

Universidad Católica de Santa María
Facultad de Ciencias e Ingenierías Físicas y Formales
**Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica
y Mecatrónica**



**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE
CLASIFICACIÓN DE LÚCUMAS APLICANDO VISIÓN
ARTIFICIAL**

Tesis presentada por el Bachiller:
Fernández Medina, Alexander Jorge

Para optar por el Título Profesional de:
Ingeniero Mecatrónico

Asesor:
M.Sc. Mestas Ramos, Sergio

AREQUIPA – PERÚ

2019



Universidad Católica de Santa María

(51 54) 382038 Fax:(51 54) 251213 ✉ ucsm@ucsm.edu.pe 🌐 http://www.ucsm.edu.pe Apartado:1350

AREQUIPA - PERÚ

**ESCUELA PROFESIONAL DE INGENIERÍA MECÁNICA, MECÁNICA
ELÉCTRICA Y MECATRÓNICA**

INFORME DICTAMINATORIO

VISTO

EL BORRADOR DE TESIS TITULADO:

**“DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO
DE CLASIFICACIÓN DE LÚCUMAS APLICANDO
VISION ARTIFICIAL”**

Presentado por el Bachiller:

FERNANDEZ MEDINA ALEXANDER JORGE

Nuestro **DICTAMEN** es:

FAVORABLE

OBSERVACIONES: *NINGUNA*

Arequipa, *01 de Julio* 2019



ING. SERGIO MESTAS RAMOS



ING. MARCELO QUISPE CCACHUCO

AGRADECIMIENTOS

Agradezco a Dios por guiar mi camino y darme fuerzas para superar obstáculos y dificultades a lo largo de toda mi vida.

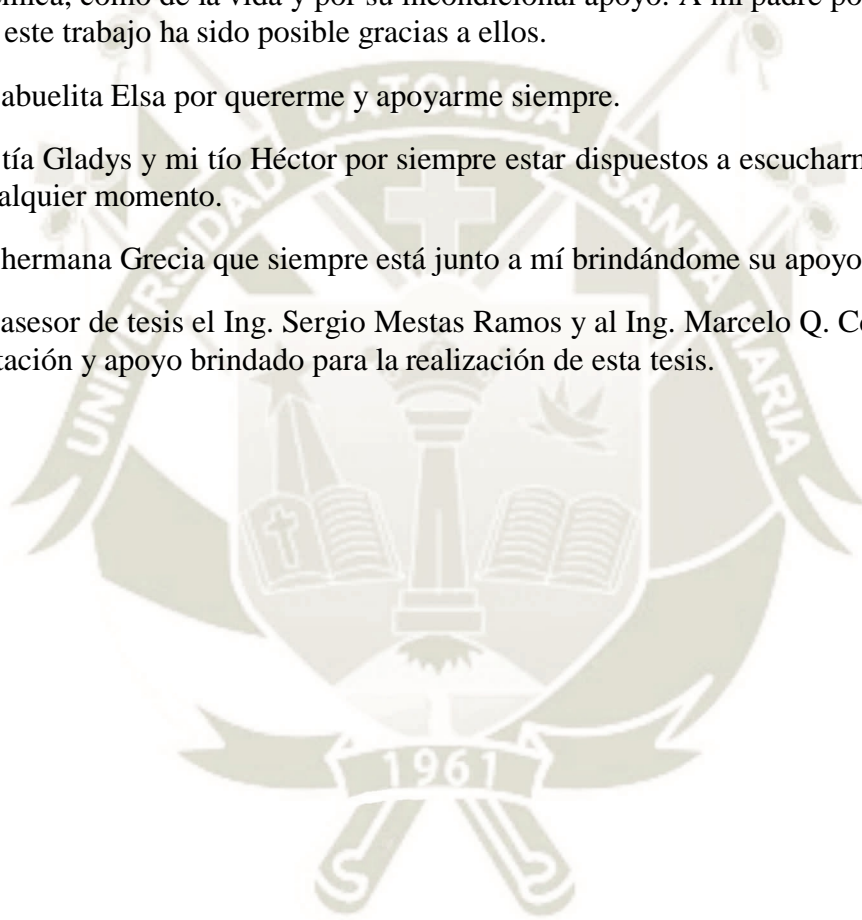
A mi madre por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida y por su incondicional apoyo. A mi padre por sus consejos. Todo este trabajo ha sido posible gracias a ellos.

A mi abuelita Elsa por quererme y apoyarme siempre.

A mi tía Gladys y mi tío Héctor por siempre estar dispuestos a escucharme y ayudarme en cualquier momento.

A mi hermana Grecia que siempre está junto a mí brindándome su apoyo.

A mi asesor de tesis el Ing. Sergio Mestas Ramos y al Ing. Marcelo Q. Ccachuco, por la orientación y apoyo brindado para la realización de esta tesis.



ÍNDICE

AGRADECIMIENTOS	i
INDICE	ii
INDICE DE FIGURAS	v
INDICE DE TABLAS	vii
RESUMEN	viii
ABSTRACT	ix
INTRODUCCIÓN	x
CAPITULO I	
1. MARCO METODOLÓGICO	1
1.1. Descripción del Problema.	1
1.2. Justificación	1
1.3. Objetivos.	2
1.4. Hipótesis.	2
1.5. Alcances.	3
1.6. Limitaciones.	3
1.7. Antecedentes.	3
CAPITULO II	
2. MARCO TEÓRICO	8
2.1. Visión artificial	8
2.1.1. Antecedentes	8
2.1.2. Visión humana y artificial	9
2.1.3. Aplicaciones de la visión artificial.	10
2.1.3.1. Visión artificial en la automatización de procesos	10
2.1.4. Procesamiento de imágenes en Visión Artificial	11
2.1.5. Elementos de los sistemas de visión artificial	12
2.1.5.1. Cámara de captura de imagen	12
2.1.5.2. Iluminación artificial	13
2.1.5.2.1. Técnicas de iluminación	14
2.1.5.2.2. Fuentes de iluminación artificial	16
2.1.5.3. Controlador	17
2.1.5.4. Sensores y actuadores	17
2.1.6. Software de Visión artificial	19
2.1.6.1. Matlab	20
2.1.6.1.1. Interfaz gráfica GUI	21
2.1.6.2. OpenCV	21

2.1.7. Etapas en el proceso de visión artificial	23
2.1.7.1. Adquisición de imagen	24
2.1.7.2. Pre procesamiento digital de imagen	25
2.1.7.2.1. Filtros de imagen	25
2.1.7.3. Segmentación	27
2.1.7.3.1. Umbralización	28
2.1.7.4. Extracción de características	30
2.1.7.5. Reconocimiento e interpretación	30
2.2. Red neuronal artificial	31
2.2.1. Antecedentes	32
2.2.2. Aplicaciones de las redes neuronales	33
2.2.3. Ventajas de las redes neuronales	35
2.2.4. Arquitectura de una RNA	36
2.2.4.1. Red neuronal monocapa	37
2.2.4.2. Red neuronal multicapa	37
2.2.5. Elementos básicos de una red neuronal	38
2.2.5.1. Funciones de entrada	38
2.2.5.2. Funciones de activación	39
2.2.5.3. Funciones de salida	39
2.2.6. Algoritmo de aprendizaje	40
2.2.6.1. Aprendizaje supervisado	40
2.2.6.2. Aprendizaje no supervisado	40
2.2.7. Backpropagation	41
2.3. La lúcumá	42
2.3.1. Generalidades	42
2.3.2. Composición química de la lúcumá	43
2.3.3. Usos y aplicaciones	45
2.4. Esquema conceptual	45
CAPITULO III	
3. DISEÑO DEL SISTEMA	47
3.1. Diseño del prototipo	47
3.1.1. Selección de hardware	48
3.1.1.1. Selección de cámara	48
3.1.1.2. Selección del sistema de iluminación	49
3.1.1.3. Selección de los instrumentos electrónicos y de control	51
3.1.2. Implementación del prototipo	53
3.2. Diseño del sistema de visión artificial	63
3.2.1. Adquisición y captura de imagen	64

3.2.2. Pre-Procesamiento de imagen	65
3.3.3. Segmentación de la imagen	66
3.2.4. Extracción de características	70
3.2.5. Reconocimiento e interpretación	71
3.3. Diseño de la red neuronal	73
3.3.1. Arquitectura de la red neuronal	73
3.3.2. Entrenamiento de la red neuronal	74
3.4. Etapa de Clasificación	82
3.4.1. Tolerancias	83
3.5. Interfaz gráfica	84
3.6. Costos	85
CAPITULO IV	
4. PRUEBAS Y RESULTADOS	87
CONCLUSIONES	97
RECOMENDACIONES	98
BIBLIOGRAFÍA	99
ANEXOS	105

ÍNDICE DE FIGURAS

Figura 1. Primeras Fotografías	8
Figura 2. Representación del Ojo mirando	9
Figura 3. Robots con aplicación de Visión Artificial	11
Figura 4. Diagrama de un sistema de Visión Artificial	12
Figura 5. Diagrama de Flujo de Captura de Imagen	13
Figura 6. Iluminación Difusa	14
Figura 7. Iluminación Frontal	15
Figura 8. Iluminación Direccional	15
Figura 9. Iluminación estructurada	16
Figura 10. Sensor ultrasónico HC-SR04	18
Figura 11. Detección de colores median OpenCV	22
Figura 12. Diagrama de bloques de las etapas de un sistema de Visión artificial	24
Figura 13. Histograma de imagen	29
Figura 14. Objeto segmentado por umbralización	29
Figura 15. Reconocimiento de objetos	31
Figura 16. Jerarquía de las redes neuronales	36
Figura 17. Perceptron Multicapa	38
Figura 18. Red neuronal feed-forward con backpropagation	42
Figura 19. Dispositivo de adquisición de imagen	49
Figura 20. Cinta de iluminación led	50
Figura 21. Pruebas de iluminación en el módulo	51
Figura 22. Arduino Mega 2560	51
Figura 23. Servomotor TowerPro MG995 implementado	52
Figura 24. Sensor ultrasónico HC-SR04 implementado	53
Figura 25. Bosquejo inicial del prototipo	54
Figura 26. Motor monofásico con reductor	55
Figura 27. Arranque por condensador	55
Figura 28. Condensadores comerciales para arranque por condensador	55
Figura 29: Relación de transmisión por cadena	56
Figura 30. Instalación de los rodamientos	57
Figura 31. Distribución de iluminación dentro del módulo de captura	57
Figura 32. Circuito de conexión de Arduino	58
Figura 33. Circuito impreso elaborado	58
Figura 34. Módulo de captura implementado	59

Figura 35. Sistema de clasificación	59
Figura 36. Diagrama de flujo de funcionamiento del prototipo	61
Figura 37: Prototipo de clasificación elaborado	62
Figura 38. Sistema de transmisión por cadena	62
Figura 39. Diagrama de flujo del algoritmo de visión artificial	63
Figura 40. Captura de imagen de una lúcumá	64
Figura 41. Conversión en escala de grises	65
Figura 42. Umbralización empleando el método de Otsu	66
Figura 43. Histograma de la captura de la lúcumá en el módulo	67
Figura 44. Corrección de ruido empleando Otsu y filtro Gaussiano	67
Figura 45. Pruebas de distintos métodos de umbralización	68
Figura 46. Umbralización por el método de Otsu	68
Figura 47. Llenado de agujeros	69
Figura 48. Operación morfológica de apertura con distintos elementos estructurales	69
Figura 49. Operación morfológica de apertura	70
Figura 50. Black Box	70
Figura 51. Cuadro de regionprops en Matlab	71
Figura 52. Principales propiedades graficadas en la imagen	71
Figura 53. Obtención de la altura inicial de la regla patrón	72
Figura 54. Obtención de las diferentes escalas	72
Figura 55. Diagrama de flujo de entrenamiento de la red neuronal	75
Figura 56. Pesado de la lúcumá	76
Figura 57. Estructura de la red neuronal diseñada	78
Figura 58. Creación de la red neuronal	79
Figura 59. Entrenamiento de la red neuronal	80
Figura 60. Regresión lineal de la red	81
Figura 61. Error cuadrático medio de la red	81
Figura 62. Variación de la gradiente en la red	82
Figura 63. Interfaz de usuario GUI	84
Figura 64. Función de visualizar el proceso por cámara	85
Figura 65. Función de visualizar segmentación en GUI	85
Figura 66. Diámetro real vs diámetro obtenido	89
Figura 67. Peso real vs peso obtenido	91
Figura 68. Coincidencia de clasificación por especificación de calibre	93
Figura 69. Coincidencia de clasificación por categorías de calibre	95

ÍNDICE DE TABLAS

Tabla 1. Composición Proximal, Minerales y Vitaminas en la Lúcumá	44
Tabla 2. Fuentes de iluminación e intensidad	50
Tabla 3: Especificaciones del microcontrolador Arduino Mega 2560	52
Tabla 4. Factor de conversión para distintos tamaños de lúcumá	73
Tabla 5. Banco de datos de la red neuronal	77
Tabla 6. Especificaciones de calibre para lúcumá	82
Tabla 7. Categorías de calibre para lúcumá	83
Tabla 8. Criterio de coincidencia empleado	83
Tabla 9. Costos del prototipo	86
Tabla 10. Registro de características seleccionadas de 50 lúcumás	88
Tabla 11. Comparación de los valores reales y obtenidos por software para el diámetro ecuatorial	90
Tabla 12. Comparación de los valores reales y obtenidos por software para el peso	92
Tabla 13. Clasificación por especificaciones de calibre	94
Tabla 14. Clasificación por categorías de calibre	96

Resumen

El desarrollo del presente trabajo de investigación propone un sistema automatizado de clasificación por tamaño y peso aplicada a la agroindustria nacional para disminuir el error en clasificación debido al trabajo manual, así como su tiempo y coste. Siendo el objetivo principal el desarrollo de un sistema que permita la clasificación por categoría y calibre empleando visión artificial y redes neuronales para el cálculo del peso.

La clasificación se realiza tomando como referencia la norma regional para la lúcuma código stan 305R-2011 de la Organización de las Naciones Unidas para la Agricultura y la Alimentación, la cual estipula una clasificación por calibre y categoría.

El sistema de visión artificial comprende la selección de la cámara y el sistema de iluminación, así como su distribución para obtener una iluminación uniforme. Se detallaron las múltiples etapas del sistema de visión artificial desde la captura de imagen, el pre procesamiento para la reducción del ruido, pasando por la segmentación para separar la lúcuma del fondo hasta la extracción e interpretación de las características. Posteriormente se realizó el entrenamiento de una red neuronal artificial para la estimación aproximada de los pesos de la lúcuma a partir de las imágenes capturadas.

Adicionalmente se elaboró un prototipo para la validación del algoritmo desarrollado, así como medir su eficiencia. El tiempo de procesamiento del algoritmo desarrollado es de 1 segundo. Se logró un acierto del 94% en la clasificación por calibre, 98% en la clasificación por categoría, un error máximo de 3.20% en el tamaño y 5.00% de peso de lúcuma para un grupo de muestra de 50 lúcumas, lográndose cumplir las tolerancias de calidad señaladas en el código.

Palabras clave: Visión artificial, lúcuma, red neuronal artificial.

Abstract

The development of this research work is an automated system of classification of size and weight applied to the agroindustry to reduce the error in the classification due to manual work, as well as its time and cost. The main objective being the development of a system that allows classification by category and size using artificial vision and artificial neural networks for weight calculation.

The classification is made taking as reference the regional norm for the lucuma codex at 305R-2011 of the United Nations Organization for Agriculture and Food, which is a classification by size and category.

The artificial vision system includes the selection of the camera and the lighting system, as well as its distribution to obtain uniform illumination. The multiple stages of the artificial vision system from the image capture, the pre-processing for noise reduction, going through the segmentation to separate the lucuma from the bottom to the extraction and interpretation of the characteristics. Later we will perform the training of an artificial neural network for the approximate estimation of the lucuma weights from the captured images.

A prototype was also developed for the validation of the algorithm developed, as well as to measure its efficiency. The processing time of the algorithm is 1 second. We achieved a success of 94% in the classification by size, 98% in the classification by category, a maximum error of 3.20% in size and 5.00% weight of lucuma for a sample group of 50 lucumas, achieving compliance with quality tolerances indicated in the codex.

Keywords: Artificial vision, lucuma, artificial neural network.

Introducción

La clasificación es algo que se puede ver en diversos aspectos de la vida cotidiana y/o profesional, como por ejemplo en el campo de la agricultura, donde, los agricultores clasifican sus productos por tamaño, color o nivel de madurez. En este caso, la clasificación toma mayor importancia pues en base a la venta de estos productos es que los agricultores empiezan a generar sus ingresos, además gracias a la clasificación es que se puede asegurar la calidad del producto que se brinda para el consumo humano.

Las empresas nacionales dedicadas a la comercialización de productos hortofrutícolas indican que el 10% de los costos de producción son debido a fallos en inspectores humanos al momento de clasificar las frutas, esto se debe a que el margen de error de un humano al clasificar frutas es del 20%.

En base a lo antes mencionado, las empresas buscan alternativas tecnológicas que puedan garantizar una alta eficiencia en el sistema de clasificación. Como respuesta tecnológica en los últimos años ha aparecido la Visión artificial como una alternativa de solución innovadora y confiable.

La visión artificial por computadora es una disciplina en creciente auge con multitud de aplicaciones, como inspección automática, reconocimiento de objetos, mediciones, robótica etc. El futuro es aún más prometedor; la creación de máquinas autónomas capaces de interactuar inteligentemente con el entorno sucede necesariamente por la capacidad de percibir éste.

CAPÍTULO I

1. MARCO METODOLÓGICO

1.1. DESCRIPCIÓN DEL PROBLEMA

La visión artificial es utilizada para el reconocimiento de objetos a través de la extracción de información cuantitativa de imágenes digitales con el objetivo de proporcionar una información específica y objetiva de lo que se está analizando. En los últimos años han ido apareciendo diversos trabajos que recomiendan el uso de esta alternativa tecnológica para la selección y manejo de diversos productos, los cuales son de importancia económica dentro del rubro de la agro exportación, como el café, la chirimoya, mango, palta, entre otros.

La lúcuma es considerada un producto peruano de bandera que busca posesionarse como valor agregado, transformando la pulpa en harina para la preparación de helados y postres, mermeladas, además de néctares y licores, entre otros derivados que aún están en investigación.

Las exportaciones de esta fruta han tenido un crecimiento sostenido en los últimos años. En el 2013, el valor de las exportaciones alcanzó los 2,8 millones de dólares, lo cual significó un incremento de casi diez veces mayor al del 2008, cuando se exportaban 297 mil dólares en este producto. A la actualidad a cobrado mayor importancia en el ámbito de la agro exportación, alcanzando un crecimiento de 16% cada año desde el 2015.

1.2. JUSTIFICACIÓN

En el Perú muchos procesos que pueden ser automatizados se siguen realizando de manera manual o semiautomática, abriendo la posibilidad de una clasificación errónea debido a algún error humano producto de factores como: cansancio y fatiga visual como ocurre con los operadores humanos. Y además de restar competitividad al mercado en lo que a tiempo de producción se refiere.

El presente trabajo plantea una alternativa para la clasificación con visión artificial de la lúcuma, la cual en los últimos tiempos ha cobrado mucha importancia económica en la rama agroexportadora.

De lo expuesto anteriormente se puede plantear la siguiente interrogante: ¿Es posible diseñar e implementar un sistema de clasificación de lúcumas con visión artificial?

1.3. OBJETIVOS

Objetivo principal

Diseñar e implementar un prototipo de clasificación de lúcumas empleando visión artificial y redes neuronales.

Objetivos específicos

- Diseñar un algoritmo de procesamiento de imágenes para todas las etapas de visión artificial.
- Diseñar y entrenar redes neuronales para la obtención del peso aproximado de la lúcuma.
- Desarrollar una interfaz de usuario amigable para la observación y control del proceso.
- Diseñar e implementar un prototipo para la validación del algoritmo desarrollado.

1.4. HIPOTESIS

Si en la actualidad la visión artificial y las redes neuronales representan una alternativa tecnológica para cumplir y realizar de manera eficaz diversas actividades relacionadas con la optimización y automatización de procesos industriales, entonces es posible clasificar las lúcumas empleando la visión artificial y las redes neuronales como una alternativa de selección tecnificada y uniforme para el sector productivo agroindustrial.

1.5. ALCANCES

El proyecto se encuentra direccionado a solucionar problemas de manejo agroindustrial puesto que nos permite determinar un sistema de clasificación de las lúcumas mediante el empleo de la visión artificial.

A nivel cuantitativo nos permite clasificar las lúcumas en función a sus dimensiones mediante algoritmos implementados.

1.6. LIMITACIONES

- Por tratarse de un prototipo, los componentes e implementación implican un bajo costo.
- El sistema está limitado a la clasificación de lúcumas y al procesamiento de una sola a la vez.

1.7. ANTECEDENTES

Del Castillo, E. (2018) Desarrollo de un sistema de visión artificial para realizar una clasificación uniforme de limones. Tesis de Pregrado. Universidad Privada del Norte. Trujillo, Perú.

La presente tesis planteó como objetivo principal la creación de un sistema de visión artificial que permita realizar una clasificación uniforme de limones. Las formas y dimensiones de los limones a ser analizados están sujetos al códex de la lima-limón de la Organización de Comida y Agricultura de las Naciones Unidas.

Actualmente se analizó el contexto internacional y nacional determinando que no existe tecnología de información asociada al proceso de clasificación de limones, esto nos brinda la posibilidad de explorar alternativas basadas en áreas de la computación que ayuden en este proceso, tal es el caso de la visión artificial.

Se diseñó una solución siguiendo las fases de visión artificial (adquisición, pre procesamiento, segmentación, descripción y reconocimiento e interpretación), donde se realizó una comparativa entre los algoritmos dentro de cada fase, para identificar cuáles son los que mejor se adaptan a la problemática planteada permitiendo obtener un resultado adecuado.

Luego de haber desarrollado el sistema de visión artificial aplicando los algoritmos seleccionados, se obtuvo como resultado que el sistema tiene una eficacia de 83.9%, una sensibilidad de 82.8% y una especificidad del 100%.

Por otro lado, el tiempo de procesamiento para clasificar un limón es de 0.33 segundos.

Gracias a los resultados obtenidos se pudo comprobar la hipótesis en la que se sostiene que un sistema de visión artificial permite realizar una clasificación uniforme de limones.

Viera, G. (2017) Procesamiento de Imágenes usando OPENCV aplicado en Raspberry Pi para la clasificación del Cacao. Tesis de licenciatura en Ingeniería Mecánico-Eléctrica. Universidad de Piura. Piura, Perú.

El presente trabajo de investigación busca establecer los conceptos básicos para el desarrollo de la visión artificial aplicada a la agroindustria nacional. Siendo su principal objetivo el desarrollo de un sistema de visión artificial para la clasificación de granos de cacao según características externas como el tamaño en la fase final del secado. La tesis está dividida en cuatro capítulos de desarrollo teórico y experimental, un capítulo de conclusiones y recomendaciones, las respectivas referencias, y finalmente los anexos. Los aspectos básicos de esta investigación se resumen en el primer capítulo de la tesis, con el estado del arte del sistema de visión artificial y la problemática de la producción de cacao. En el segundo capítulo se presenta los conceptos básicos y los elementos de un sistema de visión artificial. En el tercer capítulo, se presenta una descripción del prototipo, del software y la interfaz gráfica para la visualización de los resultados. En el cuarto capítulo se presentan los resultados del sistema implementado para la clasificación de grano de cacao orgánico según su tamaño, obteniéndose un 88.4% de acierto en la clasificación. Y finalmente en el quinto capítulo se presentan las conclusiones de la tesis. El tiempo invertido en la adquisición de la imagen más el tiempo de análisis del grano de cacao fue menor a 2 segundos por imagen.

Valdivia, C. (2016) Diseño de un sistema de Visión Artificial para la Clasificación de Chirimoyas basado en medidas. Pontificia Universidad Católica de Santa María. Lima, Perú.

El desarrollo de este trabajo, presenta el diseño de un sistema de visión artificial, capaz de medir *Annona Cherimola Mill* (Chirimoya) del ecotipo *Aurora*, las cuales provienen de la comunidad de Callahuanca, para posteriormente clasificar aquellas que cumplan con una dimensión de 10 x 12 cm empleando para la etapa de procesamiento: Ajuste de contraste y para la etapa de segmentación, *Cierre, Apertura y Código Cadena*; los métodos seleccionados cumplen con los tiempos de computo de diseño.

Fue posible clasificar exitosamente 91 de 91 Chirimoyas de Categorías Extra y Primera estudiadas con un 100 % de efectividad, una precisión de medida de 0,35mm y un error permisible de 2,18mm. Las pruebas se efectuaron en un prototipo diseñado para tal motivo y no en el equipo final; el error obtenido durante las pruebas es menor al establecido en los requerimientos en cuanto a la selección del fruto.

En la memoria descriptiva, se presentan los cálculos referentes a la selección del dispositivo de captura, lentes y requerimientos de la iluminación empleando la metodología de diseño VDI-2221. El control del banco de pruebas, empleado para validar los algoritmos realizados, se realizó con un PLC y para realizar el procesamiento de imágenes, se optó por un computador de escritorio con procesador Intel Core i5 CPU 2.53GHz junto con una cámara con comunicación USB 3.0. El diseño mecánico-eléctrico no es motivo de estudio en la presente investigación.

Castillo, J. (2015) Sistema de Visión Artificial Humanoide para reconocimiento de formas y patrones de objetos, aplicando Redes Neuronales y Algoritmos de Aprendizaje Automático. Universidad Ricardo Palma. Lima, Perú.

Esta tesis, desarrolla un sistema de Visión Artificial el cual detecta objetos bajo un ambiente a campo abierto de terreno arenoso, con un fondo incierto bajo ciertas condiciones de luminosidad, brillo saturación, contraste. Para ello se desarrolló un sistema el cual permite ver la profundidad y la distancia la cual se encuentra determinado punto de la imagen de la cámara respecto al mundo real, todo por medio de algoritmos computacionales y teoría de Visión estereoscópica, aproximándose así a un sistema de Visión artificial Humanoide al percibir la profundidad y el reconocimiento de objetos bajo el concepto que se presenta en el libro “A Humanoid Vision System for Versatile Interaction” escrita por Yasuo K., Sebastien R., Oliver

S., Gorgon C. & Akihiko N. También se desarrolló un sistema el cual permite identificar el objeto objetivo, ya sea para su posterior manipulación, o posterior acción del sistema, lo cual se implementó una red neuronal Multicapa el cual permite diferenciar de entre 3 objetos, el final deseado. El uso de la Red Neuronal Artificial guarda un papel muy importante.

Las conexiones entre neuronas tienen pesos asociados que representan la influencia de una sobre la otra. Si dos neuronas no están conectadas, el correspondiente peso de enlace es cero. Esencialmente, cada una envía su información de estado multiplicado por el correspondiente peso a todas las neuronas conectadas con ella. Luego cada una, a su vez, suma los valores recibidos desde sus dendritas para actualizar sus estados respectivos. Se emplea normalmente un conjunto de ejemplos representativos de la transformación deseada para "entrenar" el sistema, que, a su vez, se adapta para producir las salidas deseadas cuando se lo evalúa con las entradas "aprendidas". XVII Para la percepción de profundidad, se realizó con un tablero de ajedrez, al identificar las esquinas de cada cuadrado ubicados cada 2.54 cm a una distancia de la cámara de 79 cm medidos en real, obteniendo resultados muy aproximados entre [71 – 75] cm con un error de hasta 4 cm. Esto sin embargo se puede corregir aplicando mayor uso de imágenes para lectura y calibración de cámara y luego aplicar los algoritmos de estereoscopia.

Este error, es aceptable por ejemplo en sistemas de prevención de choques en los autos, los cuales tienen un retrovisor pantalla en su panel principal, mostrándose en el screen la distancia del obstáculo detectado respecto al auto a fin de evitar malas maniobras o choques. De esta manera en este proyecto de tesis se demuestra la identificación de objetos, y la percepción de la profundidad con el tablero de ajedrez.

Atencio, P. (2009) Desarrollo de un Sistema de Visión Artificial para el Control de Calidad del Mango de azúcar de exportación. Universidad del Magdalena. Santa Marta, Colombia.

En este trabajo, se describe el procedimiento para la construcción de un método que permita clasificar el fruto del “mango de azúcar” (*Mangifera Indica L.*) para exportación a través de técnicas de visión por computador y procesamiento digital de imágenes, mediante la aproximación del volumen y peso del fruto utilizando para esto el análisis de componentes principales y, por otro lado, realizando un estudio de color del fruto del mango, en base a una imagen bidimensional del fruto del mismo.



CAPÍTULO II

2. MARCO TEÓRICO

2.1. Visión artificial

2.1.1. Antecedentes

- En 1826, Niepce (1765-1833) llevó a cabo la primera fotografía, colocando una superficie fotosensible dentro de una cámara oscura para fijar la imagen.
- En 1838 el químico francés Daguerre (1787-1851) hizo el primer proceso fotográfico práctico. Empleo una placa fotográfica la cual se revelo con vapor de mercurio y se fijó con trisulfato de sodio.



Figura 1. Primeras Fotografías

Fuente: Niépce, 1826

- Años 50: La fotografía en color, los barredores multiespectrales y el radar, eran claros ejemplos de espectro electromagnético.
- 1960: Uso de la visión artificial para el estudio de recursos naturales, variables meteorológicas, entre otras.
- 1980: Aplicado para el control de calidad y para la robótica.
- 1997: Interpretación de imágenes de planetas y galaxias.
- 2000: Imágenes en teléfonos móviles, videoconferencias, sistemas de vigilancia, de control de acceso (N/A).

2.1.2. Visión humana y artificial

La visión es un sentido imprescindible para el hombre. La vista en su interior presenta más de dos millones de terminaciones nerviosas. La radiación exterior captada por el ojo es convertida en señales que son procesadas por el cerebro. La interfaz entre el cerebro y la visión del ser humano puede procesar de 10 a 12 imágenes separadas por segundo, percibiéndolas individualmente, si se excede este número la percibirá como movimiento (N/A).

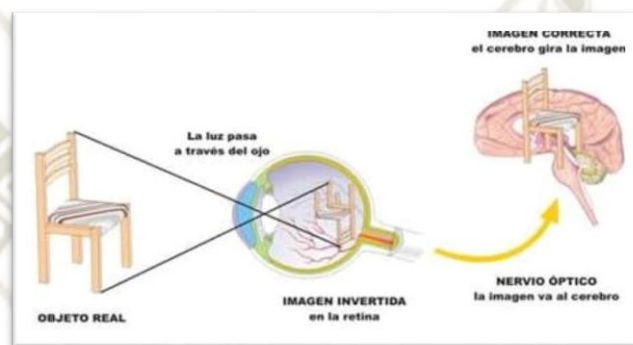


Figura 2. Representación del Ojo mirando

Fuente: ITE

La visión no es tan precisa como para determinar las diferencias que hay entre frecuencias de más de 60 Hz, aunque sí podemos distinguir, con claridad, la disparidad entre fuentes a 30 Hz y fuentes a 60 Hz. A partir de los 60 Hz, la visión humana tiene dificultades para detectar las variaciones en los FPS (fotogramas por segundo o cuadros por segundo de imágenes) que procesa, aunque algunas personas con un ojo entrenado sean capaces de distinguir diferencias muy sutiles a frames por segundo más altos. Es probable entonces que los monitores o fuentes que nos ofrezcan altas tasas de refresco no sean claramente perceptibles, pero sí puedan incrementar la percepción del movimiento al ofrecer una imagen que se aprecia más natural y sufrir una menor latencia, (un retraso de la imagen mucho menor), por lo que esta llegará antes a nuestros ojos, haciendo que no percibamos retrasos en las imágenes en movimiento (Arias, 2017).

Se puede definir la “Visión Artificial” como un campo de la “Inteligencia Artificial” que, mediante la utilización de las técnicas adecuadas, permite la obtención, procesamiento y análisis de cualquier tipo de información especial obtenida a través de imágenes digitales. La visión artificial la componen un conjunto de procesos destinados a realizar el análisis de imágenes. Estos procesos son: captación de imágenes, memorización de la información, procesamiento e interpretación de los resultados (Guevara, 2006).

2.1.3. Aplicaciones de la visión artificial.

Actualmente, las aplicaciones de la visión artificial están muy extendidas y van desde el campo de la industria (contar botellas, comprobar defectos en una cadena de montaje, interpretar un TAC médico, etc.) y el campo de la medicina (recuento y búsqueda de células), hasta los sistemas más complejos, que permiten a los robots orientarse en un entorno desconocido, pasando por el reconocimiento de patrones de la realidad aumentada, entre otras muchas aplicaciones (Maduell, 2012).

2.1.3.1. Visión artificial en la automatización de procesos

En los procesos industriales la visión artificial es una parte indispensable que expresa características de adaptabilidad, flexibilidad y capacidad de reorganización.

La visión artificial es importante para sistemas de manipulación automática y en sistemas de manufactura, pues contribuye a la retroalimentación sensorial fina que incrementa la versatilidad.

Otorga la descripción del estado que guardan los elementos del puesto de trabajo, así como sus cambios en el tiempo, información que se utiliza en la generación y cambio de su planificación de trabajo, en el monitoreo durante el desarrollo de tareas y en la detección de errores e imprevistos.

En la manipulación controlada sensorialmente permite resolver problemas de conocimiento a priori del ambiente, precisión, costo y fiabilidad y también contribuye a que los robots industriales evolucionen en distintos ambientes de trabajo.

En este contexto, un Sistema de VA debe realizar las siguientes funciones:

- Reconocer piezas y conjuntos, así como sus posiciones de equilibrio.
- Determinar la posición y orientación de piezas con respecto a una base referencial.
- Extraer y ubicar rasgos importantes de las piezas, con objeto de establecer servomecanismos visuales que contribuyan a la manipulación robotizada.
- Control de calidad de un proceso sin contacto (Sobrado, 2003).

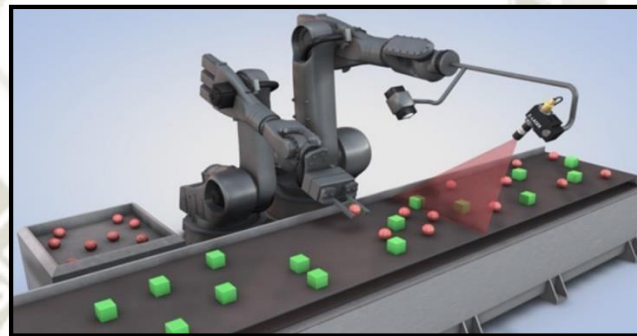


Figura 3. Robots con aplicación de Visión Artificial

Fuente: Z-laser

2.1.4. Procesamiento de imágenes en Visión Artificial.

Una imagen es una representación visual de un objeto iluminado por una fuente radiante. La naturaleza básica de una imagen se caracteriza por dos aspectos: la cantidad de luz incidente que procede de la fuente de la escena contemplada, y la cantidad de luz reflejada por los objetos de la escena. Dichas componentes reciben el nombre de iluminación y reluctancia.

Para la formación de una imagen intervienen los siguientes elementos: el objeto, la fuente radiante y el sistema de formación de la imagen (sistema óptico, un sensor y un digitalizador). Tenemos que tener en cuenta las dos principales causas que producen pérdida de información cuando se captura una imagen: la naturaleza discreta de los píxeles de la imagen y el rango limitado de valores de intensidad luminosa que es capaz de ofrecer el sistema de digitalización (N/A).

2.1.5. Elementos de los sistemas de visión artificial

- Sistema de iluminación (fluorescente, LED, polarizada, backligh, láser).
- El sensor o cámara de captura de imagen (CMOS), CCD, InGaAs).
- Tarjeta de captura de elevado ancho de banda (Firewire, Camera LINK, GbEthernet).
- Procesamiento de imagen (Procesador y algoritmos de pretratamiento y filtrado de la imagen, de segmentación, reconocimiento de formas, de extracción de descriptores y de clasificación). Sincronía con el proceso, para adquirir la imagen en el momento adecuado y para actuar con el proceso o separar unidades defectuosas (Díaz et al, 2004).

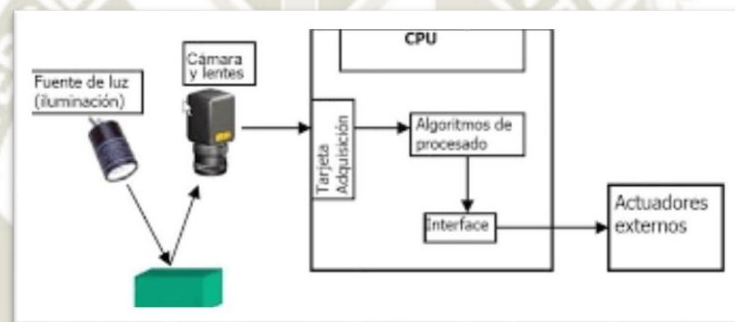


Figura 4. Diagrama de un sistema de Visión Artificial

Fuente: Duque, 2009

2.1.5.1. Cámara de captura de imagen

Transforma las señales luminosas de la escena, en señales analógicas transmitidas para ser transportadas por un cable coaxial. Presenta dos partes: el sensor, que captura las propiedades del objeto en forma de señales luminosas y lo transforma en señales analógicas; y la óptica que se encarga de proyectar los elementos adecuados de la escena ajustando una distancia focal adecuada para la captura de imagen.

Los sensores de visión usados se basan en matrices de dispositivos acoplados por carga CCD; estos transductores dan una señal con amplitud proporcional a la luminosidad de la escena y digitalizan espacialmente en dos dimensiones (líneas y columnas), pues descomponen la imagen en una matriz de puntos. Una cámara web común puede ser utilizada para la etapa de captura de imagen de los objetos a ser analizados y clasificados (Sobrado, 2003).

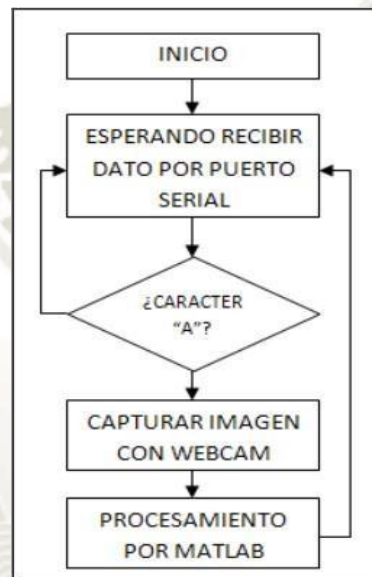


Figura 5. Diagrama de Flujo de Captura de Imagen

Fuente: Porras, s.f.

2.1.5.2. Iluminación artificial.

La iluminación de la escena es muy importante en el desarrollo de un sistema visual. Antes de intentar corregir un problema de iluminación por medio de algoritmos muy complicados, es mejor prestar atención e implantar un sistema de iluminación que permita la captura de la imagen de manera correcta (Sobrado, 2003).

Es mejor un buen sistema de iluminación, que intentar corregir ese problema por software, pues la velocidad de procesamiento será mayor con algoritmos más sencillos (Sobrado, 2003).

La fuente de luz es un aspecto fundamental en los sistemas de visión artificial y afecta la complejidad de los algoritmos de visión que se vayan a utilizar bajo estas condiciones. Una iluminación arbitraria del entorno no suele ser aceptable ya que probablemente se obtengan imágenes con bajo contraste, reflexiones especulares, además de sombras y detalles falsos. Los objetivos de una iluminación adecuada, son básicamente independizar las condiciones del entorno en la escena, y resaltar los rasgos de interés del o de los objetos bajo estudio (Cárdenas et al, 2004).

2.1.5.2.1. Técnicas de iluminación.

a. Retroiluminación Difusa.

Es adecuada en casos en casos que solo sea necesario el contorno de la pieza para su reconocimiento o medida y exista la posibilidad de apoyar dicha pieza sobre una superficie transparente.

Consiste en iluminar contra la cámara, dejando el objeto entre la cámara y la lámpara. Este método permite un alto contraste entre la pieza y el fondo, resultando fácilmente segmentable mediante una binarización, aunque se pierden los detalles de la superficie del objeto (Sobrado, 2003).

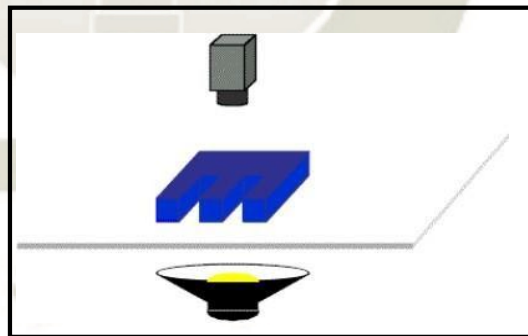


Figura 6. Iluminación Difusa

Fuente: Sobrado, 2003

b. Iluminación Frontal.

Consiste en iluminar frontalmente la pieza. Este tipo de iluminación suele presentar brillos y sombras que afectan el contraste entre la pieza y el fondo, alterando las propiedades de la pieza de estudio. Se emplea en

piezas poco reflectoras para reducir los brillos molestos, empleándose una iluminación difusa muy estudiada para piezas muy reflectoras (Sobrado, 2003).

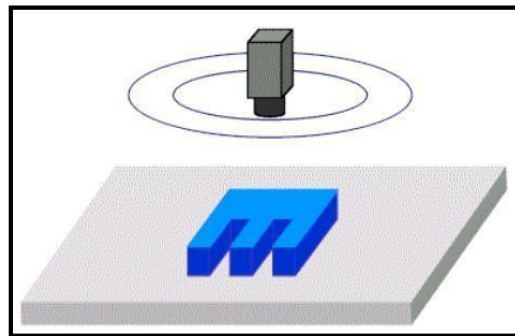


Figura 7. Iluminación Frontal

Fuente: Sobrado, 2003

c. Luz Direccional.

Se ilumina de manera direccionada en algún sentido en el espacio para destacar una característica específica del objeto. Permite la creación de sombras sobre el objeto, lo cual ayuda a aumentar el contraste de partes tridimensionales y obtener la consiguiente información 3D del elemento a analizar. De manera opuesta, al emplear una iluminación multidireccional, se reducirá la sombra del objeto (Sobrado, 2003).

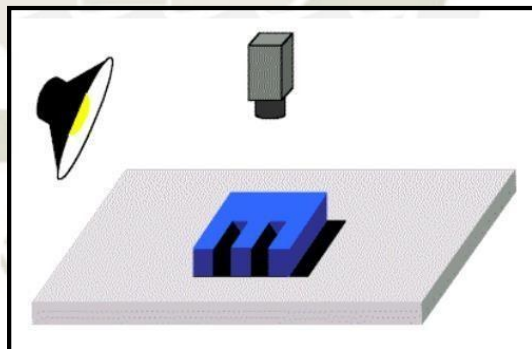


Figura 8. Iluminación Direccional

Fuente: Sobrado, 2003

d. Luz Estructurada.

Permite proyectar sobre la pieza patrones de luz modulados y observando la luz reflejada, la cual también es modulada, obtener información sobre la estructura de la superficie del objeto, la cual puede ser reconstruida

mediante triangulación. Las fuentes de luz utilizadas son especiales, pues deben de emitir luz estructurada y suelen ser láseres. Se usa para reconstrucciones 3D de objetos y para conocer su forma (Sobrado, 2003).

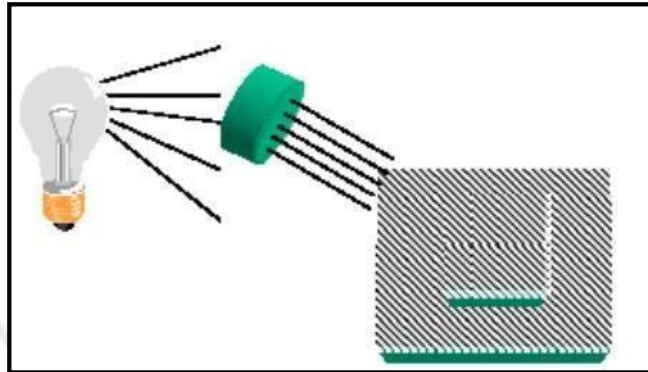


Figura 9. Iluminación estructurada

Fuente: Sobrado, 2003

2.1.5.2.2. Fuentes de iluminación artificial

Por otro lado, debemos prestar atención también a las diferentes fuentes de iluminación:

a. Lámparas incandescentes.

Es la fuente de iluminación más común y consiste en un filamento de tungsteno o halógeno-tungsteno. Como ventaja tiene que existe gran variedad de potencias y como desventaja, que reduce su luminosidad con el tiempo, lo que puede provocar problemas en algunos sistemas de visión.

b. Tubos fluorescentes.

Más eficaces que las lámparas y suministran una luz más difusa, que es bueno para piezas muy reflectoras.

c. Fibra óptica.

Para iluminar zonas de difícil acceso o extremadamente pequeñas. La iluminación es constante.

d. Láser.

Empleados para una iluminación con luz estructurada. Tiene el inconveniente de presentar un mal comportamiento frente a superficies que absorben luz.

e. Leds

La tecnología conocida como LED (por sus siglas en inglés, Light Emitting Diode, que en español significa Diodo Emisor de Luz) representa a un material semiconductor que es capaz de emitir una radiación electromagnética en forma de Luz (Sobrado, 2003).

2.1.5.3. Controlador

Un controlador es un dispositivo electrónico que permite la interacción entre el sistema operativo de una computadora con un hardware. El controlador permite una acción de interfaz que facilita el uso del hardware.

El Arduino Mega es una tarjeta de desarrollo de código libre (Open Source) que utiliza el microcontrolador ATmega 2560, posee 54 pines digitales que funcionan como entrada/salida; 16 entradas análogas, un cristal oscilador de 16 MHz, una conexión USB, un botón de reset y una entrada para la alimentación de la placa. La comunicación entre la computadora y Arduino se produce a través de un puerto serie, sin embargo, posee un convertidor USB-SERIE, por lo que sólo se necesita conectar el dispositivo a la computadora utilizando un cable USB (García, 2013).

2.1.5.4. Sensores y actuadores.

Sensor ultrasónico HC-SR04.

Los sensores ultrasónicos son muy utilizados cuando se requiere una medición no invasiva para determinar la distancia del emisor a un objeto. Entre su rango de aplicaciones esta la medición de distancias, profundidades y alturas que cambian dinámicamente, detección de objetos, etcétera. Las principales ventajas de este tipo de sensado es que, al ser una medición no invasiva, la variedad de objetos que es posible medir es muy amplia, así como medición en superficies variadas, líquidos y en ambientes hostiles. La señal ultrasónica se

puede generar mediante diferentes técnicas, como electromagnéticas, ópticas, capacitivas y piezoeléctricas; de todas, esta última es la más utilizada (Corona Ramírez, Abarca Jiménez & Mares Carreño, 2014).

Una onda sonora se puede considerar ultrasónica cuando está por encima de la frecuencia audible para el oído humano (aproximadamente 20 KHz). El sensor ultrasónico HC-SR04 puede medir la distancia a la cual se encuentra algún objeto, contando el tiempo que se demora una señal ultrasónica de 40 kHz del transmisor en rebotar sobre el objeto y retornar al receptor. Con base en esta información y considerando la velocidad del sonido en el aire, se puede obtener con mucha precisión la distancia. La descripción de los pines del sensor es la siguiente:

- VCC. Voltaje de alimentación de 5v
- Trig. Señal de entrada que habilita una medición del sensor.
- Echo. Señal de salida, emite un pulso con una duración correspondiente al tiempo que le toma a la señal salir del TX y llegar al RX (Corona Ramírez, Abarca Jiménez & Mares Carreño, 2014).

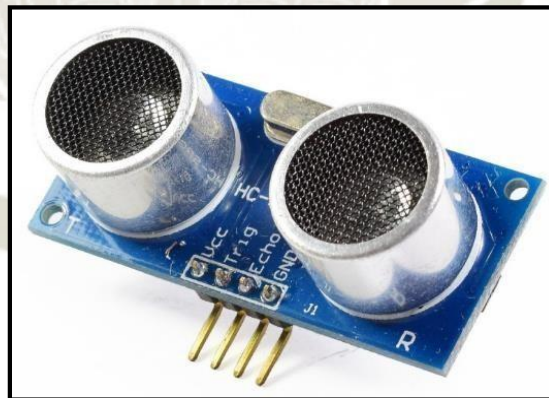


Figura 10. Sensor ultrasónico HC-SR04

Fuente: Naylamp mechatronics

Servomotor.

Un servomotor es un tipo especial de motor que permite controlar la posición del eje en un momento dado. Al hablar de un servomotor se hace referencia a un sistema compuesto por componentes electromecánicos y electrónicos. Es

capaz de moverse determinada cantidad de grados y luego mantenerse fijo en una posición establecida.

El motor en el interior es un motor DC común, y su eje se acopla a una caja de engranajes similar a una transmisión. Esto se realiza para potenciar el torque del motor y permitir mantener una posición fija cuando se requiera. La presencia del sistema de engranajes permite una inercia superior a la de un motor común y corriente. El circuito electrónico es el encargado de controlar el movimiento y la posición del motor.

A mayor consumo de corriente, mayor demanda de potencia y por extensión, mayor consumo de energía, esto es importante considerar a la hora de controlar este tipo de motores con un microcontrolador. Normalmente el consumo de corriente de un servomotor casi siempre es superior a la capacidad máxima del microcontrolador, siendo necesario una alimentación con fuentes externas. Los servomotores de modelismo operan a voltajes bajos en corriente directa, típicamente entre 4 y 6 voltios. Los servomotores industriales operan tanto en DC como en AC (monofásico o trifásico). Los servos de modelismo, en cambio, se pueden adquirir a muy bajo costo y son populares entre los usuarios de Arduino (García, 2016).

2.1.6. Software de Visión artificial

La historia de la visión artificial ha estado ligada siempre al desarrollo de la informática; este desarrollo se ha ido acentuando en los últimos años y con ello ha aparecido una notable mejora de la visión artificial. En este caso nos referimos a que el software que se utiliza ha ido mejorando y permitiéndonos una gran cantidad de nuevas posibilidades; aparte y en paralelo se ha desarrollado el hardware, pero atándose más a una mejora en las nuevas tecnologías.

Los programas de visión artificial tratan las imágenes de una manera similar, las transforman en una matriz, dividiéndolas en píxeles, y les dan un valor; Este valor varía dependiendo del programa, ya que puede tratarse de un único número, que nos indicaría el tono del color del píxel, o puede trabajar con un vector, en este

caso hemos de entender el modelo de colores RGB (red-green-blue), que representa un color con una matriz de 1×3 donde cuantifica la cantidad de rojo-verde-azul con los que se compone el color (N/A).

2.1.6.1. Matlab.

MATLAB es un programa de alto desempeño que permite realizar cálculos técnicos, integra el cálculo, la visualización y la programación en un ambiente fácil de utilizar donde los problemas y las soluciones se expresan en una notación matemática. Es un sistema interactivo cuyo elemento básico de datos es el arreglo que no requiere de dimensionamiento previo. Esto facilita y permite resolver muchos problemas computacionales, específicamente aquellos que involucren vectores y matrices, en un tiempo mucho menor al requerido para escribir un programa en un lenguaje escalar no interactivo tal como C o Fortran.

En el ámbito académico y de investigación, es la herramienta estándar para los cursos introductorios y avanzados de matemáticas, ingeniería e investigación. En la industria MATLAB es la herramienta usada para el análisis, investigación y desarrollo de nuevos productos tecnológicos.

La ventaja principal de MATLAB es el uso de familias de comandos de áreas específicas llamadas toolboxes. Lo más importante para los usuarios de MATLAB es que los toolboxes le permiten aprender y aplicar la teoría.

Los toolboxes son grupos de comandos de MATLAB (archivos M) que extienden el ambiente de MATLAB para resolver problemas de áreas específicas de la ciencia e ingeniería. Por ejemplo, existen toolboxes para las áreas de Procesamiento Digital de Señales, Sistemas de Control, Redes Neuronales, Lógica Difusa, Wavelets, etc.

La ventana de comando es la ventana principal, con la cual el usuario interactúa con MATLAB. Es la primera ventana que se abre al ejecutar MATLAB. Se utiliza para correr los comandos, correr el editor de archivos M (MEDIT, presente en la barra de herramientas), ejecutar los toolboxes, etc. (Esqueda, 2002).

2.1.6.1.1. Interfaz gráfica GUI.

Programa Informático que actúa de interfaz de usuario (GUI), utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz. Su principal uso, consiste en proporcionar al usuario un entorno visual sencillo y de fácil acceso que le permita la comunicación con el sistema operativo de una máquina o computador.

Las acciones suelen realizarse mediante manipulación directa, para facilitar la interacción del usuario con la computadora o equipo. Es resultado de la evolución de las interfaces de línea de comandos que se usaban para operar los primeros sistemas operativos y es un elemento fundamental en cualquier entorno gráfico.

En el contexto del proceso de interacción persona-ordenador, la interfaz gráfica de usuario es el artefacto tecnológico de un sistema interactivo que posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático (Castellanos, 2014).

2.1.6.2. Open CV.

La librería OpenCV es una API de aproximadamente de 300 funciones escritas en lenguaje C que se caracterizan por lo siguiente:

- Su uso es libre tanto para su uso comercial como no comercial.
- No utiliza librerías numéricas externas, aunque puede hacer uso de

alguna de ellas, si están disponibles, en tiempo de ejecución.

- Es compatible con The Intel® Processing Library (IPL) y utiliza The Intel® Integrated Performance Primitives (IPP) para mejorar su rendimiento, si están disponibles en el sistema.
- Dispone de interfaces para algunos otros lenguajes y entornos: EiC - intérprete ANSI C escrito por Ed Breen. Hawk y CvEnv son entornos interactivos (escritos en MFC y TCL, respectivamente) que utilizan el intérprete EiC; Ch - intérprete ANSI C/C++ creado y soportado por la compañía SoftIntegration; Matlab® - gran entorno para el cálculo numérico y simbólico creado por Mathworks; y muchos más.

La librería OpenCV está dirigida fundamentalmente a la visión por computador en tiempo real. Entre sus muchas áreas de aplicación destacarían: interacción hombre-máquina; segmentación y reconocimiento de objetos; reconocimiento de gestos; seguimiento del movimiento; estructura del movimiento; y robots móviles (Arévalo et al., 2004).

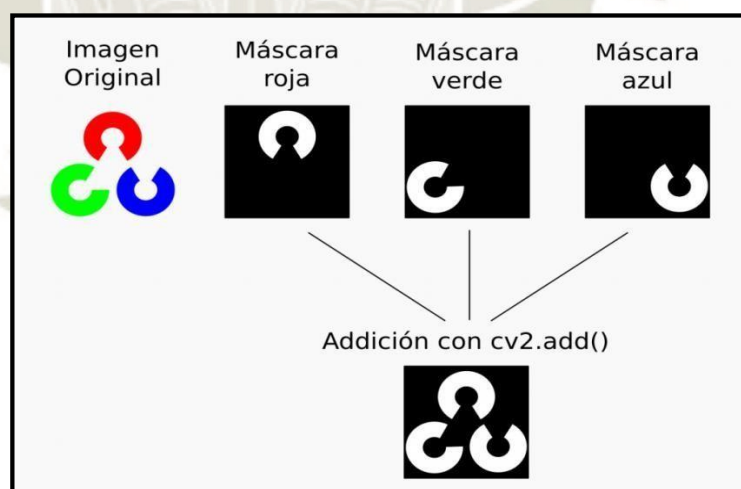


Figura 11. Detección de colores mediante OpenCV

Fuente: OpenCV

2.1.7. Etapas en el proceso de visión artificial

El primer paso en el proceso es adquirir la imagen digital. Para ello se necesitan sensores y la capacidad para digitalizar la señal producida por el sensor. Una vez que la imagen digitalizada ha sido obtenida, el siguiente paso consiste en el pre procesamiento de dicha imagen. Cuyo objetivo es mejorar la imagen de forma que el objetivo final tenga mayores posibilidades de éxito.

El paso siguiente es la segmentación, siendo su objetivo es dividir la imagen en las partes que la constituyen o los objetos que la forman. Posteriormente, es necesario convertir estos datos a una forma que sea apropiada para el ordenador. La primera decisión es saber si se va a usar la representación por frontera o región completa. La representación por la frontera es apropiada cuando el objetivo se centra en las características de la forma externa como esquinas o concavidades y convexidades. La representación por regiones es apropiada cuando la atención se centra en propiedades internas como la textura o el esqueleto. Sin embargo, en muchas aplicaciones ambas representaciones coexisten.

Finalmente, se el reconocimiento y la interpretación. El reconocimiento es el proceso que asigna una etiqueta a un objeto basada en la información que proporcionan los descriptores (clasificación). Mientras que la interpretación lleva a asignar significado al conjunto de objetos reconocidos (González et al., 2006).

Estos pasos se muestran en la siguiente imagen:

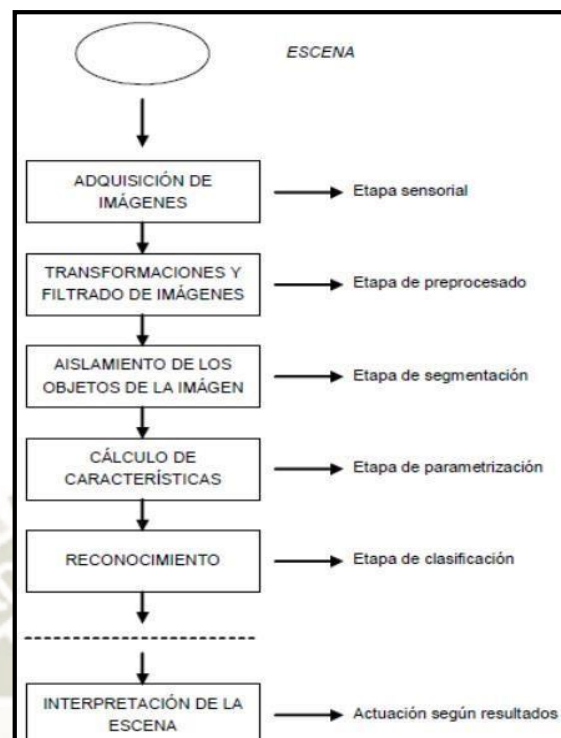


Figura 12. Diagrama de bloques de las etapas de un sistema de Visión artificial

Fuente: Gonzáles. 2006

2.1.7.1. Adquisición de imagen.

En este primer paso, se trata de conseguir que la imagen sea lo más adecuada posible para que se pueda continuar con las siguientes etapas. Una correcta adquisición de la imagen supone un paso muy importante para que el proceso de reconocimiento tenga éxito.

Dentro de esta etapa existen múltiples factores que atañen directamente al proceso de captura de la imagen, formados fundamentalmente por: el sistema hardware de visión artificial (cámara, óptica, tarjeta de adquisición de imagen, ordenador y software) y el entorno y posicionamiento de los elementos (la iluminación, el fondo, posición correcta de la cámara, ruido eléctrico-óptico externo, etc.) (Gonzáles et al., 2006).

2.1.7.2. Pre procesamiento digital de imagen.

Hace referencia al proceso inicial de una imagen la cual no ha sido trabajada, la imagen es capturada a través de los sensores y transferida hacia el computador, en aquel momento, la imagen se convierte en digital formada por dígitos que pueden ser leídos por la computadora, dichos dígitos son convertidos en pequeños puntos llamados Pixeles, donde 0 es Negro y 255 es Blanco.

Cada Pixel en Color es representado por 3 dígitos RGB (Red, Green, Blue) también en intervalos de 0 a 255 dependiente de la intensidad del color. En algunos casos la imagen capturada puede contener elementos y distorsiones no deseadas. Por tal motivo, el propósito del Pre procesamiento es mejorar la calidad de la imagen adquirida eliminando las partes indeseables o realzando las partes de mayor interés en ellas, de esta forma se aumentan las posibilidades de éxito del trabajo (Valdivia, 2016).

2.1.7.2.1. Filtros.

Se trata de métodos para resaltar o suprimir, de forma selectiva, información contenida en una imagen a diferentes escalas espaciales, para destacar algunos elementos de la imagen, o también para ocultar valores anómalos. Los filtros más utilizados son los de paso bajo (suavizan la imagen), de paso alto (aumentan el contraste), los filtros direccionales (detectan en la imagen estructuras que siguen una determinada dirección) y los de detección de bordes (permiten identificar y aislar objetos con propiedades homogéneas dentro de la imagen).

Filtros de paso bajo.

Su objetivo es suavizar la imagen, son útiles cuando se supone que la imagen tiene gran cantidad de ruido y se quiere eliminar. También pueden utilizarse para resaltar la información correspondiente a una determinada escala (tamaño de la matriz de filtrado); por ejemplo, en el caso de que se

quiera eliminar la variabilidad asociada a los tipos de cubierta presentes en la imagen uniformizando de esta manera su respuesta.

Existen varias posibilidades:

- Filtro de la media, asigna al pixel central la media de todos los pixeles incluidos en la ventana. La matriz de filtrado estaría compuesta por unos y el divisor sería el número total de elementos en la matriz.
- Filtro de media ponderada, los elementos de la matriz de filtrado no son todos, sino que se da más peso a uno de ellos (generalmente el central) para obtener un resultado más parecido a la imagen original y evitar que aparezca borrosa.
- Filtro de la mediana tiene la ventaja de que el valor final del pixel es un valor real presente en la imagen y no un promedio, de este modo se reduce el efecto borroso que tienen las imágenes que han sufrido un filtro de media. Además, el filtro de la mediana es menos sensible a valores extremos. El inconveniente es que resulta más complejo de calcular ya que hay que ordenar los diferentes valores que aparecen en los pixeles incluidos en la ventana y determinar cuál es el valor central.
- Filtros adaptativos. Son considerablemente más complejos ya que los coeficientes de ponderación se recalculan para cada uno de los pixeles en función del histograma de los ND que aparecen en la ventana.
- Filtros gaussianos. Simulan una distribución gaussiana bivalente. El valor máximo aparece en el pixel central y disminuye hacia los extremos tanto más rápido cuanto menor sea el parámetro de desviación típica s .

Filtros de paso alto.

Su objetivo es resaltar las zonas de mayor variabilidad eliminando lo que sería la componente media, precisamente la que detectan los filtros de paso bajo. Por otra parte, la respuesta de cada pixel está contaminada por la de los pixeles vecinos ya que, considerando la superficie terrestre como lambertiana, la radiación reflejada por un pixel se reparte hacia los pixeles vecinos. Los filtros de paso alto consiguen también eliminar en parte esta contaminación.

Filtros direccionales.

Se utilizan para detectar estructuras que siguen una determinada dirección en el espacio resaltando el contraste entre los píxeles situados a ambos lados de la estructura.

Filtros para la detección de bordes.

Uno de los más utilizados es el detector de bordes de Sobel que realiza la variación entre filas y columnas. (“Técnicas de filtrado”, s.f.).

2.1.7.3. Segmentación.

La segmentación permite separar una imagen en sus partes constituyentes, con el objetivo de separar el objeto de interés del resto de la imagen, por lo tanto, el nivel al que se lleva a cabo esta subdivisión depende de los requerimientos del problema. Al separar las partes constituyentes de una imagen se identifican los bordes o se separa está en regiones, líneas o curvas, etc (La Serna & Román, 2009).

La segmentación automática es una de las partes más complicadas e importantes en el procesamiento de imagen, de ser realizada correctamente la segmentación, simplificará en gran medida la solución del problema. Así mismo, una segmentación incorrecta hará que el sistema arroje un resultado incorrecto (Valdivia, 2016).

La segmentación de una imagen basado en la discontinuidad o en la similaridad de los valores del nivel de gris de sus pixels puede ser aplicada tanto a las imágenes estáticas como a las dinámicas (La Serna & Román, 2009).

Los algoritmos de segmentación de imágenes monocromáticas generalmente se basan en una de las dos propiedades básicas de valores del nivel de gris: discontinuidad y similaridad. En la discontinuidad el método consiste en extraer el objeto de interés basándose en los cambios bruscos del nivel de gris en la imagen. Los temas más importantes en la discontinuidad son: a) detección de puntos aislados, y b) detección de líneas y c) detección de bordes de una imagen. En la similaridad, se presenta la regularidad en los valores del nivel de gris, los principales métodos están basados en a) umbralización, b) crecimiento de región, y c) división y fusión de regiones (La Serna & Román, 2009).

2.1.7.3.1. Umbralización

Dada la variación de las características de luminosidad de las imágenes, no es posible determinar un umbral único para la imagen que se desea procesar. Para separar un objeto del fondo de la imagen, es necesario analizar primeramente el histograma para calcular los límites de los niveles de escalas de grises (límites de corte) a partir de los cuáles se define el objeto que está siendo analizado.

La Figura 13 muestra el histograma de un pequeño objeto sobre un fondo contrastante, el histograma nos indica que el fondo (gris oscuro) es lo que ocupa más espacio en la imagen, porque es el que contiene mayor número de pixeles en la escala de grises; luego tenemos la pieza de color claro para el caso un umbral es fácilmente identificable (Sobrado, 2003).

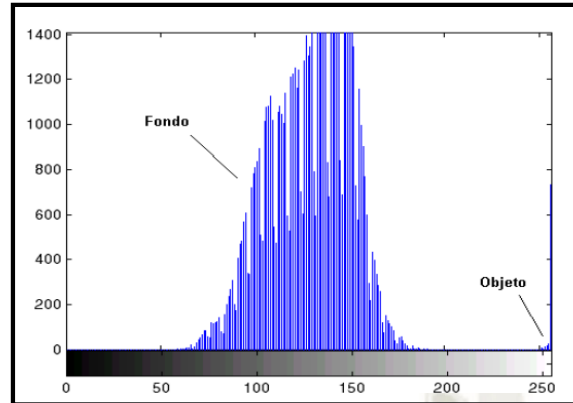


Figura 13. Histograma de imagen

Fuente: Sobrado, 2003

Posteriormente, se realiza una conversión de imagen de diferentes escalas de grises, en una imagen binaria (blanco y negro), en la cual la imagen pasa a tener todos los pixeles que son menores al límite especificado en el threshold a un valor de 0 (negro), y todos los demás a un valor de 255 (blanco) (Sobrado, 2003).

$$T_{global}(g) = \begin{cases} 0, & < t \\ 255, & \geq t \end{cases}$$

Donde:

- g: nivel de gris del pixel
- t: valor umbral preestablecido



Figura 14. Objeto segmentado por umbralización

Fuente: Sobrado, 2003

2.1.7.4. Extracción de características.

Consiste en extraer algunas medidas numéricas e información *relevante* de los datos en bruto de los patrones suministrados por los sensores (representación inicial), para la clasificación de características (de dimensión n) partiendo de los datos de entrada (Sobrado, 2003).

Este objetivo puede abordarse de dos formas:

Reduciendo la dimensionalidad de los datos: Si los patrones son de alta dimensionalidad, el coste computacional asociado a la clasificación puede ser muy alto. Como veremos, muchos clasificadores están basados en cálculos de distancias y estos cálculos pueden depender de forma cuadrática respecto a la dimensionalidad de los patrones. Como otra consideración computacional hay que considerar el espacio de almacenamiento adicional que supone guardar los valores de nuevas variables. Además, algunas de las variables pueden ser redundantes con otras y no aportar información adicional (Sobrado, 2003).

Cambiando el espacio de representación: El objetivo es obtener una nueva representación de los patrones en la que los agrupamientos aparezcan bien separados si son de diferente clase y que haya un agrupamiento por clase. Esto puede conseguirse aplicando alguna transformación sobre los datos originales. Estas transformaciones suelen ser transformaciones lineales y el objetivo suele ser maximizar la varianza (Sobrado, 2003).

2.1.7.5. Reconocimiento e interpretación.

En este procesamiento de alto nivel de reconocimiento, se le asigna una etiqueta a un objeto con base en la información que brindan sus descriptores. La interpretación les da un sentido o significado a los objetos reconocidos para finalmente llegar a clasificarlos según sus propiedades. Una técnica comúnmente usada en la etapa de clasificación se basa en clasificadores estadísticos o modelos neuronales multicapa mediante el algoritmo de retro propagación (Sabareesaan, 2013).

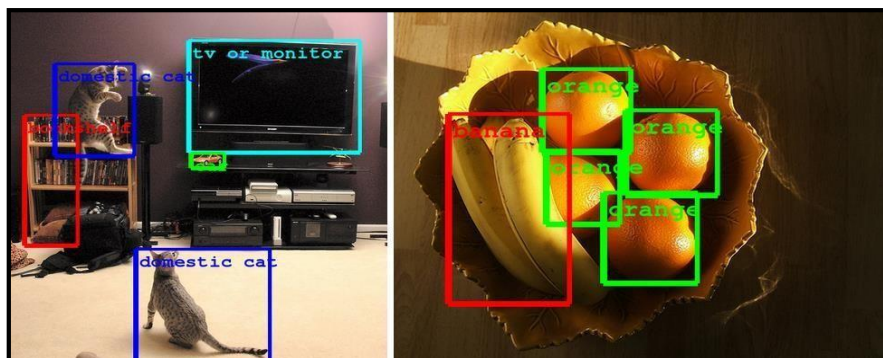


Figura 15. Reconocimiento de objetos

Fuente: Hidalgo, 2014

2.2. Red neuronal artificial.

Las redes neuronales artificiales están inspiradas en las redes neuronales biológicas, pero con unas estructuras de conexión y funcionalidades diferentes, se puede decir que las RNA (Red Neuronal Artificial) tienen tres características principales.

- Auto-Organización y Adaptabilidad: emplean algoritmos de aprendizaje adaptativo y de auto-organización para obtener mejores posibilidades de procesamiento robusto.
- Procesado no Lineal: aumenta la capacidad de la red para la clasificación de patrones, la aproximación de funciones, y el aumento de su inmunidad frente al ruido.
- Procesado Paralelo: implica el uso de un gran número de nodos de procesamiento con un alto nivel de interconectividad.

Al elemento básico se llama nodo, el cuál recibe una entrada desde la salida de otros nodos anteriores o del exterior. Cada entrada posee un peso específico asociado “w”, el valor de este aumentará o disminuirá en el proceso de aprendizaje. Cada nodo aplica una función dada f de la suma de las entradas ponderadas mediante los pesos (San Miguel, 2016).

$$y_i = \sum_j w_{ij} * y_j$$

2.2.1. Antecedentes.

-1936: Alan Turing. Fue el primero en estudiar el cerebro como una forma de ver el mundo de la computación. Sin embargo, los primeros teóricos que concibieron los fundamentos de la computación neuronal fueron Warren McCulloch, un neurofisiólogo, y Walter Pitts, un matemático, quienes, en 1943, lanzaron una teoría acerca de la forma de trabajar de las neuronas. Ellos modelaron una red neuronal simple mediante circuitos eléctricos.

-1949: Donald Hebb. Fue el primero en explicar los procesos del aprendizaje (que es el elemento básico de la inteligencia humana) desde un punto de vista psicológico, desarrollando una regla de como el aprendizaje ocurría. Aun hoy, este es el fundamento de la mayoría de las funciones de aprendizaje que pueden hallarse en una red neuronal. Su idea fue que el aprendizaje ocurría cuando ciertos cambios en una neurona eran activados. También intentó encontrar semejanzas entre el aprendizaje y la actividad nerviosa. Los trabajos de Hebb formaron las bases de la Teoría de las Redes Neuronales.

-1950: Karl Lashley. En sus series de ensayos, encontró que la información no era almacenada en forma centralizada en el cerebro, sino que era distribuida encima de él.

-1956: Congreso de Dartmouth. Este Congreso frecuentemente se menciona para indicar el nacimiento de la inteligencia artificial.

-1957: Frank Rosenblatt. Comenzó el desarrollo del Perceptron. Esta es la red neuronal más antigua; utilizándose hoy en día para aplicación como identificador de patrones.

-1960: Bernard Widroff/Marcian Hoff. Desarrollaron el modelo Adaline (ADAPtative LINear Elements). Esta fue la primera red neuronal aplicada a un problema real (filtros adaptativos para eliminar ecos en las líneas telefónicas) que se ha utilizado comercialmente durante varias décadas.

-1961: Karl Steinbeck: Die Lernmatrix. Red neuronal para simples realizaciones técnicas (memoria asociativa).

-1969: Marvin Minsky/Seymour Papert. En este año casi se produjo la “muerte abrupta” de las Redes Neuronales; ya que Minsky y Papert probaron (matemáticamente) que el Perceptron no era capaz de resolver problemas

relativamente fáciles, tales como el aprendizaje de una función no-lineal. Esto demostró que el Perceptron era muy débil, dado que las funciones no-lineales son extensamente empleadas en computación y en los problemas del mundo real.

-1974: Paul Werbos. Desarrolló la idea básica del algoritmo de aprendizaje de *propagación hacia atrás* (backpropagation); cuyo significado quedó definitivamente aclarado en 1985.

-1977: Stephen Grossberg: Teoría de Resonancia Adaptada (TRA). La Teoría de Resonancia Adaptada es una arquitectura de red que se diferencia de todas las demás previamente inventadas. La misma simula otras habilidades del cerebro: memoria a largo y corto plazo (Ruiz & Basualdo, 2001).

-1985: John Hopfield. Provocó el renacimiento de las redes neuronales con su libro: “Computación neuronal de decisiones en problemas de optimización.”

-1986: David Rumelhart/G. Hinton. Redescubrieron el algoritmo de aprendizaje de propagación hacia atrás (backpropagation). A partir de 1986, el panorama fue alentador con respecto a las investigaciones y el desarrollo de las redes neuronales.

En la actualidad, son numerosos los trabajos que se realizan y publican cada año, las aplicaciones nuevas que surgen (sobretudo en el área de control) y las empresas que lanzan al mercado productos nuevos, tanto hardware como software (sobre todo para simulación) (Ruiz & Basualdo, 2001).

2.2.2. Aplicaciones de las redes neuronales.

Las redes neuronales pueden utilizarse en un gran número y variedad de aplicaciones, tanto comerciales como militares.

Biología:

- Aprender más acerca del cerebro y otros sistemas.
- Obtención de modelos de la retina.

Empresa:

- Evaluación de probabilidad de formaciones geológicas y petrolíferas.
- Identificación de candidatos para posiciones específicas.
- Explotación de bases de datos.

- Optimización de plazas y horarios en líneas de vuelo.
- Optimización del flujo del tránsito controlando convenientemente la temporización de los semáforos.
- Reconocimiento de caracteres escritos.
- Modelado de sistemas para automatización y control.

Medio ambiente:

- Analizar tendencias y patrones.
- Previsión del tiempo.

Finanzas:

- Previsión de la evolución de los precios.
- Valoración del riesgo de los créditos.
- Identificación de falsificaciones.
- Interpretación de firmas.

Manufacturación:

- Robots automatizados y sistemas de control (visión artificial y sensores de presión, temperatura, gas, etc.).
- Control de producción en líneas de procesos.
- Inspección de la calidad.

Medicina:

- Analizadores del habla para ayudar en la audición de sordos profundos.
- Diagnóstico y tratamiento a partir de síntomas y/o de datos analíticos (electrocardiograma, encefalogramas, análisis sanguíneo, etc.).
- Monitorización en cirugías.
- Predicción de reacciones adversas en los medicamentos.
- Entendimiento de la causa de los ataques cardíacos.

Militares:

- Clasificación de las señales de radar.
- Creación de armas inteligentes.
- Optimización del uso de recursos escasos.
- Reconocimiento y seguimiento en el tiro al blanco.

La mayoría de estas aplicaciones consisten en realizar un reconocimiento de patrones, como: buscar un patrón en una serie de ejemplos, clasificar patrones, completar una señal a partir de valores parciales o reconstruir el patrón correcto partiendo de uno distorsionado. Sin embargo, está creciendo el uso de redes neuronales en distintos tipos de sistemas de control. Desde el punto de vista de los casos de aplicación, la ventaja de las redes neuronales reside en el procesado paralelo, adaptativo y no lineal. El dominio de aplicación de las redes neuronales también se lo puede clasificar de la siguiente forma: asociación y clasificación, regeneración de patrones, regresión y generalización, y optimización (Ruiz & Basualdo, 2001).

2.2.3. Ventajas de las redes neuronales.

Entre las razones de la amplia utilización de las redes neuronales, se pueden destacar las siguientes:

- a.** Teóricamente pueden determinar cualquier función, por lo que son adecuadas en aplicaciones que no son fácilmente descritas analíticamente.
- b.** Excepto los patrones de entrada, no es necesario suministrar información adicional.
- c.** Se pueden aplicar a cualquier tipo de patrones y a cualquier tipo de datos.
- d.** Se obtienen buenos resultados con datos ruidosos, como los encontrados frecuentemente en meteorología.
- e.** No se hacen hipótesis acerca de la distribución estadística de las variables de entrada, es decir los modelos neuronales no necesitan un conocimiento a priori de la distribución estadística de los datos y sus clases
- f.** Los modelos neuronales no necesitan un conocimiento a priori de los parámetros internos del clasificador.
- g.** Después de entrenadas son extremadamente rápidas y fácilmente implementarle en arquitecturas paralelas (Ruiz & Basualdo, 2001).

2.2.4. Arquitectura de una RNA.

La distribución de neuronas dentro de la red se realiza formando niveles o capas, con un número determinado de dichas neuronas en cada una de ellas. A partir de su situación dentro de la red, se pueden distinguir tres tipos de capas:

- De entrada: es la capa que recibe directamente la información proveniente de las fuentes externas de la red.
- Ocultas: son internas a la red y no tienen contacto directo con el entorno exterior. El número de niveles ocultos puede estar entre cero y un número elevado. Las neuronas de las capas ocultas pueden estar interconectadas de distintas maneras, lo que determina, junto con su número, las distintas topologías de redes neuronales.
- De salidas: transfieren información de la red hacia el exterior.

Además, las capas se organizan para formar la estructura de la red. Como se puede observar en la figura 16 las neuronas se agrupan para formar capas y las capas se unen entre ellas formando redes neuronales. Para clasificar según su topología usaremos el número de capas, pudiendo clasificar en 2 grupos: redes monocapa y redes multicapas (Ruiz & Basualdo, 2001).

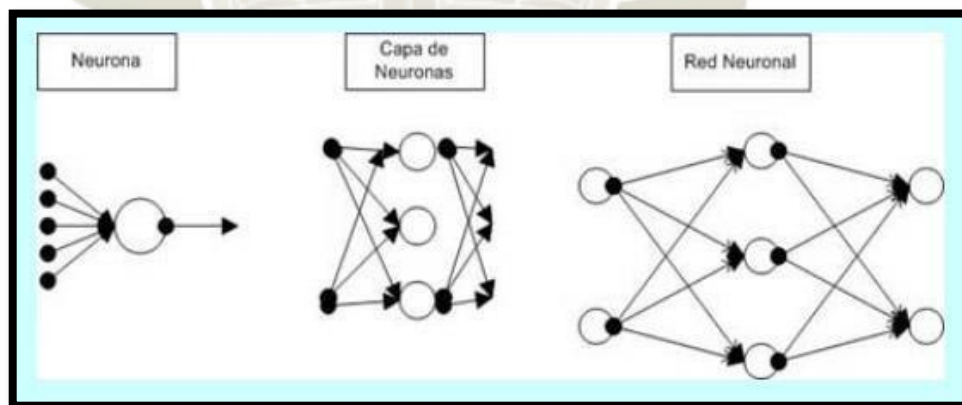


Figura 16. Jerarquía de las redes neuronales

Fuente: Aguilar, 2008

2.2.4.1. Red neuronal Monocapa

Las redes monocapa son redes que cuentan con una sola capa. Las neuronas crean conexiones laterales para conectarse con otras neuronas de su capa. Las redes más representativas son la red de Hopfield, la red BRAIN-STATE-IN-A-BOX o memoria asociativa y las maquinas estocásticas de Botzmann y Cauchy.

La conectividad entre los nodos de una red neuronal está relacionada con la forma en que las salidas están conectadas a las entradas de otras neuronas. Si la salida es la entrada de la misma, es llamada auto-recurrente.

Las redes mono capa han sido ampliamente utilizada en circuitos eléctricos ya que su topología facilita su implementacion mediante hardware, empleando matrices de diodos que representan las conexiones de las neuronas (Ruiz & Basualdo, 2001).

2.2.4.2. Red neuronal multicapa

Es una red neuronal formada por varias capas; actúa como un aproximador universal de funciones mediante el algoritmo de aprendizaje backpropagation, obligando a la red neuronal a contener al menos una capa oculta, con suficientes unidades no lineales. Es capaz de aproximar cualquier tipo de función o relación continua entre un grupo de variables de entrada y salida (San Miguel, 2016).

Por eso se considera a las redes neuronales tipo perceptrón multicapa una herramienta de propósito general, no lineal y flexible. (Rumelhart et al.,1986) presentaron un algoritmo de aprendizaje conocido como "Regla Delta Generalizada" el que permite a las redes perceptrón multicapa aprender la relación existente entre un conjunto de entrada y sus respectivas salidas (San Miguel, 2016).

El también conocido como algoritmo "backpropagation error" (propagación del error hacia atrás) tiene la capacidad de dotar a la red de generalización de tal modo que el sistema obtiene una salida correcta para un conjunto de datos de entrada que no se habían usado antes (San Miguel, 2016).

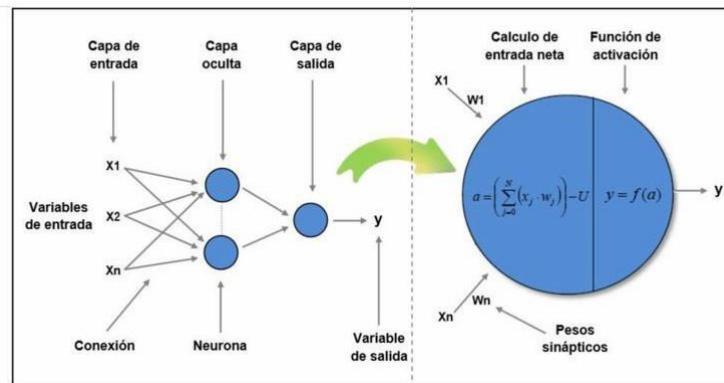


Figura 17. Perceptron Multicapa

Fuente: Barrios, 2016

2.2.5. Elementos básicos de una red neuronal

2.2.5.1. Funciones de entrada.

La neurona trata los múltiples valores de entrada como si fueran si se tratase de uno solo; esto se conoce con el nombre de *entrada global*. Para lograr esto se debe combinar las múltiples entradas simples (in_{i1}, in_{i2}, \dots) dentro de la entrada global. Esto se logra empleando la *función de entrada*, la cual se calcula a partir del *vector entrada*. La función de entrada puede representarse como la sigue ecuación:

$$input_i = (in_{i1} \cdot w_{i1}) * (in_{i2} \cdot w_{i2}) * \dots ((in_{in} \cdot w_{in})$$

dónde:

- *: Representa al operador apropiado (por ejemplo: máximo, sumatoria, productoria, etc.),
- n : Representa al número de entradas a la neurona N_i
- w_i : Representa al peso.

Los valores de entrada se multiplican por los pesos anteriormente ingresados a la neurona, estos pesos al no estar restringidos cambian la medida de influencia que tienen los valores de entrada. Es decir, que permiten que un gran valor de entrada no posea una gran influencia si el peso es pequeño (Ruiz & Basualdo, 2001).

2.2.5.2 Funciones de activación.

La función de activación combina el potencial postsináptico, proporcionado por la función de propagación, con el estado actual de la neurona para conseguir su estado futuro de activación. Es muy común que las redes neuronales no tomen su propio estado como un parámetro a tener en cuenta y que por lo tanto este no se considere. Esta función es normalmente creciente monótono y podemos citar las funciones más comunes:

Lineal:

Algunas redes neuronales usan esta función de activación como el Adeline por su eficiencia y facilidad.

Escalón:

La función escalón es la más usada para redes neuronales binarias ya que no es lineal y es muy simple. También es utilizada el perceptrón y Hopfield. Para redes que trabajan en el rango $[-1,1]$ se usa la función tansig.

Hiperbólicas o tangenciales:

Las redes con salidas continuas, como el Perceptron multicapa con retropropagación, usan esta función ya que su algoritmo de aprendizaje necesita una función derivable (Ruiz & Basualdo, 2001).

2.2.5.3. Funciones de salida.

El valor resultante de esta función es la salida de la neurona. La función de salida determina que valor se transfiere a las neuronas vinculadas. En el caso que el valor de la función de activación sea inferior del valor umbral determinado, ninguna salida se pasa a la neurona subsiguiente. Normalmente solo un rango de valores está permitido como una entrada para una neurona, los valores de salida están comprendidos en el rango $[0, 1]$ o $[-1, 1]$ (Ruiz & Basualdo, 2001).

2.2.6. Algoritmo de aprendizaje.

En cada tipo de red existe al menos un algoritmo de aprendizaje, el cual consiste en un método para encontrar un valor adecuado de los pesos.

En términos generales, se basan en definir una función objetivo implícito o explícitamente, que represente de forma global el estado de la red. A partir de ella, los pesos asignados inicialmente van evolucionando a unos valores que lleven a dicha función a un mínimo (estado estable de la red). El aprendizaje, por tanto, es algo característico del tipo de red (Romero & Calonge).

2.2.6.1. Aprendizaje supervisado.

Estos algoritmos requieren el emparejamiento de cada valor de entrada con su correspondiente valor de salida, es decir es necesario conocer el valor de salida para cada grupo de entradas que ingresan a la red. El entrenamiento consiste en presentar un vector de entrada a la red, calcular la salida de la red, compararla con la salida deseada para la obtención del error o diferencia el cual se utilizará para realimentar la red y cambiar los pesos de acuerdo con un algoritmo implementado para minimizar el error. Las parejas de vectores del conjunto de entrenamiento se aplican secuencialmente y de forma cíclica. Este proceso de calcular el error y ajustar los pesos para cada pareja se realiza hasta que el error para el conjunto de entrenamiento sea un valor pequeño y aceptable (Basogain, 2008).

2.2.6.2. Aprendizaje no supervisado

Los sistemas neuronales con entrenamiento supervisado han tenido buenos resultados en múltiples aplicaciones, pero suelen ser vistos del punto de vista biológico como no muy lógicos. Resulta difícil creer que el cerebro compare las salidas deseadas con las salidas reales y como el cerebro definiría cual es la salida deseada. Los sistemas no supervisados son modelos de aprendizaje más lógicos del punto de vista biológicos. Estos sistemas de aprendizaje no supervisado no requieren de un vector de salidas deseadas y por tanto no se

realizan comparaciones entre las salidas reales y salidas esperadas a diferencia del aprendizaje supervisado, por lo tanto, el conjunto de vectores de entrenamiento consiste únicamente de vectores de entrada. El algoritmo de entrenamiento modifica los pesos de la red de forma que produzca vectores de salida consistentes. En el proceso de entrenamiento se extraen las propiedades estadísticas del conjunto de vectores obtenidos después del entrenamiento y se agrupan en clases los vectores similares (Basogain, 2008).

2.2.7. Backpropagation.

El resurgimiento de las redes neuronales artificiales se debe en gran medida a la invención del algoritmo backpropagation. Es un método de entrenamiento de redes multicapa en el cual su potencia reside en su capacidad de entrenar capas ocultas y de este modo no verse restringida por las limitaciones de las redes de una única capa.

La unidad procesadora se caracteriza por realizar una suma ponderada de las entradas llamada S_j , presentar una salida a_j y tener un valor δ_j asociado que se utilizará en el proceso de ajuste de los pesos. El peso asociado a la conexión desde la unidad i a la unidad j se representa por w_{ji} , y es modificado durante el proceso de aprendizaje.

Normalmente, la backpropagation utiliza tres o más capas de unidades procesadoras. La capa inferior es la capa de entrada, es la única capa cuyas unidades procesadoras reciben entradas desde el exterior, cumplen la función de distribuidores, mas no realizan ninguna operación de cálculo. Las unidades procesadoras de las demás capas posteriores se encargan del procesamiento de las señales. La capa oculta cuenta con unidades procesadoras, las cuales 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 (Basogain, 2008).

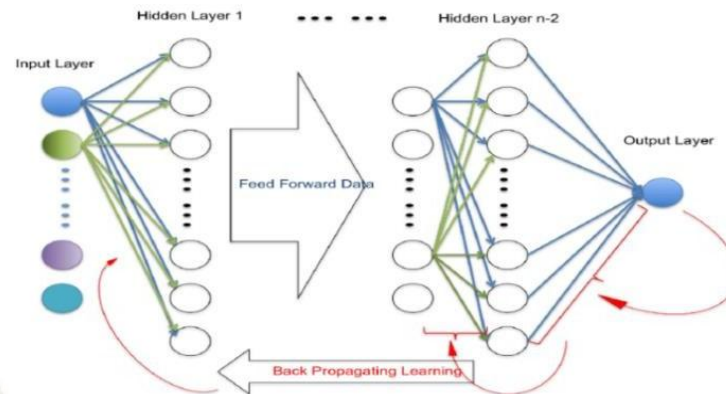


Figura 18. Red neuronal feed-forward con backpropagation

Fuente: Young, E., & Joseph, H., 2010

2.3. La Lúcumá

2.3.1. Generalidades

La lúcumá (*Pouterialucuma*) es un árbol frutal perenne que pertenece a la familia Sapotaceae nativo de los países de Perú, Ecuador y Chile (Muñoz, 1987; Jordan, 1996). En nuestro país, crece en la sierra baja, aunque puede crecer bien desde el en zonas con una altitud que no supere los 3000 msnm, que cuente con temperaturas comprendidas entre 8 y 27 grados Celsius y una humedad relativa entre 80 - 90 por ciento (MINAGRI, 2015).

La lúcumá posee forma esférica u ovalada con un ápice en punta, posee cáscara delgada de color verde o amarillo. El mesocarpio es de sabor y aroma muy agradable, color amarillo intenso, pulpa de textura harinosa, de consistencia suave en el cultivar "lúcumá seda" y dura en el cultivar "lúcumá palo". El tamaño del fruto varía de 2 a 10 cm de diámetro y un peso aproximado de 50 a 200 gramos (Navarrete & Odar, 2015).

En estos últimos años la lúcumá ha generado expectativas en los inversionistas, como consecuencia de la creciente demanda de productos naturales y exóticos en el mercado exterior. Por otra parte, la tendencia del mercado mundial de productos industrializados, se orienta a alimentos funcionales que presenten propiedades antioxidantes y anti-cancerígenas. Algunas investigaciones de mercado y pruebas de sabor dan cuenta de la creciente aceptación de la lúcumá en mercados

extranjeros tales como el europeo, norteamericano y asiático. Gracias a su agradable sabor, aroma exótico y suave textura la lúcuma se presenta como una buena alternativa para el consumidor nacional e internacional en sus diversas presentaciones (Navarrete & Odar, 2015).

Las variedades comerciales más comunes son ‘Seda’ y ‘Palo’. La primera es preferida para el consumo directo, mientras que la segunda se emplea para la producción de harina (Muñoz, 1987; MINAGRI, 2015). El color de la cáscara cambia de verde a amarillo y el color de la pulpa cambia de verde o amarillo a amarillo-anaranjado durante la maduración (Yahia et al., 2011).

Castro (1999) señala que, en estado de madurez, el color de la cáscara varía perceptiblemente de verde oscuro a verde amarillo o amarillo, sin embargo, estos cambios de color (especialmente de la cascara) pueden no ser percibidos fácilmente. Un indicador más preciso del estado de madurez es la aparición del color amarillo bajo los sépalos. Lizana (1986) citado por Castro (1999) indica que contenido de sólidos solubles y el ablandamiento de la pulpa pueden ser utilizados como índices de referencia de madurez del fruto. Se cuenta con muy poca información postcosecha acerca de la lúcuma y, por tanto, se necesita realizar investigaciones relacionadas a la fisiología del fruto (Yahia *et al.*, 2011).

2.3.2. Composición química de la lúcuma

La lúcuma es fuente de carbohidratos y fibra, además contiene minerales (calcio, fósforo y hierro) y vitaminas (tiamina, niacina, riboflavina y ácido ascórbico) Su contenido de humedad (entre 55-75 por ciento) es bajo comparado con otras frutas en las que generalmente es del 80 al 90 por ciento (Yahia et al., 2011).

El contenido de lípidos representa menos del 1% de su composición, por lo general superior al de la mayoría de frutas, con excepción de los frutos secos, mientras que los lípidos insaponificables constituyen entre el 0.1 al 0.2 por ciento. El contenido de proteína representa aproximadamente 2%, valor elevado al compararse con el presente en otros frutos (Erazo et al., 1999).

Respecto al contenido de fibra dietaria, se ha reportado que en la lúcuma ésta es mayormente insoluble, que varía del 28 al 32% según la variedad de lúcuma. Mientras que la fibra soluble representa de 3.9 a 4.0% según la variedad de lúcuma. Este contenido de fibra dietaria total es superior al de otras frutas tales como la papaya, plátano, chirimoya, mango, níspero y piña (Glorio *et al.*, 2008).

En los últimos años el número de estudios para ampliar el conocimiento de sus beneficios y características ha incrementado, estos son cada vez más detallados y emplean equipos más sensibles que han permitido ampliar el conocimiento de sus características y propiedades (Alba *et al.*, 2016; Dini, 2011; Erazo *et al.*, 1999).

En la Tabla N° 1, se presenta la composición proximal y de componentes minoritarios de la pulpa fresca de lúcuma.

Tabla 1

Composición Proximal, Minerales y Vitaminas en la Lúcuma

COMPOSICIÓN	VALOR
Energía (Kcal)	99
Agua (gr)	72.3
Proteínas (gr)	1.5
Grasa (gr)	0.5
Carbohidratos (gr)	25
Fibra (gr)	1.3
Ceniza (gr)	0.7
Calcio (mg)	16
Fósforo (mg)	26
Hierro (mg)	0.4
Retinol (ug)	355
Tiamina (mg)	0.01
Riboflavina (mg)	0.14
Niacina (mg)	19.6
Ac. Ascórbico (mg)	2.2

Fuente: Brack, 1999

2.3.3. Usos y aplicaciones.

Tradicionalmente la lúcuma es empleada tanto para consumo fresco como convertida en pulpa o harina para la elaboración de distintos productos. La mayor demanda nacional proviene del sector de helados, que la requiere en forma de harina y pulpa. En los últimos años, tanto la fruta fresca como la industrializada, se está utilizando también en la elaboración de mermeladas, yogures, pastas, papillas, batido de leche, tortas, torta de lúcuma, ravioles, bombones, pudines, galletas, licor de lúcuma, pastas, comidas y conservas (Navarrete & Odar, 2015).

Tiene un sabor intenso que recuerda al jarabe de arce, aunque según muchos, es muy superior, y se emplea cocida en: tortas, pasteles, helados, batidos, pudines y otros postres. Su consumo como fruta fresca es más raro por su peculiar retrogusto, aunque este es menos perceptible en los cultivares de mayor calidad. Es considerada por los peruanos fruta nacional y producto de bandera. Por su alto contenido en almidón, la pulpa se seca en ocasiones para su conservación; rinde una harina no perecedera, muy dulce y nutritiva, que concentra el hierro, betacaroteno y niacina contenidos en la fruta (Navarrete & Odar, 2015).

La lúcuma goza de gran fama debido a sus propiedades medicinales y nutricionales. Cien gramos de este fruto proporcionan 99 calorías y grandes nutrientes como fibra, fósforo, calcio, hierro, Tiamina, Niacina, Vitamina C, A y riboflavina (Salazar, 2018).

2.4. Esquema conceptual

➤ Redes neuronales.

Una red neuronal es un procesamiento distribuido masivamente paralelo que tiene una tendencia natural para almacenar conocimiento empírico y hacerlo disponible para el uso (Haykin, 1994).

➤ **Visión artificial.**

La visión artificial o comprensión de imágenes describe la deducción automática de la estructura y propiedades de un mundo tridimensional, posiblemente dinámico, bien a partir de una o varias imágenes bidimensionales de ese mundo (Nalwa, 1993).

➤ **Prototipo.**

Un prototipo es un modelo (representación, demostración o simulación) fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas (Floria, 2001).

➤ **Lúcuma.**

El lúcumo cuya denominación botánica es *Lúcumao bovata* H.B.K. es un frutal originario del área andina. En este ámbito todavía se encuentran ecotipos de lúcumo que crecen de manera silvestre en territorio peruano. Su fruta es rica en fibra, hierro y caroteno; un alimento altamente energético que se consume fresca e incluso es utilizada para la elaboración de deliciosos helados, postres, cócteles y un sin fin de presentaciones (N/A).

CAPÍTULO III

3. DISEÑO DEL SISTEMA

La finalidad de este proyecto es realizar la clasificación de lúcumas empleando visión artificial y redes neuronales, para lograr este objetivo se implementó un algoritmo en el software de Matlab en conjunto con la biblioteca libre de visión artificial OpenCV, asimismo su validación en un prototipo simple para determinar su eficiencia.

La clasificación de lúcumas se realizó empleando la norma nacional para lúcumas: CODEX STAN 305R-2011, la cual aplica para la especie *Pouteria lúcumas* (R. y P.) de la familia Sapotaceae, tipo “Seda” y realiza su clasificación por calibre y categoría. En esta tesis se abarcará la clasificación de lúcumas teniendo como principales parámetros su diámetro ecuatorial y peso.

En el presente capítulo se desarrollará el diseño del algoritmo y el prototipo. Primero se detallará el diseño e implantación del prototipo, así como la selección de hardware del sistema de visión artificial. Segundo se desarrollará el algoritmo de visión artificial teniendo en cuenta todas sus etapas. Y tercero el diseño y entrenamiento de la red neuronal.

3.1. Diseño del prototipo

Para que la clasificación se realice eficientemente se debe mantener constantes ciertas características tales como la iluminación y distancia de captura de imagen. Para cumplir estos requerimientos se elaboró un prototipo simple que cumpla las condiciones mencionadas previamente y permita validar el algoritmo desarrollado. A continuación, se detallará la selección del hardware empleado y diseño e implementación del prototipo.

3.1.1. Selección de hardware

Para la selección del hardware, se consideró sus especificaciones técnicas, facilidad de implementación y costo para su selección. Primeramente, se detallarán los requerimientos de la cámara para la captura de la imagen. Luego se procederá a la selección de la fuente de iluminación (características e intensidad), por último, se seleccionará el controlador, considerándose el número y tipo de entradas y salidas a utilizar, así como la facilidad de programación e implementación.

3.1.1.1. Selección de cámara

La cámara a utilizar deberá contar con una buena resolución y velocidad de captura para obtener imágenes nítidas, y teniendo en cuenta que la captura de imagen se realiza en movimiento deberá contar con enfoque automático a cortas distancias.

Se cuenta con dos posibles opciones, una cámara industrial y una cámara de uso doméstico. Entre las ventajas presentes en una cámara industrial se encuentra una alta resolución, corrección de iluminación, visión en 360°, alta velocidad y distintos tipos de conexión a ordenador (Gigabit Ethernet, etc.). La captura se realizará a una corta distancia, no siendo necesaria una alta resolución además que los tiempos de procesamiento de la imagen serían mayores, una corrección de iluminación es solucionado al contarse con un módulo para la captura el cual permite una iluminación uniforme, una alta velocidad no es necesaria al no necesitarse un alto volumen de clasificación de lúcumas, una conexión a una red y visión a 360° no son necesarios. Los costos de una cámara industrial son mayores comparados a los de una cámara de uso doméstico y al tratarse de un prototipo, se optó por una webcam de gama media.

Se optó por el modelo c525 HD Webcam de Logitech, la cual cuenta con una resolución HD de 1280 x 720 a 30 fps. Permitiendo imágenes nítidas y un buen contraste aun en movimiento además de poder variarse la resolución según se necesite. Cuenta con un enfoque automático desde los 7 cm para conservar la nitidez de la imagen para video y capturas en movimiento a distancias cortas y un costo accesible. Además, cuenta con un corrector de luz HD, ajustándose a las condiciones de iluminación para producir imágenes más luminosas y de mejor calidad. La ficha técnica de la cámara web se encuentra en el ANEXO 1.



Figura 19. Dispositivo de adquisición de imagen

Fuente: Logitech

3.1.1.2. Selección del sistema de iluminación

Una buena iluminación mejora el contraste de la imagen, en nuestro caso entre el fondo (faja) y el objeto a analizar (lúcuma) facilitando la segmentación y liberando a la imagen capturada de factores externos que pueden alterar la obtención de características.

Se realizaron pruebas en el módulo de captura con distintas fuentes de iluminación (bombillas, leds, etc.) y se observó mejores resultados empleando cintas led.

Tabla 2
Fuentes de iluminación e intensidad

Fuente de iluminación	Lumen
Bombilla led 12W	840 lumens
Bombilla incandescente 100W	1600 lumen
Cinta led	20lumen/led, 1200 lumen/metro

Fuente: Propia

Las cintas leds cuentan con buena intensidad de brillo, fácil instalación, la iluminación se distribuye de manera homogénea sobre el objeto reduciendo de manera significativa la sombra y mejorando el contraste.



Figura 20. Cinta de iluminación led

Fuente: Propia

En la figura 21 se realizó una prueba de captura en el módulo empleando cintas LEDS para la iluminación en una configuración multidireccional, se puede observar que no produce sombra y la iluminación es homogénea a lo largo de la lúcumo cumpliendo las necesidades requeridas.



Figura 21. Pruebas de iluminación en el módulo

Fuente: Propia

3.1.1.3. Selección de los instrumentos electrónicos y de control

Al tratarse de un prototipo simple, se optó por un controlador Arduino sobre un controlador industrial PLC al no ser necesario un controlador robusto, el cual se ve limitado en el número de entradas y salidas, manteniendo bajo el coste del proyecto. Arduino nos permite una alimentación externa, software gratuito, múltiples entradas y salidas digitales, además de ser de fácil uso e implementación y de menor costo comparado a un controlador industrial.

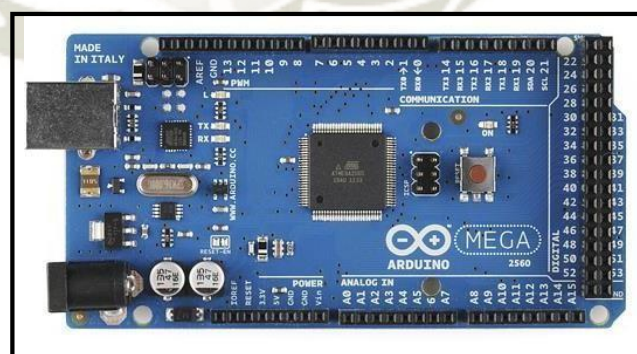


Figura 22. Arduino Mega 2560

Fuente: Arduino

Tabla 3*Especificaciones del microcontrolador Arduino Mega 2560*

Especificaciones	Arduino mega2560
Procesador	ATmega 2560
Flash Memory	256 KB
Data Memory	8 KB
EEPROM	4 KB
Digital I/O Pins	54
PWM outputs	15
Analog outputs	16
Clock Speed	16 MHz
Serial Ports	4

Fuente: Arduino

Como actuador del sistema de clasificación se empleó un servomotor TowerPro MG995, es un servo digital de un buen par de hasta 12.00 Kg/cm y 180° de giro y engranajes metálicos lo brindan de buena resistencia y puede soportar altos esfuerzos. Como sensor de presencia, se empleó un sensor ultrasónico HC-SR04 el cual cuenta con buena precisión de +- 3mm, rango de sensado desde 2 cm hasta 450 cm, es de fácil instalación y bajo consumo de energía.

*Figura 23. Servomotor TowerPro MG995 implementado*

Fuente: Propia



Figura 24. Sensor ultrasónico HC-SR04 implementado

Fuente: Propia

3.1.2. Implementación del prototipo

Con el objetivo de recrear proceso y validar los algoritmos de visión artificial en este trabajo de investigación, se implementó un prototipo de clasificación de lúcumas el cual nos permitirá medir la eficiencia del algoritmo desarrollado. La colocación y posicionamiento inicial de la lúcuma en el prototipo no son objetivos de esta tesis.

Primeramente, se realizó un bosquejo a mano alzada del prototipo para tener una idea general de los requerimientos y características a tener en cuenta al implementar el prototipo. Para la elaboración del bosquejo se consideró los principales elementos de los que deberá contar el prototipo de clasificación. Este debe contar con una zona para la captura de imagen, un sistema que permita el traslado de la lúcuma y actuadores que permitan el retiro del fruto al final de su clasificación.

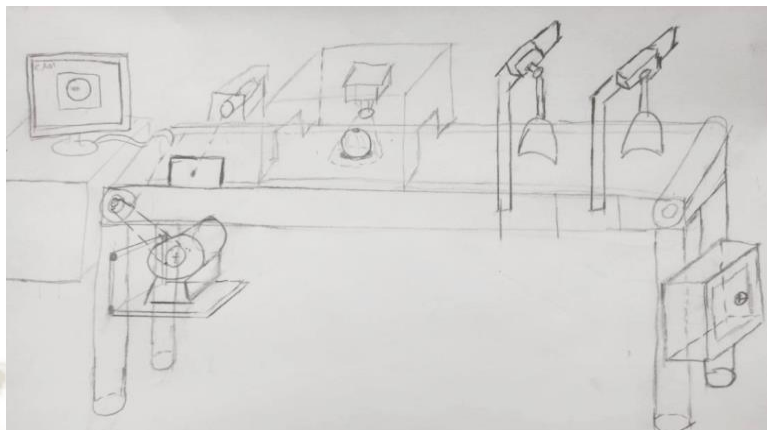


Figura 25. Bosquejo inicial del prototipo

Fuente: Propia

El prototipo de clasificación consta principalmente de 3 sistemas independientes entre sí: cinta transportadora, módulo de captura de imagen y un sistema de clasificación o separación.

La cinta transportadora se encarga del transporte de las lúcumas a través del prototipo, a una velocidad constante. La faja en la cinta transportadora se pintó de color blanco para favorecer el contraste entre la lúcumas y la faja, la cual corre sobre una plancha delgada para que el movimiento de la lúcumas en la faja sea más estable y evitar un hundimiento de la faja.

La cinta transportadora es movida por un motor monofásico, el cual cuenta con un reductor para la disminución de velocidad obteniéndose una salida a 125 RPM. El motor de 15 lb-in funciona con un arranque por condensador, para lo cual se incorporó un condensador de 240 $\mu\text{f/kW}$ en serie con el devanado auxiliar para mejorar el par y arranque del motor debido al decalado entre los campos.

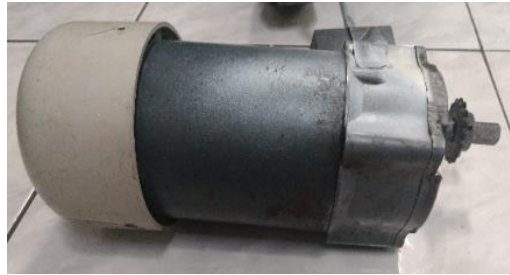


Figura 26. Motor monofásico con reductor
Fuente: Propia

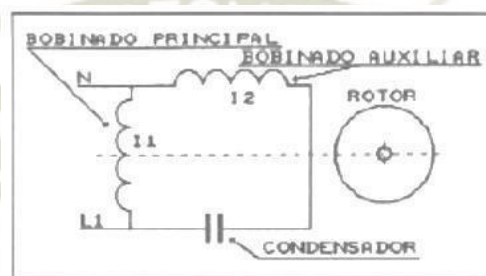


Figura 27. Arranque por condensador
Fuente: Martínez, 2001

Tensión red (V)	Condensador C	Tensión condensador U_c
220 V	~ 70 $\mu\text{F/kW}$	~ 250 V
110 V	~ 240 $\mu\text{F/kW}$	~ 125 V
380 V	~ 22 $\mu\text{F/kW}$	~ 430 V

Figura 28. Condensadores comerciales para arranque por condensador
Fuente: A. Pernía, 2011

Se empleó una transmisión por cadena desde el motor a la polea motriz de la cinta transportadora, la cual también se utilizó para la reducción de velocidad, lográndose reducir a 93.75 RPM en el eje motriz. La velocidad en la cinta transportadora está determinada en función del diámetro y RPM del eje motriz, obteniéndose una velocidad constante en la superficie de 0.20 m/s.

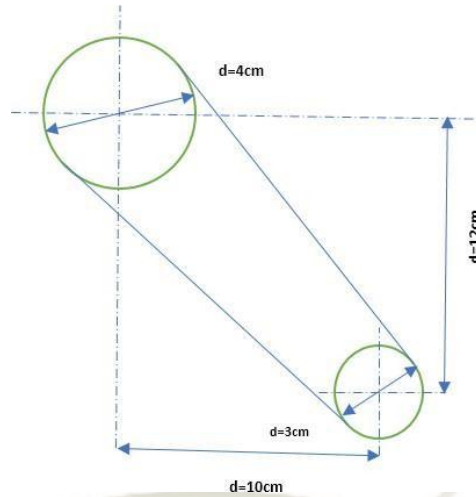


Figura 29. Relación de transmisión por cadena

Fuente: Propia

$$n_1 * d_1 = n_2 * d_2$$

$$n_2 = \frac{n_1 * d_1}{d_2} = \frac{125 \text{ RPM} * 3 \text{ cm}}{4 \text{ cm}}$$

$$n_2 = 93.75 \text{ RPM}$$

$$vp = \pi * D * \text{RPM}$$

$$v = \pi * 4 \text{ cm} * 93.75 \text{ RPM}$$

$$vp = 1178.09 \frac{\text{cm}}{\text{min}} = 0.196 \frac{\text{m}}{\text{s}} \cong 0.20 \frac{\text{m}}{\text{s}}$$

La estructura de la cinta transportadora está elaborada de acero, los rodamientos sobre los que gira las poleas se encuentran ubicados a los extremos y se encuentran fijados a la estructura.



Figura 30. Instalación de los rodamientos

Fuente: Propia

Para realizar la captura de la imagen sin verse afectada por factores externos que afecten el proceso de procesamiento de imagen, se elaboró un módulo de captura, este módulo se construyó principalmente de madera MDF rodeando la webcam, la cual se encuentra instalada en la parte superior.

El módulo de captura cuenta con tres filas de citas leds blanca en una configuración multidireccional rodeando el interior del módulo a distintas alturas para obtener una captura libre de sombras. Para garantizar una iluminación homogénea, se debe tener cuidado de que la iluminación no sea demasiado excesiva de tal manera que produzca un reflejo considerable en la superficie de la fruta.

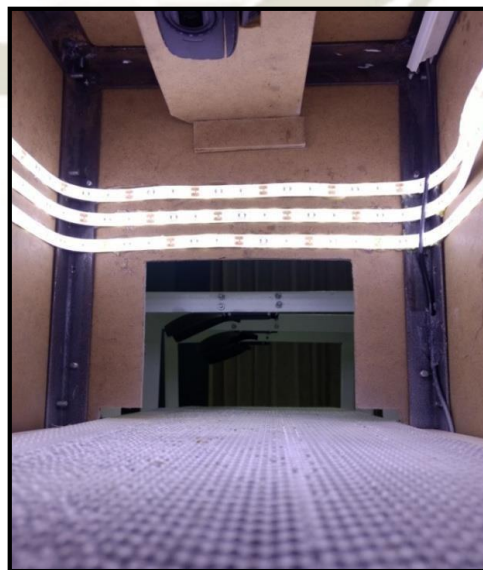


Figura 31. Distribución de iluminación dentro del módulo de captura

Fuente: Propia

Para el sistema de clasificación se tomaron en consideración los costos y facilidad de instalación, decantándose por el uso de servomotores sobre un sistema neumático. Cada uno de los servomotores está unidos a paletas que realizaran la labor de separación, están alimentados por una fuente de 5v externa y están controlados por Arduino. La figura 32 muestra el circuito de conexión del Arduino con los sensores y actuadores.

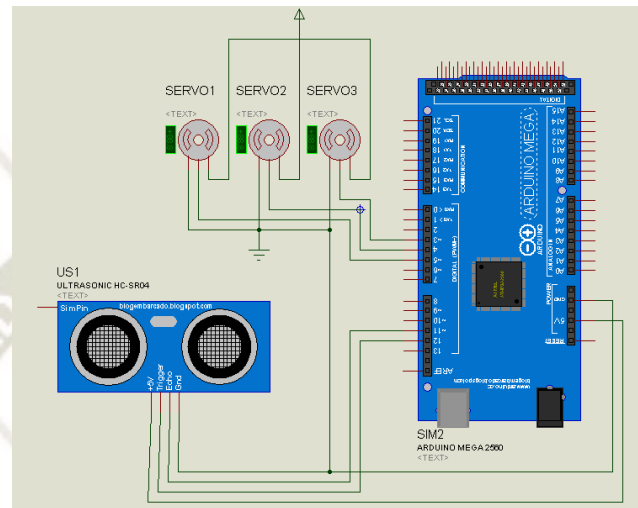


Figura 32. Circuito de conexión de Arduino

Fuente: Propia



Figura 33. Circuito impreso elaborado

Fuente: Propia

Se empleó fuentes de alimentación externa: 5v para la alimentación del Arduino, 12v para la iluminación por citas leds y un transformador de 110v-100w para alimentar el motor monofásico. Un tablero de control básico se encarga del control del motor, este cuenta con 2 botones principales:

- Botón de START: Activa el motor de la faja transportadora.
- Botón de STOP: apaga el motor, cuenta con enclavamiento.



Figura 34. Módulo de captura implementado

Fuente: Propia



Figura 35. Sistema de clasificación

Fuente: Propia

Funcionamiento

El prototipo funciona de esta manera: la lúcumas es posicionada manualmente al inicio de la cinta transportadora, donde en su transcurso será detectado por un sensor ultrasónico que cumple la función de un detector de presencia a la entrada del módulo de captura, el cual mandará una señal de inicio de programa y de realizar una captura de imagen con la webcam después de un intervalo de tiempo mientras la lúcumas se posiciona dentro del módulo captura.

La captura de imagen es procesada por múltiples etapas del algoritmo de visión artificial para la extracción de sus principales características y de redes neuronales obteniéndose como resultado los valores del peso y diámetro, así como su clasificación y separación mediante paletas movidas por servomotores para las distintas clasificaciones de lúcumas. El proceso puede ser observado y controlado mediante una interfaz gráfica en el computador.

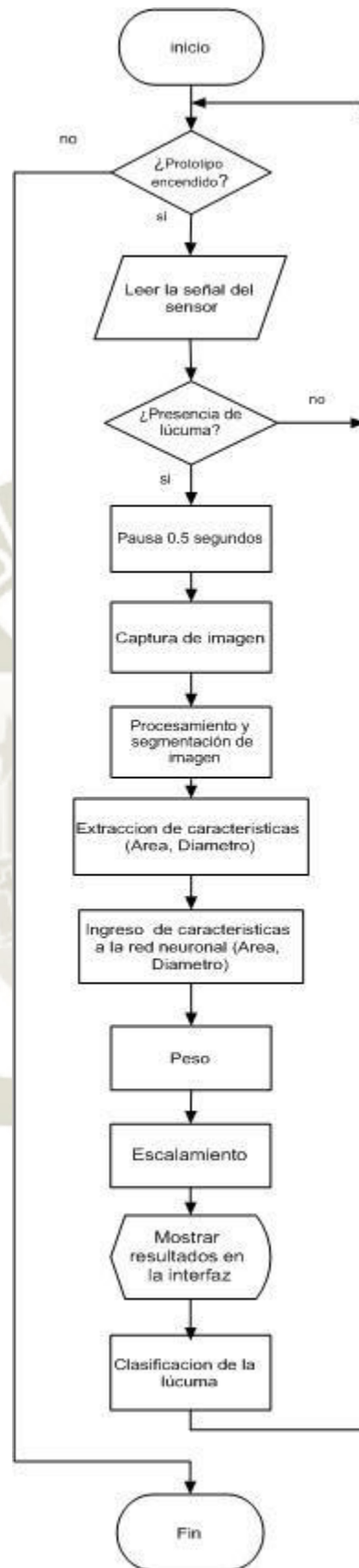


Figura 36. Diagrama de flujo de funcionamiento del prototipo

Fuente: Propia



Figura 37. Prototipo de clasificación elaborado

Fuente: Propia



Figura 38. Sistema de transmisión por cadena

Fuente: Propia

3.2. Diseño del sistema de visión artificial

El objetivo del sistema de visión artificial consiste en extraer las principales características y su interpretación a partir de la captura de imagen de la lúcumas, para lo cual es necesario que la imagen pase por múltiples etapas de procesamiento de imagen (filtros, umbralización, operaciones morfológicas, etc.) para lograr una correcta extracción de características e interpretación.

En el capítulo anterior se describieron las etapas de un sistema de visión artificial empleadas, las cuales comprendían las etapas de:

- Adquisición y captura de imagen
- Pre-procesamiento
- Segmentación
- Extracción de características
- Reconocimiento e interpretación



Figura 39. Diagrama de flujo del algoritmo de visión artificial

Fuente: Propia

3.2.1. Adquisición y captura de imagen

La obtención de la imagen se realizó utilizando una webcam. Como software de procesamiento de imágenes se empleó MATLAB en conjunto con sus apps de “MATLAB SupportPackagefor USB Webcams” para que la webcam sea reconocida por el software y “ImageAcquisitionToolbox” para configurar propiedades, realizar captura y video con la webcam y la librería de visión artificial “OPENCV”, la cual contiene múltiples funciones de procesamiento de imagen.

La captura de la imagen se realiza con unas dimensiones de 960 x 720 en el módulo de captura elaborado. En este la cámara realiza la captura en el momento en el cual la lúcuma se encuentre en el rango de visión de la cámara, para determinar cuándo se le mandara el disparo por software se colocó un sensor ultrasónico previo al ingreso de la lúcuma al módulo, el cual actuara como un sensor de presencia. La captura se realizará después de que el sensor indique la presencia de una lúcuma en la entrada del módulo.



Figura 40. Captura de imagen de una lúcuma

Fuente: Propia

Para obtener el tiempo de retardo de captura fue necesario realizar varias pruebas en el prototipo, siendo 0.5 segundos el resultado más adecuado de captura de la imagen

La captura debe realizarse con la imagen de la lúcuma en su totalidad dentro de la captura, para diferentes tamaños de lúcumas, la imagen de la lúcuma se ubicará en distintas posiciones en la captura, mas no afectará significativamente el procesamiento de la imagen.

3.2.2. Pre-Procesamiento de imagen

La imagen obtenida de la cámara se encuentra en el espacio de color RGB, a la cual debe realizarse una conversión a escala de grises. Esta conversión a escala de grises permite reducir una matriz de $[n \times m \times 3]$ a $[n \times m \times 1]$, reduciendo de tres matrices a una sola cuyo valor de pixeles tiene un rango de 0 a 255. Esta reducción disminuye considerablemente los tiempos de procesamiento de la imagen, reduce el coste computacional y facilitara la segmentación de la imagen.

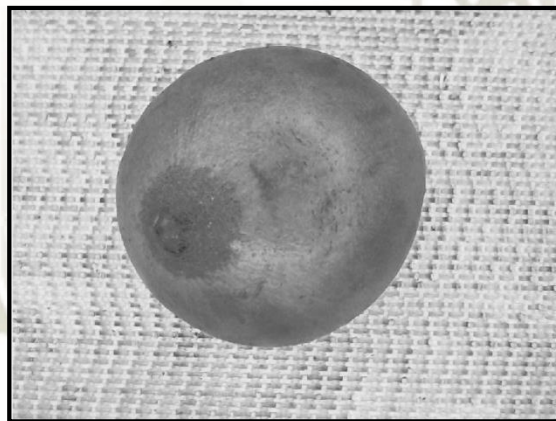


Figura 41. Conversión en escala de grises

Fuente: Propia

A la imagen capturada aplicaremos un filtro digital con el objetivo de suavizar la imagen y eliminar el ruido modificando los valores de intensidad entre pixeles vecinos muy diferentes entre sí. El filtro gaussiano emborrona la imagen y elimina el ruido, es similar al filtro de media, pero es menos sensible a cambios locales y puede producir un suavizado más uniforme.

3.2.3. Segmentación de la imagen

Luego de realizar la conversión de la imagen a escala de grises, se debe realizar la binarización de la imagen para separar la lúcuma del fondo de la imagen, esto se logró empleando una umbralización. Las técnicas de umbralización son empleada cuando hay una clara diferencia entre el objeto a extraer y el fondo, en nuestro caso la lúcuma que presenta fundamentalmente colores verdes oscuros contrasta con la faja transportadora de color blanco claro.

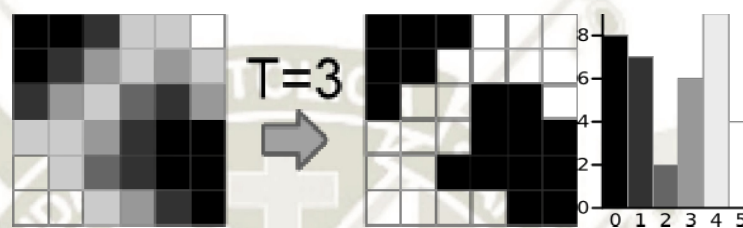


Figura 42. Umbralización empleando el método de Otsu

Fuente: OpenCV

Para realizar una correcta umbralización de la imagen, debemos revisar el contraste entre la lúcuma y la faja transportadora para realizar dicho análisis, se hace uso del histograma de la imagen. El histograma resulta útil para determinar si el contraste de una imagen digital son los adecuados.

De la figura 43 podemos observar dos picos claramente identificables, el primero se refiere a la lúcuma constituida principalmente de colores grises oscuros y el segundo refiere a la cinta transportadora con tonos más blancos. Como es posible discriminar fácilmente la lúcuma y la cinta por su distinto tono, es posible realizar su segmentación.

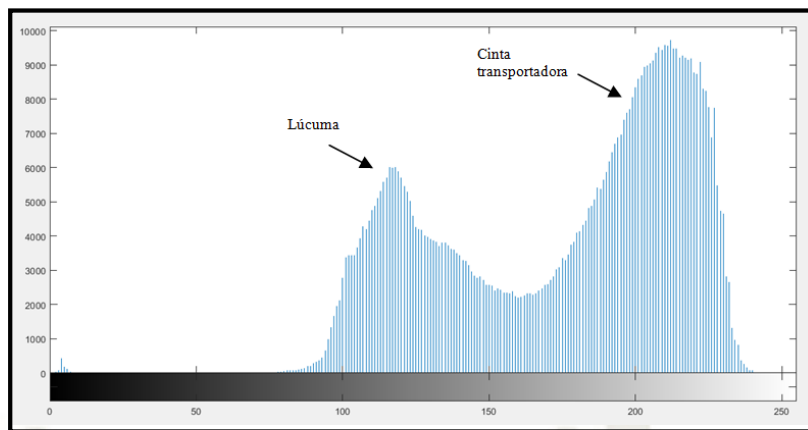


Figura 43. Histograma de la captura de la lúcuma en el módulo

Fuente: Propia

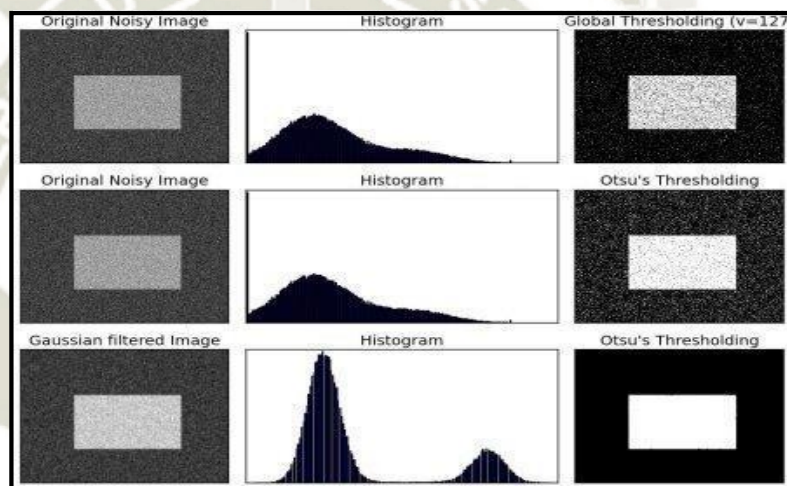


Figura 44. Corrección de ruido empleando Otsu y filtro Gaussiano

Fuente: OpenCV

Se probaron distintas opciones de umbralización para distintas muestras de lúcumas (Figura 45), donde se puede observar que la selección del valor umbral óptimo para una lúcuma no es necesariamente el mismo para otras, así mismo se puede observar que el método de Otsu en conjunto a un filtro Gaussiano producen una imagen mejor binarizada. En el método de Otsu el umbral óptimo de la imagen se halla automáticamente basado en agrupación y es definido de forma que la dispersión dentro de cada segmento sea lo más pequeña posible.

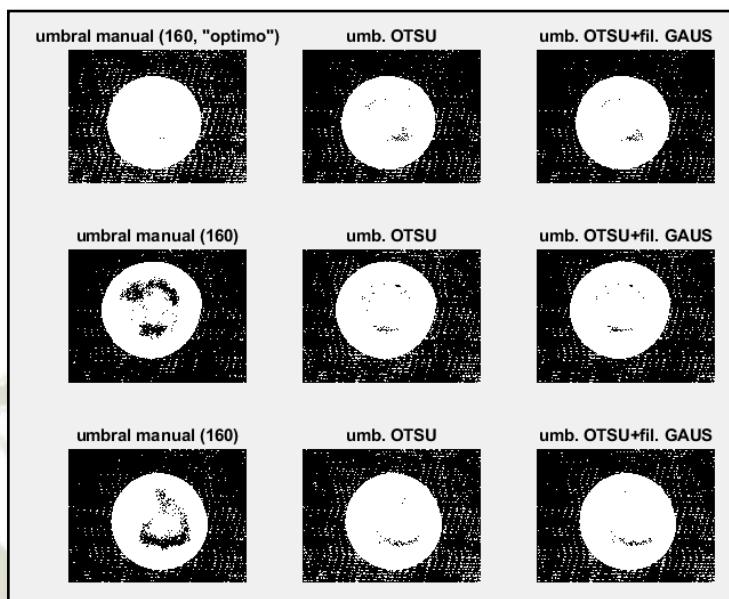


Figura 45. Pruebas de distintos métodos de umbralización

Fuente: Propia

Obtenido el umbral óptimo por el método de Otsu y una binarización inversa, todos los valores por debajo o por encima de este son reemplazados por 0 o por 255 (negro o blanco respectivamente), obteniéndose la imagen binaria.



Figura 46. Umbralización por el método de Otsu

Fuente: Propia

Como se puede observar en la figura 46, se produjeron muchos huecos o agujeros en la imagen segmentada debido a la superficie de la faja y el reflejo en la superficie de la lúcuma, estos agujeros son reconocidos como áreas u objetos diferentes con distintas características, siendo necesario aplicar operaciones morfológicas para su eliminación.

Las operaciones morfológicas simplifican imágenes para conservar las principales características de forma de los objetos. Primeramente, se aplicó los comandos *imfill* y *bwareopen* para definir restricciones de área y omitir elementos muy pequeños o huecos en la imagen binarizada. Como se observa en la figura 47, se eliminaron todos los agujeros de la imagen segmentada.



Figura 47. Llenado de agujeros

Fuente: Propia

Luego se procedió a aplicar la operación morfológica de apertura (erosión seguida de una dilatación) en la imagen binaria para suavizar contornos y eliminar pequeños salientes o conexiones en la imagen. Para lograr esto, se creó un elemento estructural para realizar la operación morfológica de apertura, como se observa en la figura 48, se probaron distintos elementos estructurales en la imagen para realizar la apertura, obteniéndose un contorno más suavizado con las estructuras disco y octágono. La figura 49 muestra una correcta segmentación de la lúcuma.

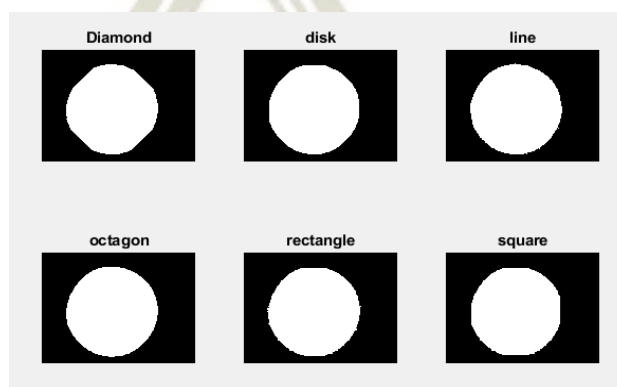


Figura 48. Operación morfológica de apertura con distintos elementos estructurales

Fuente: Propia

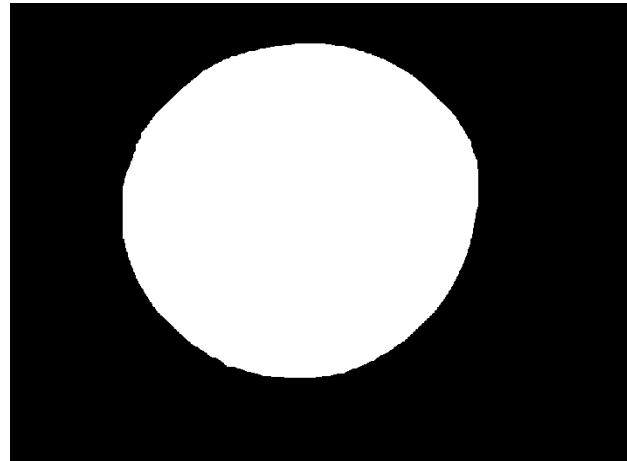


Figura 49. Operación morfológica de apertura

Fuente: Propia

3.2.4. Extracción de características

La extracción de características necesarias para la clasificación se realizó utilizando el comando *regionprops* de MATLAB. Todas las características extraídas de las muestras se encuentran en el ANEXO 2. Las características o descriptores más relevantes son dos: el diámetro equivalente para determinar el diámetro ecuatorial y el área, ambos representados en número de píxeles y necesarios para determinar el peso aproximado.

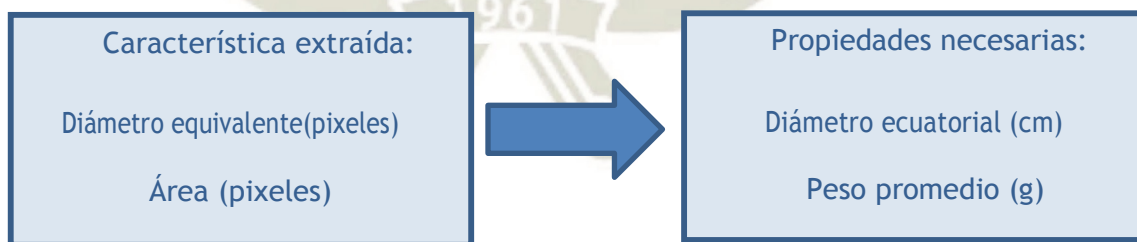


Figura 50. Black Box

Fuente: Propia

1	2		3			4	5	6	7	8	
Area	Centroid		BoundingBox			MajorAxisLength	MinorAxisLength	Orientation	EquivDiameter	Perimeter	
220789	449.1403	324.5988	180.5000	65.5000	538	523	542.6601	518.2457	27.6806	530.2049	1.6751e+03

Figura 51. Cuadro de regionprops en Matlab

Fuente: Propia

Para una mejor representación de las características obtenidas, se graficó encima de la imagen las principales características obtenidas: Bounding box (cuadrado verde), centroide (punto gris), orientación, diámetro (círculo rojo), longitud (línea verde) y ancho (línea celeste).

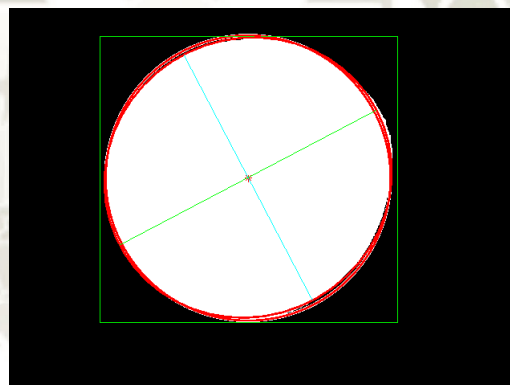


Figura 52. Principales propiedades graficadas en la imagen

Fuente: Propia

3.2.5. Reconocimiento e interpretación

Hasta ahora las características extraídas previamente se encuentran como número de píxeles, siendo necesario realizar su conversión a centímetros empleando una escala.

Para obtener dicha escala se realizó una captura de una regla patrón en el módulo en las mismas condiciones que las lúcumas las cual se utilizó como estándar, donde se revisó el número de píxeles en un centímetro de la regla patrón, se

observó que para obtener correctamente dicha escala es necesario que la regla patrón se encuentre a la mitad de la altura de la lúcuma, la cual es la zona del mayor diámetro y área. Para lograr esto se añadió una altura inicial base desde la cinta transportadora hasta la regla patrón.

Inicialmente se optó por poner una sola altura inicial base que sería un valor promedio de las alturas de diferentes tamaños de lúcumas, pero se observó errores significativos en los valores de diámetros de las lúcumas de mayor y menor tamaño, entonces se decidió optar por tener tres distintas alturas base en función del tamaño de las lúcumas para obtener resultados más exactos.



Figura 53. Obtención de la altura inicial de la regla patrón

Fuente: Propia

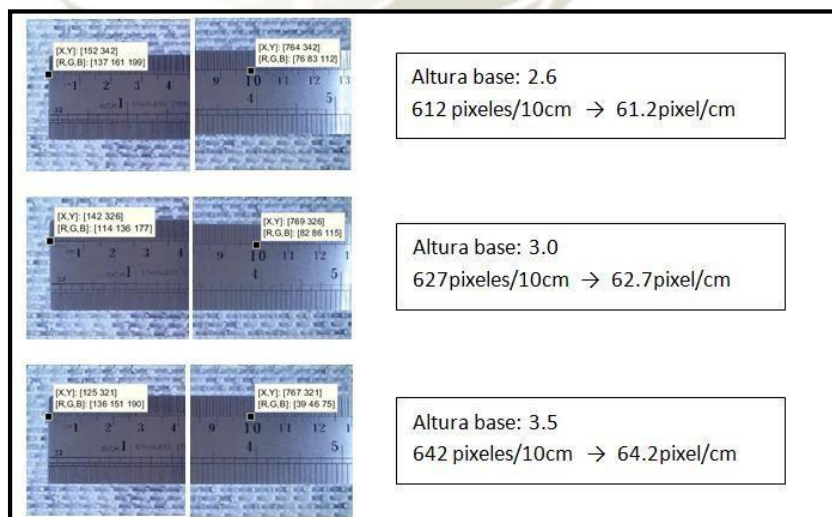


Figura 54. Obtención de las diferentes escalas

Fuente: Propia

Luego de analizar la figura 54, se puede obtener que 1cm equivale a 64.2 pixeles en una lúcum grande, 62.7 en una lúcum mediana y 61.2 en una lúcum pequeña aproximadamente, pudiéndose obtener las dimensiones de la lúcum en sistema métrico.

Tabla 4

Factor de conversión para distintos tamaños de lúcum

Tamaño de lúcum	Rango (pixeles)	Factor de conversión
Grande	Área \geq 150000	64.2
Mediana	100000 < Área < 150000	62.7
Pequeña	Área \leq 100000	61.2

Fuente: Propia

3.3. Diseño de la red neuronal

Al momento de selección de la red neuronal se tomó en cuenta la velocidad de entrenamiento, facilidad de implementación y el error, optándose por una red neuronal prealimentada (Feed-Forward Neural Network o FFNN), esta cuenta con una capa de entrada, una capa intermedia oculta y una capa de salida, en este tipo de red las neuronas de entrada están totalmente conectadas a las neuronas de la capa intermedia oculta y estas a su vez totalmente conecta a las neuronas de la capa de salida. La red FFNN tendrá como entrada las características extraídas previamente y de salida los valores de peso aproximado de la lúcum.

3.3.1. Arquitectura de la red neuronal

La red neuronal implementada está compuesta por 3 capas:

Capa de entrada: considerándose la forma semiesférica de la lúcum se optó ingresar como entrada a la red neuronal las características más representativas extraídas: el diámetro equivalente y el área. El número de neuronas en las entradas

es igual al número de entradas a la red neuronal, siendo dos neuronas en la capa de entrada.

Capa oculta: o capa intermedia, neuronas cuyas entradas proceden de capas previas y sus salidas pasan a las capas posteriores.

Capa de salida: La salida de la red neuronal será el peso aproximado de la lúcumá. El número de neuronas en la salida es igual al número de salidas de la red neuronal, siendo una neurona en la capa de salida.

3.3.2. Entrenamiento de la red neuronal

El aprendizaje es el proceso por el cual la RNA modifica el valor de sus pesos en respuesta a la información en la entrada. El aprendizaje en FFNN con propagación hacia atrás ocurre durante la fase de entrenamiento en la cual los valores de entrada del banco de datos se aplican a la capa de entrada y luego se aplica para adelante. El patrón de activación en la capa de salida es comparado con el valor correcto del banco de datos para calcular el error, el cual es propagado hacia atrás desde la capa de salida a las neuronas de entrada para ajustar los pesos en cada capa de la red neuronal reduciendo el error y mejorando la eficiencia de la red. Posteriormente de realizado el entrenamiento de la red, es necesario probarla para revisar su eficiencia y volver a entrenar de ser necesario.

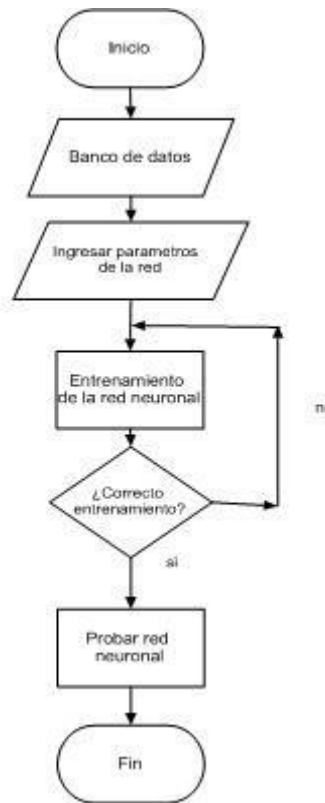


Figura 55. Diagrama de flujo de entrenamiento de la red neuronal

Fuente: Propia

Como se mencionó previamente, para efectuar el entrenamiento de red neuronal es necesario contar con una base de datos y tratándose de una red supervisada, es necesario conocer los valores de entradas y salidas. Estos se obtuvieron seleccionando un grupo de 31 lúcumas de distintas dimensiones obteniendo sus características de área, diámetro y peso de cada lúcumas. A partir de estas características se obtuvo la base de datos necesarios para el entrenamiento de la red neuronal. El peso se obtuvo utilizando una balanza electrónica.



Figura 56. Pesado de la lúcuma

Fuente: Propia

Para ingresar los valores de entrada y salida a la red se tiene que realizar previamente un escalamiento para que los valores se encuentren comprendidos entre 0 y 1. En la entrada al área se realizó multiplicación por $1/350000$, al diámetro se realizó una multiplicación por $1/700$ mientras que a la salida se realizó una multiplicación por $1/400$. Se puede expresar las entradas y salidas después de aplicar el escalamiento en funciones. Los datos se ingresan de la siguiente forma:

$$[P1, P2, P3, Pn...] = \{(A1, A2, A3, An...); (D1, D2, D3, Dn...)\}$$

Donde:

- A representa el área escalada,
- D representa el diámetro escalado y
- P representa el valor de peso escalado medido en la balanza.

Luego de la obtención de las características de las lúcumas, pesado individual y después de su escalamiento correspondiente, se obtuvo el banco de datos de la tabla 5, con el cual se procedió a entrenar la red neuronal:

Tabla 5

Banco de datos de la red neuronal

N°	Área escalada	Diámetro escalado	Peso escalado
1	0.917	0.913	0.955
2	0.759	0.831	0.728
3	0.743	0.822	0.725
4	0.714	0.806	0.715
5	0.668	0.779	0.713
6	0.637	0.761	0.688
7	0.635	0.760	0.658
8	0.625	0.754	0.625
9	0.595	0.735	0.575
10	0.590	0.733	0.565
11	0.531	0.695	0.505
12	0.516	0.685	0.493
13	0.510	0.681	0.478
14	0.482	0.662	0.473
15	0.481	0.661	0.463
16	0.463	0.649	0.458
17	0.452	0.641	0.448
18	0.443	0.645	0.438
19	0.415	0.614	0.403
20	0.407	0.609	0.383
21	0.385	0.592	0.370
22	0.378	0.586	0.353
23	0.355	0.568	0.348
24	0.354	0.574	0.335
25	0.327	0.545	0.280
26	0.317	0.537	0.270
27	0.313	0.534	0.250
28	0.283	0.507	0.248
29	0.256	0.482	0.225
30	0.254	0.481	0.223
31	0.238	0.465	0.180

Fuente: Propia

No existe un método estándar aceptable para definir el número exacto de neuronas y capas ocultas en una red neuronal, entonces se procedió a determinar su número realizando múltiples pruebas para distinto número de neuronas y capas ocultas en la red, seleccionando la red que genere el menor error. A más neuronas en la capa oculta, producirán mejores resultados, pero pueden volverse inestable y tendrá mayores tiempos de entrenamiento. Se encontró que empleando 50 neuronas y 2 capas ocultas generaban el menor error entre las pruebas realizadas.

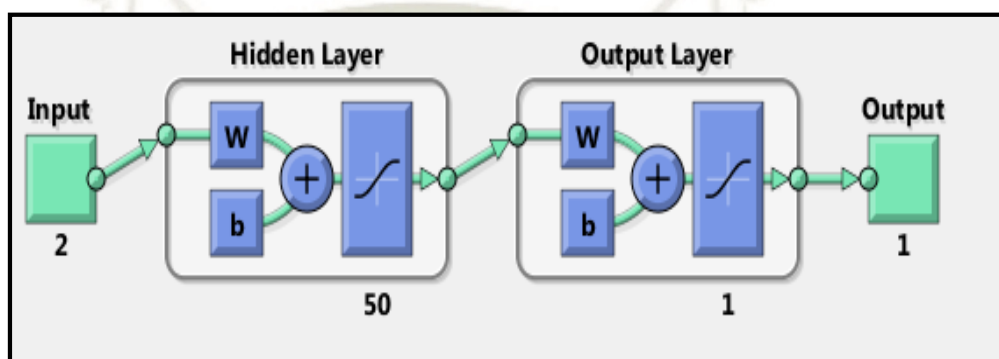


Figura 57. Estructura de la red neuronal diseñada

Fuente: Propia

Para el entrenamiento de la red neuronal se utilizó el comando `nntool` de matlab, La función de aprendizaje empleada es `Traingdm` (Gradient descent with momentum backpropagation), el cual actúa como un filtro pasa bajos, el momento permite a RNA ignorar pequeños valores de error en la superficie, sino podría quedar atorada en un mínimo local.

Para la función de entrenamiento se seleccionó el algoritmo Levenberg-Marquardt, la cual es una función de entrenamiento con bias y pesos actualizables empleando la optimización de Levenberg-Marquardt, suele ser la primera opción para redes supervisadas, tiene una rápida convergencia, buena precisión y es capaz de lograr un menor error cuadrático medio (MSE) siempre que el número de pesos en la red no sea demasiado alto.

La función tangente sigmoidea (tansig) ofrece mayor flexibilidad al tener un mayor rango de resultados, de valores de -1 a 1. El entrenamiento se efectuará hasta lograr un valor mínimo de error establecido o al alcanzar el número de 1000 épocas establecida.

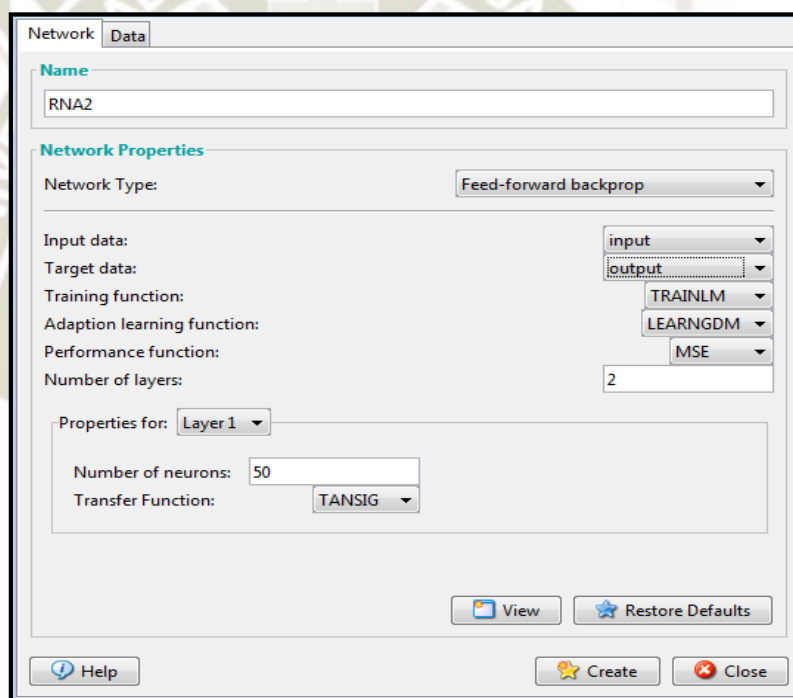


Figura 58. Creación de la red neuronal

Fuente: Propia

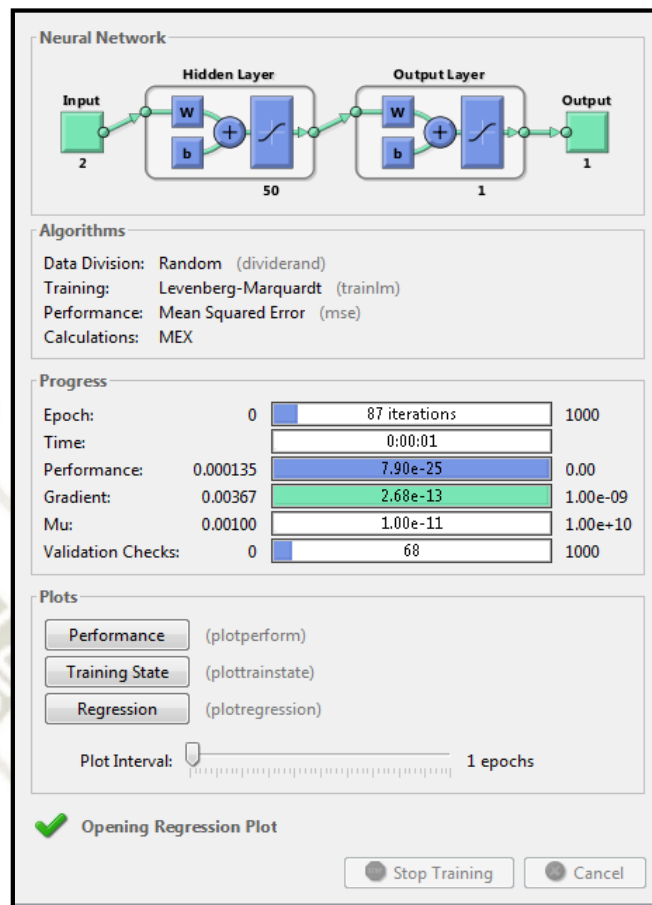


Figura 59. Entrenamiento de la red neuronal

Fuente: Propia

En la figura 60 se puede observar la regresión lineal entre la salida de la red y sus valores objetivos correspondientes. Se puede observar que la salida sigue al valor objetivo en el entrenamiento, validación y prueba con un valor R superior a 0.99 en todos los casos.

La figura 61 representa la gráfica del error cuadrático medio (MSE) de entrenamiento, validación y prueba. Se puede observar que los resultados del entrenamiento son favorables debido a que el error cuadrático medio obtenido es bajo, además el error de pruebas y el error de validación tienen similares características. Tampoco se produjo un overfitting significativo en la iteración 19 (donde se produce el mejor rendimiento de validación).

La figura 62 muestra algunas estadísticas de entrenamiento de la red y la variación del gradiente. Se puede observar que la gradiente a sobrepasado el valor mínimo local de la función objetivo.

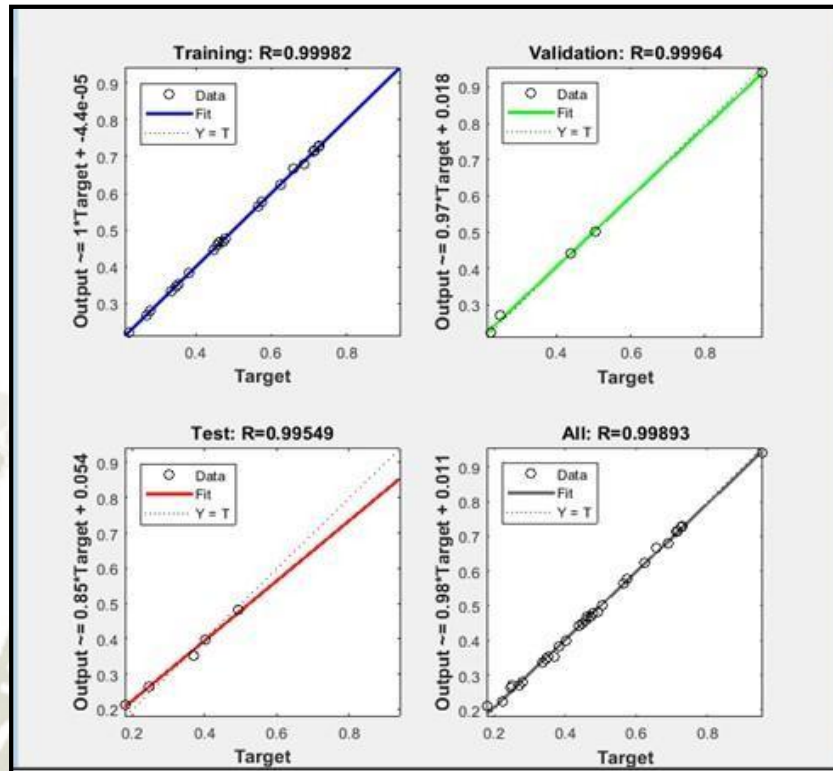


Figura 60. Regresión lineal de la red

Fuente: Propia

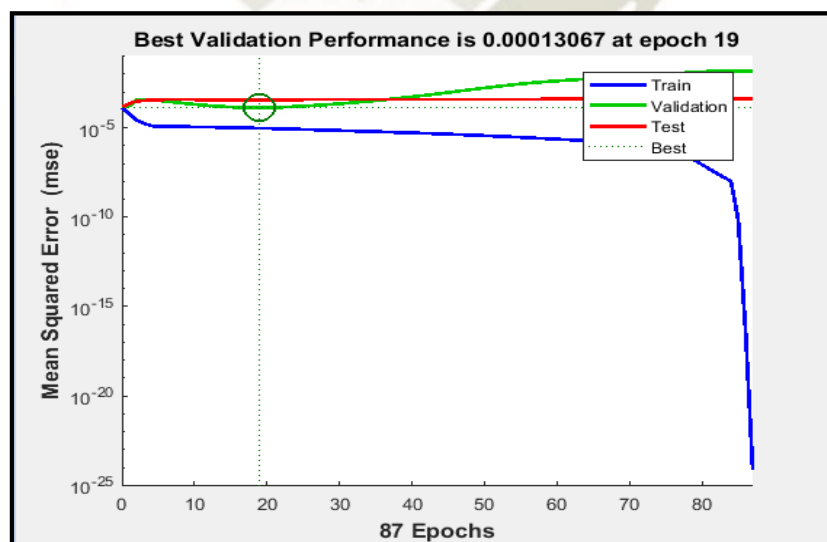


Figura 61. Error cuadrático medio de la red

Fuente: Propia

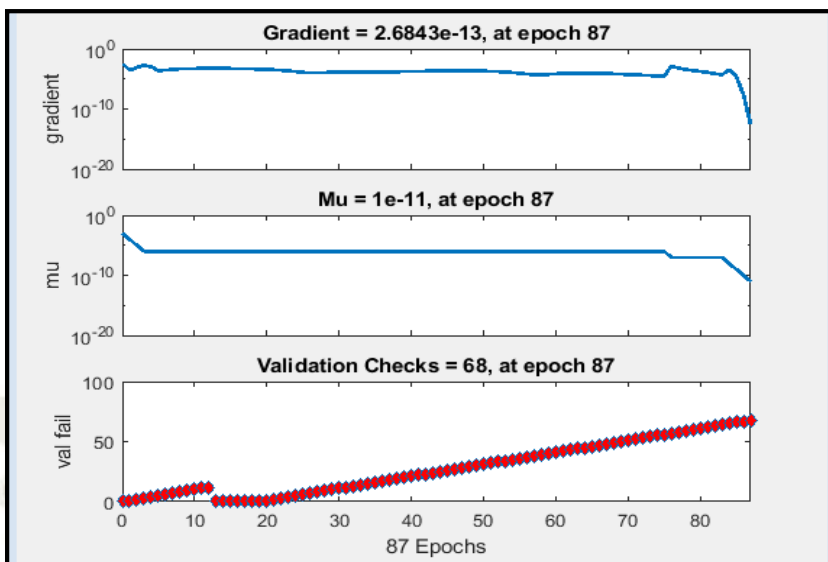


Figura 62. Variación de la gradiente en la red

Fuente: Propia

3.4. Etapa de Clasificación

La clasificación de lúcumas se realiza empleando la norma del CODEX para la lúcuma: CODEX STAN 305R-2011, la cual aplica para la especie *Poteria lúcuma* (R. y P.) de la familia *Sapotaceae* la cual incluye a la lúcuma tipo “seda”. En la norma se estipula las disposiciones relativas a la clasificación por calibres y categorías:

Tabla 6

Especificaciones de calibre para lúcuma

Calibre	Peso unitario promedio (g)	Rango de pesos (g)
A	-	Menor a 55
B	90	50 – 130
C	160	120 – 200
D	230	190 - 270
E	300	260 – 340
F	-	Mayor a 330

Fuente: FAO, 2011

Los rangos especificados de pesos incluyen la tolerancia de calibre, que es de $\pm 10\%$ en peso de la fruta.

Tabla 7

Categorías de calibre para lúcuma

	Categorías		
	Extra	I	II
Diametro ecuatorial (cm)	De 6 a 9	Mayor a 9, o mayor o igual a 4 y menor a 6	Menor a 4
Peso (g)	120 a 300	Mayor a 300, o mayor o igual a 50 y menor a 120	Menor a 50

Fuente: FAO, 2011

Como se puede observar de la tabla 6, ciertos rangos cuentan con una o dos posibilidades de clasificaciones debido a las tolerancias por lo que se empleó el criterio descrito en la tabla 8 como ejemplo para distinguir si la clasificación se realizó correctamente, parcial o incorrectamente entre la clasificación real y la obtenida por software.

Tabla 8

Criterio de coincidencia empleado

Clasificación		Coincidencia
Real	Software	
A	A	Si
A	B	No
A, B	A	Si
A	A, B	Parcial
A, B	A, B	Si

Fuente: Propia

3.4.1. Tolerancias

La norma CODEX STAN 305-2011 indica las tolerancias para el calibre y categoría a tener en cuenta para su clasificación y envasado:

- Las tolerancias de calibre para todas las categorías son de $\pm 10\%$ el peso de la lúcuma

- Para la categoría “Extra” la tolerancia es de 5% del peso de la lúcuma que no satisfaga esta categoría pero que si satisfaga los requerimientos de la categoría I
- Para la categoría “I” la tolerancia es de 10% del peso de la lúcuma que no satisfaga esta categoría pero que si satisfaga los requerimientos de la categoría II
- Para la categoría “II” la tolerancia es de 10% del peso de la lúcuma que no satisfaga esta categoría.

3.5. Interfaz gráfica

La interfaz gráfica fue diseñada en el programa MATLAB empleando la interfaz de usuario GUIDE para permitir al usuario interactuar fácilmente con el programa por lo tanto este se diseñó de modo que sea agradable y de fácil uso para el usuario. En la interfaz se puede visualizar los siguientes elementos: video del proceso, la captura de imagen, visualizar la correcta segmentación, botón de encendido y apagado de la cámara, botón de cierre de programa, selector de clasificación a usar, diámetro y peso aproximados.

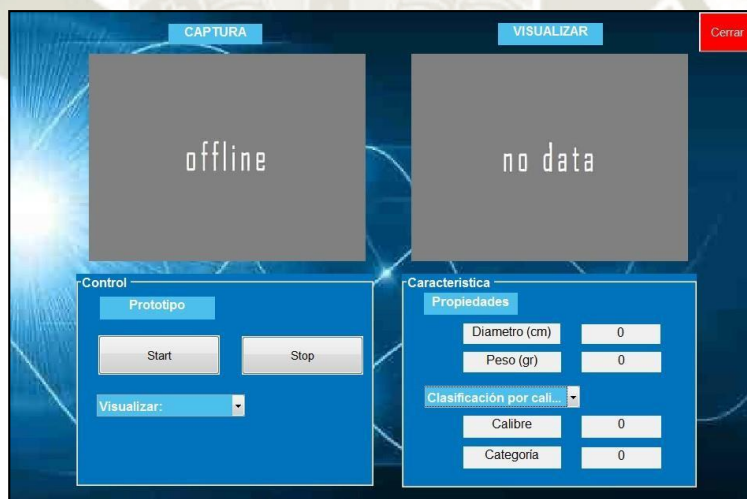


Figura 63. Interfaz de usuario GUI

Fuente: Propia

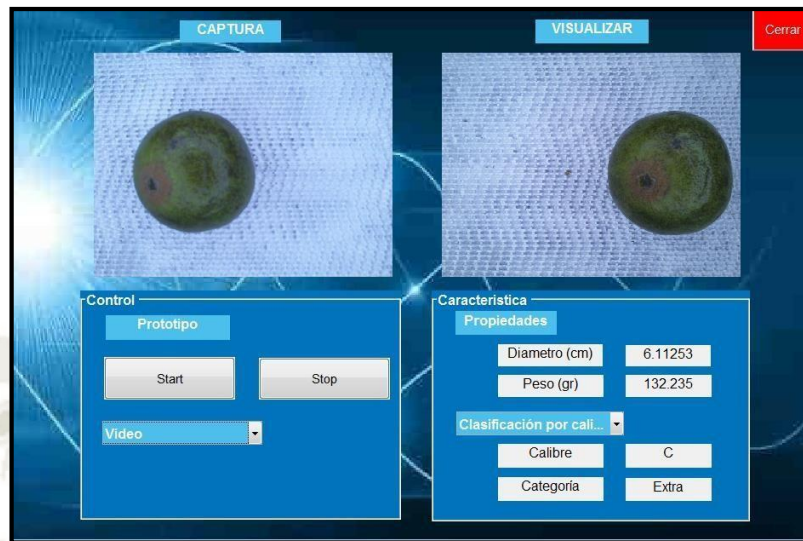


Figura 64. Función de visualizar el proceso por cámara

Fuente: Propia

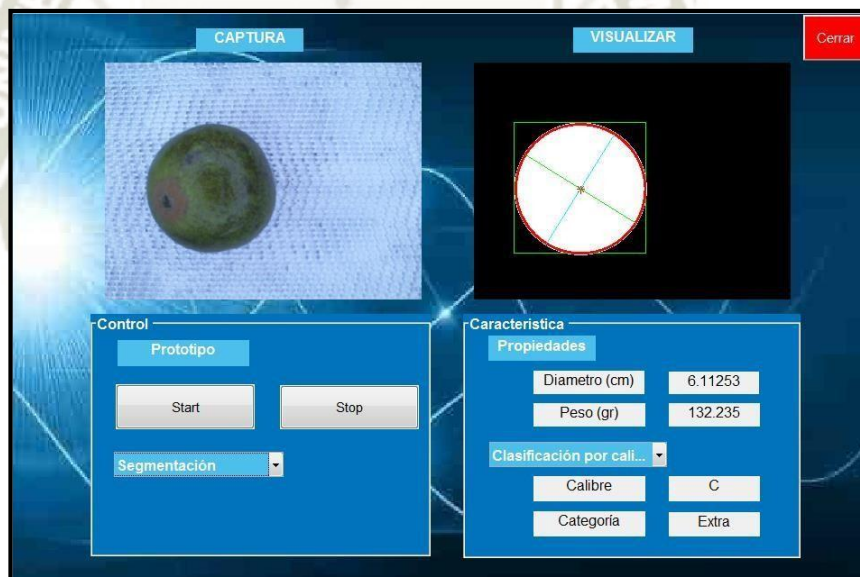


Figura 65. Función de visualizar segmentación en GUI

Fuente: Propia

3.6. Costos

Se realizó un análisis de costos del proyecto tomándose en cuenta todos los componentes necesarios para la elaboración del sistema de visión artificial y del prototipo, los cuales se encuentran detallados en la tabla 9, la suma total de todos los componentes es de S/1721 soles.

Tabla 9

Costos del prototipo

Componente	Cantidad	Material	Precio (S/.)
Componentes electrónicos	1	Arduino Mega 2560	80.00
	1	Logitech c525 HD Webcam	250.00
	3	Servomotor mg 995 tower pro	81.00
	1	HC-SR04 Módulo sensor ultrasónico	13.00
	30	Cables de conexión I/O	8.00
	1	Circuito impreso	20.00
Componentes mecánicos	1	Motor monofásico 0.5 HP 110v	225.00
	1	Condensador 240 µf/kW	8.00
	1	Cadena de transmisión	60.00
	1	Pulsador Start / Stop	12.00
		Materiales de fabricación (planchas y tubos)	180.00
		Fabricación de la base del módulo	400.00
	Fabricación del módulo de captura	80.00	
Software	1	Licencia Matlab	120.00
Varios	1	Cintas Led 5 metros	55.00
	1	Transformador 220/5v	15.00
	1	Transformador 220/12v	20.00
	1	Transformador 220/110v	35.00
	1	Cinta transportadora	30.00
	2	Canaletas de cables, 2metros	7.00
	3	Clavijas	6.00
	1	Cable robusto AWG	12.00
	1	Cinta aislante	4.00
	TOTAL		

Fuente: Propia

CAPITULO IV

4. PRUEBAS Y RESULTADOS

Para la etapa de pruebas, se utilizaron lúcumas cuyas características no se emplearon para el entrenamiento de la red neuronal con el objetivo de validar los resultados y realizar acciones correctivas de ser el caso.

A continuación, en la tabla 10, se muestran las principales características extraídas para la clasificación: área y diámetro equivalente para una muestra de 50 lúcumas, las características obtenidas y la clasificación correspondiente de cada lúcumas pueden ser observadas en la computadora empleando la interfaz de usuario en Matlab.

El tiempo de procesamiento del algoritmo desde la captura de imagen hasta su clasificación es de aproximadamente 1 segundo y el tiempo de análisis y clasificación de la lúcumas desde su ingreso al prototipo hasta su salida es de aproximadamente 3.3 segundos.

Tabla 10

Registro de características seleccionadas de 50 lúcumas

N° Lúcuma	Área (píxeles)	Diámetro equivalente (píxeles)
1	224717	534.90
2	187857	489.07
3	195825	499.33
4	170527	465.96
5	295534	613.42
6	137270	418.06
7	127443	402.82
8	119665	390.34
9	105592	366.67
10	97816	352.91
11	283481	600.78
12	235689	547.80
13	238313	550.84
14	220916	530.36
15	213167	520.97
16	200795	505.63
17	195370	498.75
18	191196	493.39
19	169613	464.71
20	141578	424.57
21	116345	384.88
22	82810	324.71
23	213051	520.83
24	148980	435.53
25	148727	435.16
26	134973	414.55
27	117508	386.80
28	116401	408.84
29	115311	383.17
30	161964	454.11
31	200446	505.19
32	72806	304.47
33	168597	463.32
34	169319	464.31
35	177140	474.91
36	216525	525.06
37	119682	390.36
38	122604	395.10
39	122344	394.68
40	126625	401.53
41	143014	426.72
42	143449	427.37
43	153715	442.40
44	174551	471.43
45	178721	477.03
46	185016	485.36
47	209133	516.02
48	118109	387.79
49	151596	439.34
50	184451	484.61

Fuente: Propia

Comparación y obtención del error

Diámetro ecuatorial

Los valores de diámetro equivalente en pixeles obtenidos previamente se convierten a centímetros con la escala correspondiente. La tabla 11 muestra la comparación del diámetro entre el valor obtenido por software, el valor real medido manualmente con un vernier y la diferencia o error entre ambas mediciones para las 50 muestras, se puede observar que el error máximo obtenido es de 2.4 milímetros equivalente a 3.20% del diámetro de la lúcuma. Obteniéndose una eficiencia aproximada del 96.80%.

En la figura 66 se graficó el diámetro equivalente real y el obtenido por software para las muestras tomadas, estas fueron ordenadas de menor a mayor diámetro. Se puede observar que ambas tienen valores muy cercanos y la diferencia o error no es muy alto entre ambas.

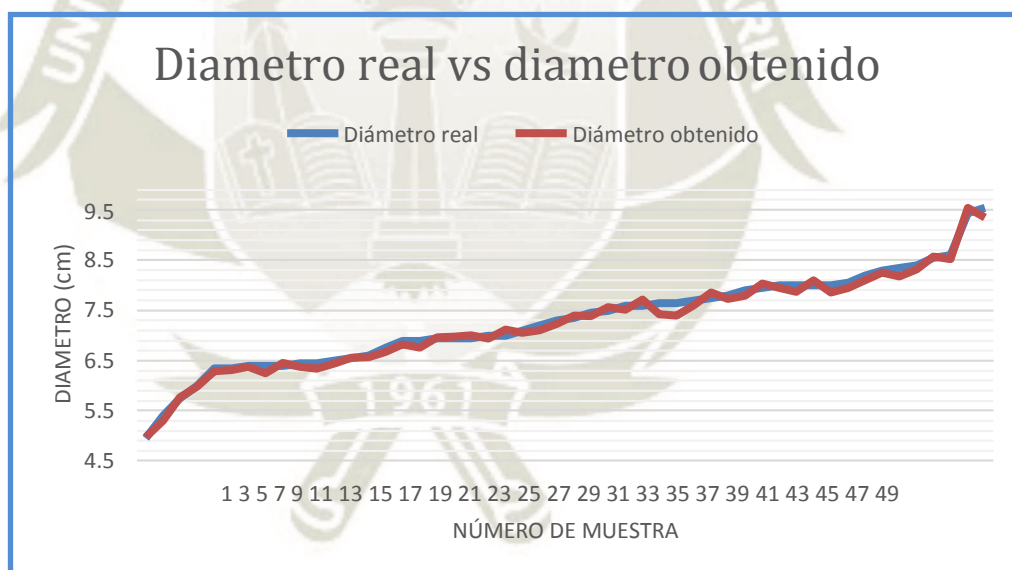


Figura 66. Diámetro real vs diámetro obtenido

Fuente: Propia

Tabla 11

Comparación de los valores reales y obtenidos por software para el diámetro ecuatorial

N° Muestra	Diámetro ecuatorial (cm)			
	Real (cm)	Obtenida por software (cm)	Error (cm)	Error %
1	8.40	8.33	0.07	0.81
2	7.90	7.80	0.10	1.26
3	8.00	7.96	0.04	0.45
4	7.65	7.43	0.22	2.86
5	9.45	9.55	-0.10	-1.11
6	6.90	6.83	0.07	1.00
7	6.60	6.58	0.02	0.27
8	6.40	6.38	0.02	0.34
9	6.00	5.99	0.01	0.14
10	5.75	5.77	-0.02	-0.29
11	9.55	9.36	0.19	2.01
12	8.60	8.53	0.07	0.78
13	8.55	8.58	-0.03	-0.35
14	8.30	8.26	0.04	0.47
15	8.20	8.11	0.09	1.04
16	8.00	7.88	0.12	1.55
17	8.05	7.95	0.10	1.19
18	7.75	7.87	-0.12	-1.54
19	7.35	7.41	-0.06	-0.84
20	7.00	6.94	0.06	0.89
21	6.35	6.29	0.06	0.96
22	5.40	5.31	0.09	1.75
23	8.00	8.11	-0.11	-1.41
24	7.00	7.12	-0.12	-1.66
25	7.20	7.11	0.09	1.24
26	6.90	6.77	0.13	1.83
27	6.35	6.32	0.03	0.47
28	6.75	6.68	0.07	1.03
29	6.40	6.26	0.14	2.17
30	7.30	7.24	0.06	0.79
31	8.0	7.87	0.13	1.64
32	4.95	4.98	-0.03	-0.51
33	7.45	7.39	0.06	0.81
34	7.65	7.41	0.24	3.20
35	7.5	7.57	-0.07	-0.99
36	8.35	8.18	0.17	2.05
37	6.45	6.38	0.07	1.11
38	6.40	6.46	-0.06	-0.87
39	6.50	6.45	0.05	0.78
40	6.55	6.56	-0.01	-0.17
41	6.95	6.97	-0.02	-0.32
42	6.95	6.98	-0.03	-0.48
43	7.10	7.06	0.04	0.62
44	7.60	7.52	0.08	1.07
45	7.70	7.61	0.09	1.19
46	7.80	7.74	0.06	0.76
47	7.95	8.04	-0.09	-1.10
48	6.45	6.34	0.11	1.76
49	6.95	7.01	-0.06	-0.82
50	7.60	7.73	-0.13	-1.70

Fuente: Propia

Peso

Los valores de diámetro y área en píxeles obtenidos previamente son escalados e ingresan a una red neuronal para la obtención del peso aproximado. La tabla 12 muestra la comparación del peso entre el valor real medido manualmente con una balanza electrónica, el valor obtenido por software y la diferencia o error entre ambas mediciones para las 50 muestras. Analizando, se puede observar que el error máximo obtenido es de 12.09 gramos y un error máximo porcentual de 5.00% de peso, dando como resultado una eficiencia promedio de 95.00%. El error de 5% se encuentra dentro de los rangos de tolerancias de la norma empleada. Se puede observar que los errores son mayores en las frutas que presentan formas atípicas.

En la figura 67 se graficó el peso real y el obtenido por software para las muestras tomadas, estas fueron ordenadas de menor a mayor peso. Se puede observar que ambas tienen valores muy cercanos y la diferencia o error no es muy alto entre ambas.

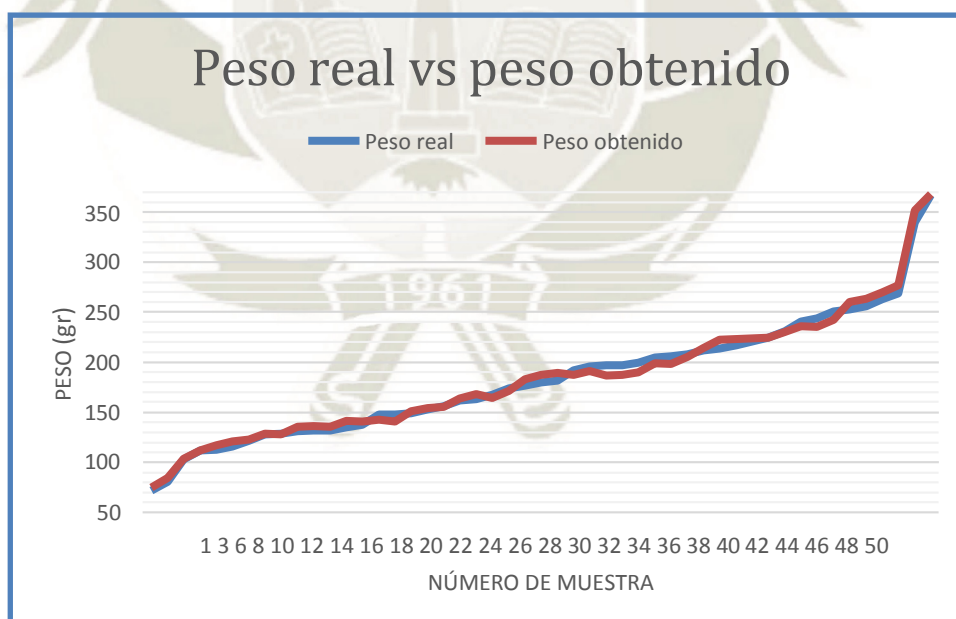


Figura 67. Peso real vs peso obtenido

Fuente: Propia

Tabla 12

Comparación de los valores reales y obtenidos por software para el peso

N° Muestra	Peso (gr)			
	Real (gr)	Obtenida por software (gr)	Error (gr)	Error %
1	256	263.63	-7.63	-2.98
2	208	205.55	2.45	1.18
3	217	223.53	-6.53	-3.01
4	192	187.81	4.19	2.18
5	367	368.29	-1.29	-0.35
6	148	143.06	4.94	3.34
7	135	141.46	-6.46	-4.79
8	129	128.38	0.62	0.48
9	112	112.02	-0.02	-0.02
10	103	103.58	-0.58	-0.56
11	340	352.09	-12.09	-3.56
12	269	276.87	-7.87	-2.93
13	263	270.42	-7.42	-2.82
14	253	259.80	-6.80	-2.69
15	241	236.00	5.00	2.07
16	221	224.31	-3.31	-1.50
17	214	223.00	-9.00	-4.21
18	212	214.31	-2.31	-1.09
19	180	187.50	-7.50	-4.17
20	149	151.14	-2.14	-1.44
21	113	117.26	-4.26	-3.77
22	81	84.86	-3.86	-4.77
23	244	235.82	8.18	3.35
24	168	164.56	3.44	2.05
25	162	164.22	-2.22	-1.37
26	148	141.08	6.92	4.68
27	116	120.87	-4.87	-4.20
28	131	136.01	-5.01	-3.82
29	110	114.18	-4.18	-3.80
30	177	183.4	-6.40	-3.62
31	225	224.42	0.58	0.26
32	72	74.94	-2.94	-4.08
33	197	187.15	9.85	5.00
34	197	187.4	9.60	4.87
35	200	190.45	9.55	4.78
36	251	242.43	8.57	3.41
37	128	128.43	-0.43	-0.34
38	132	136.26	-4.26	-3.23
39	132	135.71	-3.71	-2.81
40	138	141.18	-3.18	-2.30
41	153	154.61	-1.61	-1.05
42	156	155.55	0.45	0.29
43	174	171.79	2.21	1.27
44	182	189.21	-7.21	-3.96
45	196	191.47	4.53	2.31
46	205	201.32	3.68	1.80
47	231	230.31	0.69	0.30
48	122	123.17	-1.17	-0.96
49	163	168.29	-5.29	-3.25
50	206	198.29	7.71	3.74

Fuente: Propia

Especificaciones de calibre

La tabla 13 muestra la comparación entre el calibre determinado manualmente por medición y el calibre determinado mediante software basados en la norma para una muestra de 50 lúcumas. El criterio de coincidencia empleado se encuentra en la tabla N°8.

Analizando la figura 68 la cual muestra las proporciones de coincidencia de clasificación respecto a la especificación de calibre, se puede observar que 47 de 50 lúcumas tuvieron una clasificación correcta siendo equivalentes al 94% del total, 3 lúcumas tuvieron una clasificación parcial igual al 6% del total y 0 lúcumas una clasificación incorrecta, obteniéndose como resultado una eficiencia global de 94% para la clasificación por especificación de calibre.

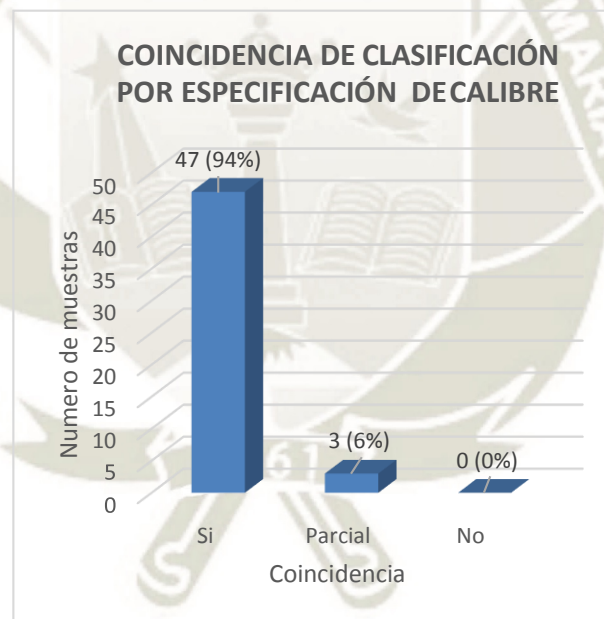


Figura 68. Coincidencia de clasificación por especificación de calibre

Fuente: Propia

Tabla 13

Clasificación por especificaciones de calibre

N° Muestra	Calibre		
	Real	Obtenida por software	Coincidencia
1	D	D, E	parcial
2	D	D	si
3	D	D	si
4	C, D	C	si
5	F	F	si
6	C	C	si
7	C	C	si
8	B, C	B, C	si
9	B	B	si
10	B	B	si
11	E, F	F	si
12	D, E	E	si
13	D, E	D	si
14	D	D	si
15	D	D	si
16	D	D	si
17	D	D	si
18	D	D	si
19	C	C	si
20	C	C	si
21	B	B	si
22	B	B	si
23	D	D	si
24	C	C	si
25	C	C	si
26	C	C	si
27	B	B, C	parcial
28	B, C	C	si
29	B	B	si
30	C	C	si
31	D	D	si
32	B	B	si
33	C, D	C	si
34	C, D	C	si
35	C, D	C, D	si
36	D	D	si
37	B, C	B, C	si
38	C	C	si
39	C	C	si
40	C	C	si
41	C	C	si
42	C	C	si
43	C	C	si
44	C	C	si
45	C, D	C, D	si
46	D	D	si
47	D	D	si
48	B, C	B, C	si
49	C	C	si
50	D	C, D	parcial

Fuente: Propia

Categorías de calibre

La tabla 14 muestra la comparación entre las categorías de calibre determinadas manualmente por medición y las determinado mediante software basados en la norma para una muestra de 50 lúcumas.

Analizando la figura 69 la cual muestra las proporciones de coincidencia de clasificación respecto a las categorías de calibre, se puede observar que 49 de 50 lúcumas tuvieron una clasificación correcta siendo equivalentes al 98% del total y solamente 1 lúcumas tuvo una clasificación incorrecta siendo equivalente al 2% del total, obteniéndose como resultado una eficiencia global de 98% para la clasificación por categorías de calibre.

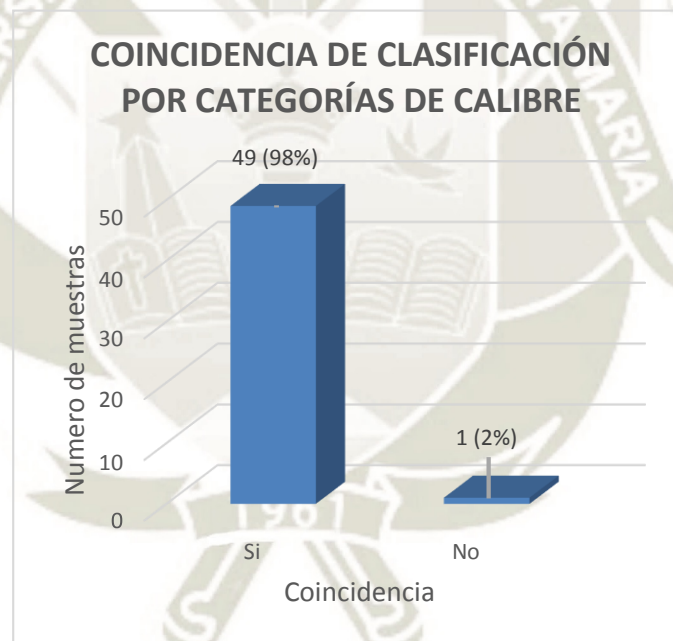


Figura 69. Coincidencia de clasificación por categorías de calibre

Fuente: Propia

Tabla 14
Clasificación por categorías de calibre

N° Muestra	Categorías		
	Real	Obtenida por software	Coincidencia
1	Extra	Extra	si
2	Extra	Extra	si
3	Extra	Extra	si
4	Extra	Extra	si
5	I	I	si
6	Extra	Extra	si
7	Extra	Extra	si
8	Extra	Extra	si
9	I	I	si
10	I	I	si
11	I	I	si
12	Extra	Extra	si
13	Extra	Extra	si
14	Extra	Extra	si
15	Extra	Extra	si
16	Extra	Extra	si
17	Extra	Extra	si
18	Extra	Extra	si
19	Extra	Extra	si
20	Extra	Extra	si
21	I	I	si
22	I	I	si
23	Extra	Extra	si
24	Extra	Extra	si
25	Extra	Extra	si
26	Extra	Extra	si
27	I	Extra	no
28	Extra	Extra	si
29	I	I	si
30	Extra	Extra	si
31	Extra	Extra	si
32	I	I	si
33	Extra	Extra	si
34	Extra	Extra	si
35	Extra	Extra	si
36	Extra	Extra	si
37	Extra	Extra	si
38	Extra	Extra	si
39	Extra	Extra	si
40	Extra	Extra	si
41	Extra	Extra	si
42	Extra	Extra	si
43	Extra	Extra	si
44	Extra	Extra	si
45	Extra	Extra	si
46	Extra	Extra	si
47	Extra	Extra	si
48	Extra	Extra	si
49	Extra	Extra	si
50	Extra	Extra	si

Fuente: Propia

CONCLUSIONES

- Se ha logrado clasificar las lúcumas estrictamente por su tamaño y peso, empleando la visión artificial y las redes neuronales.
- Se elaboró un algoritmo de procesamiento de imágenes en el software de Matlab en conjunto con la librería de visión artificial OPENCV. El algoritmo es capaz de obtener el tamaño y peso aproximados de una lúcumas, así como su clasificación por calibre o categoría. El tiempo de procesamiento de imagen del algoritmo es de aproximadamente 1 segundo. La clasificación de lúcumas logro obtener un porcentaje de acierto de 94% para la clasificación por calibre y de 98% para la clasificación por categoría de un total de 50 muestras.
- La red neuronal diseñada es capaz de dar un valor aproximado del peso teniendo como entradas los valores de diámetro y área. La red neuronal logra cumplir los requerimientos de tolerancia estipulados en la norma CODEX STAN 305R-2011.
- El desarrollo de una interfaz de usuario nos permite visualizar y controlar el proceso, así como observar los valores de peso, diámetro y su clasificación correspondiente de cada lúcumas.
- Se diseñó e implemento un prototipo el cual permitió realizar las pruebas de funcionamiento necesarias para la validación del algoritmo desarrollado y determinar que mejoras realizar para su implementación industrial. La implementación de un módulo de captura permite una iluminación uniforme, reducción de sombras y maximizar el contraste.

RECOMENDACIONES

- Como trabajo futuro se sugiere implementar una cámara adicional para obtener la altura de la lúcumá para un mejor entrenamiento de la red neuronal y reducir el error del peso.
- Se debe tener en cuenta que cada producto agrícola tiene diferentes formas, textura, tamaños y colores, estas características influyen directamente en la obtención del peso. También se debe considerar que, a menor uniformidad del producto, será necesario un mayor número de características para un correcto entrenamiento de la red neuronal.
- La aplicación de distintos componentes como lo son actuadores neumáticos para la distribución en la faja transportadora o un controlador más robusto como un PLC mejorarían considerablemente la eficiencia del sistema y son aconsejables para una posible implementación industrial.
- Como trabajo futuro se propone la detección de defectos superficiales, así como la determinación del estado de madurez del producto. Para casos en que sea necesario un análisis o inspección de color más profundo, un cambio a un modelo de color HSV o Lab permite una mayor facilidad para diferenciar los distintos valores de color y tono.
- Un dispositivo de adquisición de imagen de mayor resolución producirá datos más exactos, pero en consecuencia tendrá un coste computacional mucho mayor, es necesario determinar si el nuevo tiempo de procesamiento de imagen cumplirá los requerimientos de producción planteados.

BIBLIOGRAFÍA

- Aguilar, D. (2008). Jerarquía de las redes neuronales [Digital image]. Retrieved from http://www.revistasbolivianas.org.bo/scielo.php?script=sci_arttext&pid=S1997-40442008000200022&lng=es&nrm=iso
- Arévalo, M.; Gonzáles, J. & Ambrosio, G. (2004). La Librería de Visión Artificial OpenCV Aplicación a la Docencia e Investigación. Dpto. De Ingeniería de Sistemas y Automática, Universidad de Málaga. España.
- Arias, M. (2017). Que no te engañen tus ojos: toda la verdad sobre los FPS. Publicados en tecnología.
- Atencio, P. (2009). Desarrollo de un Sistema de Visión Artificial para el Control de Calidad del Mango de azúcar de exportación. Tesis de Pregrado. Universidad de Magdalena. Santa Marta, Colombia.
- Basogain, X. (2008). *Redes neuronales artificiales y sus aplicaciones* [Ebook] (pp. 32-34). UPV/EHU. Retrieved from https://ocw.ehu.eus/pluginfile.php/9047/mod_resource/content/1/redes_neuro/contenidos/pdf/libro-del-curso.pdf
- Barrios, C. (2016). Arquitectura de una Red Neuronal Artificial del tipo Perceptron Multicapa.]. Retrieved from https://www.researchgate.net/publication/31752441_5_Zonificacion_agroecologica_para_el_cultivo_de_arroz_de_riego_Oryza_Sativa_L_en_Colombia
- Blake, A. (1999). Introduction to Active Contours and Visual Dynamics. Visual Dynamics Group, Dept. Engineering Science, University of Oxford.
- Blasco, J., Aleixos, N. & Moltó, E. (2007). Computer vision detection of peel defects in citrus by means of a region oriented segmentation algorithm. *Journal of Food Engineering* 81, 535-543.
- Brack. A. (1999). Diccionario Enciclopédico de Plantas Útiles del Perú. Cusco. CBC, Pág. 222
- Cárdenas, A.; Díaz, R. & Tirado, M. (2004). Visión por Computador Lessing. Monografía del Minor de Automatización Industrial del Programa de Ingeniería Electrónica. Universidad Tecnológica de Bolívar. Facultad de Ingeniería Electrónica. Cartagena D. T. y C.
- Castellanos, L. (2104). Sistemas Operativos. Revisado en: <https://lcsistemasoperativos.wordpress.com/2015/02/02/02-04-complemento-graphical-user-interface-interfaz-grafica-de-usuario/>.

- Castillo, J. (2015). Sistema de Visión Artificial Humanoide para reconocimiento de formas y patrones de objetos, aplicando Redes Neuronales y Algoritmos de Aprendizaje Automático. Tesis de Pregrado. Universidad Ricardo Palma. Lima, Perú.
- Castro, C. (1999). Evaluación de algunos índices de madurez para la cosecha en lúcuma (Pouterialucuma) cv. Lucuna B-2 en el valle de Chancay Huaral. Tesis Ing. Agrónomo. Lima, PE, UNALM. 139 p.
- Corona Ramírez, L., Abarca Jiménez, G., & Mares Carreño, J. (2014). *Sensores y actuadores: Aplicaciones con Arduino* (1st ed., pp. 142-143). Mexico D.F.: Patria.
- Del Castillo, E. (2018). Desarrollo de un Sistema de Visión Artificial para realizar una Clasificación uniforme de limones. Tesis de Pregrado. Universidad Privada del Norte. Trujillo, Perú.
- Detectar múltiples colores con OpenCV y Python [Digital image]. (2016, May 18). Retrieved April 25, 2019, from <https://robologs.net/2016/05/18/detectar-multiples-colores-con-opencv-y-python/>
- Diagrama de bloques de las etapas de un sistema de Visión artificial [Digital image]. (n.d.). Retrieved April 25, 2019, from <http://www.roboticaeducativa.org/mod/resource/view.php?id=2051>
- Díaz, R., Gil, L., Serrano, C., Blasco, M., Moltó, E. & Blasco, J. (2004). Comparison of three algorithms in the classification of table olives by means of computer vision. *Journal of Food Engineering* 61, 101-107.
- Dini, I. 2011. Flavonoid glycosides from Pouteriaobovata (R. Br.) fruit flour. *FoodChemistry* 124: 884-888.
- Duque. (2009). Sistema de atención visual para la detección de puntos topológicos de referencia - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Figura-5-Eschema-de-un-sistema-de-vision-artificial_fig5_40804360 [accessed 2 May, 2019]
- Erazo, S; Escobar, A; Olaeta, J; Undurraga, P. (1999). Determinación proximal y carotenoides totales de frutos de seis selecciones de lúcuma (Pouteria lúcuma). *Alimentos* 24(1): 67-75.
- Esqueda, J. (2002). Matlab e Interfaces Gráficas. Evento: CONATEC 2002 Instituto Tecnológico De Ciudad Madero. Universidad Autónoma de Baja California, Unidad Tijuana.
- Floria, A. (2001). *Manual de Técnicas para el Diseño Participativo de Interfaces de Usuario de Sistemas basados en Software y Hardware* [Ebook] (pp. 83-84). Zaragoza.

Retrieved from http://www.disenomovil.mobi/multimedia_un/trabajo_final/01_que_es_el_prototipado.pdf

- Fuentealba, C; Gálvez, L; Cobos, A; Olaeta, J; Defilippi, B; Chirinos, R; Campos, D; Pedreschi, R. (2016). Characterization of main primary and secondary metabolites and in vitro antioxidant and antihyperglycemic properties in the mesocarp of three biotypes of *Pouterialucuma*. *FoodChemistry* 190: 403-411.
- García, A. (2013). Arduino Mega: Características, Capacidades y donde conseguirlo en Panamá. Revisado en: <http://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/>.
- García, A. (2016). ¿Qué es y cómo funciona un servomotor? | Panama Hitek. Retrieved 7 May 2019, from <http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>
- Glorio, P; Repo-Carrasco, R; Velezmoro, C; Anticona, S; Huaranga, R; Martínez, P; Melgarejo, S; Astuhuaman, L; Huamán, N; Icochea, J; Peña, J. 2008. Fibra dietaria en variedades peruanas de frutas, tubérculos, cereales y leguminosas. *Revista de la Sociedad Química del Perú* 74(1): 46-56.
- Gonzáles, A.; Martínez, F.; Pernía, A.; Alba, F.; Castejón, M.; Ordieres, J. & Vergara, E. (2006). Técnicas y Algoritmos básicos de Visión Artificial. Servicios de Publicaciones. Universidad de Rioja.
- Guevara A., Jornales F., Alpha V. y Alvarez F. (2006) Técnicas y Algoritmos Básicos de Visión Artificial. Universidad de la Roja. España
- Haykin, S. (1994), *Neural Networks: A comprehensive foundation*, NY, McMillan, P.2
- Hidalgo. (2014, September 05). Detección de objetos [Digital image]. Retrieved April 26, 2019, from <https://www.whatsnews.com/2014/09/05/la-red-neuronal-del-equipo-de-googlenet-gana-una-de-las-tres-pistas-del-desafio-de-imagenet/>
- Isasi, P. & Galván, I. (2004). *Redes de neuronas artificiales. Un enfoque práctico*. Madrid: Pearson educación, S.A.
- ITE. (n.d.). Mecanismo de la visión humana [Digital image]. Retrieved May 2, 2019, from http://www.ite.educacion.es/formacion/materiales/129/cd/unidad_1/mo1_mecanismo_de_la_vision.htm
- Jo, R. (2016). *Diseño Mecánico-Eléctrico de una Máquina Mecatrónica para seleccionar Uvas de Mesa*. Tesis de Posgrado. Pontificia Universidad Católica del Perú. Lima, Perú.
- Jordan, M. 1996. *Pouteria species in Biotechnology in Agriculture and Forestry*. Vol. 35. Springer. pp. 291-307.

- Klette, R. (2014). *Concise Computer Vision: An Introduction into Theory and Algorithms*. Springer. London.
- La Serna, N., & Román, U. (2009). *Técnicas de Segmentación en Procesamiento Digital de Imágenes*. Retrieved 21 May 2019, from http://200.62.146.19/BibVirtual/publicaciones/risi/2009_n2/v6n2/a02v6n2.pdf
- Li, Q., Wang, M. & Gu, W. (2002). Computer vision based system for apple surface defect detection. *Computers and Electronics in Agriculture* 36, 215-223.
- Ling, Y. & Hurlbert, A. (2008). Role of color memory in successive color constancy," *Journal of the Optical Society of America A*, vol. 25, no. 6, pp. 1215-1226.
- Maduell, E. (2012). *Visión artificial*. [ebook] Cataluña: FUOC. Fundació para la Universitat Oberta de Catalunya, pp.5-6. Disponible en: <https://openlibra.com/es/book/download/vision-artificial> [Accessed 3 May 2019].
- Martínez Sánchez, V. (2001). *Automatización industrial moderna* (pp. 135-136), Alfaomega, Mexico.
- MINAGRI (Ministerio de Agricultura y Riego, PE). 2015. Ficha técnica: Lúcumo.
- Muñoz, M. 1987. Nomenclatura del lúcumo cultivado en Chile. *Agricultura Técnica (Chile)* 47(4): 416-418.
- Nalwa. *A Guided Tour to Computer Vision*. Addison-Wesley, 1993.
- Navarrete, J., & Odar, C. (2015). *Efecto de la cantidad de pulpa fresca y la temperatura del aire de secado en la obtención de harina de lúcumo (Poutería Obovata)* (Profesional). Pedro Ruiz Gallo.
- Oyama, P. (2014). Método para la clasificación de granos de café por imágenes digitales para uso de atributos seleccionados de morfología, color y textura, Tesis de maestría, Universidad de Sao Paulo-Brasil.
- Palomino, J. (2014). *Software Automático de Reconocimiento de Patrones para la Discriminación de Granos de Arroz Pulido basado en procesamiento digital de imágenes*. Tesis de Pregrado. Universidad Nacional José María Arguedas. Andahuaylas, Perú.
- Pernia, M. (2011). Retrieved from https://www.researchgate.net/profile/Marino_Pernia/publication/235752028_Motores_Monofasicos-conceptos_basicos/links/02bfe5131e98cea5f8000000.pdf.
- Platero, C. (2013). *Introducción a la Visión Artificial*. Dpto. Electrónica, Automática e Informática Industrial.

- Porras, J. Clasificación System Based on Computer Vision. Curso: CE 1002 Taller de Electrónica . Escuela Profesional de Ingeniería Electrónica. Universidad Ricardo Palma.
- Ramírez, J. (2017). Reconocimiento de granos de café verde Arabiga sin defectos físicos en muestras usando visión artificial. Tesis presentada para optar el título de ingeniero de sistemas. Universidad Nacional de San Agustín de Arequipa Facultad de ingeniería de producción y servicios. Arequipa -Perú.
- Rayo, J. A., y Chimbo, J. A. (2013). Utilización de la minicomputadora Raspberry Pi para la adquisición y evaluación de datos de consumo de energía eléctrica de equipos a 220 voltios. Tesina de Seminario previa a la obtención del Título de Ingeniero en Electrónica y Telecomunicaciones, Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador.
- Rojas, P. (2014). Regeneración de patrones PR. Revisado en: <https://prezi.com/tjqw3s8z5gcf/regeneracion-de-patrones/>
- Romero, L. & Calonge, T. Redes Neuronales y Reconocimiento de Patrones. Dpto. de Informática y Automática. Universidad de Salamanca. España.
- Rudas, C. (2015). Diseño de un Sistema Mecatrónico para la Selección de Arándanos. Tesis de Posgrado. Pontificia Universidad Católica del Perú. Lima, Perú.
- Ruiz & Basualdo (2001). Redes Neuronales: Conceptos Básicos y Aplicaciones. Universidad Tecnológica Nacional – Facultad Regional Rosario. Argentina.
- Rumelhart, D. E., Hinton, G. E. & Williams, R. J. in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol. 1: *Foundations* (eds Rumelhart, D. E. & McClelland, J. L.) 320–360 (MIT, Cambridge, 1986).
- Sabareesaan, J.; Janardhana, G. & Jaya, J. (2013). Computer aided inspection system for food products using machine vision. A review. 2013 International Conference on Current Trends in Engineering and Technology (ICCTET), pages 29–33. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6675906>
- Salazar, M. (2018). Lúcura, el superalimento peruano. Retrieved 9 June 2019, from <http://www.gastronosfera.com/es/tendencias/lucuma-el-superalimento-peruano>
- San Miguel, J. (2016). *Desarrollo con matlab de una red neuronal para estimar la demanda de energía eléctrica* (Master). Escuela de ingenierías industriales universidad de valladolid.
- Sobrado, E. (2003). Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot. Tesis para optar el grado de magister en ingeniería

de control y automatización. Pontificia Universidad Católica. Escuela de graduados.
Lima -Perú.

- Técnicas de filtrado (s.f.). Recuperado de <https://www.um.es/geograf/sigmur/teledet/tema06.pdf>
- Tutorial de Arduino y sensor ultrasónico HC-SR04. Disponible en: https://naylampmechatronics.com/blog/10_tutorial-arduino-y-sensor-ultrasonico-hc-sr04.html.
- Valdivia, C. (2016). Diseño de un Sistema de Visión Artificial para la Clasificación de Chirimoyas basado en medidas. Tesis de Posgrado. Pontificia Universidad Católica del Perú. Lima, Perú.
- Vargas, V. (2009). Sistema de Visión Artificial para el Control de Calidad en Pieza Cromadas. Tesis de Posgrado. Escuela Superior de Ingeniería Mecánica y Eléctrica Unidad Azcapotzalco. Instituto Politécnico Nacional. México.
- Viera, G. (2017). Procesamiento de Imágenes usando OPENCV aplicado en Raspberry Pi para la clasificación del Cacao. Tesis de Pregrado. Universidad Nacional de Piura. Piura, Perú.
- Vives, L., Mejía, H. Vilcherrez, K., Vassallo, M. (2014). Visión Artificial: Aplicación de filtros y segmentación en imágenes de hojas de café – Rev. Ingeniería: Ciencia, Tecnología e Innovación VOL 1/N° 2, ISSN: 2313-1926/.
- Yahia, E; Gutiérrez-Orozco, F; Arvizu-de Leon, C. 2011. Phytochemical and antioxidant characterization of mamey (Pouteriasapota) fruit. FoodResearch International 44(7): 2175-2181.
- Young, E., & Joseph, H. (2010). A Feed-forward network with backpropagation learning [Digital image]. Retrieved from https://www.researchgate.net/publication/221983860_Multi-structure_Segmentation_of_Multi-modal_Brain_Images_using_Artificial_Neural_Networks
- Z-laser. (2018). Láser industrial para posicionamiento, alineación y visión artificial [Digital image]. Retrieved May 1, 2019, from <https://www.z-laser.es/aplicaciones/robotica.html>

ANEXO 1: Ficha técnica HD Webcam Logitech c525
Fuente: Logitech




Logitech®
HD Webcam C525

For portable HD video calling and recording – with autofocus

- HD 720p video calling and recording plus autofocus – on most major IM applications and Logitech Vid™ HD
- 1-click Facebook™, Twitter™ and YouTube™ HD uploading
- Fold-and-go plus swivel design
- 8-megapixel photos (software enhanced)
- Logitech® Fluid Crystal™ Technology

The Logitech® HD Webcam C525. For portable HD video calling and recording – with autofocus. Stay close to the people you care about in smooth HD 720p with autofocus – on your favourite IM or Logitech Vid™ HD – thanks to Logitech More HD. Upload your latest clip to Facebook™, Twitter™ or YouTube™ with just one click. Logitech Video Effects™ and the included MAGIX™ photo and video editor help you look good. The fold-and-go plus swivel design makes it easy to take it with you. You can send your friends and family a dazzling 8MP pic. And it includes Logitech Fluid Crystal™ Technology – delivering smoother video, sharper pictures, richer colours and clearer sound in real-world conditions (requires installation of included software).



Logitech® Logitech® HD Webcam C525

Logitech

Package specifications

	Primary pack	Master shipper carton
Part # WER	960-000721	n/a
Bar Code	5099206027121 (EAN-13)	50992060271214 (SCC-14)
Part # Central	960-000722	n/a
Bar Code	5099206027138 (EAN-13)	50992060271313 (SCC-14)
Part # EER	960-000723	n/a
Bar Code	5099206027145 (EAN-13)	50992060271412 (SCC-14)
Weight	226.8 g	2.147 kg
Length	20.95 cm	31.50 cm
Width	7.62 cm	31.60 cm
Height / depth	15.24 cm	22.40 cm
Volume	2.433 dm ³	0.02230 m ³
1 primary pack	1	n/a
1 intermediate pack	0	n/a
1 master shipper carton	8	1
1 pallet EURO	432	54
1 container 20ft	10080	1260

Technical Specifications

- HD video calling (1280 x 720 pixels) with recommended system
- HD video capture: up to 1280 x 720 pixels
- Logitech Fluid Crystal™ Technology
- Logitech More HD technology
- Autofocus
- Photos: up to 8 megapixels (software enhanced)
- Built-in microphone with Logitech RightSound™ technology
- Hi-Speed USB 2.0 certified (recommended)
- Universal clip fits laptops, LCD or CRT monitors

Logitech webcam software:

- Logitech Vid™ HD

- Logitech RightLight™ 2 technology
- Video and photo capture
- MAGIX™ photo and video editing software
- 1-click Facebook™, Twitter™ and YouTube™ HD upload (registration required)
- Logitech Video Effects™: fun filters, avatars, face accessories, video masks and mask maker*

Works with most instant messaging applications

* Intel® Pentium® 4 (2.8 GHz) recommended.

Package Contents

- Webcam with 1.5-m (5-foot) cable
- Logitech webcam software with Logitech Vid™ HD
- User documentation

2-year manufacturer's guarantee with full product support

System Requirements

- Windows® XP (SP2 or higher), Windows Vista® or Windows® 7 (32-bit or 64-bit)

Basic requirements:

- 1 GHz
- 512 MB RAM
- 200 MB hard drive space
- Internet connection
- USB 1.1 port (2.0 recommended)

For HD 720p video calling on Logitech Vid™ HD and HD video sharing on Facebook™/Twitter™/YouTube™:

- 2.4 GHz Intel® Core™2 Duo
- 2 GB RAM
- 200 MB hard drive space
- USB 2.0 port
- 1 Mbps upload speed or higher
- 1280 x 720 screen resolution

For other video-calling clients, please check the recommended requirements.



ANEXO 2: Registro de características de 50 muestras de lúcumas en pixeles

Fuente: Elaboración propia

Muestra	Área	Centroide		Bounding box				Largo	Ancho	Orientación	Área convexa	Diámetro equivalente	Extensión	Perímetro
		X	Y	X1	X2	Y1	Y2							
1	224717	467.42	386.05	195.5	125.5	546.0	526.0	545.046	525.114	13.154	225623	534.90052	0.782	1,712.962
2	187857	397.60	375.00	154.5	126.5	496.0	493.0	497.776	481.005	-38.859	188466	489.06744	0.768	1,541.542
3	195825	443.68	424.73	194.5	173.5	499.0	504.0	503.928	494.859	89.212	196796	499.33169	0.778	1,592.372
4	170527	439.44	421.08	208.5	178.5	456.0	485.0	482.185	450.674	85.741	171299	465.96322	0.771	1,476.696
5	295534	380.85	388.48	675.0	855.0	630.0	611.0	624.355	602.946	-80.728	297673	613.42121	0.767	1,981.816
6	137270	442.37	367.47	225.5	163.5	433.0	406.0	440.947	396.492	-23.204	137882	418.06410	0.780	1,331.356
7	127443	472.64	421.94	270.5	216.5	395.0	410.0	419.587	387.201	-60.781	128093	402.82188	0.786	1,281.832
8	119665	463.05	401.94	270.5	199.5	386.0	404.0	399.266	381.761	-87.570	120290	390.33602	0.767	1,238.422
9	105592	459.84	435.02	275.5	249.5	368.0	371.0	373.108	360.518	61.001	106334	366.66594	0.773	1,173.868
10	97816	465.43	359.13	282.5	187.5	366.0	345.0	363.172	343.110	-7.169	98431	352.90679	0.774	1,130.192
11	283481	416.33	373.94	975.0	885.0	635.0	572.0	632.995	570.340	-9.164	285352	600.78217	0.780	1,939.066
12	235689	396.51	357.83	116.5	895.0	564.0	539.0	558.236	537.737	4.372	237280	547.80339	0.775	1,769.550
13	238313	447.02	332.45	160.5	625.0	574.0	537.0	569.026	533.337	2.569	239973	550.84438	0.773	1,781.600
14	220916	449.09	324.73	178.5	605.0	542.0	534.0	542.683	518.565	28.357	222623	530.35741	0.763	1,718.724
15	213167	490.78	393.68	229.5	126.5	523.0	535.0	528.333	513.805	-53.896	215186	520.97280	0.761	1,690.688
16	200795	462.77	392.70	204.5	140.5	514.0	507.0	509.675	501.690	3.208	202399	505.62845	0.770	1,634.134
17	195370	438.69	396.36	192.5	139.5	490.0	518.0	513.594	484.509	-74.512	197045	498.75125	0.769	1,611.540
18	191196	443.00	444.82	191.5	197.5	506.0	498.0	496.364	490.540	-17.648	193232	493.39468	0.758	1,612.686
19	169613	479.31	383.27	239.5	151.5	479.0	465.0	471.343	458.320	26.671	171068	464.71279	0.761	1,502.994
20	141578	493.83	343.42	285.5	124.5	418.0	442.0	436.614	412.986	86.771	142820	424.57356	0.766	1,370.230
21	116345	512.78	363.50	319.5	166.5	387.0	399.0	390.438	379.615	73.967	117367	384.88317	0.753	1,243.166
22	82810	458.95	349.38	289.5	188.5	342.0	326.0	331.278	318.633	7.738	83954	324.71059	0.742	1,064.194
23	213051	424.23	400.02	159.5	140.5	525.0	516.0	527.295	514.600	25.101	213922	520.83103	0.786	1,653.198
24	148980	450.61	365.84	229.5	146.5	443.0	441.0	443.026	428.514	57.122	150083	435.53097	0.762	1,395.784
25	148727	439.85	381.48	216.5	163.5	442.0	437.0	441.291	429.293	-28.193	149538	435.16100	0.769	1,381.606

Muestra	Área	Centroide		Bounding box				Largo	Ancho	Orientación	Área convexa	Diámetro equivalente	Extensión	Perímetro
26	134973	404.45	445.28	196.5	240.5	418.0	415.0	419.016	410.380	40.415	135718	414.55152	0.778	1,326.744
27	117508	626.35	394.25	424.5	199.5	400.0	391.0	391.004	382.881	-1.359	119092	386.80206	0.751	1,276.882
28	131280	498.30	404.79	290.5	198.5	417.0	408.0	418.822	399.198	-31.201	132739	408.84091	0.771	1,331.196
29	115311	461.81	457.02	277.5	255.5	373.0	402.0	399.451	367.706	-82.738	116387	383.16906	0.769	1,238.866
30	161964	435.70	433.35	201.5	206.5	467.0	448.0	470.177	439.133	19.879	163676	454.11339	0.774	1,472.666
31	200446	415.27	388.00	167.5	124.5	502.0	517.0	517.369	493.557	-62.632	202157	505.18885	0.772	1,644.242
32	72806	445.15	481.86	285.5	329.5	316.0	308.0	307.547	301.916	-22.822	73402	304.46589	0.748	9,725.220
33	168597	437.69	399.53	203.5	167.5	473.0	462.0	469.032	457.750	27.442	169920	463.31886	0.771	1,492.348
34	169319	505.53	423.74	269.5	182.5	469.0	478.0	484.022	445.619	58.085	170473	464.30986	0.755	1,491.232
35	177140	512.63	414.13	268.5	177.5	488.0	474.0	480.364	469.615	5.919	178175	474.91226	0.765	1,513.592
36	216525	551.15	412.01	297.5	141.5	514.0	538.0	539.336	511.390	-72.925	217726	525.06018	0.783	1,671.476
37	119682	465.66	411.65	270.5	212.5	392.0	397.0	393.806	387.040	59.857	120557	390.36375	0.769	1,255.554
38	122604	388.16	405.51	193.5	206.5	395.0	400.0	398.626	391.852	-63.454	123168	395.10032	0.775	1,255.038
39	122344	469.44	362.36	265.5	162.5	400.0	400.0	395.329	394.337	-20.304	122875	394.68116	0.764	1,249.182
40	126625	345.93	435.40	141.5	242.5	409.0	395.0	405.857	397.564	10.152	127554	401.52703	0.783	1,284.398
41	143014	359.28	459.47	143.5	242.5	426.0	433.0	431.103	422.649	-66.647	143973	426.72131	0.775	1,367.904
42	143449	431.15	412.43	215.5	192.5	428.0	436.0	435.557	419.593	63.775	144212	427.36979	0.768	1,361.204
43	153715	406.51	462.98	184.5	239.5	447.0	446.0	447.166	437.823	61.694	154555	442.39803	0.771	1,406.806
44	174551	509.65	384.13	271.5	149.5	475.0	468.0	480.571	462.566	40.813	175602	471.42893	0.785	1,505.784
45	178721	509.13	417.94	261.5	180.5	495.0	476.0	486.557	468.139	13.319	179414	477.02688	0.758	1,526.110
46	185016	441.84	458.87	197.5	216.5	486.0	495.0	490.688	480.603	-81.278	185964	485.35522	0.769	1,540.858
47	209133	407.01	397.21	144.5	135.5	517.0	522.0	519.623	512.755	74.766	210298	516.01977	0.774	1,643.624
48	118109	378.67	413.89	178.5	216.5	400.0	390.0	388.980	386.843	80.325	119843	387.78996	0.757	1,274.904
49	151596	433.99	440.08	209.5	219.5	448.0	438.0	449.709	429.369	35.987	152941	439.33816	0.772	1,422.334
50	184451	468.34	400.56	216.5	163.5	499.0	473.0	496.053	473.787	-34.256	185629	484.61356	0.781	1,548.110

ANEXO 3: Código elaborado en Matlab

Fuente: Elaboración Propia

```
function varargout = sonico_servos_guide(varargin)
% SONICO_SERVOS_GUIDE MATLAB code for sonico_servos_guide.fig
% SONICO_SERVOS_GUIDE, by itself, creates a new SONICO_SERVOS_GUIDE or raises the existing
% singleton*.

% H = SONICO_SERVOS_GUIDE returns the handle to a new SONICO_SERVOS_GUIDE or the handle to
% the existing singleton*.
%
% SONICO_SERVOS_GUIDE('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in SONICO_SERVOS_GUIDE.M with the given input arguments.
%
% SONICO_SERVOS_GUIDE('Property','Value',...) creates a new SONICO_SERVOS_GUIDE or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before sonico_servos_guide_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to sonico_servos_guide_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help sonico_servos_guide
% Last Modified by GUIDE v2.5 25-Mar-2019 12:24:02
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @sonico_servos_guide_OpeningFcn, ...
    'gui_OutputFcn', @sonico_servos_guide_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before sonico_servos_guide is made visible.
function sonico_servos_guide_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to sonico_servos_guide (see VARARGIN)

% Choose default command line output for sonico_servos_guide
handles.output = hObject;

%closepreview
ah = axes('unit', 'normalized', 'position', [0 0 1 1]);
bg = imread('fondo5.jpg'); imagesc(bg);
```

```

set(ah,'handlevisibility','off','visible','off')
uistack(ah,'bottom');

%%% Pantallas de guide
fondo1=imread('fondo8.jpg');
axes(handles.gui_captura);
imshow(fondo1);
fondo2=imread('fondo9.jpg');
axes(handles.gui_visualizar);
imshow(fondo2);

%clear all port
delete(instrfind({'Port'},{'COM4'}))

global a
global sensor
global buclegeneral
global s3
global s4
global s5
buclegeneral=0;

% Declarando variables arduino y servos
a = arduino('COM4', 'Mega2560', 'Libraries', {'Servo','JRodrigoTech/HCSR04'}); % arduino
s3=servo(a,'D3','MinPulseDuration',5.20e-04,'MaxPulseDuration',2.1e-03); % servo 1
s4=servo(a,'D4','MinPulseDuration',5.20e-04,'MaxPulseDuration',2.1e-03); % servo 2
s5=servo(a,'D5','MinPulseDuration',5.20e-04,'MaxPulseDuration',2.1e-03); % servo 3

writePosition(s3,0.80); %posicion inicial servo 1
writePosition(s4,0.85); %posicion inicial servo 2
writePosition(s5,0.85); %posicion inicial servo 3

% Creando variable del sensor ultrasonico HCR04
% Donde: pin D12 = TRIGGER, pin D13= ECHO,
sensor = addon(a, 'JRodrigoTech/HCSR04', 'D12', 'D13')

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes sonico_servos_guide wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = sonico_servos_guide_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in start.
function start_Callback(hObject, eventdata, handles)

% Declaracion de variables
global sensor %sensor ultrasonico
global buclegeneral % condicionante 1
global s3 %servo 1
global s4 %servo 2
global s5

```

```

global stats_aper
global img_final
global vid
global vidRes
global nBands
global hImage

%Configuracion de la camara
closepreview
%axes(handles.gui_visualizar);
vid = videoinput('winvideo', 1, 'MJPEG_960x720'); % Resolucion de la camara
vid.FramesPerTrigger=1; % Captura una frame por trigger
vid.ReturnedColorspace='rgb'; % Salida de la Imagen RGB
triggerconfig(vid,'manual');
vidRes=get(vid,'VideoResolution');
nBands = get(vid, 'NumberOfBands');
hImage = image(zeros(720, 960, nBands), 'parent', handles.gui_visualizar);
%preview(vid, hImage);

objsen=0;

while buclegeneral==0

while objsen==0
%%%%%%%%%% Sensor ultrasonico HCR04
tsen = readTravelTime(sensor);
sprintf('Current distance is %.4f meters\n', 340*tsen/2)
tclasi=340*tsen*100/2;

pause(0.03) %pausa entre lecturas del sensor ultrasonico

if tclasi<=18 && tsen>=0 && objsen==0 %presencia de objeto en la faja
    objsen=1;
end
end

while objsen==1
pause(0.40) %%tiempo de retardo de toma de foto
foto_luc=getsnapshot(vid); %captura de imagen con la camara

%Captura guardada con el nombre "prueba1"
imwrite(foto_luc,'prueba1','jpg');
%Procesamiento de imagen
img=imread('prueba1'); % Carga de imagen
axes(handles.gui_captura); % Mostrar en el guide la captura
imshow(img); % segundo cuadro guide

%Transformacion a escala de grises
src = cv.cvtColor(img, 'RGB2GRAY');

% Aplicando el metodo de OTSU
%Umbralizacion de Otsu despues de un filtro Gausiano
blur_otsu = cv.GaussianBlur(src);
[th_otsu3,ots3] = cv.threshold(src,'Otsu', 'Max Value',255,'Method','BinaryInv');

%Mejora de señal binaria ( rellenar agujeros, apertura )
acon=imfill(th_otsu3,'hole'); %llenar los espacios vacios
acon=bwareaopen(acon,1000); %eliminar valores pequeños

%Apertura

```

```

st1=strel('diamond', 50);
imfin(:,:,1)=imopen(acon,eval('st1'));

%OBTENCION DE PROPIEDADES DE IMAGEN FINAL
%obtencion de la imagen anterior
img_final=imfin(:,:,1);

%Extraccion de características
stats_aper=regionprops('table',img_final,'Area','BoundingBox','Centroid','EquivDiameter','MajorAxisLength','MinorAxisLength','Perimeter','Orientation');
area_total=(stats_aper.Area(1))

%Escalamiento de diametro segun area
if area_total<150000
    f_d=61.2;
elseif area_total>=150000 && area_total<=200000
    f_d=62.7;
elseif area_total>200000
    f_d=64.2;
end

% Extrayendo datos (Diametro "ancho", Area "N° de pixeles") del regionprop
stats_aper
area_red=(stats_aper.Area(1))/350000; % Acondicionamiento Area
diametro_red=(stats_aper.EquivDiameter(1)/700); % Acondicionamiento Diametro

load RNA2 % Cargando red neuronal

%Ingresa valores de Diametro y area a la red neuronal
peso_aprox=RNA2([area_red; diametro_red])*400; % Peso_aprox=red28([Area de la foto
segmentada])*320(proporcion usada, excel)
diametro_ecu=(diametro_red*700)/f_d; %Diametro aproximado en cm

set(handles.gui_peso,'string',peso_aprox);
set(handles.gui_diametro,'string',diametro_ecu);

%Clasificacion segun calibre para lucumas
if peso_aprox<=50
    fprintf('CALIBRE: A --- PESO APROX: %d gr. \n', peso_aprox)
    set(handles.guicalibre,'string','A');
    pause(2.1);
    writePosition(s3,0.25);
    pause(1);
    writePosition(s3,0.8);
elseif peso_aprox>50 && peso_aprox<=55
    fprintf('CALIBRE: A, B --- PESO APROX: %d gr. \n', peso_aprox)
    set(handles.guicalibre,'string','A, B');
    pause(2.1);
    writePosition(s3,0.25);
    pause(1);
    writePosition(s3,0.8);
elseif peso_aprox>55 && peso_aprox<=120
    fprintf('CALIBRE: B --- PESO APROX: %d gr. \n', peso_aprox)
    set(handles.guicalibre,'string','B');
    pause(2.1);
    writePosition(s3,0.25);
    pause(1);
    writePosition(s3,0.8);
elseif peso_aprox>120 && peso_aprox<=130
    fprintf('CALIBRE: B, C --- PESO APROX: %d gr. \n', peso_aprox)
    set(handles.guicalibre,'string','B, C');

```

```

    pause(1.17);
    writePosition(s4,0.2);
    pause(1);
    writePosition(s4,0.85);
elseif peso_aprox>130 && peso_aprox<=190
    fprintf('CALIBRE: C --- PESO APROX: %d gr. \n', peso_aprox)
    set(handles.guicalibre,'string','C');
    pause(1.17);
    writePosition(s4,0.2);
    pause(1);
    writePosition(s4,0.85);
elseif peso_aprox>190 && peso_aprox<=200
    fprintf('CALIBRE: C, D --- PESO APROX: %d gr. \n', peso_aprox)
    set(handles.guicalibre,'string','C, D');
    pause(1.17);
    writePosition(s4,0.2);
    pause(1);
    writePosition(s4,0.85);
elseif peso_aprox>200 && peso_aprox<=260
    fprintf('CALIBRE: D --- PESO APROX: %d gr. \n', peso_aprox)
    set(handles.guicalibre,'string','D');
    pause(1.17);
    writePosition(s4,0.2);
    pause(1);
    writePosition(s4,0.85);
elseif peso_aprox>260 && peso_aprox<=270
    fprintf('CALIBRE: D, E --- PESO APROX: %dr g. \n', peso_aprox)
    set(handles.guicalibre,'string','D, E');
    pause(0.55);
    writePosition(s5,0.2);
    pause(1);
    writePosition(s5,0.85);
elseif peso_aprox>270 && peso_aprox<=330
    fprintf('CALIBRE: E --- PESO APROX: %d gr. \n', peso_aprox)
    set(handles.guicalibre,'string','E');
    pause(0.55);
    writePosition(s5,0.2);
    pause(1);
    writePosition(s5,0.85);
elseif peso_aprox>330 && peso_aprox<=340
    fprintf('CALIBRE: E, F --- PESO APROX: %d gr. \n', peso_aprox)
    set(handles.guicalibre,'string','E, F');
    pause(0.55);
    writePosition(s5,0.2);
    pause(1);
    writePosition(s5,0.85);
elseif peso_aprox>340
    fprintf('CALIBRE: F --- PESO APROX: %d gr. \n', peso_aprox)
    set(handles.guicalibre,'string','F');
    pause(0.55);
    writePosition(s5,0.2);
    pause(1);
    writePosition(s5,0.85);
end

%Clasificacion segun categoria para lucumas
if peso_aprox>=120 && peso_aprox<300
    fprintf('CATEGORIA: EXTRA --- PESO APROX: %d gr. \n', peso_aprox)
    pause(0.55);
    writePosition(s5,0.2);
    pause(1);

```

```

    writePosition(s5,0.85);
elseif peso_aprox>=50 && peso_aprox<120
    fprintf('CATEGORIA: I --- PESO APROX: %d gr. \n', peso_aprox)
    pause(1.17);
    writePosition(s4,0.2);
    pause(1);
    writePosition(s4,0.85);
elseif peso_aprox<50
    fprintf('CATEGORIA: II --- PESO APROX: %d gr. \n', peso_aprox)
    pause(2.1);
    writePosition(s3,0.25);
    pause(1);
    writePosition(s3,0.8);
elseif diametro_ecu>300
    fprintf('CATEGORIA: RANGO NO VALIDO --- PESO APROX: %d gr. \n', peso_aprox)
end

%CLASIFICACION EMPLEANDO SU CATEGORIA
if peso_aprox<50 && diametro_ecu<4
    set(handles.guicateg,'string','II');
    pause(2.1);
    writePosition(s3,0.25);
    pause(1);
    writePosition(s3,0.8);
elseif peso_aprox<50 && diametro_ecu>=4 && diametro_ecu<6
    set(handles.guicateg,'string','II');
    pause(2.1);
    writePosition(s3,0.25);
    pause(1);
    writePosition(s3,0.8);
elseif peso_aprox>=50 && peso_aprox<120 && diametro_ecu<4
    set(handles.guicateg,'string','II');
    pause(1.17);
    writePosition(s4,0.2);
    pause(1);
    writePosition(s4,0.85);
elseif peso_aprox>=50 && peso_aprox<120 && diametro_ecu>=4 && diametro_ecu<6
    set(handles.guicateg,'string','I');
    pause(1.17);
    writePosition(s4,0.2);
    pause(1);
    writePosition(s4,0.85);
elseif peso_aprox>=50 && peso_aprox<120 && diametro_ecu>=6 && diametro_ecu<9
    set(handles.guicateg,'string','I');
    pause(1.17);
    writePosition(s4,0.2);
    pause(1);
    writePosition(s4,0.85);
elseif peso_aprox>=120 && peso_aprox<300 && diametro_ecu>=4 && diametro_ecu<6
    set(handles.guicateg,'string','I');
    pause(0.55);
    writePosition(s5,0.2);
    pause(1);
    writePosition(s5,0.85);
elseif peso_aprox>=120 && peso_aprox<300 && diametro_ecu>=6 && diametro_ecu<9
    set(handles.guicateg,'string','Extra');
    pause(0.55);
    writePosition(s5,0.2);
    pause(1);
    writePosition(s5,0.85);
elseif peso_aprox>=300 || diametro_ecu>=9

```

```

    set(handles.guicateg,'string','T');
    pause(0.55);
    writePosition(s5,0.2);
    pause(1);
    writePosition(s5,0.85);
end
objsen=0;
end
end

% --- Executes on button press in stop.
function stop_Callback(hObject, eventdata, handles)
% hObject handle to stop (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
closepreview
axes(handles.gui_visualizar);
fondo1=imread('fondo8.jpg');
axes(handles.gui_visualizar);
imshow(fondo1);
fondo2=imread('fondo9.jpg');
axes(handles.gui_captura);
imshow(fondo2);

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject handle to popupmenu2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

pop_selec=get(handles.popupmenu2,'Value');
if pop_selec==1
    set(handles.text_calibre,'visible','off');
    set(handles.guicalibre,'visible','off');
    set(handles.text_categoria,'visible','off');
    set(handles.guicateg,'visible','off');
elseif pop_selec==2
    set(handles.text_calibre,'visible','on');
    set(handles.guicalibre,'visible','on');
    set(handles.text_categoria,'visible','on');
    set(handles.guicateg,'visible','on');
elseif pop_selec==3
    set(handles.text_calibre,'visible','on');
    set(handles.guicalibre,'visible','on');
    set(handles.text_categoria,'visible','off');
    set(handles.guicateg,'visible','off');
elseif pop_selec==4
    set(handles.text_calibre,'visible','off');
    set(handles.guicalibre,'visible','off');
    set(handles.text_categoria,'visible','on');
    set(handles.guicateg,'visible','on');
end

% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject handle to popupmenu2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.

```

```

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes during object creation, after setting all properties.
function text29_CreateFcn(hObject, eventdata, handles)
% hObject handle to text29 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

global buclegeneral
buclegeneral=1;
clear all port, close all force
delete(instrfind({'Port'},{'COM4'}))
close(sonico_servos_guide);

% --- Executes during object creation, after setting all properties.
function text_calibre_CreateFcn(hObject, eventdata, handles)
% hObject handle to text_calibre (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject handle to popupmenu1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

%Declaracion de variables
global stats_aper
global vid
global hImage
global nBands
global img_final

pop_selec=get(handles.popupmenu1,'Value');

if pop_selec==1
closepreview
fondo2=imread('fondo9.jpg');
axes(handles.gui_visualizar);
imshow(fondo2);

elseif pop_selec==2
closepreview
fondo2=imread('fondo9.jpg');
axes(handles.gui_visualizar);
imshow(fondo2);

elseif pop_selec==3
    preview(vid, hImage);

elseif pop_selec==4
axes(handles.gui_visualizar);
imshow(img_final);
%Grafica de las características de regionprops

centro=stats_aper.Centroid;
diametro=stats_aper.EquivDiameter;
radio=diametro/2;
boundingbox=stats_aper.BoundingBox;

```

```

hold on
viscircles(centro,radio);
plot(stats_aper.Centroid(:,1),stats_aper.Centroid(:,2),'*r');

for k=1:size(boundingBox,1)
    rectangle('position',boundingbox(k,:),'EdgeColor','g')
end
hold off

%GRAFICA DE LONGITUD MAYOR Y DIAMETRO SOBRE LA IMAGEN
s = regionprops(img_final, 'Orientation', 'MajorAxisLength', 'MinorAxisLength', 'Eccentricity', 'Centroid');
hold on
phi = linspace(0,2*pi,50);
cosphi = cos(phi);
sinphi = sin(phi);
for k = 1:length(s)
    xbar = s(k).Centroid(1);
    ybar = s(k).Centroid(2);
    a = s(k).MajorAxisLength/2;
    b = s(k).MinorAxisLength/2;
    theta = pi*s(k).Orientation/180;
    R = [ cos(theta) sin(theta)
        -sin(theta) cos(theta)];
    xy = [a*cosphi; b*sinphi];
    xy = R*xy;
    x = xy(1,:) + xbar;
    y = xy(2,:) + ybar;
    plot(x,y,'r','LineWidth',2);
    hold on
    % Major Axis Values
    xMajor1 = xbar + (((s(k).MajorAxisLength)/2) * cosd(s(k).Orientation));
    yMajor1 = ybar - (((s(k).MajorAxisLength)/2) * sind(s(k).Orientation));
    xMajor2 = xbar - (((s(k).MajorAxisLength)/2) * cosd(s(k).Orientation));
    yMajor2 = ybar + (((s(k).MajorAxisLength)/2) * sind(s(k).Orientation));
    line([xMajor1 xMajor2],[yMajor1 yMajor2],'color','g')
    % Minor Axis Values
    xMinor1 = xbar + (((s(k).MinorAxisLength)/2) * sind((180-s(k).Orientation)));
    yMinor1 = ybar - (((s(k).MinorAxisLength)/2) * cosd((180-s(k).Orientation)));
    xMinor2 = xbar - (((s(k).MinorAxisLength)/2) * sind((180-s(k).Orientation)));
    yMinor2 = ybar + (((s(k).MinorAxisLength)/2) * cosd((180-s(k).Orientation)));
    line([xMinor1 xMinor2],[yMinor1 yMinor2],'color','c')
end
hold off
end

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject handle to popupmenu1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

ANEXO 4: NORMA REGIONAL PARA LA LÚCUMA

Fuente: FAO

NORMA REGIONAL PARA LA LÚCUMA (CODEX STAN 305R - 2011)

1. DEFINICIÓN DEL PRODUCTO

Esta Norma se aplica a las frutas de las variedades comerciales de lúcuma, obtenidas de la especie *Pouteria lucuma* (R. y P.), de la familia *Sapotaceae*, que habrán de suministrarse frescas al consumidor, después de su acondicionamiento y envasado. Se excluye la lúcuma destinada a la elaboración industrial.

2. DISPOSICIONES RELATIVAS A LA CALIDAD

2.1 REQUISITOS MÍNIMOS

En todas las categorías, a reserva de las disposiciones especiales para cada categoría, y las tolerancias permitidas, la lúcuma deberá estar:

- entera, y preferentemente manteniendo la base del pedúnculo que siempre va adherida al cáliz; de contar con el pedúnculo, este no deberá ser mayor a 5 mm, cortado limpiamente;
- sana, deberán excluirse los productos afectados por podredumbre, moho o deterioro que hagan que no sean aptos para el consumo;
- limpia, y prácticamente exenta de cualquier materia extraña visible;
- • prácticamente exenta de plagas, y daños causados por ellas, que afecten el aspecto general del producto;
- exenta de humedad externa anormal, salvo la condensación consiguiente a su remoción de una cámara frigorífica;
- exenta de cualquier olor y/o sabor extraño;
- exenta de daños causados por bajas y/o altatemperaturas;
- exenta de daños físicos tales como: magulladuras profundas, cortes, golpes etc. Se aceptan en el momento de comercialización rajaduras naturales superficiales causadas por la maduración;
- • exenta de quemaduras de sol.

Asimismo deberá presentar un aspecto fresco y la forma y color externo característico de la variedad.

2.1.1 La lúcuma deberá haberse recolectado cuidadosamente, y haber alcanzado un grado apropiado de desarrollo y madurez fisiológica, de conformidad con las características de la variedad, el momento de cosecha, y la zona en que se produce.

El desarrollo y condición de la lúcuma deberán ser tales que le permita:

- soportar el transporte y la manipulación; y
- llegar en estado satisfactorio al lugar de destino.

2.2 CLASIFICACIÓN

La lúcuma se clasifica en tres categorías, según se definen a continuación:

2.2.1 Categoría “Extra”

La lúcuma de esta categoría deberá ser de calidad superior. Deberá presentar la forma, desarrollo y coloración característica de la variedad, teniendo en cuenta la zona de producción. No deberá tener defectos, salvo defectos superficiales muy leves siempre y cuando no afecten el aspecto general del producto, su calidad, estado de conservación, y presentación en el envase. En ningún caso, los defectos deberán afectar la pulpa del producto.

El número máximo de semillas en esta categoría es de 2, la firmeza de la lúcuma tipo “seda” será blanda y para el tipo “palo” será semidura.

2.2.2 Categoría I

La lúcuma de esta categoría deberá ser de buena calidad. Deberá presentar la forma, desarrollo y coloración característica de la variedad, teniendo en cuenta la zona de producción. Podrán permitirse, sin embargo, los siguientes defectos leves¹, siempre y cuando no afecten al aspecto general del producto, su calidad, estado de conservación, y presentación en el envase:

- defectos leves de forma que no correspondan a la variedad, defectos leves en la piel que no superen 1 cm² para todos los calibres delúcuma.

En ningún caso, los defectos deberán afectar la pulpa del producto.

El número máximo de semillas en esta categoría es de 3, la firmeza de la lúcuma tipo “seda” será blanda y para el tipo “palo” será semidura.

2.2.3 Categoría II

Esta categoría comprende la lúcuma que no puede clasificarse en las categorías superiores, pero satisface los requisitos mínimos especificados en la Sección 2.1. Podrán permitirse, sin embargo, los siguientes defectos, siempre y cuando la lúcuma conserve sus características esenciales en lo que respecta a su calidad, estado de conservación y presentación:

- defectos de forma que no correspondan a la variedad, defectos leves en la piel que en total no superen 2 cm² para todos los calibres delúcuma.

El número máximo de semillas en esta categoría es de 3, la firmeza de la lúcuma tipo “seda” será semidura y para el tipo “palo” será dura.

3. DISPOSICIONES RELATIVAS A LA CLASIFICACIÓN POR CALIBRES

El calibre se determina por el peso de la fruta. Las lúcumas se clasifican de acuerdo con la siguiente Tabla:

Cuadro 1
Especificaciones de calibre para lúcuma²

Calibre	Peso unitario promedio (g)	Rango de pesos (g)
A	-	Menor a 55
B	90	50 – 130
C	160	120 – 200
D	230	190 - 270
E	300	260 – 340
F	-	Mayor a 330

¹ Defectos leves: rajaduras, rozadura debida al viento (rameado), manchas superficiales debido a aplicaciones o plagas que no ocasionen daño degenerativo que afecten la pulpa del fruto.

² Los rangos especificados incluyen la tolerancia de calibre, que es de $\pm 10\%$ en peso de la fruta contenida en cada envase.

Cuadro 2
Categorías de calibre para lúcuma

	Categorías		
	Extra	I	II
Diametro ecuatorial (cm)	De 6 a 9	Mayor a 9, o mayor o igual a 4 y menor a 6	Menor a 4
Peso (g)	120 a 300	Mayor a 300, o mayor o igual a 50 y menor a 120	Menor a 50

4. DISPOSICIONES RELATIVAS A LAS TOLERANCIAS

En cada envase se permitirán tolerancias de calidad y calibre para los productos que no satisfagan los requisitos de la categoría indicada.

4.1 TOLERANCIAS DE CALIDAD

4.1.1 Categoría “Extra”

El 5%, en número o en peso, de la lúcuma que no satisfaga los requisitos de esta categoría, pero satisfaga los de la Categoría I o, excepcionalmente, que no superen las tolerancias establecidas para esta última.

4.1.2 Categoría I

El 10%, en número o en peso, de la lúcuma que no satisfaga los requisitos de esta categoría, pero satisfaga los de la Categoría II o, excepcionalmente, que no superen las tolerancias establecidas para esta última.

4.1.3 Categoría II

El 10%, en número o en peso, de la lúcuma que no satisfaga los requisitos de esta categoría, ni los requisitos mínimos, con excepción de los productos afectados por podredumbre, o cualquier otro tipo de deterioro, que haga que no sean aptos para el consumo.

4.2 TOLERANCIAS DE CALIBRE

Para todas las categorías, el 10 %, en número o en peso, de la lúcuma que corresponda al calibre inmediatamente superior y/o inferior al indicado en el envase.

5. DISPOSICIONES RELATIVAS A LA PRESENTACIÓN

5.1 HOMOGENEIDAD

El contenido de cada envase deberá ser homogéneo y estar constituido únicamente por lúcuma del mismo origen, variedad, calidad, calibre, y visiblemente del mismo grado de madurez y desarrollo. La parte visible del contenido del envase deberá ser representativa de todo el contenido.

5.2 ENVASADO

La lúcuma deberá envasarse de tal manera que el producto quede debidamente protegido. Los materiales utilizados en el interior del envase deberán ser nuevos³, estar limpios y ser de calidad tal que evite cualquier daño externo o interno al producto. Se permite el uso de materiales, en particular papel o sellos, con indicaciones comerciales, siempre y cuando estén impresos o etiquetados con tinta o pegamento no tóxico.

La lúcuma deberá disponerse en envases que se ajusten al Código Internacional de Prácticas Recomendado para el Envasado y Transporte de Frutas y Hortalizas Frescas (CAC/RCP 44-1995).

5.2.1 Descripción de los Envases

Los envases deberán satisfacer las características de calidad, higiene, ventilación y resistencia necesarias para asegurar la manipulación, el transporte y la conservación apropiados de la lúcuma. Los envases deberán estar exentos de cualquier materia y olor extraños.

6. MARCADO O ETIQUETADO

6.1 ENVASES DESTINADOS AL CONSUMIDOR

Además de los requisitos de la Norma General del Codex para el Etiquetado de Alimentos Preenvasados (CODEX STAN 1-1985), se aplicarán las siguientes disposiciones específicas:

6.1.1 Naturaleza del Producto

Si el producto no es visible desde el exterior, cada envase deberá etiquetarse con el nombre del producto “Lúcuma” y, facultativamente, con el de la variedad.

6.2 ENVASES NO DESTINADOS A LA VENTA AL POR MENOR

Cada envase deberá llevar las siguientes indicaciones en letras agrupadas en el mismo lado, marcadas de forma legible e indeleble y visibles desde el exterior, o bien en los documentos que acompañan el envío.

6.2.1 Identificación

Nombre y dirección del exportador, envasador y/o expedidor. Código de identificación (facultativo)⁴.

6.2.2 Naturaleza del Producto

Nombre del producto “Lúcuma” si el contenido no es visible desde el exterior. Nombre de la variedad (facultativo).

6.2.3 Origen del Producto

País de origen y, facultativamente, nombre del lugar, distrito o región de producción.

6.2.4 Especificaciones Comerciales

- Categoría;
- Calibre (código de calibre o diámetro mínimo y máximo en milímetros);
- Peso neto;
- Número de lote.

6.2.5 Marca de Inspección Oficial (facultativa)

7. CONTAMINANTES

7.1 El producto al que se aplica las disposiciones de la presente Norma deberá cumplir con los niveles máximos de la Norma General del Codex para los Contaminantes y las Toxinas presentes en los Alimentos y Piensos (CODEX STAN 193-1995).

7.2 El producto al que se aplica las disposiciones de la presente Norma deberá cumplir con los límites máximos de residuos de plaguicidas establecidos por la Comisión del Codex Alimentarius.

⁴ La legislación nacional de algunos países requiere una declaración expresa del nombre y la dirección. Sin embargo, en caso de que se utilice una marca en clave, habrá de consignarse muy cerca de ella la referencia al “envasador y/o expedidor” (o a las siglas correspondientes).

8. HIGIENE

8.1 Se recomienda que el producto regulado por las disposiciones de la presente Norma se prepare y manipule de conformidad con las secciones apropiadas del Código Internacional Recomendado de Prácticas - Principios Generales de Higiene de los Alimentos (CAC/RCP 1-1969), Código de Prácticas de Higiene para Frutas y Hortalizas Frescas (CAC/RCP 53-2003) y otros textos pertinentes del Codex, tales como códigos de prácticas y códigos de prácticas de higiene.

8.2 El producto deberá ajustarse a los criterios microbiológicos establecidos de conformidad con los Principios para el Establecimiento y la Aplicación de Criterios Microbiológicos a los Alimentos (CAC/GL 21-1997)

9. DESCRIPCIÓN

La lúcuma es una fruta originaria del área andina del Perú, donde se le encuentra distribuida, presentando numerosos biotipos, muchos de ellos en forma silvestre. El fruto es una baya o drupa redonda, ovalada, cónica, achatada o en forma de trompo, de 4 a 9 cm. aprox. de diámetro con olor y color característico, generalmente de forma asimétrica con una punta apical que suele estar rodeada de una areola bruna a verde claro. El exocarpo o cáscara es muy delgado y quebradizo, lampiño o algo escamoso. El mesocarpo o pulpa es de grosor variable y textura de harinosa a pastosa, y de firmeza blanda a dura. La semilla es redonda achatada de 2 a 4 cm. de diámetro y está cubierta por un epispermo grueso de color marrón claro u oscuro, con un hilio u ombligo oblongo de color blanco opaco y cubierta por un endocarpo u ollejo delgado de color amarillo claro.

Para la presente Norma se aplican las siguientes definiciones:

- 1.1 **biotipo:** Conjunto de individuos cuyos fenotipos corresponden al mismo genotipo (patrimonio hereditario).
- 1.2 **areola:** Pequeño reborde, depresión o mancha, más o menos circular que aparece en ciertos órganos o pared de algunas células.
- 1.3 **lúcuma seda:** Lúcuma que posee una pulpa de firmeza blanda a semi-dura. Véase Tabla 1.
- 1.4 **lúcuma palo:** Lúcuma que posee una pulpa de firmeza dura. Véase Tabla 1.

Cuadro 1
Estado de firmeza de la pulpa

Firmeza		Estado de firmeza
En Kg-f	En N	
0 a 6	0 a 60	Blanda
Mayor a 6 y menor a 11	Mayor a 60 y menor a 110	Semi-dura
Mayo o igual a 11	Mayor o igual a 110	Dura

Nota: 1 kg-f = 9,80665 N (\approx 10 N).