visión artificial.

5.1. Adquisición de Imágenes

La adquisición de las imágenes es uno de los pasos más importantes en un sistema de visión artificial, ya que la calidad de las imágenes a analizar debe tener una calidad óptima, que no afecte al rendimiento del sistema.

A continuación, las imágenes son digitalizadas mediante la toma de fotografías con una cámara de celular Motorola G de 13 Mega Píxeles en un entorno especialmente diseñado para mitigar cualquier sombra que pudiese proyectar el mango.



Figura 10: *Instrumento empleado para la captura de las imágenes digitales.*

El entorno se crea a partir de una base donde reposa la cámara, una base con fondo blanco. A los costados, se ubican dos fuentes de luz blanca que son proyectadas hacia el mango como se muestra en la [Fig. 10].

Los defectos de los mangos que se ha estudiado son los siguientes:

- a) Antracnosis: El fruto está afectado por enfermedad.



Figura 11: *Mango con defecto de Antracnosis*

- b) Lenticelas oscuras esparcidas: debido a una exposición directa a la luz solar.



Figura 12: *Mango con defecto de Lenticelas oscuras esparcidas*

- c) Mancha Necrótica: causada por bacterias.



Figura 13: *Mango causada por bacterias*

- d) Manchas Oscuras: causada por golpes cuando caen del árbol o por bacterias.



Figura 14: *Mango con defecto de Manchas oscuras*

5.2. Pre – Procesamiento

En la etapa de pre procesamiento se pretende corregir defectos que puedan producirse durante la captura y digitalización de las imágenes de los mangos. Si bien se ha tomado las precauciones para obtener una muestra lo más limpia posible, debido a la forma del mango, es posible que, aun así, proyecten una sombra indeseada, la cual de no ser removida al procesar las muestras, pueden inducir errores no forzados en los resultados

A. Filtrado de imagen

Es necesario filtrar la imagen para quitar ruido, para realizar esto es necesario reducir la amplitud de las variaciones de la imagen, una forma simple de hacer esto, es reemplazar cada píxel por la media del valor de los píxeles de alrededor, de esta forma las variaciones rápidas de intensidad pueden ser suavizadas y reemplazadas por una transición más gradual. El filtro elegido para realizar este suavizado ha sido GaussianBlur, esta función tiene en cuenta el peso de los píxeles más cercanos que los alejados. La función realizada por este algoritmo de suavizado es la siguiente:

$$G(x, y) = \frac{1}{2 * \pi * \sigma^2} * e^{\frac{-x^2+y^2}{2*\sigma^2}}$$

Donde la “x” es la distancia desde el origen (punto del píxel a tratar) en el eje horizontal, e “y” es la distancia desde el origen en el eje vertical y “σ” es la desviación estándar de la distribución Gaussiana. Cuando se aplica en dos dimensiones, esta fórmula produce una superficie cuyos contornos son círculos concéntricos con la distribución Gaussiana desde el punto central.

Si aplicamos la función GaussianBlur sobre la imagen escalada a grises obtenemos el siguiente resultado:



a) Imagen a escala de grises

b) Imagen suavizada

Figura 15: *Utilizando Filtro GaussianBlur a la imagen del mango*

5.3. Segmentación

El objetivo de la segmentación es simplificar la representación de una imagen en otra más significativa y más fácil de analizar, resaltando características relevantes según se desee. Es importante realizar la segmentación previa de las imágenes a analizar ya que, de la imagen, solo nos interesa conservar la región correspondiente al mango, despreciando otro tipo de información que no corresponda a esta.

Para la segmentación se estudiaron dos métodos: K-Means y Segmentación por umbralización, utilizando el método de Otsu.

Los resultados obtenidos con las técnicas son mostrados a continuación

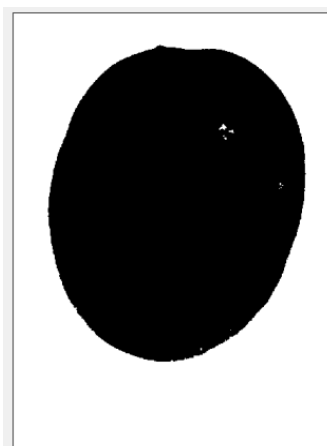
A) Segmentación por Umbralización

Se utiliza el método de Umbralización de Otsu debido a que es un algoritmo muy eficiente cuando existe una diferencia muy marcada entre los objetos a extraer respecto del fondo de la imagen. En este caso, se desea separar el mango del fondo de color uniforme, eliminando sombras. Como resultado de aplicar el algoritmo obtenemos

una imagen binarizada donde, jugando con el umbral, se obtiene la segmentación deseada.



a) imagen digitalizada



b) imagen binarizada

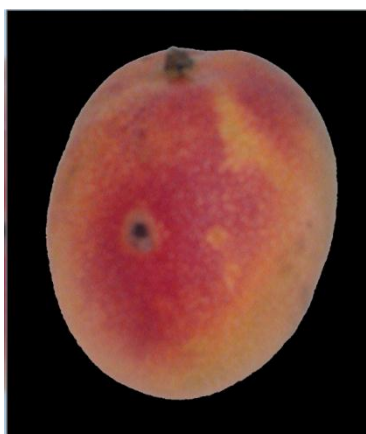


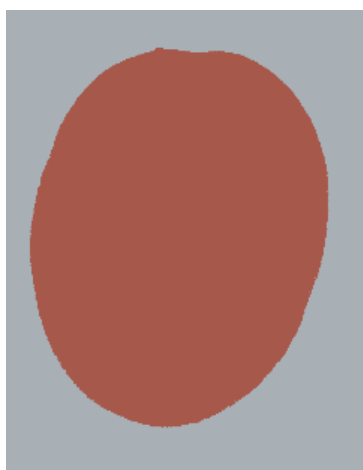
Figura 16: *Segmentación por Umbralización a una imagen digital*

c) Imagen segmentada

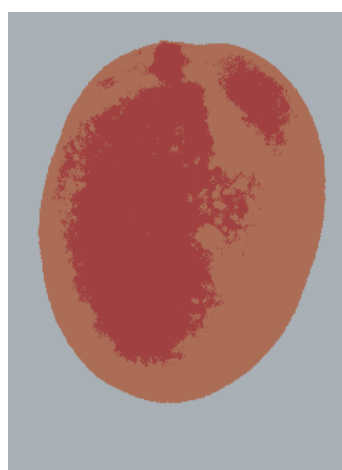
El mango se muestra totalmente negra, mientras que el resto adquiere un fondo blanco. Como se desea mantener los colores del mango, se

toma como referencia los pixeles con valor negro de la imagen binarizada para obtener los pixeles a conservar de la imagen original.

B) K-Means



a) Clusters K = 2



b) Clusters k = 3



c) Clusters K = 10



d) Clusters K = 20

Figura 17: Segmentación de imágenes utilizando K-Means con distinta cantidad de clústeres (2, 3, 10 y 20 clusters respectivamente).

El proceso empieza recibiendo la cantidad clúster a utilizar, definiendo la variable K, cada uno con su respectivo centroides. Luego, para cada objeto a agrupar se calcula su distancia y se asigna cada objeto al grupo o clúster con la distancia mínima. Por último, para cada grupo se actualiza la ubicación de su centroides en función a los objetos agrupados a él, todo esto hasta que el algoritmo se considere convergido, es decir, no presenta variación con respecto a las asignaciones a cada centroides. Aplicado a procesamiento de imágenes, la cantidad de clústeres representa la cantidad de colores en la cual será segmentada la imagen. Cada pixel de la imagen es agrupado utilizando una medida de diferencia, en este caso, la distancia cuadrática Euclídea (Y., 2010).

$$d(x_i, x_j) = \sum_{j=1}^p (x_{ij} - x_{ij})^2 = \|x_i - x_j\|^2$$

Los resultados se muestran en la [Fig. 17], siendo 2 la cantidad de clúster indicada para la segmentación según se requiere en este caso, de esta manera se logra segmentar correctamente el mango del fondo que la rodea. Utilizar más clústeres significa perder precisión para la segmentación deseada, además que al usar más clústeres, el costo computacional se eleva proporcionalmente. Se utilizará este tipo de segmentación para obtener información correspondiente solo a los contornos exteriores del mango, discriminando información respecto a los contornos internos como de los defectos del mango.

5.4. Extracción de Características

a) Algoritmo Canny

El primer paso de este descriptor consiste en la detección de bordes en la imagen. Para esto, utiliza el algoritmo de Canny, debido a ser un algoritmo de detección con una buena localización y pocos errores de localización.

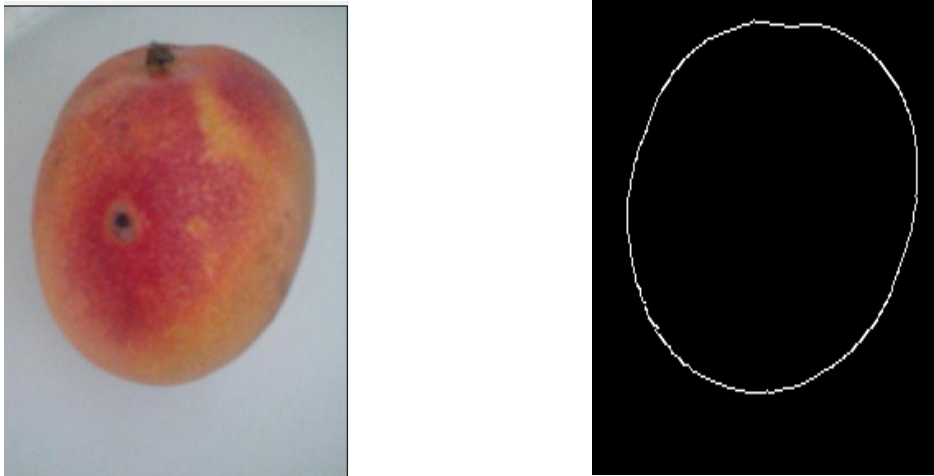


Figura 18: *Aplicando el algoritmo Canny para detectar bordes*

El borde de una imagen puede apuntar en diferentes direcciones, por lo que el algoritmo de Canny utiliza cuatro filtros para detectar horizontal, vertical y diagonal en los bordes de la imagen borrosa. El operador de detección de bordes devuelve un valor para la primera derivada en la dirección horizontal (G_y) y la dirección vertical (G_x). A partir de éste, se pueden determinar el gradiente de borde y la dirección:

$$G = \sqrt{G_x^2 + G_y^2}$$

a) Blurred Shape Model (BSM)

Los descriptores de forma se utilizan para describir un objeto valiéndose de las características que se pueden obtener según su forma, lo cual es posible gracias al uso de distintas técnicas de detección de bordes. Para esto generalmente, se utilizan características de bajo nivel debido a que estas pueden ser extraídas

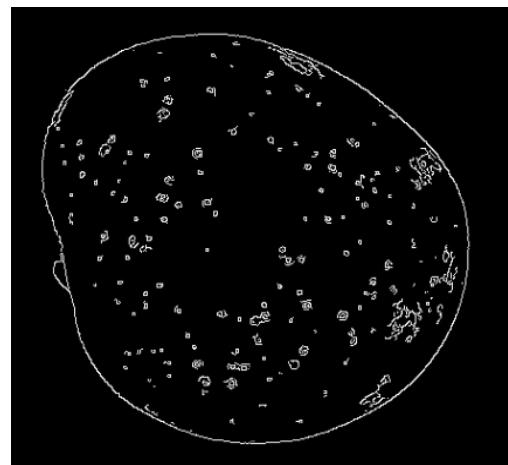


Figura 19: A la izquierda está la imagen original y a la derecha se ha utilizado el algoritmo BSM para detectar el contorno y defectuosidad del mango

automáticamente de una imagen sin tener previamente información sobre la forma del objeto (Nixon, 2002).



a) Histogramas de color

Escala de Grises

Primero la imagen debe convertirse en un formato más apropiado para extraer las Características. Se consideró que

en esta primera aproximación, que lo más relevante de las imágenes a tratar era su contorno, y para hallar su contorno no es necesario tener en cuenta el color de las imágenes. Por lo que es más conveniente convertir la imagen a una escala de grises. En este proyecto de investigación se trabajó con el modelo RGB. En este modelo se asigna una intensidad a cada uno de los tres colores primarios de luz: rojo, verde y azul, de esta manera cada pixel en una fotografía se representa mediante un valor que identifique la intensidad de cada uno de estos tres colores que, mezclados por adición, se acerque más al color verdadero del píxel, las cámaras digitales modernas utilizan valores entre 0 y 255 (2^8) permitiendo de esta manera más de 16 millones de colores distintos. En una escala de grises en cambio, cada píxel es representado con un único valor entre 0 y 255, el cual representa que tan oscuro es el mismo. Para convertir un píxel de una imagen en color a uno en escala de grises se suele hacer un

promedio ponderado de la intensidad de cada uno de los tres colores en donde a cada color se le asigna un peso. Al ser un promedio ponderado la suma de los tres pesos debe ser igual a 1.

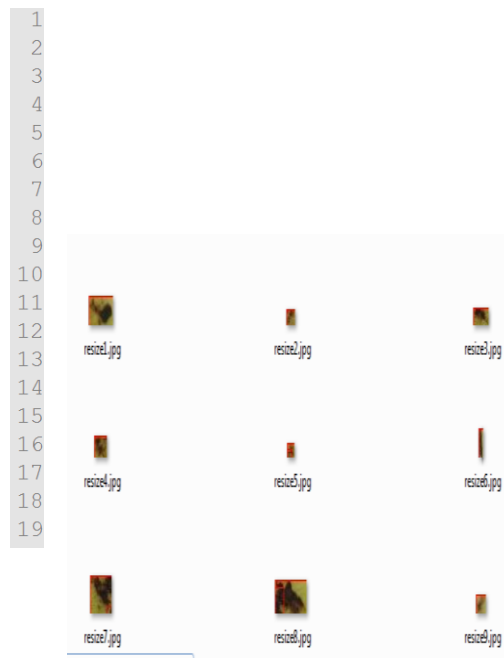


Figura 21: Fueron extraídos todas las

Figura 20: En esta imagen segmentada se detectaron 9 defectos los cuales serán extraídos sus características de Color.

características de color solo a los defectos que se encontraron de la figura 20



```

1 @relation GLCM
2
3 @attribute 'correlacion' real
4 @attribute 'contraste' real
5 @attribute 'homogeneidad' real
6 @attribute 'energia' real
7 @attribute 'class' {Antracnosis, Manchas_Necroticas, Manchas_Oscuras }
8 @data
9
10 0.805537,427.909,0.11859,0.00130298,Manchas_Necroticas
11 0.557991,547.099,0.0773942,0.00474694,Manchas_Necroticas
12 0.770116,443.099,0.0845501,0.00224395,Manchas_Necroticas
13 0.545986,458.918,0.0838385,0.00239139,Manchas_Oscuras
14 0.17429,787.711,0.0468179,0.00617284,Manchas_Oscuras
15 -0.269459,3932.11,0.071091,0.00545,Manchas_Oscuras
16 0.793298,409.853,0.101209,0.000954964,Manchas_Necroticas
17 0.867585,380.747,0.122669,0.000976452,Manchas_Necroticas
18 0.655803,588.959,0.0694753,0.00320367,Manchas_Oscuras

```

Figura 22: Extracción de características de textura utilizando descriptor GLCM correspondiente a la figura 19.

Gray Level Co-Ocurrence Matrix (GLCM)

La matriz de co-ocurrencia y uno de los métodos conocidos para el análisis de textura, las estadísticas de segundo orden almacenan en esta matriz de la medición de la dependencia espacial de niveles de gris, el uso de repetida matriz de co-ocurrencia de cualquier relación entre los niveles de gris imagen que varía rápidamente con la distancia en texturas pausadamente finas y gruesas (Haralickto, 1979)

Para poder obtener las características de textura se utilizan los siguientes descriptores:

Energía

$$E = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} [p(i,j)]^2$$

Cuanto más suave es la textura, mayor valor toma Energía - (Segundo Momento Angular), (p.e. intensidad constante \Rightarrow ASM = 1, distribución unif. De parejas \Rightarrow ASM = $p < 1$)

Homogeneidad

$$C = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{ij p_{d\theta}(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

Es una medida de la homogeneidad de una imagen. Una escena homogénea voluntad contener sólo unos pocos niveles de grises, dando una GLCM con sólo unos pocos pero relativamente altos valores de $p(i,j)$. Por lo tanto, la suma de cuadrados será alta.

Contraste

$$Co = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} (i-j)^2 p(i,j)$$

Esta medida de contraste o variación de la intensidad local favorecerá contribuciones de $p(i,j)$ de la diagonal, es decir, $i \neq j$.

Cuanto mayor es el contraste de la imagen, mayor es el valor de Co al ponderar las diferencias (i,j) con $(i-j)^2$

Correlación

$$C = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \frac{ij p_{d\theta}(i,j) - \mu_x \mu_y}{\theta_x \theta_y}$$



La correlación es una medida de la dependencia lineal de nivel de gris entre los píxeles en las posiciones especificadas respecto a la otra.

Se extraen características de color y textura.

5.5. Clasificación

Para la implementación de estos clasificadores, se usó el conjunto de librerías desarrollada en Java, contiene clases propias como el clasificador K-NN, así como también brinda la posibilidad de utilizar los clasificadores contenidos en las librerías de WEKA (**Mark, 2009**), ampliamente utilizadas en estudios sobre Aprendizaje automático. Se utilizaron los clasificadores la red neuronal multi-capas implementadas en esta librería.

Para la experimentación, se han tenido en cuenta las siguientes especificaciones para los clasificadores:

K-NN: Se ha tomado en consideración realizar pruebas con un número de $K = 1$ vecinos más cercanos.

Red neuronal: La red neuronal es de tipo Back Propagation, conformada por 3 capas: de entrada, oculta y de salida.

ARQUITECTURA DE LA RED BACKPROPAGATION

La unidad procesadora básica de la red Backpropagation se representa en la Figura 16. Las entradas se muestran a la izquierda, y a la

derecha se encuentran unidades que reciben la salida de la unidad procesadora situada en el centro de la figura.

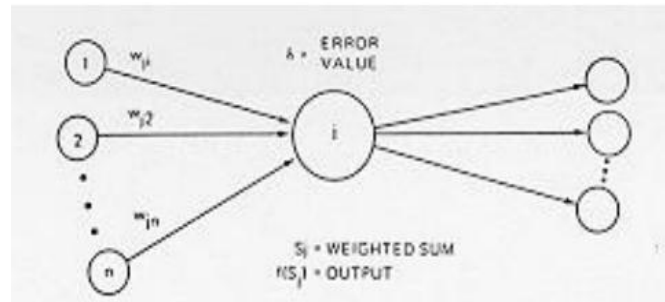


Figura 23: Unidad procesadora básica Backpropagation

Normalmente, la Backpropagation utiliza tres o más capas de unidades procesadoras. La Figura 23 muestra la topología backpropagation típica de tres capas. La capa inferior es la capa de entrada, y se caracteriza por ser la única capa cuyas unidades procesadoras reciben entradas desde el exterior. Sirven como puntos distribuidores, no realizan ninguna operación de cálculo. Las unidades procesadoras de las demás capas procesan las señales como se indica en la figura 23. La siguiente capa superior es la capa oculta, y todas sus unidades procesadoras están interconectadas con la capa inferior y con la capa superior. La capa superior es la capa de salida que presenta la respuesta de la red.

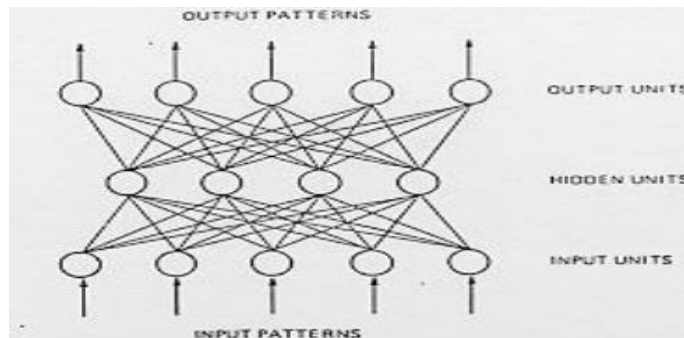


Figura 24: Red Backpropagation completamente interconectada

Propagación hacia Adelante

Esta fase de propagación hacia adelante se inicia cuando se presenta un patrón en la capa de entrada de la red. Cada unidad de la entrada se corresponde con un elemento del vector patrón de entrada. Las unidades de entrada toman el valor de su correspondiente elemento del patrón de entrada y se calcula el valor de activación o nivel de salida de la primera capa. A continuación las demás capas realizarán la fase de propagación hacia adelante que determina el nivel de activación de las otras capas.

La unidad procesadora j obtiene la cantidad S_j según la ecuación (Figura 23)

$$S_j = \sum_i a_i W_{ji}$$

Y genera la salida o nivel de activación según la ecuación (Figura 24)

$$Salida = f(S_j)$$

La función f es una función umbral genérica, entre las que cabe destacar la función Sigmoid y la función Hiperbólica.

Propagación hacia Atrás

Una vez se ha completado la fase de propagación hacia adelante se inicia la fase de corrección o fase de propagación hacia atrás. Los cálculos de las modificaciones de todos los pesos de las conexiones empiezan por la capa de salida y continua hacia atrás a través de todas las capas de la red hasta la capa de entrada. Dentro de los tipos de ajuste de pesos se puede clasificar dos grupos, ajuste de unidades procesadoras de la capa de salida y ajuste de unidades procesadoras de las capas ocultas.

Ajuste de Pesos de la Capa de Salida: el ajuste de estos pesos es relativamente sencillo debido a que existe y se conoce el valor deseado para cada una de las unidades de la capa de salida. Cada unidad de la capa de salida produce un número real como salida y se compara con el valor deseado especificado en el patrón del conjunto de entrenamiento.

La Figura 25 muestra el cálculo de los valores δ_j de las unidades de la capa de salida.

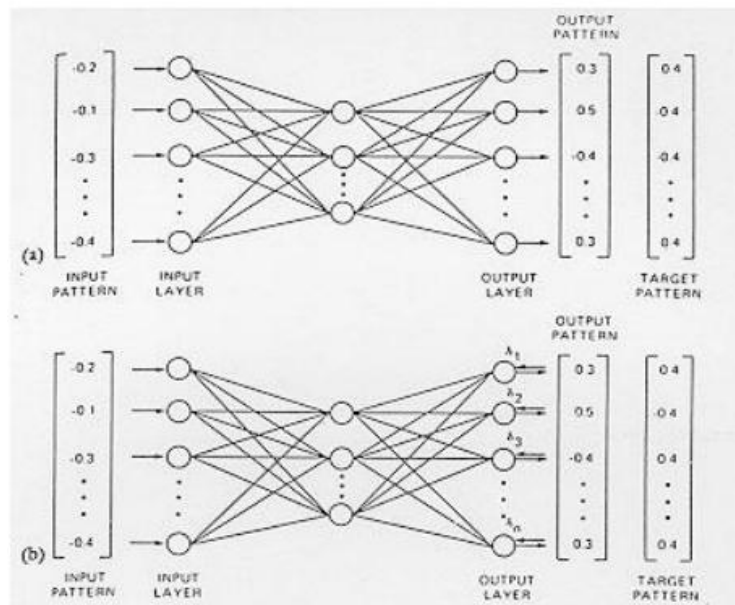


Figura 25: Cálculo de valores δ_J de la capa de salida

Ajuste de Pesos de las Capas Ocultas: estas capas no tienen un vector de salidas deseadas y por tanto no se puede seguir el método de propagación de error mencionado en el caso de unidades procesadoras de la capa de salida. El valor de error calculado para este tipo de unidades procesadoras se obtiene a partir de la ecuación de la Figura 25.

$$\delta_j = \left[\sum_k \delta_k W_{kj} \right] f'(S_j)$$

La Figura 26 representa la obtención del valor δ_j para las unidades de las capas ocultas.

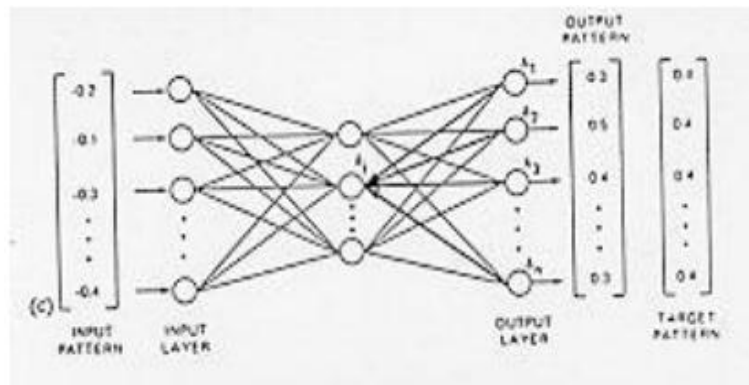


Figura 26: Cálculo de valores δ_j de las capas ocultas

El ajuste de los pesos asociados a las conexiones se realiza a partir del valor de cada unidad de proceso. Cada peso es ajustado según la ecuación (Figura 26) conocida como la regla generalizada (Rumelhart, 1988).

$$\Delta W_{ji} = \eta \delta_j a_i$$

La variable η es el coeficiente de aprendizaje. Este coeficiente, normalmente entre 0.25 y 0.75 refleja el grado de aprendizaje de la red. Algunas veces este coeficiente es modificado de un valor inicial alto a valores progresivamente menores durante la sesión del entrenamiento con el objetivo de lograr un mejor aprendizaje.

Pantalla de Prueba

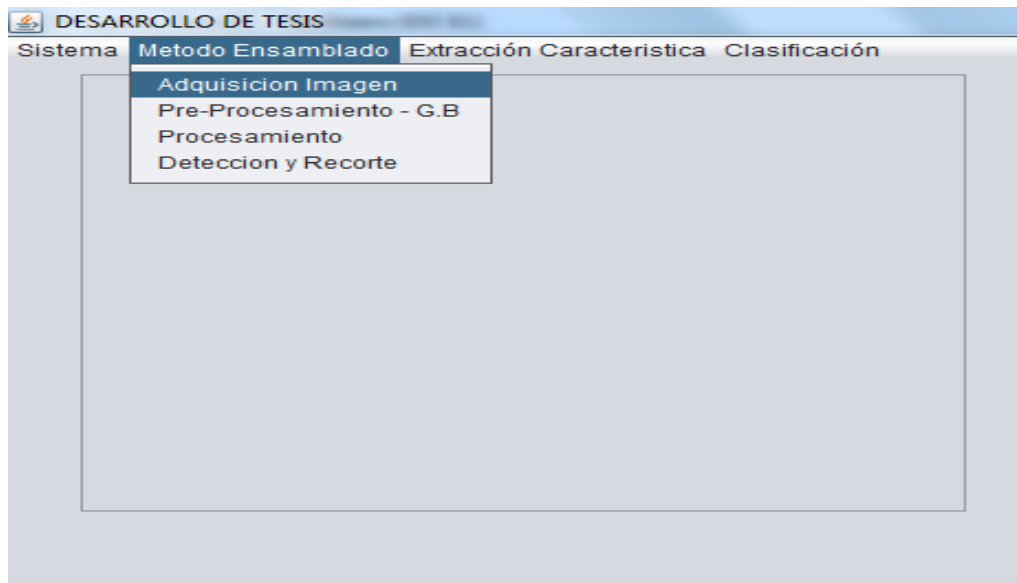


Figura 27: *Adquisición de Imágenes y Pre - Procesamiento*

1. En la pestaña Adquisición tenemos un botón Abrir donde se seleccionara la imagen a evaluar.
2. En el Pre- Procesamiento se evaluara la imagen cargada para ello se pasara por el filtro Gaussiano para quitar ruido a la imagen.
3. En el procesamiento se segmento con OTSU para separar la imagen del fondo.
4. Detección y recorte de solo los defectos externos del mango
5. En el menú extracción de características se evaluaran textura y color.
6. En la clasificación se utilizó el algoritmo que mejor reconoce que es la Red Neuronal.

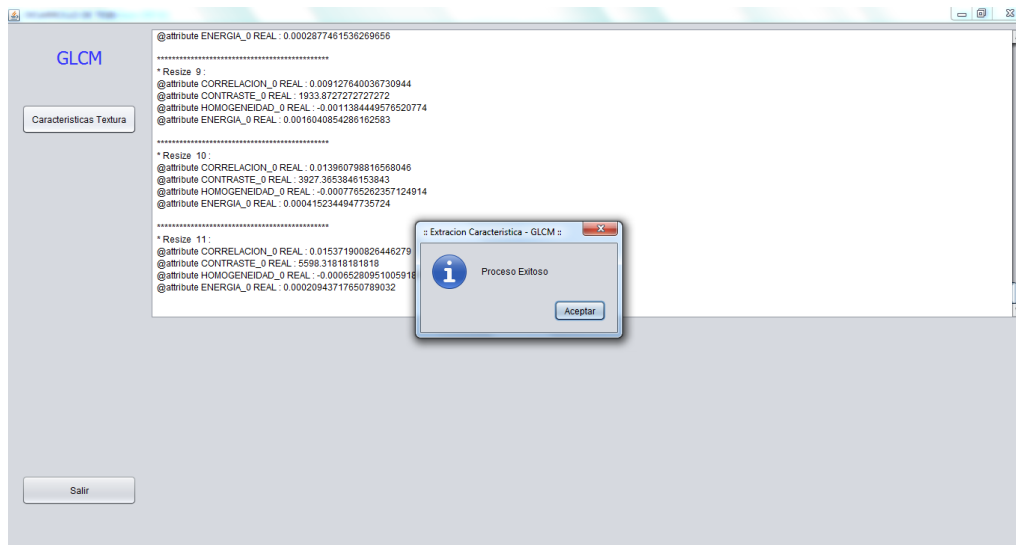


Figura 28: Extracción de características de textura utilizando GLCM

7. Se seleccionara el botón Características de Textura para la extracción de características de Textura.

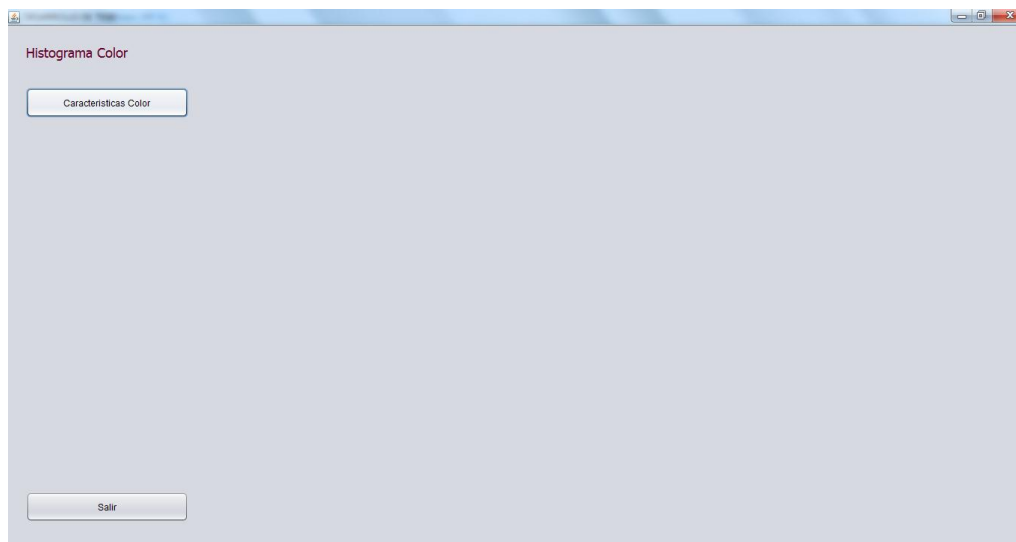


Figura 29: Extracción de Características de Color utilizando Histogramas de color

8. Se selecciona el botón características de color para que se extraigan todas sus características.

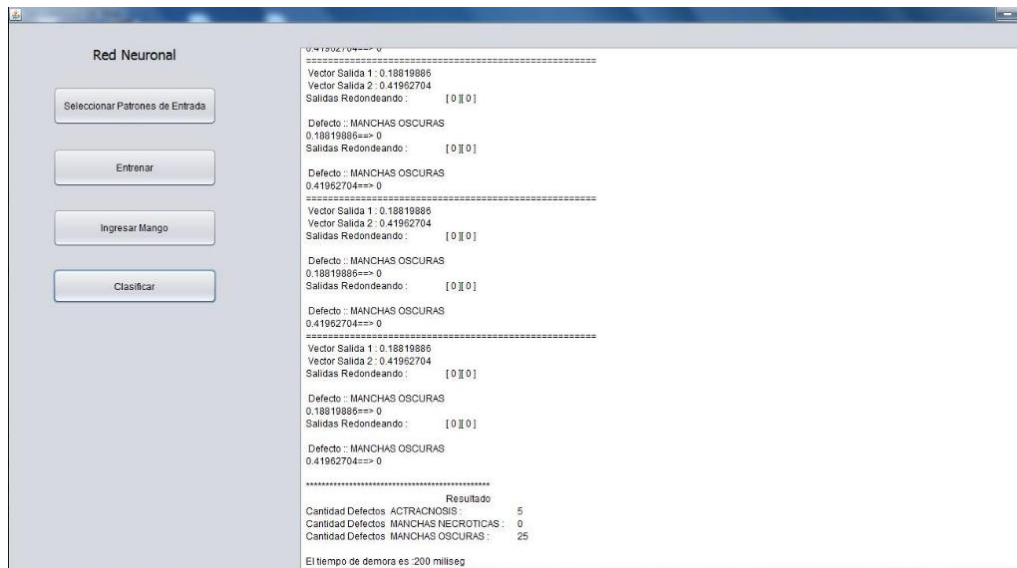


Figura 30: Clasificación con la Red Neuronal

9. Se selecciona el botón seleccionar patrones de entrada.
10. Luego se selecciona el botón entrenar.
11. Luego se selecciona el botón Ingresar Mango.
12. Por último se selecciona el botón Clasificar y se obtendrá los resultados.

Método 2: Dada la necesidad de Detectar los defectos externos del mango en imágenes digitales mediante un sistema computacional, se planteó la misma infraestructura que del método 1, solo que cambia que en este método solo se detectara y cuantificara mas no clasificara.

Para realizar la detección se siguieron estos pasos:

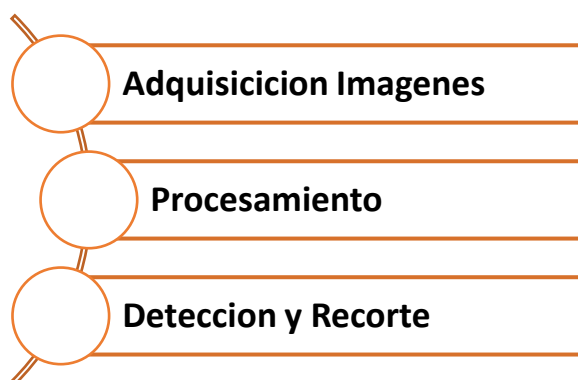


Figura 31: Método de Etiquetado de píxeles conexos

Detección de Contornos

El objetivo es encontrar las características necesarias para poder establecer una discriminación entre mangos sanos y defectuosos. Este paso se determina a partir de un cierto valor que relaciona los píxeles totales del mango con aquellos que pertenecen a un defecto.

Cuando el mango ha sido segmentado del resto de la imagen el siguiente paso se basa en la búsqueda de defectos o imperfecciones en la pieza. Éstos se caracterizan por una coloración y textura diferenciales dentro de la Homogeneidad del mango.

Una vez realizado la segmentación en el método 1, se pasó a este método y realizo lo siguiente: detectó, cuantificó y recorto automáticamente solo los defectos encontrados en cada imagen digitalizada, luego se pasó al método 1 para que culmine el proceso de extracción de características y clasificación.

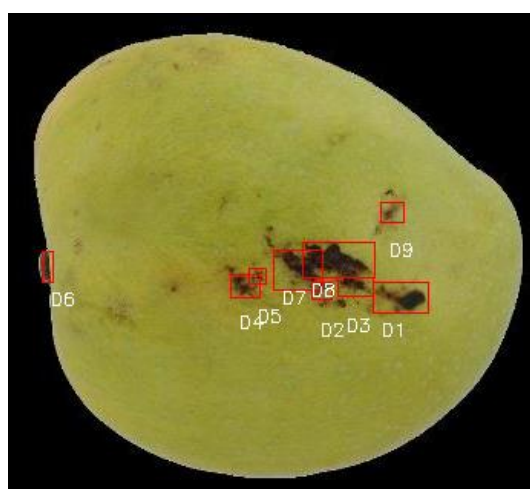


Figura 32: *Detección automática de defectos externos del mango*

Para ello se utilizo la tecnica de etiquetado de pixeles conexos.

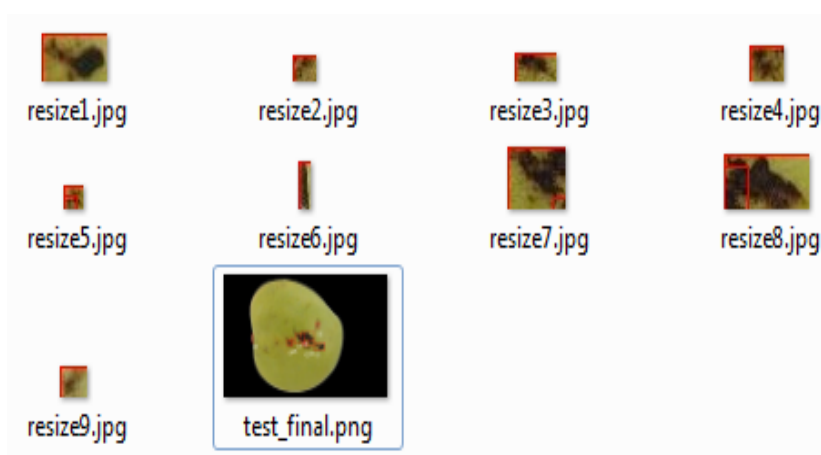


Figura 33: *Recorte automático de los defectos encontrados en la imagen digitalizada*

Método Re curativo

El tipo de grafos a los que está orientado son matrices binarias. Lo que hará será devolvernos una matriz con las componentes conexas numeradas. Cada punto puede estar conectado con los 8 ó 4 de alrededor. Por ejemplo, le pasamos esta matriz:

```

0 1 1 1 0 0 0
1 1 1 0 0 0 0
1 0 0 0 1 0 1
1 1 0 1 1 0 1
0 0 0 0 1 1 1
1 1 1 0 0 0 0
1 1 1 1 1 0 0
    
```

Y nos devuelve esta:

```

0 1 1 1 0 0 0
1 1 1 0 0 0 0
1 0 0 0 2 0 2
1 1 0 2 2 0 2
0 0 0 0 2 2 2
3 3 3 0 0 0 0
3 3 3 3 3 0 0
    
```



Crearemos una función que reciba la matriz. Esta función crea una matriz de ceros de igual tamaño. Esta función tendrá una variable que empiece en 1 y vaya subiendo. Recorrerá la matriz de casilla en casilla. Cuando llegue a una diferente de 0, le pasa esa casilla a la función recursiva, que se encarga de todo. Después le suma 1 a la variable de etiquetas y sigue recorriendo la matriz.

La función recursiva recibe la matriz de ceros, la matriz original, la etiqueta, y los índices de la casilla. En la casilla a rellenar, en la matriz original la pone a 0, y le pone la etiqueta a la misma casilla de la matriz de ceros. Después recorre los 8 vecinos y se llama a si misma si encuentra alguno diferente de 0, pasándole la misma etiqueta.

La función recursiva puede parecer un poco liosa, pero no lo es. Lo que hace es, en las dos primeras líneas, borrar de la matriz original y meter en la de ceros la etiqueta (label). El resto es buscar vecinos, teniendo cuidado de no salirse de los límites de la matriz. Por ejemplo, antes de mirar si la casilla $(x-1, y)$ está ocupada, se asegura de que x sea mayor que 1, si no mirará en $x=0$, y eso en matlab es estar fuera. Si lo haces con arrays en C, entonces se tendrá que mirar que sea mayor que 0, y menor que el número de filas, para no salirte. **(White-hat hacking, 2012)**

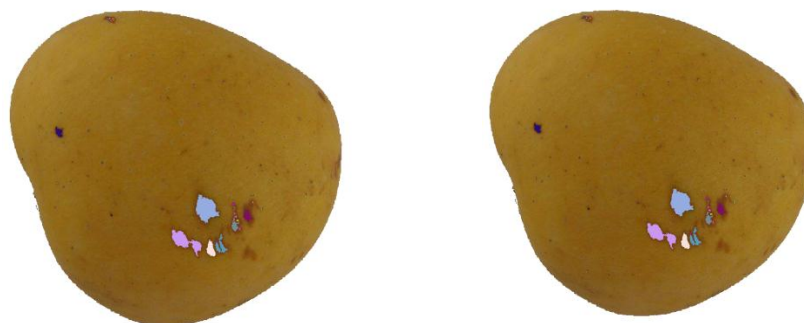


Figura 34: *Detección y cuantificación automático de defectos externos del mango utilizando la técnica de etiquetado de pixeles conexos.*

Método Ensamblado

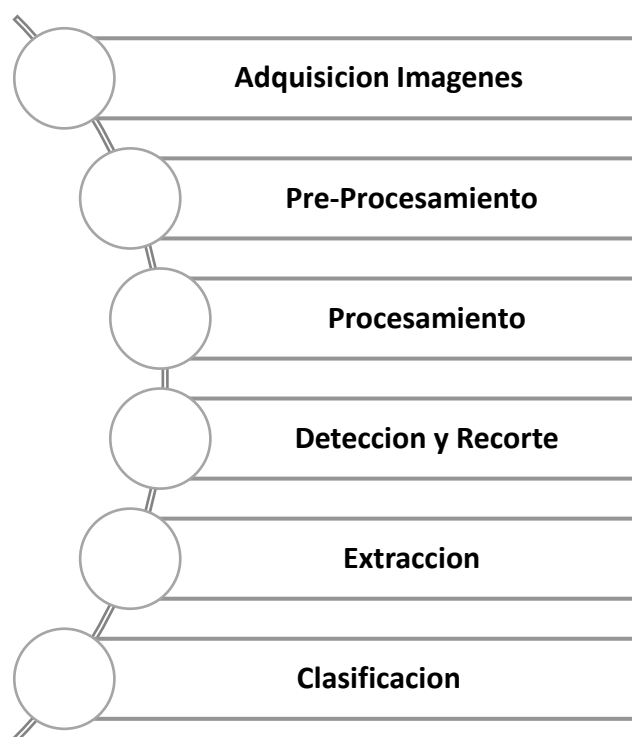


Figura 35: *Método Ensamblado*

Resultados

Correspondiente a la Figura 32:

Tabla 5: *Defectos encontrados automáticamente por la técnica de Conexos.*

Descripción del Defecto	Cantidad
Lenticelas	0
Manchas Oscuras	4
Manchas Necróticas	5
Antracnosis	0
Total de defectos	9

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

- a) Se identificó distintas técnicas de procesamiento de imágenes, las cuales fueron: conversión a escala de grises; redimensionado de imagen; filtros (Gaussiano, Mediano, Media); Binarización (Otsu), algoritmos de clustering (K-means); descriptores de textura (Energía, correlación, contraste, entropía, homogeneidad); descriptor de forma (canny y Bluerred Shape Model); descriptores de color (Histograma de color); algoritmo de detección (etiquetado de píxeles conexos); algoritmos de clasificación (K-Nearest Neighbor y NeuralNet).
- b) Se determinó los defectos externos del mango según su categoría extra, categoría I y categoría II, basándose en la norma técnica peruana (NTP 011.010 MANGO, Mango fresco. Requisitos).
- c) Se seleccionó los algoritmos de procesamiento de imágenes que han permitido tener una adecuada descripción e identificación de características de los objetos en tiempo real. Los algoritmos para el procesamiento de imágenes usados, fueron K-Means y Otsu. Los algoritmos de segmentación pudieron separar efectivamente un objeto del fondo que le rodea y facilitando el análisis por parte de los descriptores en ese único objeto. El algoritmo que dio un mejor resultado fue el algoritmo de Umbralización por el método de Otsu.
- d) Se implementó algoritmos en el lenguaje de programación en c++. De acuerdo a los papers consultados se instaló el IDE QT, se incorporó las librerías opencv 2.4.11 y 3.0 para el mejor desarrollo de la

investigación por lo tanto permitió obtener resultados favorables.

e) Se evaluó los algoritmos de visión artificial teniendo como clasificadores Red Neuronal y K-NN. El clasificador que obtuvo mejor resultado para el análisis de vectores de características fue Red neuronal con un 87.79% de efectividad para color y 85.84 % clasifico correctamente para textura. Por lo tanto se utilizó otro método que detecto los defectos utilizando la técnica de etiquetado de pixeles conexos dando como resultado al objetivo general que es detectar automáticamente los defectos externos del mango.

6.2. Recomendaciones

a) Al adquirir imágenes digitalizadas deben tenerse algunas condiciones como por ejemplo tener un ambiente controlado y que se capture con una cámara de alta resolución con el objetivo de no tener inconvenientes a la hora de procesar las imágenes.

b) Es importante investigar otros defectos y desarrollar herramientas computacionales que puedan desempeñar esta tarea, que actualmente la llevan a cabo los mismo agricultores en el campo, que debido a sus limitaciones como observador humano, es susceptible a cometer errores.

REFERENCIAS

- Adolfo Moreno Bermúdez, D. B. (2012). Un prototipo mecánico para la automatización del proceso de selección del mango tipo exportación.
- Ajoy, A. &. (2005). Segmentación de regiones homogéneas.
- Alejandro Díaz Sotolongo, D. B. (2013). Combinación de técnicas de procesamiento digital de imágenes e inteligencia artificial para el análisis de imágenes de mamografía.
- APEM. (20 de Febrero de 2014). *RPP Noticias*. Obtenido de <http://rpp.pe/economia/economia/produccion-peruana-de-mangos-superara-las-140000-toneladas-noticia-671272>
- ASERCA. (16 de ABRIL de 2014). *SAGARPA*. Obtenido de <http://www.sagarpa.gob.mx/saladeprensa/2012/Paginas/2014B289.asp>
x
- Azuela, S. (2006). procesamiento de imágenes digitales.
- Bermúdez, A. M., Padilla, D. B., & Torres, G. S. (2013). Un prototipo mecánico para la automatización del proceso de selección del mango tipo exportación.
- Boullosa, O. (2011). Estudio comporativo de descriptores visuales para la detección de escenas cuasi-duplicadas. *Madrid: Universidad Autónoma de Madrid*.
- Corral Martín, J. M. (2011). Fusser J. Invariant shape description and measure of object similarity. *Colores y Sombras*, 139 - 142.
- Coto. (2003). imágenes multicanales en la umbralización.



- Cuenca, J. S. (2008). , reconocimiento de objetos por descriptores de forma.
- Graf. (2012). Square Model Shape Matrix.
- INDECOPI. (2002). NTP 011.010 .
- Julesz, B. (1975). Experiments in the Visual Perception of texture. . *Sci. American vol. 232(4)*, 34 - 43.
- La Serna Palomino .N, R. C. (2007). Técnicas de segmentación en procesamiento digital de imágenes.
- Ledda I. Larchera, P. M. (2013). Ponderación de calidad en frutas usando técnicas de visión artificial para la estimación de daños.
- Mark, H. E. (2009). The WEKA data Mining Software.
- Moreno Bermúdez .A, B. P. (2012). Un prototipo mecánico para la automatización del proceso de selección del mango tipo exportación.
- Nixon, M. &. (2002). Feature Extraction and Image Processing.
- Rumelhart, M. &. (1988). Explorations in Parallel Distributed Processing MIT Press Cambridge.
- Sánchez Martínez. H, M. G. (2009). Sistema de visión artificial para la inspección, selección y control de calidad de fresas.
- Sons, J. W. (2000). Digital Image Processing Algorithms and Applications.
- Sunat. (2013). *Proyectos Peruanos*. Obtenido de <http://www.proyectosperuanos.com/mangos.html>
- U.N.Q. (2005). Segmentación de umbralización.
- Vassallo Barco, M. (2015). Detección automática de deficiencias nutricionales en imágenes de hojas de café usando descriptores de forma y textura.

Vicherrez Chavarry, K. (2015). Procesamiento de Imágenes utilizando Descriptores de Color MPEG-7,.

Vidal A, T. P.-M. (2013). *cofilab*. Obtenido de <http://www.cofilab.com/portfolio/mangoesdb/>

Y., G. (2010). Clasificación no supervisada. *El método de K-medias*.

ANEXOS

Anexo 1

Imágenes\Descriptoros	COLOR	TEXTURA	DEFECTOS ENCONTRADOS
Img. 01	X		Manchas oscuras Lenticelas oscuras esparcidas
Img. 02	X	X	Lenticelas oscuras esparcidas Manchas necróticas Antracnosis
Img. 03	X	X	Cicatriz
Img. 04	X	X	Antracnosis Lenticelas oscuras esparcidas
Img. 05	X	X	Manchas oscuras causadas por bacterias Ralladura
Img. 06		X	Ralladura
Img. 07	X		Lenticelas oscuras esparcidas Manchas oscuras causadas por bacterias.
Img. 08	X	X	Antracnosis Manchas oscuras causadas por bacterias. Lenticelas oscuras esparcidas
Img. 09	X	X	Manchas necróticas
Img. 10	X		Lenticelas oscuras esparcidas. Larvas de mosca
Img. 11	X		Manchas oscuras
Img. 12	X		Lenticelas oscura



			Manchas oscura
Img. 13	X	X	Lenticelas oscura esparcidas Manchas oscura Rozaduras
Img. 14	X		Manchas oscura Lenticelas oscuras.
Img. 15	X	X	Manchas necróticas Manchas oscura
Img. 16	X		Manchas oscura
Img. 17	X		Lenticelas oscuras esparcidas. Manchas oscura
Img. 18	X		Lenticelas oscuras esparcidas. Manchas oscura
Img. 19	X		Manchas oscura
Img. 20	X		Larvas de mosca
Img. 21	X		Manchas oscuras. Lenticelas oscuras esparcidas.
Img. 22	X	X	Cicatriz Manchas oscuras
Img. 23	X	X	Manchas oscuras causadas por bacterias. Lenticelas oscuras esparcidas. Ralladura
Img. 24	X	X	Manchas oscuras causadas por bacterias. Lenticelas oscuras esparcidas. Rozaduras Cicatriz
Img. 25	X		Manchas oscuras



Img. 26	X		Secreción de látex Manchas oscuras
Img. 27	X	X	Manchas oscuras Ralladura
Img. 28	X	X	Manchas oscuras Manchas necróticas Secreción de látex
Img. 29	X		Manchas oscuras
Img. 30	X	X	Manchas oscuras Manchas necróticas Secreción de látex Ralladura Lenticelas oscuras esparcidas
Img. 31	X		Manchas oscuras Secreción de látex Lenticelas oscuras esparcidas
Img. 32	X		Manchas oscuras Secreción de látex Lenticelas oscuras esparcidas Larvas de mosca
Img. 33	X		Secreción de látex Lenticelas oscuras esparcidas Manchas oscura
Img. 34	X		Secreción de látex Lenticelas oscuras esparcidas



Img. 35	X		Larvas de mosca Lenticelas oscuras esparcidas
Img. 36	X	X	Lenticelas oscuras esparcidas Secreción de látex Ralladura
Img. 37	X	X	Lenticelas oscuras esparcidas Ralladura manchas necróticas
Img. 38	X	X	Lenticelas oscuras esparcidas manchas necróticas manchas oscuras
Img. 39	X		Manchas oscuras. Secreción de látex
Img. 40	X		Secreción de látex Manchas oscuras
Img. 41	X		Secreción de látex Manchas oscuras
Img. 42	X		Larva de mosca Secreción de látex Manchas oscuras
Img. 43	X		Secreción de látex Manchas oscuras
Img. 44	X		Manchas oscuras
Img. 45	X	X	Manchas oscuras Secreción de látex Manchas necróticas

Img. 46	X		Manchas oscuras
Img. 47	X		Manchas oscuras Lenticelas oscuras esparcidas Secreción de látex.
Img. 48	X		Manchas oscuras Lenticelas oscuras esparcidas
Img. 49	X		Manchas oscuras
Img. 50	X		Manchas oscuras
Img. 51	X	X	Manchas oscuras Ralladura Larva de mosca
Img. 52	X		Manchas oscuras
Img. 53	X	X	Manchas necróticas Cicatriz Lenticelas oscuras esparcidas
Img. 54	X		Manchas oscuras
Img. 55	X		Manchas oscuras
Img. 56	X		Lenticelas oscuras esparcidas
Img. 57	X		Manchas oscuras
Img. 58	X		Manchas oscuras Lenticelas oscuras esparcidas
Img. 59	X		Manchas oscuras Lenticelas oscuras esparcidas
Img. 60	X		Manchas oscuras Lenticelas oscuras esparcidas.
Img. 61	X	X	Manchas oscuras Ralladuras



Img. 62	X		Lenticelas oscuras esparcidas Manchas oscuras
Img. 63	X	X	Manchas oscuras Manchas necróticas
Img. 64	X	X	Manchas oscuras Cicatrices
Img. 65	X	X	Manchas necróticas Manchas oscuras
Img. 66	X		Lenticelas oscuras esparcidas Manchas oscuras causadas por bacterias.
Img. 67	X	X	Antracnosis Manchas oscuras causadas por bacterias. Lenticelas oscuras
Img. 68	X	X	Ralladura Lenticelas Rojas
Img. 69	X	X	Lenticelas rojas esparcidas. Secreción de látex Ralladura
Img. 70	X	X	Lenticelas Rojas esparcidas Ralladura Secreción de látex
Img. 71	X		Lenticelas oscura esparcidas
Img. 72	X	X	Lenticelas oscura esparcidas Mancha necrótica Cicatriz
Img. 73	X		Defecto leve en la cascara



Img. 74	X		Defecto leve en la cascara
Img. 75	X	X	Manchas oscuras Lenticelas oscuras esparcidas Cicatrices
Img. 76	X	X	Lenticelas oscuras esparcidas. Manchas oscuras Cicatrices
Img. 77	X		Manchas oscuras
Img. 78	X	X	Manchas necróticas Lenticelas oscuras esparcidas Cicatriz
Img. 79	X		Lenticelas oscuras esparcidas Manchas oscuras causada por bacterias
Img. 80	X		Manchas oscuras. Lenticelas oscuras esparcidas.
Img. 81	X		Lenticelas oscuras esparcidas Manchas oscuras
Img. 82	X		Manchas oscuras. Lenticelas oscuras esparcidas.
Img. 83	X	X	Manchas oscuras causadas por bacterias. Lenticelas oscuras esparcidas.



			Cicatrices
Img. 84	X	X	Cicatrices
Img. 85	X	X	Cicatrices
Img. 86	X		Manchas oscuras Lenticelas oscuras esparcidas
Img. 87	X	X	Manchas oscuras Cicatrices
Img. 88	X	X	Manchas oscuras Cicatrices
Img. 89	X		Manchas oscuras
Img. 90	X		Larva de mosca
Img. 91	X		Manchas oscuras Lenticelas oscuras esparcidas
Img. 92	X	X	Manchas oscuras Cicatrices Lenticelas oscuras esparcidas
Img. 93	X		Lenticelas oscuras esparcidas Manchas oscuras
Img. 94	X		Lenticelas oscuras esparcidas
Img. 95	X		Lenticelas oscuras esparcidas Manchas Oscuras Manchas necróticas
Img. 96	X		Lenticelas oscuras esparcidas Manchas oscuras
Img. 97	X		Lenticelas oscuras esparcidas



Manchas Oscuras = 62 imágenes de mangos

Lenticelas oscuras esparcidas = 51 imágenes de mangos

Manchas Necróticas = 14 imágenes de mangos

Antracnosis = 4 imágenes de mangos

Cicatriz = 15 imágenes de mangos

Ralladura = 11 imágenes de mangos

Manchas oscuras causadas por bacterias = 9 imágenes de mangos

Secreción de latex = 17 imágenes de mangos

Larva de mosca = 7 imágenes de mangos

Rozadura = 2 imágenes de mangos

Lenticelas rojas esparcidas = 3 imágenes de mangos

Para nuestra investigación y poner a prueba los algoritmos se tuvo en cuenta los 4 primeros defectos, porque es lo que más se repite en nuestro repositorio y por sus grandes defectos en comparación con otros.

CONSTANCIA DE VALIDACIÓN

EXPERTO EN: Agronomía (Defectos externos del mango)

Quien suscribe _____,
con DNI N° _____ mediante la presente hago
constar que el instrumento de recolección de datos de reconocimiento
de defectos externos del mango: **“UTILIZACIÓN DE TÉCNICAS DE
VISIÓN ARTIFICIAL PARA LA DETECCIÓN AUTOMÁTICA DE
DEFECTOS EXTERNOS DEL MANGO”**, cuyos autores son los
estudiantes de pre-grado Heiner Anibal Saldaña Saldaña y Erick David
Lucero Carrillo, reúne los requisitos suficientes y necesarios para ser
considerados válidos y confiables; por lo tanto están aptos para ser
aplicados en el logro de los objetivos que desea obtener.

Atentamente

Ing. Walter Suyón Maza
DNI N°: 17425461

Anexo 2

Algoritmo Canny

```

54 void MainWindow::on_btnCanny_clicked()
55 {
56
57     QSize sz = ui->original->frameSize();
58     cv::resize(imagen, imagen, Size(sz.width(), sz.height()));
59
60
61     // Canny(imagen, imagen, 0, 30, 3);
62
63
64
65     cvtColor(imagen, imagen, CV_RGB2GRAY); // se transforma la imagen, de RGB a una imagen en escala de grises
66     //Aplicamos filtro gaussiano
67     // GaussianBlur(imagen,imagen,Size(11,7),1.5);
68
69     threshold(imagen,imagen,0,255,THRESH_OTSU);
70     //Canny(imagen, imagen, 50, 200, 3);
71     Canny(imagen, imagen, 0, 30, 3);
72     // se aplica el método de canny para la detección de contornos
73
74
75     const uchar *qImage1 = (const uchar*)imagen.data;
76     //const uchar *qImage2 = (const uchar*)org.data;
77
78     QImage img1(qImage1, imagen.cols, imagen.rows, imagen.step, QImage::Format_Indexed8 );
79     // QImage img2(qImage2, org.cols, org.rows, org.step, QImage::Format_RGB888 );
80
81     ui->nueva->setPixmap(QPixmap::fromImage(img1.rgbSwapped()));
82     // ui->original->setPixmap(QPixmap::fromImage(img2.rgbSwapped()));
83
84
85
86 }
87

```

Algoritmo K-Means

```

239 void MainWindow::on_btn_Kmeans_clicked()
240 {
241     int REDUCEC_SIZE = 460;
242     /*
243     // IplImage* original = cvLoadImage("c://Users/usuario/Documents/build-Mango1-Desktop_Qt_5_3_MinGW_32bit-Debug/mango1.jpg
244     // IplImage* original = cvLoadImage("c://Users/usuario/Documents/build-Mango1-Desktop_Qt_5_3_MinGW_32bit-Debug/mango2.jpg
245     // IplImage* original = cvLoadImage("c://Users/usuario/Documents/build-Mango1-Desktop_Qt_5_3_MinGW_32bit-Debug/mango3.jpg
246     // IplImage* original = cvLoadImage("c://Users/usuario/Documents/build-Mango1-Desktop_Qt_5_3_MinGW_32bit-Debug/mango4.jpg
247     IplImage* original = cvLoadImage("c://Users/usuario/Pictures/mango2.jpg");
248     //IplImage* original = cvLoadImage("c://Users/usuario/Documents/build-Mango1-Desktop_Qt_5_3_MinGW_32bit-Debug/mango10.jp
249
250     Mat src = Mat(original);
251     */
252
253     QString nombre= QFileDialog::getOpenFileName();
254     QByteArray nombre2 = nombre.toLocal8Bit();
255     char* filename=nombre2.data();
256     Mat src = imread(filename);
257
258     Mat samples(src.rows * src.cols, 3, CV_32F);
259     for( int y = 0; y < src.rows; y++ )
260         for( int x = 0; x < src.cols; x++ )
261             for( int z = 0; z < 3; z++ )
262                 samples.at<float>(y + x*src.rows, z) = src.at<Vec3b>(y,x)[z];
263
264
265     int clusterCount = 3;
266     Mat labels;
267     int attempts = 5;
268     Mat centers;
269     kmeans(samples, clusterCount, labels, TermCriteria(CV_TERMCRIT_ITER|CV_TERMCRIT_EPS, 10000, 0.0001), attempts, KMEA
270
271
272     Mat new_image( src.size(), src.type() );
273     for( int y = 0; y < src.rows; y++ )

```



```

274     for( int x = 0; x < src.cols; x++ )
275     {
276         int cluster_idx = labels.at<int>(y + x*src.rows,0);
277         new_image.at<Vec3b>(y,x)[0] = centers.at<float>(cluster_idx, 0);
278         new_image.at<Vec3b>(y,x)[1] = centers.at<float>(cluster_idx, 1);
279         new_image.at<Vec3b>(y,x)[2] = centers.at<float>(cluster_idx, 2);
280     }
281
282
283     float scale = new_image.cols / (float) REDUCEC_SIZE;
284
285     if ( new_image.cols > REDUCEC_SIZE ){
286         int scaledHeight = cvRound(new_image.rows / scale);
287         cv::resize(new_image, new_image , Size(REDUCEC_SIZE,scaledHeight ));
288         imshow( " image", new_image );
289         waitKey( 0 );
290     }else{
291         cv::resize(new_image, new_image , Size(new_image.cols,new_image.rows ));
292         imshow( " image", new_image );
293         waitKey( 0 );
294     }
295 }
296

```

Algoritmo Otsu

```

296
297 void MainWindow::on_btnUmbral1_clicked() {
298     int REDUCEC_SIZE = 460;
299     // obteniendo la imagen
300     QString nombre= QFileDialog::getOpenFileName();
301     QByteArray nombre2 = nombre.toLocal8Bit();
302     char* filename=nombre2.data();
303
304     Mat original = imread(filename);
305     Mat Img_Grey = original.clone();
306     blur(original, original, Size(10,20));
307
308     cvtColor(original, Img_Grey, CV_RGB2GRAY);
309
310     Mat img_bw = Img_Grey < 128;
311
312
313
314
315
316     // Reducir el tamaño de la imagen para mejor rendimiento.
317     float scale = original.cols / (float) REDUCEC_SIZE;
318     if (original.cols > REDUCEC_SIZE && Img_Grey.cols > REDUCEC_SIZE && img_bw.cols > REDUCEC_SIZE){
319         int scaledHeight = cvRound(original.rows / scale);
320         cv::resize(original, original , Size(REDUCEC_SIZE,scaledHeight ));
321         imshow("Original 1",original);
322
323         int scaledHeight2 = cvRound(Img_Grey.rows / scale);
324         cv::resize(Img_Grey, Img_Grey , Size(REDUCEC_SIZE,scaledHeight2 ));
325         imshow("Gris 1",Img_Grey);
326
327         int scaledHeight3 = cvRound(img_bw.rows / scale);
328         cv::resize(img_bw, img_bw , Size(REDUCEC_SIZE,scaledHeight3 ));
329         imshow("Binario 1",img_bw);
330
331     }else{
332         cv::resize(original, original , Size(original.cols,original.rows ));
333         imshow("Original 2",original);
334
335
336         cv::resize(Img_Grey, Img_Grey , Size(Img_Grey.cols,Img_Grey.rows ));
337         imshow("Grises 2 ",Img_Grey);
338
339         cv::resize(img_bw, img_bw , Size(img_bw.cols,img_bw.rows ));
340         imshow("Binario 2",img_bw);
341
342     }
343
344 }
345

```



Extracción de características de textura utilizando GLCM

```
CvGLCM* glcm;
Mat gray = cr.clone();
cvtColor(cr, gray, CV_RGB2GRAY);
imshow("asdasd",gray);
IplImage* copycr = cvCloneImage(&(IplImage) gray);

glcm = cvCreateGLCM(copycr, 2, NULL,4, CV_GLCM_OPTIMIZATION_NONE);
cvCreateGLCMDescriptors(glcm, CV_GLCMDESC_OPTIMIZATION_ALLOWDOUBLENES);
cout << "Correlacion : " << cvGetGLCMDescriptor(glcm,0,CV_GLCMDESC_CORRELATION) << endl;
cout << "Contraste : " << cvGetGLCMDescriptor(glcm,0,CV_GLCMDESC_CONTRAST) << endl;
cout << "Homogeneidad : " << cvGetGLCMDescriptor(glcm,0,CV_GLCMDESC_HOMOGENITY) << endl;
cout << "Energia : " << cvGetGLCMDescriptor(glcm,0,CV_GLCMDESC_ENERGY) << endl;
// cout << "Entropia : " << cvGetGLCMDescriptor(glcm,0,CV_GLCMDESC_ENTROPY) << endl;

cvWaitKey(0);
```

Extracción de características de Forma utilizando Blurred Shape

Model (BSM)

```
30 void BlurredShapeModel::featuresExtraction()
31 {
32     features.clear();
33     //vector<double> features;
34     //medir tiempo de procesamiento
35     Mat imgg = this->sourceImage.clone();
36     cvtColor(this->sourceImage, imgg, CV_RGB2GRAY);
37     /// automatic canny
38     double otsu_thresh_val = cv::threshold(
39         imgg, imgg, 0, 255, CV_THRESH_OTSU
40     );
41     double high_thresh_val = otsu_thresh_val,
42         lower_thresh_val = otsu_thresh_val * 0.5;
43     Canny(this->sourceImage, this->edges, lower_thresh_val, high_thresh_val);
44
45     int width = nX, height = nY;
46     vector<Mat> imageSegment = ImageProcessor::splitImage(this->edges, width, height);
47     vector<Segment> segments;
48     vector<vector<int>> matriz = initializeArray(width, height);
49     int i = 0;
50     for (int j = 0 ; j < width; j++)
51     {
52         for ( int k = 0 ; k < height; k++)
53         {
54             vector<int> nb = neighbours(j,k,matriz);
55             Segment sg (imageSegment[i],nb,j,k);
56             segments.push_back(sg);
57             i++;
58         }
59     }
60
61     //Procedimiento para obtener el vector de caracteristicas
62     vector<double> vectorr = initializeVector(width,height);
63
64     cout << "Segments: " << segments.size() << endl;
65
```



```

66  for ( int j = 0 ; j < segments.size() ; j++)
67  {
68      Segment seg = segments.at(j);
69      Mat image = seg.getSegment();
70
71      for ( int a = 0; a < image.cols; a++)
72      {
73          for ( int b = 0 ; b < image.rows ; b++)
74          {
75              //cout << "a: " << a << " , b: " << b << endl;
76              int color = image.at<char>(a,b);
77              //cout << "color : " << color << endl;
78              ///CORREGIR?
79              if (color == -1) // Detectando un pixel de borde
80              { // Obteniendo las coordenadas absolutas del pixel
81
82                  int coordWidth = (a + (seg.width * 2 * seg.getPosX()));
83                  int coordHeight = (b + (seg.height * 2 * seg.getPosY()));
84                  // Obteniendo las regiones vecinas a dicho pixel
85                  vector<int>neighbors = seg.getNeighbors();
86                  // Inicializando vector de características
87                  vector<int> ds = initializeIntVector(width, height);
88                  for (int k = 0; k < neighbors.size(); k++)
89                  {
90                      // Calculando las distancias entre el pixel
91                      ds[(neighbors[k] - 1)] = distanceTo(coordWidth, segments[neighbors[k] - 1].centroidWidth
92                                                         , coordHeight, segments[neighbors[k] - 1].centroidHeight);
93                      ///CORREGIR?
94                  }
95                  // Calculando sumatoria entre todas las distancias obtenidas
96                  int sum = 0;
97                  for (int l = 0; l < ds.size(); l++) {
98                      sum += ds[l];
99                  }
100                 // Realizando la primera normalizacion. Se obtienen valores decimales comprendidos entre [0, 1]
101                 vector<double> n1 = initializeVector(width, height);
102                 for (int k = 0; k < n1.size(); k++)
103                 {
104                     n1[k] = (double)ds[k] / (double)sum;
105
106                     ///CORREGIR?
107                 }
108                 // Calcular la inversa de cada distancia
109                 vector<double> ids = initializeVector(width, height);
110                 for (int e = 0; e < ds.size(); e++) //IDS O DS?
111                 {
112                     if(n1[e] != 0){
113                         // ids[e] = 1 - n1[e];
114                         ids[e] = pow(n1[e], -1); // Correccion
115                     }
116                     // Calculando sumatoria entre todas las distancias inversas obtenidas
117                     double sum2 = 0;
118                     for (int l = 0; l < ids.size(); l++) {
119                         sum2 += ids[l];
120                     }
121                     // Realizando la normalizacion de los valores obtenidos en las inversas
122                     vector<double> n2 = initializeVector(width, height);
123                     for (int k = 0; k < n2.size(); k++) {
124                         n2[k] = (double)ids[k] / (double)sum2;
125                     }
126                     // Actualizar el vector salida sumando a las posiciones de las regiones afectadas el
127                     // valor de las distancias obtenido en los pasos anteriores.
128                     for (int k = 0; k < vectorr.size(); k++) {
129                         vectorr[k] = vectorr[k] + n2[k];
130                     }
131                     ///
132                 }
133             }
134         }
135     }
136
137     }
138
139     double sum3 = 0;
140     // cout << "VECTOR SIZE : " << vectorr.size();
141     for (int l = 0; l < vectorr.size(); l++) {
142         sum3 += vectorr[l];
143     }

```

```
137
138     }
139     double sum3 = 0;
140     // cout << "VECTOR SIZE : " << vectorr.size();
141     for (int l = 0; l < vectorr.size(); l++) {
142         sum3 += vectorr[l];
143     }
144
145     for (int k = 0; k < vectorr.size(); k++) {
146         vectorr[k] = (double) vectorr[k] / (double) sum3;
147     }
148
149     for (int k = 0; k < vectorr.size(); k++) {
150         this->features.push_back(vectorr[k]);
151     }
152
153     // cout << "Fin de la extraccion de características" << endl;
154     // cout << "features Size: " << this->features.size();
155 }
156
157 string BlurredShapeModel::featuresToString(ofstream& fs, String tipo)
158 {
159     QString outString;
160
161     cout << "features Size : " << features.size() << endl;
```