Portada

INDICE / Contenido

Capítulo 1 Introducción

Resumen

Palabras clave

Introducción

Objetivo

Capítulo 2 Desarrollo

Desarrollo General (Software y tecnología usada)

Esquema General

Desarrollo Aplicación

Librerías

Librerías para procesamiento de imágenes

Otras librerías Usadas

Descripción General

Cámara web

Iluminación

Esp-32

Software Implementado

Hardware

Capítulo 3 Resultados

Capítulo 4 Análisis de costos

Capítulo 5 Conclusiones

Anexos

Bibliografía

# Resumen

En el presente proyecto se realizó un sistema que utiliza el procesamiento de imágenes para la clasificación automática de limones

Para llevarlo a cabo se utilizó el lenguaje de programación Python, junto a la librería OpenCV, con los cuales se procesan las imágenes procedentes del cuarto de visión, conformado por una web cam y el sistema de iluminación, el resultado del procesamiento es enviado al microcontrolador esp-32 mediante comunicación serial para que se encargue del accionamiento de los servomotores.

Se utilizó el entorno de desarrollo Visual Studio Code para el desarrollo del programa de PC, y QtDesigner para el desarrollo de la interfaz gráfica.

Una vez finalizado se logró clasificar los limones entre los que son aptos para exportación, los que son para consumo local y los que son descartados, conocer el diámetro de cada limón y calcular cuántos y que cantidad de kilogramos son procesados.

**Palabras Clave:** Clasificación de limones, procesamiento de imágenes, Python, OpenCV

Introducción

\*Para introducción hablar un poco sobre el limón, clasificación, innovación, etc.

# Objetivo General

Diseñar y construir un sistema clasificador de limones, implementando la selección de color y tamaño por medio de visión artificial para la distinción del fruto tipo exportación.

# Objetivos Específicos

* Diseñar y construir un sistema que permita la clasificación de limones para seleccionar el limón adecuado según la norma de calidad internacional.
* Diseñar un programa de adquisición de imágenes que permita la clasificación del limón por el color y tamaño.
* Diseñar la interfaz gráfica (GUI) que permita ver la información resultante.
* Diseñar los circuitos de control para cada uno de los actuadores del sistema.
* Construir un prototipo funcional que permita realizar pruebas de funcionamiento de los sistemas que clasifican el fruto.

# Capítulo 2 – Desarrollo

## 2.1 Software Utilizado

## 2.1.1 Python

Es un lenguaje de programación interpretado de alto nivel, interactivo y orientado a objetos. El código fuente está disponible bajo Licencia Publica General de GNU. La sintaxis de Python es muy limpia, con énfasis en la legibilidad y utiliza palabras clave estándar en inglés.

Python fue creado a finales de la década de los años 80 por un europeo llamado Guido Van Rossum en los Países Bajos. El objetivo de Guido era cubrir la necesidad de un lenguaje orientado a objetos de sencillo uso que sirviese para tratar diversas tareas dentro de la programación que habitualmente se hacía en Unix usando C.

Su formato no utiliza corchetes para delimitar bloques, sino que son las tabulaciones y los espacios los que definen en que bloque se encuentra una instrucción. Una de sus características es el uso de palabras donde otros lenguajes utilizarían símbolos. Por ejemplo, los operadores lógicos ‘!’, ‘||’ y ‘&&’ en Python se escriben ‘not’, ‘or’ y ‘and’, respectivamente.



*Figura 1 – Logo de Python*

2.1.2 Librerías

2.1.2.1 OpenCV

OpenCV (Open Source Computer Vision Library) es una biblioteca de maching learning y visión por computadora de código abierto. Al ser un producto con licencia BSD, OpenCV facilita a las empresas a utilizar y modificar el código.

La biblioteca fue lanzada oficialmente en 1999 y posee más de 2000 algoritmos optimizados que se pueden utilizar para detectar y reconocer colores, caras, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos de cámara, etc.

Es actualmente utilizado por grandes compañías como Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda y Toyota.



*Figura 2 – Logo de OpenCV[[1]](#footnote-1)*

2.1.2.1 PyQT

Es una biblioteca que nos permite la conexión entre la interfaz gráfica desarrollada en el software QT Designer y el código desarrollado en Python.

2.1.3 QT Desingner

Es una herramienta para diseñar y crear interfaces gráficas de usuario (GUI), permite componer y personalizar ventanas y cuadros de diálogos de manera *“what-you-see-is-what-you-get”* (Lo que vemos es lo que optenemos).



*Figura 3 – Logo de QT[[2]](#footnote-2)*

1. https://opencv.org/ [↑](#footnote-ref-1)
2. https://www.qt.io/ [↑](#footnote-ref-2)