UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES

TRABAJO DE TITULACIÓN

Propuesta tecnológica, previo a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

DISEÑO DE UN PROTOTIPO DE SISTEMA DE CLASIFICACIÓN DE MANZANAS SEGÚN SU DEFECTO MEDIANTE TÉCNICAS DE VISIÓN ARTIFICIAL PARA LA INSPECCIÓN DE CALIDAD EN PROCESOS INDUSTRIALES

AUTOR
PAÚL MICHAEL SUÁREZ RICARDO

PROFESOR TUTOR
ING. LUIS ENRIQUE CHUQUIMARCA JIMÉNEZ, MSC

LA LIBERTAD – ECUADOR
2024

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de titulación denominado: “DISEÑO DE UN PROTOTIPO DE SISTEMA DE CLASIFICACION DE MANZANAS SEGÚN SU DEFECTO MEDIANTE TECNICAS DE VISION ARTIFICIAL PARA LA INSPECCION DE CALIDAD EN PROCESOS INDUSTRIALES” Elaborado por el estudiante **Paúl Michael Suárez Ricardo,** de la carrera de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes y autorizo a el estudiante para que inicie los trámites legales correspondientes.

La Libertad, 6 de Diciembre del 2023

Firmado electrónicamente por:
LUIS ENRIQUE CHUQUIMARCA JIMENEZ

Ing. Luis Enrique Chuquimarca Jiménez, MSc.

TUTOR

DEDICATORIA.

Quiero dedicar el presente trabajo a mi familia, quienes me han apoyado desde el inicio de mi carrera y siempre han creído en mí, en especial a mi mami Jaqueline porque ella me trajo a la vida y me incentivó a seguir estudiando, de ella aprendí a no rendirme, sin lugar a duda toda mi carrera universitaria se la debo a ella.

A mi amada Belén y a mi bebé Alessandro quienes son mi fortaleza para levantarme día a día.

A todos mis docentes quien fueron parte del desarrollo de mi carrera en especial a mi tutor Ing. Luis Enrique Chuquimarca Jiménez quien a más de haber sido mi docente en varias materias y tutor de mi tesis es un gran amigo y estuvo siempre disponible.

Al Ing. Daniel Gómez quien a pesar de que ya no está presente físicamente entre nosotros sus palabras: “sacrificio, trabajo y estudio duro serán una satisfacción para uno mismo”, resuenan en mi presente y resonaran para la eternidad.

Paúl M. Suárez Ricardo.

AGRADECIMIENTO.

Agradezco a Dios todo poderoso, a mi familia, a mis amigos, a mis docentes por estar siempre presente y brindarme todo el soporte en mi desarrollo hacia el final de mi carrera universitaria.

A Dios por darme sabiduría y permitirme aprender cada día mas, por cuidar de mí y estar siempre presente a mi lado en todo el camino de mi carrera universitaria.

A mi Familia por estar siempre apoyándome y siempre creer en mí, su calor me ayudó mucho en tiempos difíciles, en especial agradezco mucho a mi mami por todo el esfuerzo que hizo para encaminarme hacia mi meta.

A mis queridos amigos con quienes nos sacrificamos en el estudio para lograr nuestro mismo objetivo siempre unidos, y por darme de los mejores recuerdos de mi vida en esta fabulosa etapa de la universidad, los considero también mi familia.

A mis Docentes quienes siempre estuvieron dispuestos a enseñar y estar siempre disponibles ante cualquier duda, sin lugar a duda los mejores docentes que pude tener.

Paúl M. Suárez Ricardo.

TRIBUNAL DE GRADO

Firmado electrónicamente por:
WASHINGTON DANIEL TORRES GUIN

Ing. Washington Torres Guin, Mgt.

DECANO DE LA FACULTAD

Firmado electrónicamente por:
JOSE MIGUEL SANCHEZ AQUINO

Ing. José Sánchez Aquino, Mgt.

DIRECTOR DE LA CARRERA

SHENDRY BALMORE ROSERO VASQUEZ

Firmado digitalmente por SHENDRY BALMORE ROSERO VASQUEZ
DN: cn=SHENDRY BALMORE ROSERO VASQUEZ, o=EC
Motivo: Soy el autor de este documento
Ubicación:
Fecha: 2024-04-03 08:27:05:00

Ing. Shendry Rosero Vásquez, Mgt.

DOCENTE DEL ÁREA

Firmado electrónicamente por:
LUIS ENRIQUE CHUQUIMARCA JIMENEZ

Ing. Luis Chuquimarca Jiménez, Mgt.

DOCENTE TUTOR

Firmado electrónicamente por:
MARIA MARGARITA RIVERA GONZALEZ

Ab. María Rivera González, Mgt.

SECRETARIA GENERAL.

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CARRERA ELECTRÓNICA Y TELECOMUNICACIONES

"DISEÑO DE UN PROTOTIPO DE SISTEMA DE CLASIFICACIÓN DE MANZANAS SEGÚN SU DEFECTO MEDIANTE TÉCNICAS DE VISIÓN ARTIFICIAL PARA LA INSPECCIÓN DE CALIDAD EN PROCESOS INDUSTRIALES"

Autor: **PAÚL SUÁREZ RICARDO**

Tutor: **LUIS CHUQUIMARCA JIMÉNEZ**

RESUMEN

El presente trabajo de titulación propone diseñar e implementar un prototipo de clasificadora a través de sistemas artificiales para la selección de manzanas con defecto y sin defecto con el fin de reducir costos y mano de obra utilizados en los sistemas de clasificación que se implementan en la post-cosecha de manzanas. La implementación está compuesta por una etapa de transportación donde un Motorreductor AC conectado a poleas moviliza una banda que transportara la fruta hacia la zona de visión de la cámara y hará la captura de video en tiempo real para el análisis de la fruta en el modelo entrenado, posterior pasará a la etapa de clasificación donde se cuenta con un servomotor conectado a una paleta montada en la base de la estructura de la clasificación en forma horizontal, así el sistema detecta si la fruta está en mal estado de tal manera que la paleta se abrirá y la fruta caerá hacia una gaveta de frutas malas, caso contrario la fruta continuara su recorrido si está en perfectas condiciones. El modelo fue entrenado en Teachable Machine con módulos de TensorFlow, la CNN que se aplica es MobileNet, dicho modelo se usa en un sistema creado en Python haciendo uso de OpenCv para la captura de imagen en tiempo real de la fruta, también bibliotecas para la comunicación serial con el controlador que hará girar el servomotor de acuerdo al análisis del estado de la fruta y también bibliotecas de QTdesigner para la interfaz de usuario. Se obtuvieron resultados de precisión por época de 0,9998 así como pérdida por época de 0.00098 es decir casi nula y 99% de predicción en las clases de tal manera que el sistema funciona correctamente de acuerdo a los parámetros planteados.

Palabras Clave: CNN, MobileNet, TensorFlow, Python, Opencv, GUI, Serial, Controlador.

ABSTRACT

The present degree work proposes to design and implement a prototype of a classifier through artificial systems for the selection of apples with defects and without defects in order to reduce costs and labor used in the classification systems that are implemented in the post -apple harvest. The implementation is composed of a transportation stage where an AC Gear Motor connected to pulleys mobilizes a belt that will transport the fruit to the camera's viewing area and capture video in real time for the analysis of the fruit in the trained model. , later it will go to the classification stage where there is a servomotor connected to a pallet mounted at the base of the classification structure horizontally, thus the system detects if the fruit is in poor condition in such a way that the pallet is It will open and the fruit will fall towards a drawer of bad fruits, otherwise the fruit will continue its journey if it is in perfect condition. The model was trained in Teachable Machine with TensorFlow modules, the CNN that is applied is MobileNet, this model is used in a system created in Python using OpenCv for real-time image capture of the fruit, also libraries for the serial communication with the controller that will rotate the servomotor according to the analysis of the state of the fruit and also QTdesigner libraries for the user interface. Precision results per epoch of 0.9998 were obtained as well as loss per epoch of 0.00098, that is, almost zero, and 99% prediction in the classes in such a way that the system works correctly according to the proposed parameters.

Key words: CNN, Mobilenet, TensorFlow, Python, Opencv, GUI, Serial, Controler.

DECLARACIÓN

El contenido del presente Trabajo de Graduación es de mi responsabilidad; el patrimonio intelectual del mismo pertenece a la Universidad Estatal Península de Santa Elena.

Paúl Michael Suárez Ricardo

ÍNDICE DE CONTENIDO

Contenido

APROBACIÓN DEL TUTOR II
DEDICATORIA. III
AGRADECIMIENTO. IV
TRIBUNAL DE GRADO V
RESUMEN VI
ABSTRACT VII
DECLARACIÓN VIII
ÍNDICE DE CONTENIDO IX
ÍNDICE DE FIGURAS XIII
ÍNDICE DE TABLAS XIV
INTRODUCCIÓN 1
CAPÍTULO I 2
1.1 ANTECEDENTES 2
1.2 DESCRIPCIÓN DEL PROYECTO 3
1.3 OBJETIVOS DEL PROYECTO 5
1.3.1 OBJETIVO GENERAL 5
1.3.2 OBJETIVOS ESPECÍFICOS. 5
1.4 RESULTADOS ESPERADOS 6
1.5 JUSTIFICACIÓN 6
1.6 ALCANCE 7
1.7 METODOLOGÍA 7
1.7.1 INVESTIGACIÓN BIBLIOGRÁFICA 7
1.7.2 INVESTIGACIÓN EXPLORATORIA 8
1.7.3 INVESTIGACIÓN DIAGNÓSTICA 8
CAPÍTULO II 9
2.1 MARCO CONTEXTUAL 9
2.2 MARCO CONCEPTUAL 10
2.2.1 MANZANA 10
2.2.2 INSPECCIÓN DE CALIDAD DE MANZANAS 10
2.2.3 VISIÓN ARTIFICIAL 10

2.2.4 APRENDIZAJE PROFUNDO 11
2.2.5 RED NEURONAL CONVOLUCIONAL 11
2.2.6 MODELO MOBILENET V2 13
2.2.7 MICROPROCESADOR 14
2.2.8 MICROCONTROLADOR 14
2.2.9 MOTORES ELÉCTRICOS AC 15
2.2.10 SERVOMOTORES 15
2.2.11 CONTROL DE SERVOMOTORES 15
2.2.12 Señal PWM 16
2.2.13 COMUNICACIÓN SERIAL 16
2.2.14 CÁMARA 17
2.2.15 BANDA TRANSPORTADORA 17
2.2.16 LENGUAJE DE PROGRAMACIÓN 17
2.2.17 INTERFAZ GRAFICA DE USUARIO 17
2.2.18 SISTEMAS AUTOMATIZADOS 18
2.3 MARCO TEÓRICO 19
CAPITULO III 20
3.1 COMPONENTES DE LA PROPUESTA 20
3.1.1 COMPONENTES FÍSICOS 20
3.1.1.1 MOTOREDUCTOR AC 110V 20
3.1.1.2 SERVOMOTOR MG995 21
3.1.1.3 PLACA ARDUINO R3 21
3.1.1.4 MICROPROCESADOR INTEL CORE I5 1135G7 22
3.1.1.5 FACECAM VGA 23
3.1.2 COMPONENTES LÓGICOS 24
3.1.2.1 PYTHON 24
3.1.2.2 OPENCV 24
3.1.2.3 TENSORFLOW 25
3.1.2.4 MEDIAPIPE 25
3.1.2.5 KERAS 26
3.1.2.6 PROTEUS 26
3.1.2.7 VIRTUAL SERIAL PORT 27
3.1.2.8 ARDUINO IDE 27
3.1.2.9 TEACHABLE MACHINE 28

3.1.2.10 PYSERIAL 3.4 28
3.1.2.11 QTDESIGNER 29
3.2 DISEÑO DE LA PROPUESTA 29
3.2.1 DISEÑO DE LA BANDA TRANSPORTADORA 29
3.2.2 IMPLEMENTACIÓN DE LA BANDA TRANSPORTADORA TIPO CINTA 30
3.2.3 DISEÑO DE LA CLASIFICADORA DEL SISTEMA. 31
3.2.4 IMPLEMENTACIÓN DE LA CLASIFICADORA 32
3.2.5 MONTAJE DE LA CÁMARA 33
3.2.6 DIAGRAMA DE CONEXIONES DEL SISTEMA 34
3.2.7 DIAGRAMA DE CONEXIONES DE ALIMENTACION 35
3.2.8 DIAGRAMA DE ENTRENAMIENTO DE LA RED NEURONAL. 36
3.2.9 DATASEET 37
3.2.10 DISEÑO CON PROTEUS 39
3.2.11 CONFIGURACIÓN DE TEACHABLE MACHINE 40
3.2.11.1 SELECCIÓN DEL TIPO DE PROYECTO 40
3.2.11.2 SELECCIÓN DEL MODELO 42
3.2.11.3 AÑADIR CLASES 42
3.2.11.4 PREPARACIÓN Y CONFIGURACIÓN DEL MODELO 43
3.2.11.5 VISTA PREVIA DEL MODELO 44
3.2.11.6 EXPORTACIÓN DEL MODELO 44
3.2.12 DISEÑO DE GUI EN QT DESIGNER 45
3.3 ESTUDIO DE FACTIBILIDAD 48
3.3.1 FACTIBILIDAD TÉCNICA 48
3.3.2 ANÁLISIS DE COSTOS DEL PROYECTO 48
CAPÍTULO IV 50
4.1 PRUEBAS Y RESULTADOS 50
4.1.1 ENTRENAMIENTO DEL MODELO 50
4.1.2 PRUEBA DEL MODELO CON EL SISTEMA DE VISIÓN ARTIFICIAL 56
4.1.3 PRUEBA DEL SISTEMA CON EL CONTROLADOR ELECTRÓNICO DE GIRO DEL MOTOR 58
4.1.4 DISEÑO FINAL CON LA INTERFAZ DE USUARIO 59
4.2 CONCLUSIONES Y RECOMENDACIONES 63
4.2.1 CONCLUSIONES 63

4.2.2 RECOMENDACIONES.....64
4.3 BIBLIOGRAFÍAS.....65
4.4 ANEXOS.....70

ÍNDICE DE FIGURAS

Figura 1. Diagrama del sistema (Fuente: El autor). 4
Figura 2. *Arquitectura de red convolucional [15]*. 12
Figura 3. Conectividad dispersa [16]. 12
Figura 4. Mapa de caracteres [18]. 13
Figura 5. Arquitectura MobilenetV2 [20]. 14
Figura 7. Motorreductor Ac 20
Figura 6. Servomotor MG995. 21
Figura 7. Placa Arduino UNO R3. 22
Figura 8. Intel Core 1135G7 23
Figura 9. FACECAM VGA. 23
Figura 10. Logotipo de Python 24
Figura 11. Logotipo de OpenCv. 25
Figura 12. Logotipo de Tensorflow 25
Figura 13. Logotipo del Módulo MediaPipe. 26
Figura 14. Logotipo de la biblioteca Keras 26
Figura 15. Logotipo del software Proteus. 27
Figura 16. Logotipo de la herramienta Virtual Serial Port. 27
Figura 17. Logotipo del software ARDUINO IDE. 28
Figura 18. Logotipo del software TECHEABLE MACHINE 28
Figura 19. Logotipo de PySerial. 29
Figura 20. Logotipo de QT Designer. 29
Figura 21. Vista lateral del diseño de la banda transportadora 30
Figura 22. Vista Superior del diseño de la banda transportadora 30
Figura 23. Banda transportadora del sistema. 31
Figura 24. Vista frontal del Diseño de la clasificadora. 32
Figura 25. Vista superior del Diseño de la clasificadora. 32
Figura 26. Clasificadora del sistema. 33
Figura 27. Posición de la cámara para captura de imagen. 34
Figura 28. Visualización en tiempo real de la manzana. 34
Figura 29. Diagrama de conexiones del sistema. 35
Figura 30. Diagrama de conexiones de alimentación 36
Figura 31. Diagrama del entrenamiento. 37
Figura 32. Parte de la selección de dataset para manzanas buenas. 38
Figura 33. Parte de la selección de dataset para manzanas dañadas. 38
Figura 34. Ambiente sin fruta para el entrenamiento. 39
Figura 35. Diseño con Proteus del sistema 40
Figura 36. Vista selección Proyecto tipo imagen de Teachable Machine 41
Figura 37. Vista de clases de proyecto imagen. 41
Figura 39. Vista dataset generado a través de python. 42
Figura 40. Vista selección Modelo de imagen estándar 42
Figura 41. Vista del apartado para añadir clases. 43
Figura 42. Vista de configuración del modelo. 43
Figura 43. Vista previa de un modelo entrenado. 44
Figura 44. Selección para descargar modelo compatible con KERAS. 45
Figura 45. Selección de la plantilla Window. 45
Figura 46. Vista del Frame de captura de video. 46
Figura 47. Etiquetas principales del modelo entrenado. 46
Figura 48. Botones principales del GUI. 47
Figura 49. Configuración de parámetros. 47
Figura 50. Vista de agregar recursos. 47
Figura 51. Parte del código del GUI en Python. 48
Figura 52. Muestras de imágenes para las clases. 50
Figura 53. Preparación del modelo. 50

Figura 54. Matriz de confusión del modelo entrenado. 51
Figura 55. Precisión en época 2. 52
Figura 56. Precisión en época 10. 52
Figura 57. Perdida en época 5. 53
Figura 58. Perdida en época 10. 53
Figura 59. Perdida en época 49. 54
Figura 60. Vista previa del modelo en clase 3. 54
Figura 61. Vista previa del modelo en clase 2. 55
Figura 62. Vista previa del modelo en clase 1. 55
Figura 63. Sistema de visión computarizada manzana buena. 56
Figura 64. Sistema de visión computarizada manzana mala. 57
Figura 65. Sistema de visión computarizada ambiente sin manzana. 57
Figura 66. Etapa de clasificación simulada manzana buena. 58
Figura 67. Etapa de clasificación simulada manzana mala 58
Figura 68. Etapa de clasificación simulada estado vacío 59
Figura 69. Vista 1 ambiente sin manzana. 59
Figura 70. Vista 2 ambiente sin manzana. 60
Figura 71. Vista 3 ambiente sin manzana. 60
Figura 72. Vista 1 clasificación Manzana buena 61
Figura 73. Vista 2 clasificación Manzana buena 61
Figura 74. Etapa de clasificación manzana Buena. 61
Figura 75. Vista 1 clasificación manzana con defecto. 62
Figura 76. Vista 2 clasificaciones manzana con defecto. 62

ÍNDICE DE TABLAS

Tabla 1. Especificaciones Motorreductor AC 20
Tabla 2. Especificaciones Servomotor MG995 21
Tabla 3. Especificaciones Arduino uno R3 22
Tabla 4. Especificaciones Intel Core 1135g7 23
Tabla 5. Especificaciones FACECAM VGA 24
Tabla 6. Costos de Equipos 48
Tabla 7. Costos de Licencias de Software 49
Tabla 8. Costos de otros Elementos 49
Tabla 9. Costos Finales del proyecto 49

INTRODUCCIÓN

En la actualidad el uso de tecnologías para clasificación de manzanas es escasa en el Ecuador, esto debido a que las empresas prefieren contratar personal para que realicen ese proceso de forma manual, lo cual implica un gasto enorme para la empresa ya que requiere una gran cantidad de mano de obra para lograr resultados óptimos para el proceso de clasificación. Este proyecto pretende innovar el proceso de clasificación de manzanas llevándolo a otro nivel usando tecnología de ingeniería mediante el diseño e implementación de un sistema de visión artificial compatible con actuadores eléctricos para el proceso de clasificación de manzanas según su defecto enfocado en redes neurales de tal manera que el sistema aprenda y de acuerdo a su entrenamiento cumpla los requerimientos dados optimizando el sistema de clasificación.

Se entrena una red neuronal convolucional para que aprenda a distinguir que manzana está en buen estado, o en mal estado. Para la parte de software se usa un computador, el cual se encargará de realizar los diferentes procesos lógicos a través de la aplicación Python que es en donde se montaran los demás recursos para el sistema. Se usa visión artificial con una cámara conectada al computador para el respectivo procesamiento de imágenes en tiempo real de las manzanas, se pone en marcha el modelo entrenado aplicando bibliotecas de Python, luego de acuerdo al resultado que arroje el modelo entrenado se activará el actuador que clasificará la manzana. Se usará una banda transportadora para el desplazamiento de la manzana hacia la sección de clasificación.

CAPÍTULO I

1.1 ANTECEDENTES

Dentro de Ecuador existen diferentes tipos de frutas tropicales que se pueden encontrar en todos los mercados durante todo el transcurso del año. Una de las tantas frutas que se cultiva es la manzana la cual es muy jugosa y a su vez nutritiva. Se la suele usar por su poder antioxidante, para mejorar el sistema inmune, disminuir la fatiga, etc.

La manzana es un cultivo muy importante en varias localidades (Ecuador) debido a que existe un área extensa y concentrada dentro de la misma, una gran parte del total que se consume en Ecuador de este fruto se produce en dicha zona que cuenta con escasa información actualizada sobre los productores que tienen bajos rendimientos y pobre desarrollo tecnológico [1]. Además, la inspección de calidad en el Ecuador es muy rígida esto debido a que debe exportar frutas de calidad, en donde el producto no debe tener cicatrices, ningún tipo de estropeo, debe ser una fruta lo más impecable posible [2].

Ecuador existen empresas a pesar de que cuentan con la tecnología necesaria para cumplir los requisitos de exportación, los métodos aplicados en todos los casos no son los más eficientes [3]. Por ello cada día la investigación en el país sobre visión artificial busca implementar técnicas que faciliten las normativas de alta calidad para la exportación.

Los procesos de inspección de calidad en el sector industrial de alimentos han mejorado con el uso de técnicas de inteligencia artificial por visión computarizada. De esta manera, los productos finales son óptimos para la exportación y consumo. Existen tareas como la inspección de productos cárnicos y agrícolas están siendo realizadas por sistemas de visión capaces de clasificarlas según sus características y evaluar su calidad [4].

Por esta razón, las empresas de la provincia de Santa Elena pueden optar por instalar sistemas de visión artificial con algoritmos inteligentes que permitan establecer una adecuada inspección de calidad de sus productos finales.

1.2 DESCRIPCIÓN DEL PROYECTO

En el proyecto propuesto plantea un prototipo de sistema de clasificación de manzanas en pos cosecha mediante técnicas de visión artificial de tal forma que el producto final tenga una calidad de alto nivel para la posterior exportación y consumo.

Un computador será el encargado de ejecutar las aplicaciones y comunicaciones con los dispositivos del proyecto. Para empezar el proceso de inspección, las manzanas pasarán por una banda de transporte.

Durante el recorrido de la fruta, una cámara estará capturando y enviando la imagen en tiempo real para el respectivo procesamiento con el sistema de clasificación por visión artificial mediante OpenCV en la plataforma Python.

Para el sistema de clasificación, se utilizará una red neuronal entrenada en la plataforma Tensorflow junto con el módulo Keras, la cual permite el aprendizaje profundo. De esta manera, la red neuronal detectará el estado de la fruta, posteriormente se ejecutará el proceso de control actuadores electromecánicos ubicados al final de la banda, los cuales tienen la función de clasificar la fruta, ubicándolas en gavetas de acuerdo a su estado.

Para mover la banda transportadora se usará un motor AC junto a un variador de frecuencia, en la etapa de clasificación se usarán actuadores eléctricos controlados por arduino para mover la manzana hacia la gaveta respectiva de acuerdo a la predicción de la red neuronal. Para la comunicación Computador-arduino se usarán algoritmos de la biblioteca Serial en Python, la cual permite controlar las salidas de los puertos del arduino.

El sistema está formado por 3 etapas (ver Figura 1), las cuales especifican en forma ordenada los procesos que ejecuta el sistema que se detalla a continuación:

1. **Desplazamiento. -** ingreso de la manzana a la banda transportadora y captura de imagen de la misma.

2. **Validación. -** La imagen es analizada por la red neuronal y según el resultado validado se envían las respectivas señales hacia la placa arduino mediante el software de comunicación.

3. **Clasificación. -** De acuerdo a la validación de los datos la placa arduino enviara las respectivas señales hacia el actuador eléctrico de clasificación (Servomotor conectado a un eje con una paleta).

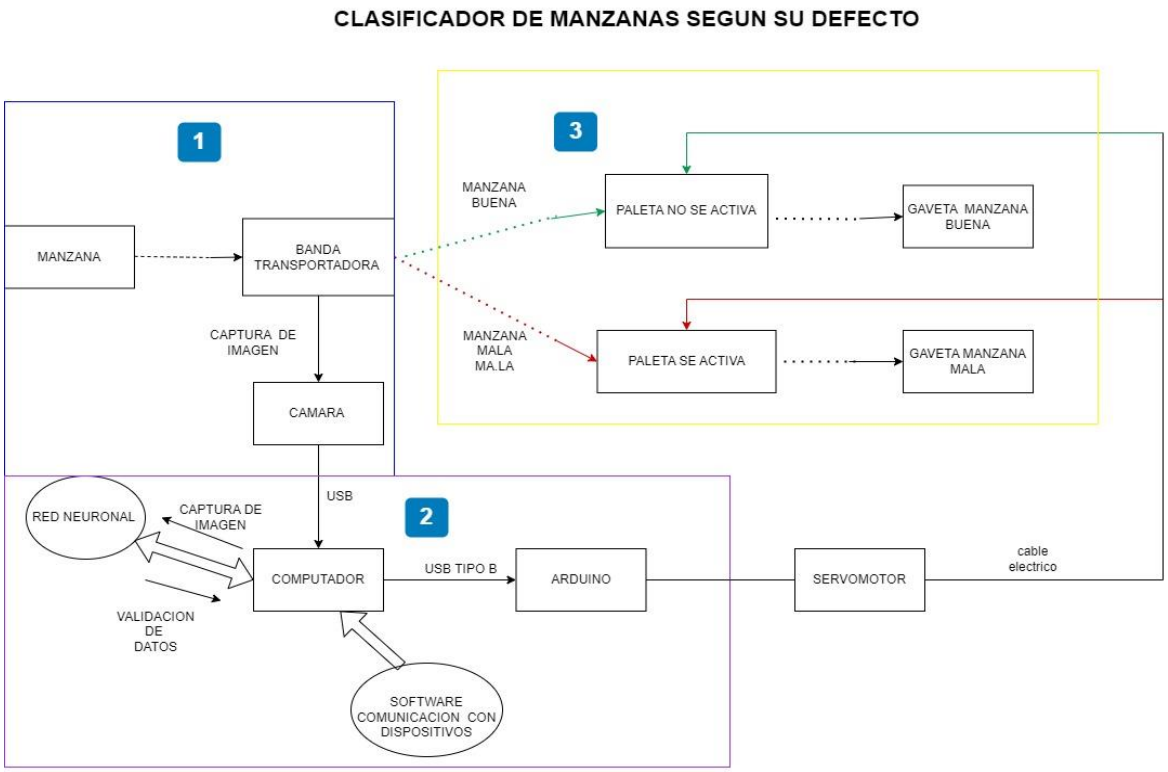


Figura 1. Diagrama del sistema (Fuente: El autor).

1.3 OBJETIVOS DEL PROYECTO

1.3.1 OBJETIVO GENERAL

Implementar un prototipo de control de sistema de clasificación de manzanas mediante técnicas de visión artificial para inspección de calidad en procesos industriales.

1.3.2 OBJETIVOS ESPECÍFICOS.

- Entrenar una red neuronal artificial utilizando la librería TensorFlow y KERAS para la clasificación de manzanas según su defecto (buenas y malas).

- Implementar un sistema de visión artificial en Python mediante el módulo OpenCV para la adquisición de imágenes de las manzanas en la banda transportadora.

- Utilizar un control de giro motores eléctricos mediante un controlador electrónico para la clasificación de manzanas.

- Diseñar una interfaz de usuario de nivel básico para el monitoreo del sistema

.

1.4 RESULTADOS ESPERADOS

En el presente proyecto, se tiene como resultados esperados los siguientes:

- Una red neuronal entrenada en Tensorflow con un porcentaje aceptable de predicción para manzanas buenas y para frutas malas o putrefactas.
- Un sistema de visión artificial en Python usando red neuronal entrenada la cual debe ser capaz de clasificar las manzanas en tiempo real.
- Un control de giro mediante un controlador de motores que manipule los mecanismos para la clasificación de manzanas.
- Una interfaz de usuario sencilla donde se puedan visualizar la captura de video en tiempo real y los parámetros principales del sistema de clasificación.

1.5 JUSTIFICACIÓN

Durante el manejo pos cosecha de frutos climatéricos como el mango, plátano, aguacate, manzanas, papaya y cítricos, ocurren diversos cambios fisicoquímicos, como la pérdida de firmeza y turgencia y aumento del contenido de sólidos solubles totales, modificación del contenido de lípidos, la disminución de la acidez, degradación y biosíntesis de pigmentos, como clorofilas, xantofilas y carotenoides [5].

Gracias a la visión artificial en la inspección de calidad se puede mencionar varios aspectos importantes entre uno de los cuales se tiene la flexibilidad de procesos productivos debido a la adaptabilidad y su entorno de escalabilidad que exigen los cambiantes procesos productivos, lo cual quiere decir que se puede modificar un mismo sistema de visión artificial para inspeccionar diferentes tipos de frutas. Otro factor importante es que existe la subjetividad durante la inspección, esto se debe a los ciclos repetitivos de trabajos automatizados, los cuales ofrecen mejor rendimiento en tiempo mínimo, también se puede mencionar que este tipo de tecnología es asequible, ya que debido a la gran cantidad de aplicaciones de la Visión artificial se puede hacer uso de gran variedad de aplicaciones de acuerdo a los sistemas que se plantea. Las soluciones de visión artificial aplicadas a la industria permiten obtener, procesar y analizar imágenes, para más tarde ser analizadas por un ordenador. Todo este proceso permite automatizar una gran variedad de procesos y tareas al aportar a las máquinas la información que necesitan para la toma de decisiones en cada proceso designado [6].

1.6 ALCANCE

El proyecto se direcciona hacia la inspección de calidad de manzanas donde se analiza si la fruta tiene algún tipo de enfermedad es decir si está en buen estado o no, se omite el peso de la fruta ya que no tiene relevancia en el proceso de clasificación usando la red neuronal.

Cuando se trata de frutas frescas se asume que las frutas están maduras, sin embargo, se puede mencionar que un sistema sin una red neuronal es capaz de detectar si una fruta está madura o verde según su tono de color, pero no puede detectar si la fruta madura posee algún defecto en su consistencia o posee algún tipo de enfermedad.

Se aprovecha la inclinación de la clasificadora ya que esto permite a la fruta tener un avance en caída libre. La cámara empleada es capaz de captar la fruta y se tiene toma diferente para ser analizadas en la red neuronal.

Para el control y uso de actuadores se contemplan diferentes tipos de sistemas, sin embargo, para la finalidad que se requiere, lo cual es hacer un movimiento simple como el movimiento de un servomotor se optó hacer un sistema sencillo utilizando un driver para el control de los motores.

El proyecto se direcciona hacia la inspección de calidad de manzanas donde se analiza si la fruta tiene algún tipo de enfermedad es decir si está en buen estado o no, se omite el peso de la fruta ya que no tiene relevancia en el proceso de clasificación usando la red neuronal.

Cuando se trata de frutas frescas se asume que las frutas están maduras, sin embargo, se puede mencionar que un sistema sin una red neuronal es capaz de detectar si una fruta está madura o verde según su tono de color, pero no puede detectar si la fruta madura posee algún defecto en su consistencia o posee algún tipo de enfermedad.

1.7 METODOLOGÍA

Para realizar el proyecto se aplican las metodologías bibliográfica, exploratoria y diagnóstica.

1.7.1 INVESTIGACIÓN BIBLIOGRÁFICA

La investigación bibliográfica puede definirse como cualquier investigación que requiera la recopilación de información a partir de materiales publicados [7]. Se recopila información de artículos y trabajos en donde se implemente redes neuronales en visión artificial, entre más trabajos analizados mejores resultados se obtendrán, debido a que es un tema poco usado en el área industrial.

El uso de arduino también es muy amplio entre los trabajos se estudiarán de los cuales destaca

su gran aceptación para proyectos pequeños basados en temáticas industriales, al obtener gran cantidad de referencias de artículos y trabajos el proyecto estará mejor fundamentado y justificado.

1.7.2 INVESTIGACIÓN EXPLORATORIA

La investigación exploratoria es un tipo de investigación utilizada para estudiar un problema que no está claramente definido, por lo que se lleva a cabo para comprenderlo mejor, pero sin proporcionar resultados concluyentes [8].

Este tipo de investigación se enfoca en el conocimiento adquirido y analizado sobre el tema de redes neuronales que ya empezó a usarse en el área industrial pero actualmente los estudiantes tienen poco conocimiento, por lo que el resultado del proyecto es único e innovador. Como esta investigación no exige una estructura obligatoria, el investigador tiene la opción de escoger la idea, proceso o conclusión que le ha parecido más sencillo.

Aquí podemos encontrar una solución al problema con los motores DC que se usaran pues se requiere que asimilen el movimiento y giro como el de un servomotor.

1.7.3 INVESTIGACIÓN DIAGNÓSTICA

La investigación diagnóstica supone análisis de situaciones. El análisis de la situación es un momento complejo que nos desafía a conocer lo que está sucediendo en una determinada representación de la realidad que denominamos situación, incluyendo lo que está sucediendo a quienes están actuando sobre y dentro de los límites de esa realidad [9]. Teniendo en claro la problemática se comienza a enfocar en los diferentes procesos que requieren soluciones ya sean sencillas o complejas. En el caso de soluciones complejas se puede mencionar el uso de Tensorflow que es una biblioteca de aprendizaje de máquina, existen otras bibliotecas que cumplen la misma función sin embargo por ser sencillas no implica que puedan dar los resultados esperados, la mayoría suele tener redes neuronales ya entrenadas, pero es mejor y más eficiente tener una red neuronal entrenada con un propio Dataset y entre otras funciones que ofrece Tensorflow.

CAPÍTULO II

2.1 MARCO CONTEXTUAL

En la mayor parte del Ecuador la producción de manzanas incluyendo la exportación de la misma es muy esencial ya que produce trabajo para muchos agricultores y proporciona una gran ganancia para las empresas que se dedican a la cosecha de la misma.

En el Ecuador la producción de manzanas se define por su alto grado de calidad, por lo cual es importante tener en cuenta un muy buen control de calidad para su respectiva exportación, la mayor producción de manzanas se las realiza en las zonas altas del Ecuador que corresponden a las provincias de, Chimborazo, Cotopaxi, Cañar, Tungurahua, Loja y Azuay. En la mayoría de las empresas productoras de manzanas cuentan con procesos de selección del fruto de forma manual, y su selección va de acuerdo al grado de estado de la fruta, es decir se toma en cuenta la manzana que esta buena o esta mala, cuando se refiere a mala es porque da indicios de podredumbre, plaga, etc. Este tipo de proceso se lo realiza a través de personal encargado de verificar fruta a fruta el estado de la misma a través de una banda transportadora industrial.

El proyecto propuesto se encuentra ubicado en el laboratorio de neumática, cuenta con la implementación de un sistema automatizado de clasificación de manzanas buenas y malas mediante técnicas de visión artificial con redes neuronales que facilitan la manera como se clasifican e incrementa de manera eficiente los tiempos de trabajo y reduce los costos por recurso de operación manual. Se usaron alrededor de 300 muestras en cada clase para el entrenamiento, donde se obtuvieron resultados positivos.

2.2 MARCO CONCEPTUAL

2.2.1 MANZANA

La manzana es una fruta tropical que posee alto contenido en hidratos de carbono, fibra, vitaminas y potasio, se la recomienda mucho para toda persona de cualquier edad por sus altos nutrientes.

Debido a su gran composición nutricional la manzana aporta grandes beneficios para la salud de las personas. La manzana es un fruto que proviene del árbol del manzano, no se conoce exactamente el origen de la manzana pero se puede mencionar que pertenece a la familia de las rosáceas, así como también se suele decir que proviene del cruce de manzanos de las tierras de Europa y Asia [10].

2.2.2 INSPECCIÓN DE CALIDAD DE MANZANAS

En pos-cosecha si es adecuada la técnica para la recolección de la fruta es siempre fundamental para asegurar la comercialización. La clasificación ayuda a mantener el mejor aspecto y se utiliza para separarlos en lo que es normal venda cajas que puedan contener frutas del mismo tipo y tamaño. Existen diferentes necesidades los mercados de frutas hacen que su clasificación sea muy importante y la solución común es hacer esto a mano, para los que se especializan en este trabajo, sin embargo, hacerlo a mano no es suficiente o rentable. Para evitar complicaciones y retrasos en la línea de producción que se producen durante la ejecución usualmente se tiende a montar clasificadores manuales, en vez de lo mencionado se puede usar clasificadores automáticos pero debido a que son complejos los mecanismos se retractan y proceden a usar los más comunes ya que lo que supone un coste alto [11].

Las normas internacionales para la inspección de manzanas se establecen principalmente a través comisiones como la OECD. Estas normas se aplican a nivel internacional y se conocen como las Normas Generales para la Inspección y Clasificación de Manzanas [12].

2.2.3 VISIÓN ARTIFICIAL

Desde hace muchos años la visión artificial ha estado presente en el área de la electrónica sin embargo en la actualidad existe una gran demanda por crear sistemas y diseños que empleen este tipo de tecnología. Es una ciencia que se ocupa de los métodos de captura, procesamiento, análisis y comprensión de imágenes del mundo real para obtener información digital o símbolos que los humanos pueden manejar. Esencialmente, esto les da a las máquinas la capacidad de procesar imágenes que imitan la visión humana, pero tienen características

completamente diferentes. Esto ciertamente tiene algo que ver con el desarrollo de IA [13].

2.2.4 APRENDIZAJE PROFUNDO

El aprendizaje profundo (deep learning, en inglés) es una subdisciplina de la inteligencia artificial (IA) que se centra en el entrenamiento de redes neuronales artificiales profundas para realizar tareas específicas. Estas redes neuronales están inspiradas en la estructura y funcionamiento del cerebro humano y están diseñadas para aprender y tomar decisiones a partir de datos.

Existen diversas arquitecturas de redes neuronales profundas, como redes neuronales convolucionales (CNN) para tareas de visión por computadora, redes neuronales recurrentes (RNN) para secuencias de datos, y redes neuronales generativas (GAN) para generar datos nuevos y creativos.

El aprendizaje profundo ha demostrado ser muy efectivo en una variedad de aplicaciones, como reconocimiento de voz, visión por computadora, procesamiento de lenguaje natural, juegos, conducción autónoma, diagnóstico médico y mucho más. Ha revolucionado la forma en que abordamos problemas de IA complejos y ha llevado a avances significativos en muchos campos [14].

2.2.5 RED NEURONAL CONVOLUCIONAL

A inicios de los años noventa fueron presentadas al mundo por LeCun et al, este tipo de redes son diseñadas con el propósito de ejemplificar una arquitectura de redes especializadas según sea su uso, la cual contiene información sobre cualquier tipo de invariancia de tipos bidimensionales usando patrones con conexiones locales y se restringen los casos según la información en los pesos. Muestran un tipo de arquitectura multicapa, en donde cada capa presenta una cantidad determinada de convoluciones las cuales contiene funciones de activación no lineales.

Esta forma de conexiones entre neuronas en una red convolucional se diseña a partir de la constitución de la corteza visual en los animales, en donde el resultado de cada neurona se puede demostrar mediante una operación de convolución de manera matemáticamente.

Las características más significativas de las redes neuronales convolucionales se muestran a continuación (ver Figura 2) como son su conectividad dispersa y también los pesos compartidos

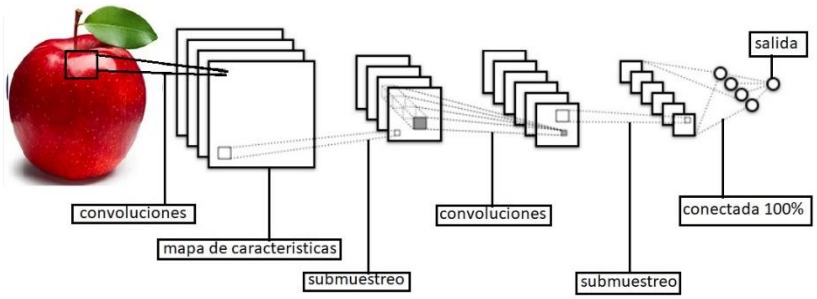


Figura 2. *Arquitectura de red convolucional [15].*

Cuando se habla de conectividad dispersa es como las redes neuronales convolucionales pueden aprovechar las correlaciones del espacio local intensificando aquellos patrones de conectividad local con las neuronas de capas cercanas.

Una neurona como también se denomina capa M+1 las entradas son el subconjunto de las neuronas de la capa M, entonces se puede decir que una neurona de capa M+1 da como resultado un campo receptivo 3 que es un campo adyacente a la neurona de capa M+1 (ver Figura 3).

Debido a que cada neurona posee un tamaño fijo no se valoran las variaciones que se encuentren fuera del espacio cercano a esta.

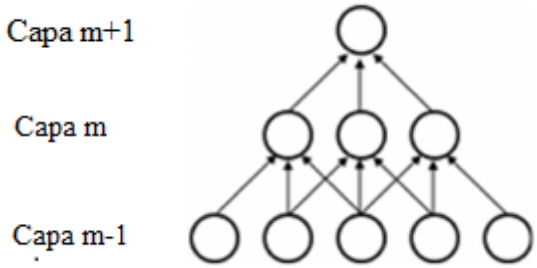


Figura 3. Conectividad dispersa [16].

Para este diseño de red neuronal en cada filtro se usa por completo el campo visual de la capa. Estos filtros o unidades poseen una misma configuración de pesos y sesgos de tal manera que se crea un mapa de caracterización. En la capa m aparecen 3 filtros que corresponden al mismo mapa de características. Un mismo color representa el peso utilizado. De tal manera que al usar estos filtros a través del campo de visión se puede identificar cualquier característica independientemente de la posición. Se puede decir que al utilizar pesos compartidos se optimiza el aprendizaje debido a que se reduce el número de características para ser aprendidas, así también disminuye la brecha entre la fase de entrenamiento y etapa de generalización. Adicional disminuye la memoria usada para utilizar la red neuronal.

En conclusión, cuando una capa convolucional cuenta con muchos mapas de características (ver figura 4) y son capaces de identificar cualquier característica a través de su filtro por todo el campo de visión entonces la capa está completa [17].

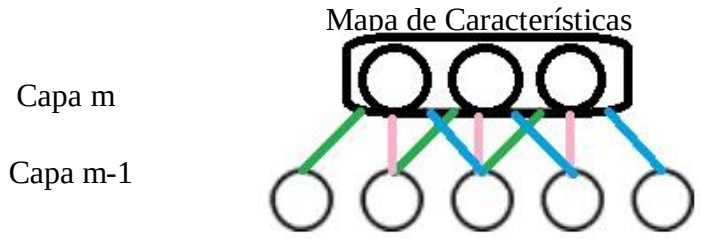


Figura 4. Mapa de caracteres [18].

2.2.6 MODELO MOBILENET V2

Fue presentado por la compañía Google para la satisfacción de emplear redes neuronales en proyectos libres y de código abierto.

MobileNet V2 es la versión que usa el presente proyecto, gracias a esta modelo se consigue reducir el tiempo de ejecución de procesos computacionales para obtener una alta fidelidad de precisión.

El hecho de poder utilizar redes neuronales profundas en microcontroladores y cualquier otro dispositivo de manera fácil y eficaz para el usuario hace que el aprendizaje ya no sea tan complejo como lo visto en décadas pasadas. Al ser un modelo de código abierto se abre a varias posibilidades de ser usado con cualquier aplicación o dispositivo debido a su alta compatibilidad por ser software libre.

MobileNetV2 no solo potencia la nueva generación de aplicaciones con redes neuronales para aplicaciones de visión móvil sino también potencia el aprendizaje de esta área redes neuronales convolucionales. Este modelo se puede encontrar como bibliotecas o módulos en Tensor Flow HUB como también modelos entrenados en github.

MobileNet presenta una arquitectura que factoriza convoluciones presentes en 2 capas la una llamada “depthwise” y la otra capa convolucional 1x1 llamada “pointwise”, gracias a esto se logra reducir el tiempo de ejecución de los procesos matemáticos como también el tamaño en memoria del modelo.

El MobileNetV2 que se usa en el presente proyecto utiliza parte de la configuración de la

primera generación de MobileNet ya mencionada utilizando una convolución que se puede separar profundamente como bloques de edificación de alta fidelidad. Con esto la versión añade 2 especificaciones en su arquitectura (ver figura 5) que son cuellos de botella lineales en medio de las capas convolucionales y las conexiones para el acceso de botella [19].

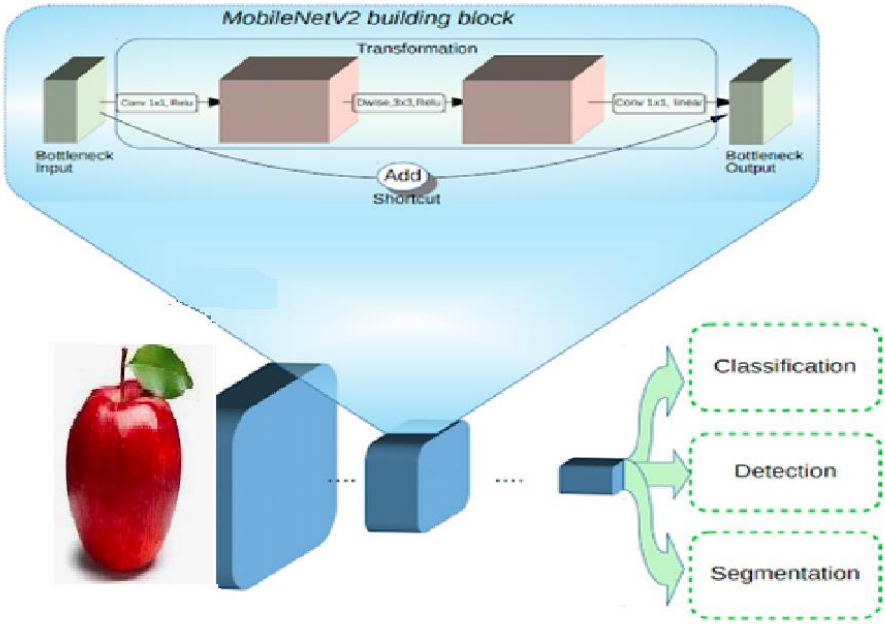


Figura 5. Arquitectura MobilenetV2 [20].

2.2.7 MICROPROCESADOR

Es aquel que ejecuta las funciones de una CPU. Funciona en el momento que se el computador y es responsable de ejecutar el sistema operativo y los programas relacionados. También puede ejecutar diversos tipos de procesos solicitados por el usuario de acuerdo al use que se le esté dando.

Puede alojar una o más CPU (Unidades centrales de procesamiento), ALU (Unidad aritmética lógica) y una Unidad informática de coma flotante. Está conectado a la placa base mediante un conector especial. A menudo se le añade un disipador de calor para evitar que se sobrecalienten [21].

2.2.8 MICROCONTROLADOR

Es un circuito integrado digital al cual se le puede asignar para diferente tipo de tarea debido a que tiene una gran funcionabilidad programable. Consta de una unidad central de procesamiento (CPU), memoria (ROM y RAM) y periféricos tanto de entrada como de salida. Como se entiende, existe diversos tipos de micro controladores, los más comunes son usados

en computadores o en celulares los cuales debido a su gran alcance proporcionan grandes beneficios a los diversos dispositivos electrónicos [22].

2.2.9 MOTORES ELÉCTRICOS AC

Es un actuador capaz de convertir energía eléctrica en una energía totalmente mecánica con una rotación mediante un campo magnético que se genera en sus devanados. Se trata de máquinas eléctricas rotativas formadas por un estator y un rotor. Algunos motores eléctricos son reversibles porque pueden convertir la energía mecánica en energía eléctrica actuando como generador o generador. Los motores de tracción utilizados en locomotoras o vehículos híbridos, diseñados de manera adecuada, normalmente realizan ambas tareas. Se utilizan en multitud de campos como instalaciones industriales, comerciales y privadas. Es ampliamente utilizado en ventiladores, vibradores de teléfonos móviles, bombas de agua, vehículos eléctricos, electrodomésticos, amoladoras angulares y otras herramientas eléctricas, accionamientos, etc. Los motores eléctricos pueden funcionar con fuentes de corriente alterna (CA) [23].

2.2.10 SERVOMOTORES

Un servomotor es un tipo de motor eléctrico que se utiliza para controlar con precisión la posición, velocidad y en algunos casos, la fuerza de un mecanismo. Estos motores son ampliamente utilizados en aplicaciones donde se requiere un control preciso del movimiento, como en robótica, automatización industrial, sistemas de control de vuelo de aeronaves y otros dispositivos electromecánicos.

Los servomotores son conocidos por su capacidad para proporcionar un control preciso y rápido del movimiento, lo que los hace ideales para tareas que requieren alta precisión, como la automatización de maquinaria, el seguimiento de objetos en robótica, el control de la dirección en vehículos a control remoto, y muchas otras aplicaciones [24].

2.2.11 CONTROL DE SERVOMOTORES

Primero se tiene un mecanismo de retroalimentación el cual se usa para controlar con precisión la posición o la velocidad del servomotor, se utiliza un dispositivo de retroalimentación, como un potenciómetro o un codificador. Este dispositivo proporciona información sobre la posición actual del motor y permite al sistema de control ajustarla según sea necesario.

Ahora el controlador es el cerebro del sistema y se encarga de recibir señales de entrada, como

las órdenes de posición o velocidad deseadas, y compara esta información con la retroalimentación del motor. Luego, calcula la diferencia (error) entre la posición deseada y la posición actual y ajusta la señal de control enviada al motor para minimizar ese error y lograr la posición deseada [25].

2.2.12 Señal PWM

La señal PWM (Pulse Width Modulation, en inglés) es una técnica de modulación utilizada en electrónica y electricidad para controlar la cantidad de energía entregada a un dispositivo, como un motor o una luz, al variar el ancho de un pulso de señal digital. En una señal PWM, se modula el ancho (duración) del pulso de la señal, mientras que la frecuencia se mantiene constante.

La señal PWM se utiliza para controlar la velocidad de motores, el brillo de luces LED, la intensidad de calefactores y muchas otras aplicaciones en las que se necesita variar la potencia suministrada. Al cambiar el ciclo de trabajo de la señal PWM, se puede controlar la cantidad de energía entregada al dispositivo. Por ejemplo, en un motor DC, un ciclo de trabajo más alto aceleraría el motor, mientras que un ciclo de trabajo más bajo lo haría funcionar más lentamente.

La señal PWM es especialmente útil en sistemas de control, como en microcontroladores y sistemas embebidos, donde se necesita un control preciso de la potencia entregada a los dispositivos. También es ampliamente utilizada en sistemas de regulación y control de procesos industriales [26].

2.2.13 COMUNICACIÓN SERIAL

La comunicación serial se le denomina al envío de datos de un bit en una misma instancia en manera secuencial ya sea en un canal de comunicación como también por un bus.

La comunicación serial tiene su ventaja, y esta es que solo necesita un pequeño número de líneas de transmisión para que una comunicación paralela pueda transmitir dicha información o datos. A diferencia de la comunicación en paralelo la cual si necesita más líneas de transmisión al igual que el número de bits que conforman la información. La desventaja de proponer una comunicación en paralelo es que pueden surgir muchos problemas a la hora de transmitir un gran número de bits como es el caso de interferencia, así como como también el de sincronización [27].

2.2.14 CÁMARA

Es una pequeña cámara digital la cual puede ser conectada a cualquier computador que soporte el video en tiempo real y la cual puede capturar imágenes y transmitirlas a través de Internet, o algunos otros dispositivos mediante módulos. Existen otras cámaras que de forma autónoma necesitan un punto mismo de acceso a la red, es bien usada para diversas tareas cotidianas como videoconferencia, conforme pasan los años las empresas que las fabrican van mejorando en la calidad de imagen [28].

2.2.15 BANDA TRANSPORTADORA

La banda o también llamada cinta transportadora se usa mayormente en la industria y su función principal es trasladar objetos de un lugar a otro según sea el requerimiento lo cual beneficia a la mano de obra ya que el avance es rápido y no requiere utilizar algún tipo de fuerza humana.

Se usa una banda transportadora diseñada para satisfacer las especificaciones del proyecto, el cual consiste únicamente en desplazar la manzana hacia la sección de clasificación. Esto se logra mediante el uso de un motorreductor DC conectado mediante un sistema con poleas a los rodillos que moverán la banda [29].

2.2.16 LENGUAJE DE PROGRAMACIÓN

Lenguaje de programación es aquel conjunto de líneas de código o instrucciones mediante la cual el ser humano es capaz de interactuar con las computadoras, esto de manera general.

Mediante el uso de algoritmos específicos se detalla el tipo de instrucción que se desea que realice una máquina, para ello estas instrucciones deben ser ordenadas y concretas para que una vez compiladas puedan ser leídas por la máquina [30].

2.2.17 INTERFAZ GRAFICA DE USUARIO

Se llama GUI por sus siglas en inglés (Graphical User Interface) y su denominación proviene debido a la relación o entre la persona y la máquina. Tiene como objetivo representar el código backend de algún sistema con tal que se le pueda facilitar el su uso y la visualización del mismo. Esto se logra mediante el uso de iconos, imágenes, tablas, diagramas, y demás herramientas para complementar la facilidad de uso. Una GUI se usa comúnmente para facilitar el uso y control de PC, Tablet y demás dispositivos tecnológicos, así como también

aplicaciones web o de escritorio [31].

2.2.18 SISTEMAS AUTOMATIZADOS

Un sistema automatizado es un conjunto de componentes, dispositivos o software que realiza tareas o procesos de forma autónoma o semiautónoma sin una intervención humana constante. Estos sistemas están diseñados para ejecutar acciones de manera sistemática y predefinida, siguiendo instrucciones programadas o reglas preestablecidas. La automatización se utiliza en una amplia variedad de aplicaciones y sectores, desde la industria manufacturera y la tecnología de la información hasta la domótica y la robótica.

Algunas características clave de los sistemas automatizados incluyen:

- Control preprogramado: Los sistemas automatizados funcionan según un conjunto de instrucciones o reglas predefinidas. Estas instrucciones pueden ser proporcionadas mediante software, programación, sensores, o una combinación de estos.
- Reducción de la intervención humana: Los sistemas automatizados minimizan la necesidad de intervención o supervisión humana, lo que puede aumentar la eficiencia, reducir errores y, en algunos casos, mejorar la seguridad.
- Repetibilidad: Los sistemas automatizados pueden ejecutar tareas de manera consistente y repetitiva, lo que es especialmente valioso en procesos industriales y de fabricación.
- Velocidad y precisión: Los sistemas automatizados pueden realizar tareas a velocidades constantes y con alta precisión, lo que es importante en aplicaciones que requieren control exacto.
- Monitoreo y retroalimentación: Muchos sistemas automatizados incluyen sensores y sistemas de retroalimentación que les permiten monitorear su entorno y ajustar su comportamiento en función de las condiciones cambiantes.

Algunos ejemplos comunes de sistemas automatizados incluyen líneas de ensamblaje de fábricas, sistemas de control de tráfico, sistemas de gestión de inventario, sistemas de climatización y control de iluminación en edificios, sistemas de automatización de procesos químicos, sistemas de control de vuelo en aeronaves y sistemas robóticos.

La automatización puede mejorar la eficiencia, la productividad y la calidad en una variedad de aplicaciones, y es una parte fundamental de la industria 4.0 y la revolución industrial actual, donde la interconexión y la automatización avanzada están transformando la fabricación y otros sectores [32].

2.3 MARCO TEÓRICO

Para la realización del presente trabajo se llevó a cabo un profundo a análisis de artículos así como también proyectos de titulación tanto locales como de otras instituciones, los cuales aportaron de manera productiva a la elaboración de este proyecto.

"Clasificación de manzanas utilizando visión artificial y redes neuronales artificiales", publicación de la revista INGENIERIA Y REGION del país de COLOMBIA, aquí inicia la investigación y lleva a evaluar que tan factible es usar redes neuronales para la clasificaron de frutas, la información obtenida del trabajo mencionado es fructífera porque se llega al punto de usar redes Neuronales Convolucionales logrando porcentajes de predicción demasiado altos a comparación de otros tipos de técnicas usados en visión artificial. En otros trabajos se ve cómo se desenvuelven mejor las redes convolucionales en diversos tipos de sistemas de predicción como es el caso del uso de TensorFlow, que contiene un gran número de módulos, modelos pre entrenados, e información para que el sistema que se requiera usar sea estable y confiable [33].

"Procesamiento de imágenes para la clasificación masiva de frutos basado en el color" En el proyecto mencionado se emula un modelo tipo escala de una maquina industrial la cual contiene varias líneas, dichas líneas compuestas por rodillos los cuales proporcionan la movilización del objeto a clasificar, en este tipo de trabajo se lo realiza a través de una cámara para la captura de imágenes en tiempo real. Consiguiendo así imágenes precisas para ser analizadas y posteriormente predecir con alto índice de aceptación [34].

CAPITULO III

3.1 COMPONENTES DE LA PROPUESTA

En este capítulo existen dos secciones las cuales corresponden a componentes físicos y componentes lógicos del proyecto.

3.1.1 COMPONENTES FÍSICOS

3.1.1.1 MOTOREDUCTOR AC 110V

El motor AC de inducción (ver figura 7) se lo utiliza para movilizar la cinta transportadora de la banda a través de un sistema de poleas, ofrece los engranajes de eje paralelo de alta resistencia que maximizan el rendimiento del motor con un cableado sencillo y opciones de montaje flexibles. Debido a su voltaje de operación 110V, como se indica en la tabla 1 correspondiente a las especificaciones del motorreductor AC, puede operar en cualquier lugar sin problemas de alimentación [35].

Figura 7. Motorreductor Ac

Tabla 1. Especificaciones Motorreductor AC

Motorreductor AC	
Motor	AC de inducción
Tipo	De dientes Rectos
Orientación de los ejes	Coaxial
Potencia	10 W – 50W
Voltaje de operación	110V – 220V
Par	20-50 Nm

3.1.1.2 SERVOMOTOR MG995

Un servomotor es aquel actuador eléctrico que dispone de un motor de corriente continua pero que se diferencia de los demás motores Dc porque está diseñado para conseguir un giro con determinado Angulo en vez de un giro continuo. El modelo de servomotor a usar es el TOWER PRO MG995 (ver figura 6) el cual es compatible con arduino y su función principal se encuentra en la sección de clasificación que va a ayudar a mover un plato rectangular a través de un eje que se acopla al servomotor. Debido a su voltaje de operación de 4,8V se puede fácilmente energizar a más de su torque de 9,4 kg/cm como se ve en la tabla 2 de especificaciones del servomotor MG995 suficiente para hacer girar la paleta [36].

Figura 6. Servomotor MG995.

Tabla 2. Especificaciones Servomotor MG995

SERVOMOTOR	
Modelo	MG995
Peso	55g
Dimensiones	40,7 x 19,7 x 42,9 mm
Par de torsión	9,4 kg/cm (4,8V) ; 11kg/cm(6v)
Voltaje de operación	4,8V - 6,6V
Tipo de engranaje	Metal Gear

3.1.1.3 PLACA ARDUINO R3

La placa arduino R3 (ver figura 7) es un dispositivo electrónico compatible con el micro controlador ATmega328p como se muestra en la tabla 3, con el cual se tiene habilitado 14 pines de entrada y salida. Se dispone de este dispositivo para realizar el correcto enlace con

los actuadores eléctricos del proyecto, una de sus salidas está conectada directamente al servomotor MG995, el cual debido a su alta compatibilidad con estos dispositivos brinda una alta confiabilidad al momento del uso [37].

Figura 7. Placa Arduino UNO R3.

Tabla 3. Especificaciones Arduino uno R3

PLACA ARDUINO	
Modelo	Uno R3
Micro controlador	ATmega328p
Voltaje de operación	5V; 7v-12V
Pines de E/S	14 pines
Memoria Flash	32kb
EEPROM	1KB
Velocidad de reloj	16 MHz
Peso	25g

3.1.1.4 MICROPROCESADOR INTEL CORE I5 1135G7

El Intel Core i5-1135G7 (ver figura 8) es un procesador de la familia Tiger Lake de Intel, que se lanzó en 2020. Pertenece a la 11ª generación de procesadores Intel Core, que se basa en la arquitectura Willow Cove [38]. Tiene una frecuencia base de reloj de 2.4 GHz y puede alcanzar una frecuencia máxima de hasta 4.2 GHz en modo turbo tal como se muestra en sus especificaciones en la tabla 4.

Figura 8. Intel Core 1135G7

Tabla 4. Especificaciones Intel Core 1135g7

PROCESADOR INTEL CORE	
Tipo	I5
Modelo	1135G7
Cantidad de núcleos	4
Cantidad de subprocesos	8
Frecuencia	4,20 GHz
Cache	8MB
Velocidad de bus	4GT/s
TDP	28W

3.1.1.5 FACECAM VGA

La cámara a usar es el modelo FaceCam VGA (ver figura 9) de la marca Genius con resolución de 1000x 720 HD como se puede observar en la tabla 5 de especificaciones. Es ideal para conferencia la cual se puede experimentar conferencias en alta definición, posee enfoque manual para una mejor visibilidad de acuerdo al trabajo de visión que se requiere usar, además de traer un micrófono incorporado. La definición es suficiente para usar en visión artificial. Por lo cual fue escogida para el procesamiento de imágenes en tiempo real usado en el proceso de clasificación de manzanas [38].

Figura 9. FACECAM VGA.

Tabla 5. Especificaciones FACECAM VGA

FACECAM VGA	
Modelo	Facecam 1000x
Interfaz	USB
Longitud cable	1,5M
Sensor de imagen	CMOS de pixeles de alta definición 720p
Tipo de Lente	enfoque manual
Blanco y Balance	Automático y Manual
Angulo de visión	60°
Micrófono	si

3.1.2 COMPONENTES LÓGICOS

Se detallan a continuación todo componente lógico utilizado en el proyecto tales como aplicaciones, módulos, simuladores y demás herramientas de software.

3.1.2.1 PYTHON

Lenguaje de programación de alto nivel cuya licencia de código es abierta (ver logotipo en figura 10), es capaz de soportar programación orientada a objetos, programación imperativa, también se usa para programación funcional, una de sus ventajas es que se lo conoce como lenguaje interpretado, así como dinámico. Su uso también se radica en ser multiplataforma con lo cual hace posible que sea compatibles con diversas aplicaciones, su uso en el presente proyecto data como base del sistema de clasificación, comunicación y visión artificial en tiempo real [39].

Figura 10. Logotipo de Python

3.1.2.2 OPENCV

Desarrollada como biblioteca libre aplicable para visión artificial (ver logotipo en figura 11), su compatibilidad con python permite el uso de todos sus módulos para el uso de visión por

computadora, su lenguaje de programación se basa en C ++ optimizado, su interfaz es sencilla logrando un manejo fácil de aprender para el usuario [40].

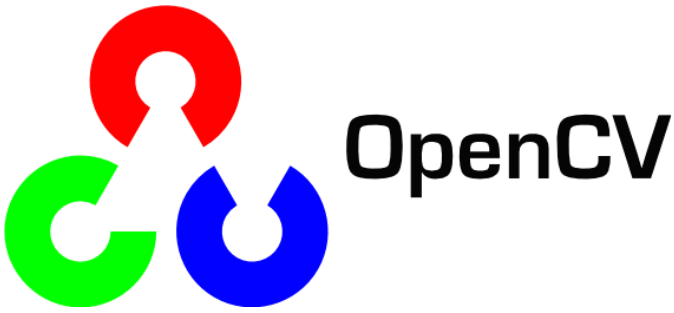


Figura 11. Logotipo de OpenCv.

3.1.2.3 TENSORFLOW

Biblioteca de código abierto que se usa para el aprendizaje automático mediante un gran número de tareas designadas para un resultado específico (ver logotipo en figura 12). Desarrollada por google para crear sistemas capaces de construir redes neuronales que son capaces de detectar y decifrar patrones alcanzando un alto grado de predicción, su amplia gama y su compatibilidad con python la hacen eficaz para el uso en el presente proyecto [41].

Figura 12. Logotipo de Tensorflow

3.1.2.4 MEDIAPIPE

Biblioteca multiplataforma que fue creada por la compañía de Google (ver logotipo en figura 13) la cual puede ejecutar gran variedad de aplicaciones de aprendizaje según el diseño que se requiera dar en el área de visión computarizada. Además, la biblioteca permite realizar las ejecuciones de las aplicaciones de una manera óptima de acuerdo al rendimiento de nuestro CPU, lo cual facilita el uso de modelos entrenados complejos sin ningún problema de delay [42].

Figura 13. Logotipo del Módulo MediaPipe.

3.1.2.5 KERAS

En el presente proyecto se la conoce como una biblioteca de python que trabaja con machine learning y redes neuronales (ver logotipo en figura 14), la cual se la puede ejecutar sobre TensorFlow. Está diseñada para reducir el tiempo de ejecución de algoritmos complejos basados en redes de aprendizaje profundo. En general posee soporte para redes neuronales convolucionales y redes neuronales recurrentes [43].

Figura 14. Logotipo de la biblioteca Keras.

3.1.2.6 PROTEUS

Es una herramienta de software usada para diseño de circuitos electrónicos (ver logotipo en figura 15) y diversas aplicaciones de la misma área como diseño de esquemas electrónicos, elaboración de la placa del circuito, depuración y programación, etc. Con este software se desarrolla el diseño y simulación de la etapa de clasificación para nuestro sistema, el cual tiene un punto importante ya que así se puede corroborar el sistema de forma simulada.
Teniendo una simulación bien elaborada en su diseño se puede pasar a la etapa del diseño físico, así se evitará dañar cualquier componente electrónico antes de subir algún código [44].

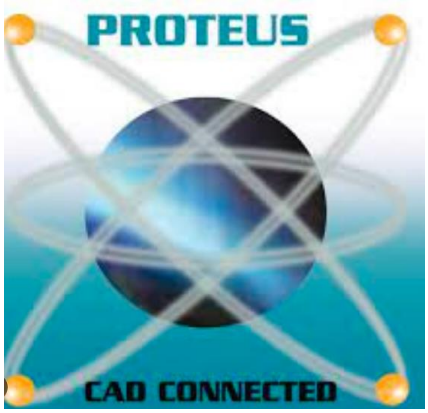


Figura 15. Logotipo del software Proteus.

3.1.2.7 VIRTUAL SERIAL PORT

Es una aplicación capaz de simular puertos seriales COM de manera virtual en un computador (ver logotipo en figura 16), con esta aplicación se logra la comunicación entre nuestro sistema y el diseño de clasificación desarrollado en Proteus. De tal forma que se logra tener una mejor visualización del funcionamiento del sistema de forma simulada [45].

Figura 16. Logotipo de la herramienta Virtual Serial Port.

3.1.2.8 ARDUINO IDE

Software multiplaforma en el entorno de desarrollo JAVA (ver logotipo en figura 17) para los componentes electrónicos de la compañía ARDUINO, permite programar bloques para entradas y salidas de los micro controladores en este caso el ATmega328p el cual montado sobre la placa ARDUINO UNO R3 formara parte del sistema de la clasificadora, además de sus algoritmos multiplataforma permite la interacción con otras aplicaciones y componentes [46].

Figura 17. Logotipo del software ARDUINO IDE.

3.1.2.9 TEACHABLE MACHINE

Desarrollado por google a través de una comunidad de personas dedicadas al machine learning (ver logotipo en figura 18), permite realizar diferentes tipos de entrenamientos de redes neuronales artificiales para que un computador sea capaz de reconocer imágenes, sonidos, posturas según lo requerido.

Se puede crear modelos automatizados para aplicaciones web, móviles, computadores personales, etc., de una manera fácil. La mayor ventaja es que el modelo es enviado a servidores de alto rendimiento de google para ser entrenado en menos tiempo, de tal manera que no se requiere un computador con componentes sofisticados [47].

Figura 18. Logotipo del software TECHEABLE MACHINE

3.1.2.10 PYSERIAL 3.4

Es una biblioteca de python (ver logotipo en figura 19) capas de poder comunicar el entorno de desarrollo de Python con cualquier puerto serial conectado al computador donde se está trabajando.

Esta biblioteca es usada para establecer la comunicación entre python y uno de los puertos donde estará conectado el arduino UNO REV3 ya sea que la placa arduino este de forma física o simulada [48].

Figura 19. Logotipo de PySerial.

3.1.2.11 QTDESIGNER

Es una herramienta para diseñar y graficar interfaces de usuario (ver logotipo en figura 20) con soporte para sistemas operativos LINUX, MAC, WINDOWS, además de tener bibliotecas compatibles con plataformas de programación como python, lenguaje C, java script, entre otros.

Debido a su alta fiabilidad con bibliotecas de python es capaz que enlazar la interfaz gráfica y añadirla al proyecto con redes neuronales ya que sus bibliotecas permiten la ejecución mediante hilos y crea una alta tasa de respuesta en tiempo real si se usa algún tipo cámara [49].

Figura 20. Logotipo de QT Designer.

3.2 DISEÑO DE LA PROPUESTA

3.2.1 DISEÑO DE LA BANDA TRANSPORTADORA

La banda transportadora se diseñó de tal manera que pueda desplazar la manzana, esto se logra mediante un motorreductor el cual conectado a sistema de poleas logra el movimiento y el torque necesario para los propósitos requeridos. Como se puede ver en la figura 21 y figura 22 sus dimensiones son compactas debido a que el tamaño de la manzana no es muy grande.

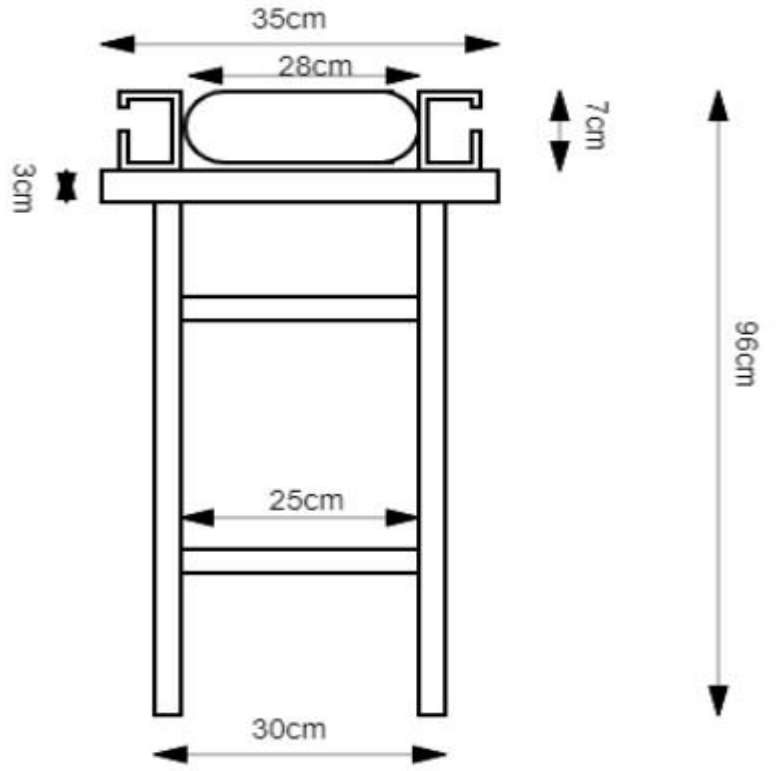


Figura 21. Vista lateral del diseño de la banda transportadora

Figura 22. Vista Superior del diseño de la banda transportadora

3.2.2 IMPLEMENTACIÓN DE LA BANDA TRANSPORTADORA TIPO CINTA

De acuerdo al diseño un técnico especializado en trabajos y soldadura industrial fabricó toda la estructura con las medidas solicitadas. Para la cinta se usó un tipo de plancha de goma sintética la cual fue cocida y puesta sobre rodillos, los cuales están conectados al motorreductor AC a través de un sistema de poleas, dicho circuito se enciente con un pulsador de señal fija.

El diseño final se lo puede apreciar en la figura 23 donde se muestra ya la banda transportadora tipo cinta implementada.

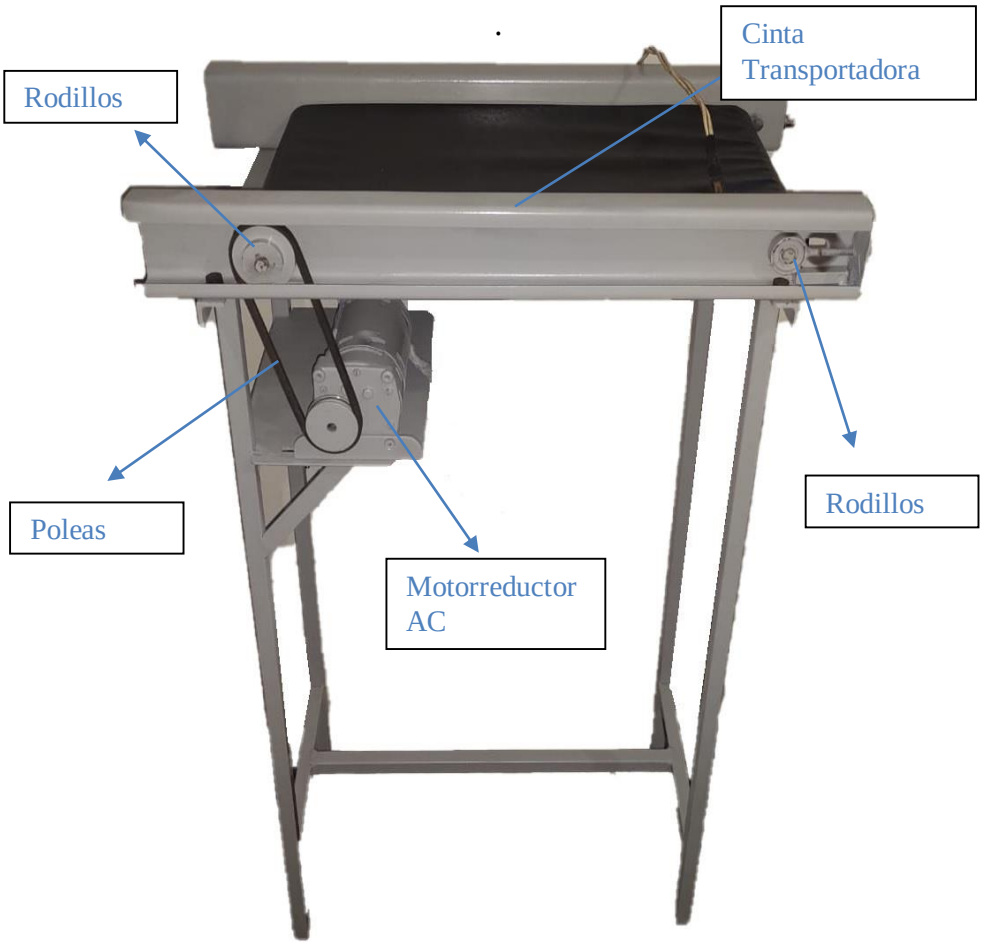


Figura 23. Banda transportadora del sistema.

3.2.3 DISEÑO DE LA CLASIFICADORA DEL SISTEMA.

Se aplica un diseño de clasificadora inclinada, se usa la inclinación de la banda para la caída de la manzana después de pasar a través de la banda transportadora, se tiene conectada una paleta con eje a un servomotor el cual el cual hará girar la paleta de tal manera que queda un orificio en el centro del clasificador por la cual caerá la manzana, caso contrario si la manzana está en buen estado pasará por encima de la paleta hacia el final de la clasificadora. Las dimensiones se las puede observar tanto en la figura 24 como figura 25.

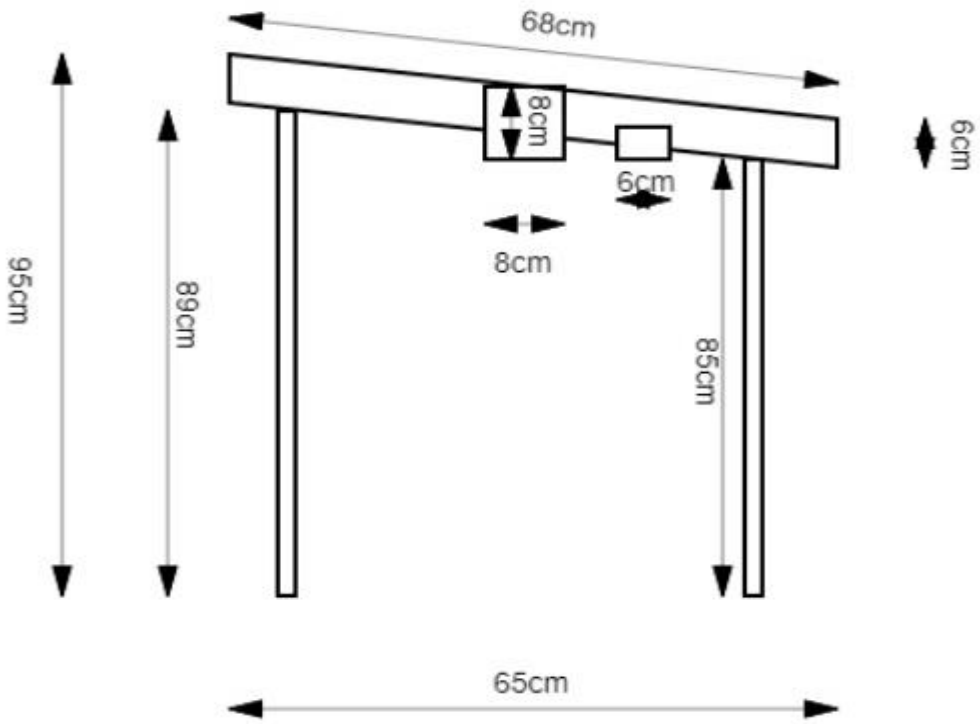


Figura 24. Vista frontal del Diseño de la clasificadora.

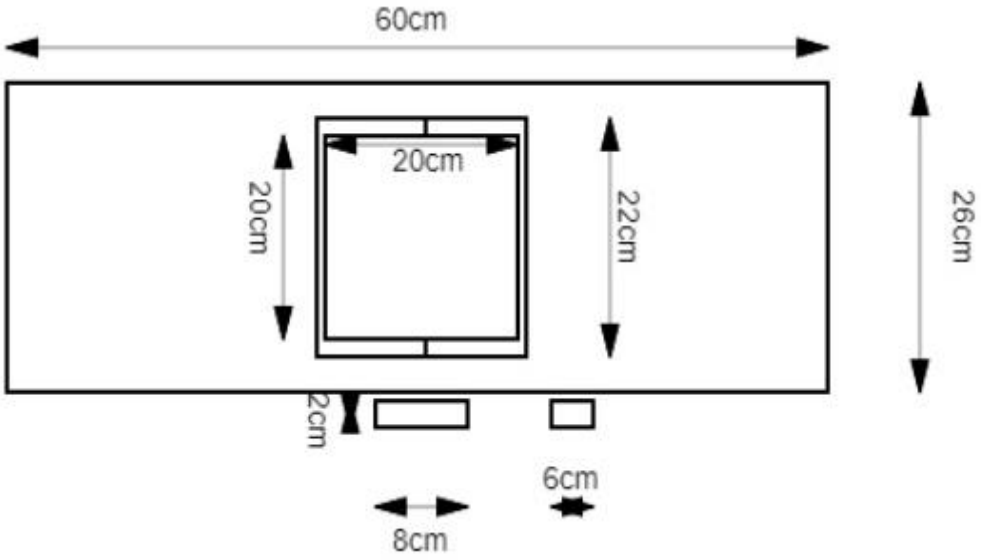


Figura 25. Vista superior del Diseño de la clasificadora.

3.2.4 IMPLEMENTACIÓN DE LA CLASIFICADORA

De acuerdo al diseño se procedió a fabricar la estructura de la clasificadora con las medidas propuestas, dejando una caja en una de las paredes laterales para introducir la placa arduino R3 quien hará la de controlar el giro del servomotor. El eje del Servomotor esta soldado al eje centrado de la paleta, el cual dejara caer la manzana en mal estado, caso contrario no se abrirá

y dejara la manzana seguir su recorrido. Se deja la pared al final del recorrido para recoger las manzanas en buen estado, logrando así el diseño esperado para la etapa de clasificación. La implementación se puede observar en la figura 26.

Figura 26. Clasificadora del sistema.

3.2.5 MONTAJE DE LA CÁMARA

Para una correcta captura de la manzana es necesario tener una excelente posición de la cámara (ver figura 27), de tal forma que se logre captar la mayor cantidad de rasgos de la fruta para su respectiva validación, en este caso la posición optima fue colocar la cámara sobre uno de los rieles de la banda transportadora casi al final de la misma.

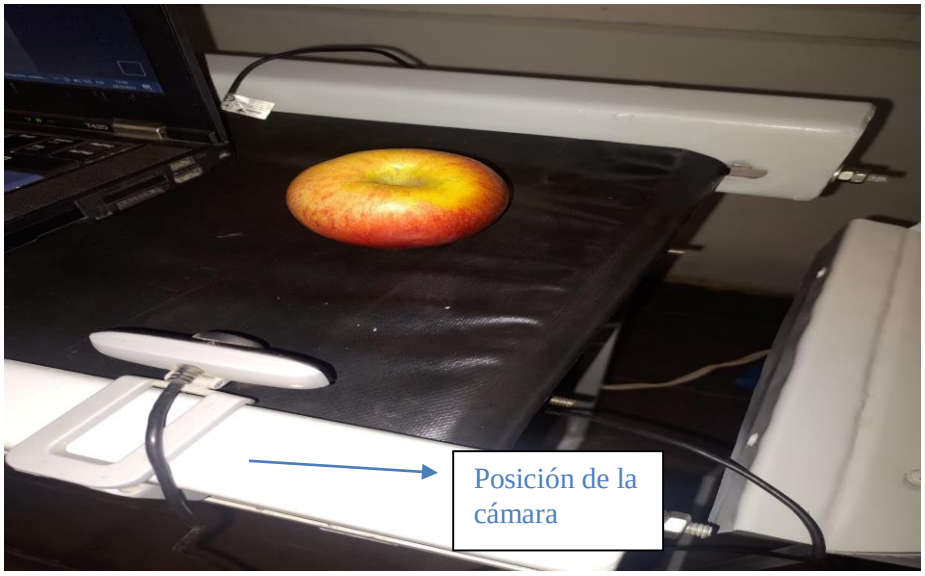


Figura 27. Posición de la cámara para captura de imagen.

En tiempo real se puede apreciar que la captura de imagen (ver figura 28) logra concentrar gran parte de visualización en la manzana, así se obtiene gran cantidad de información para ser validada a través de la red neuronal ya entrenada.

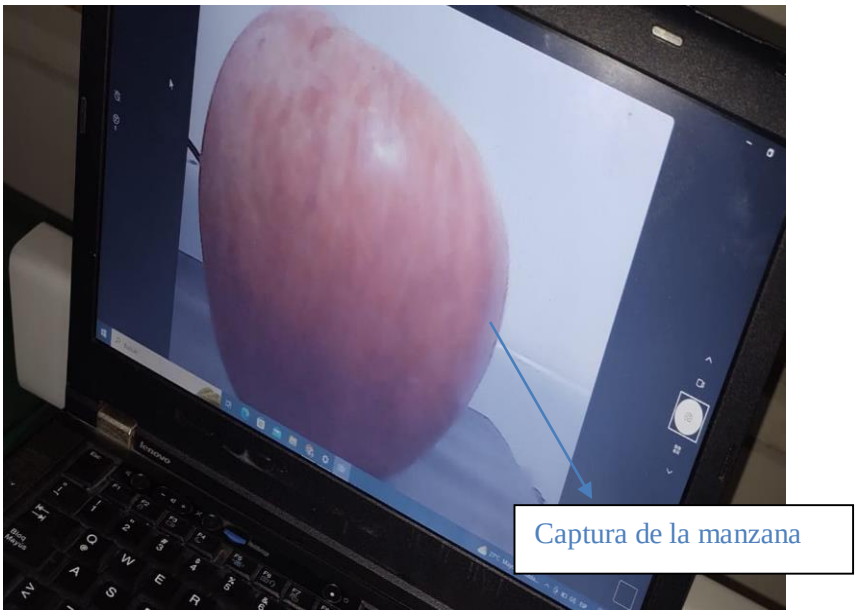


Figura 28. Visualización en tiempo real de la manzana.

3.2.6 DIAGRAMA DE CONEXIONES DEL SISTEMA

La manzana es desplazada hacia la sección de validación y luego hacia la sección de clasificación, se usa una banda transportadora diseñada solo para el desplazamiento de la manzana. El computador a usar es una laptop Lenovo Thinkpad T420, la cual se va encargar de ejecutar los procesos de cómputo usados por el sistema en su parte lógica, así como también de controlar los dispositivos de hardware conectados al sistema. Tanto la parte lógica como la

parte de hardware son ejecutadas y controlada a través de la aplicación python. Dentro de python se logra manipular los datos de imagen adquiridos por la cámara para que sean validados por la red neuronal ya entrenada la cual también se puede usar dentro de python a través de bibliotecas de Tensorflow y Keras, una vez validados los datos se procede a la sección de clasificación que es controlada por un servomotor conectado a arduino, este control se logra controla través de la Liberia PySerial para tener comunicación entre python y arduino. El diagrama de conexiones se lo puede apreciar en la figura 29.

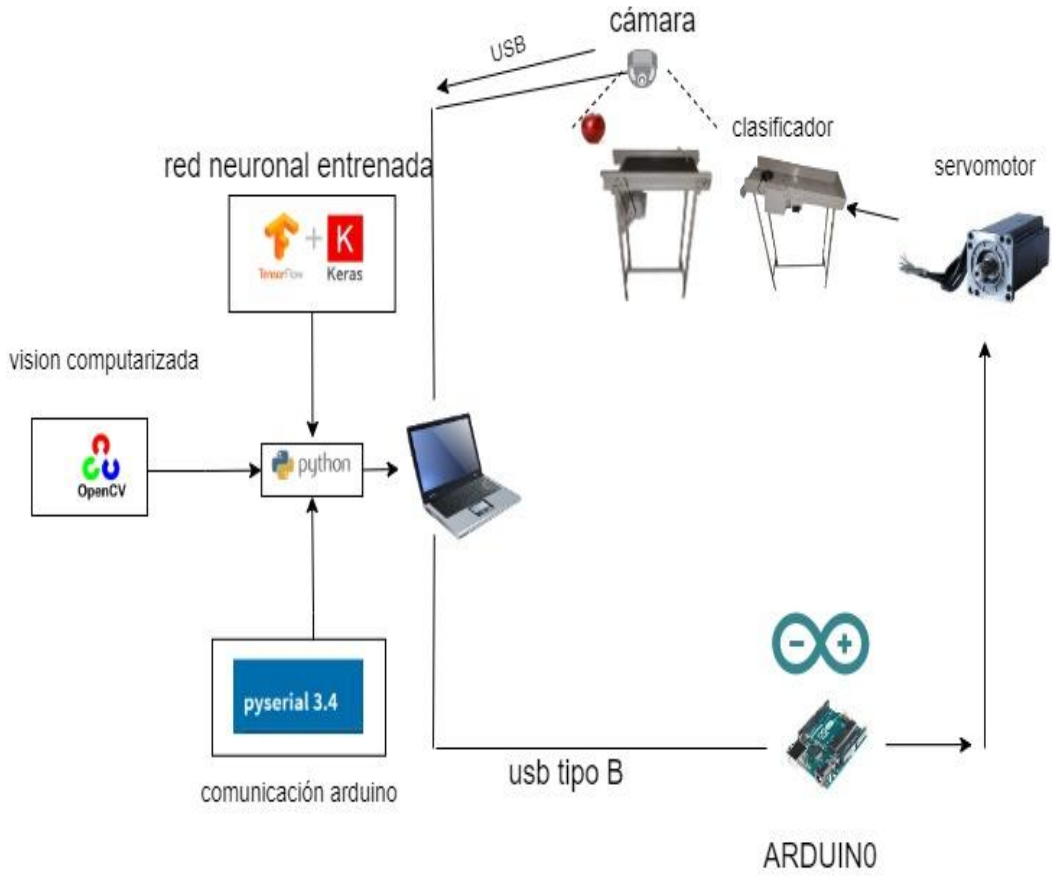


Figura 29. Diagrama de conexiones del sistema.

3.2.7 DIAGRAMA DE CONEXIONES DE ALIMENTACION

En la figura 30 se puede apreciar las conexiones de alimentación, solo basta con una línea de baja tensión de 110 V AC para poder alimentar nuestro sistema. Desde el Motorreductor AC hasta el arduino a través de un adaptador de 110V AC a 12 V Dc.

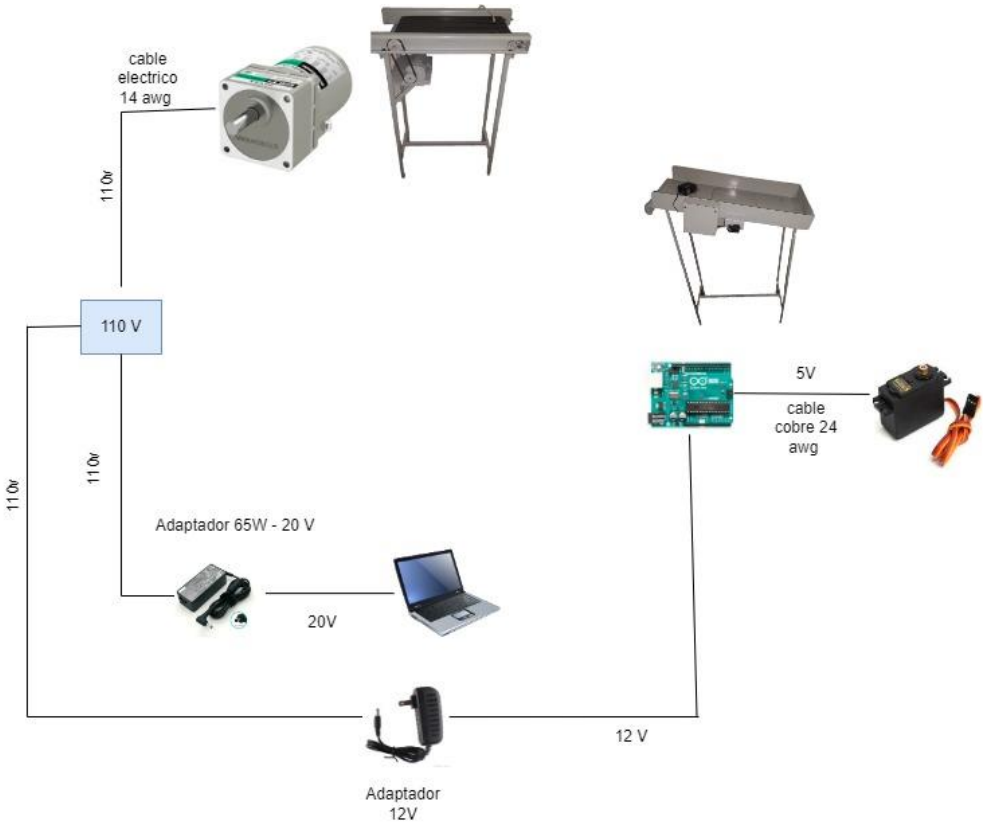


Figura 30. Diagrama de conexiones de alimentación

3.2.8 DIAGRAMA DE ENTRENAMIENTO DE LA RED NEURONAL.

El dataset de las manzanas tanto buenas como malas es enviado a través del internet hacia la aplicación TECHEABLE MACHINE, la cual posee equipos de cómputo especializados en entrenamientos de redes neuronales, estos equipos de alta gama y alto rendimiento logran alcanzar un tiempo récord en entrenar este tipo de sistemas solo en cuestión de minutos, luego la red neuronal ya entrenada es devuelta hacia la aplicación TECHEABLE MACHINE para que el usuario verifique los resultados y pueda importarla desde la aplicación web directamente. En la figura 31 se puede apreciar el diagrama de entrenamiento de la red neuronal.

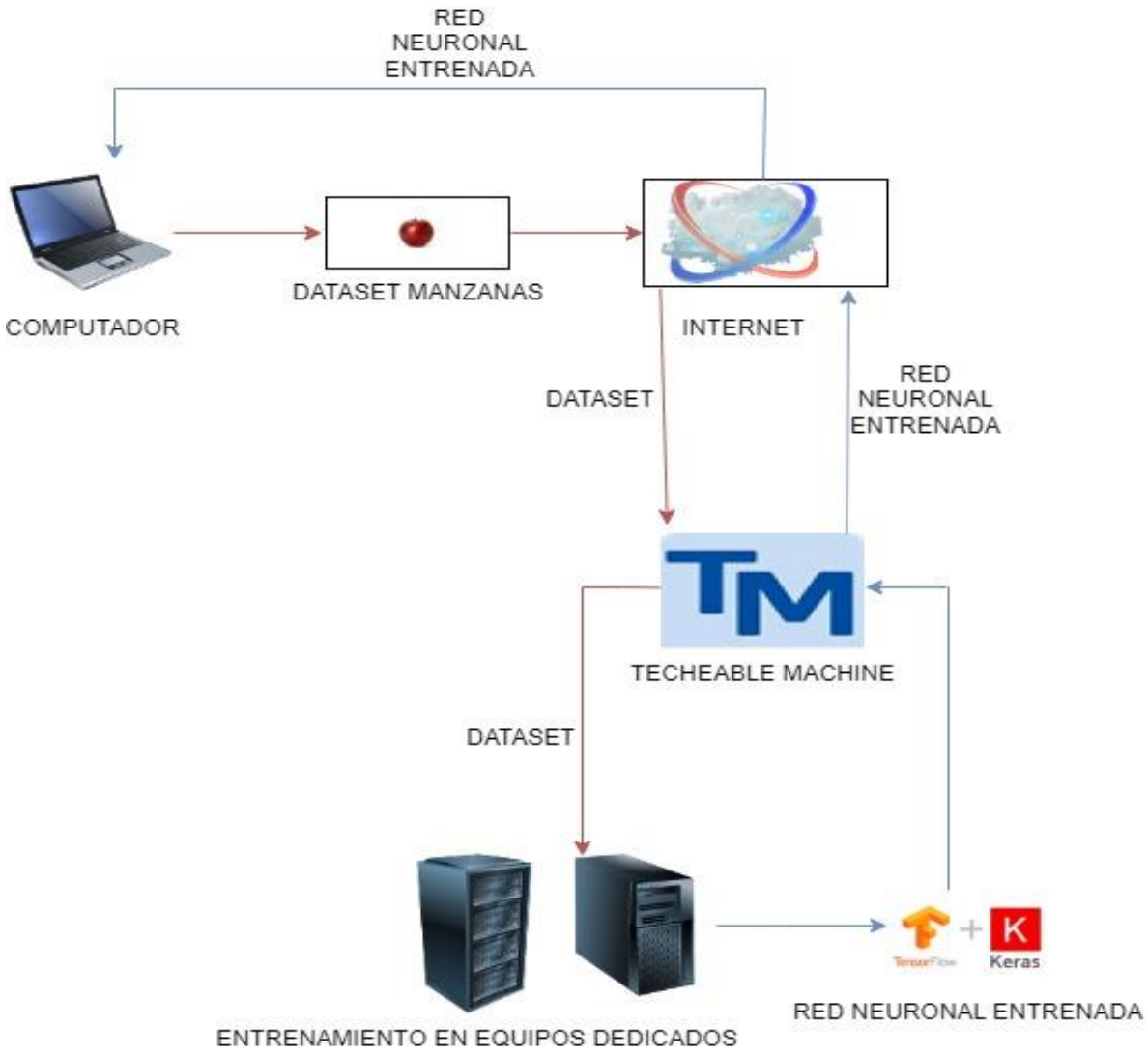


Figura 31. Diagrama del entrenamiento.

3.2.9 DATASEET

3.2.9.1 ADQUISICIÓN DE DATOS

Dataset es el conjunto de muestras recolectadas para el entrenamiento de la red neuronal, para el sistema se adquirió 3 tipos de dataset, uno para manzanas buenas, otro para manzanas malas, y otro para ambiente vacío o cuando no existe fruta.

Esto es importante porque la red neuronal debe aprender a reconocer inclusive cuando no existe la manzana dentro de la captura de imagen en tiempo real, de lo contrario se corre el riesgo de que arroje resultados de predicción erróneos. La selección se hizo a base a la norma para inspección de calidad de manzanas de la OECD.

3.2.9.2 SELECCIÓN DE MANZANAS BUENAS

Es necesario seleccionar las manzanas que mejor están conservadas y en mejor estado (ver figura 32), se debe escoger aquellas que no tienen golpes, tampoco puntos blandos o presentan algún tipo de corte, no deben tener algún tipo de plaga y mucho menos estar en estado inicial de podredumbre, deben tener buen color de madurez para este caso se trabaja con manzanas rojas.

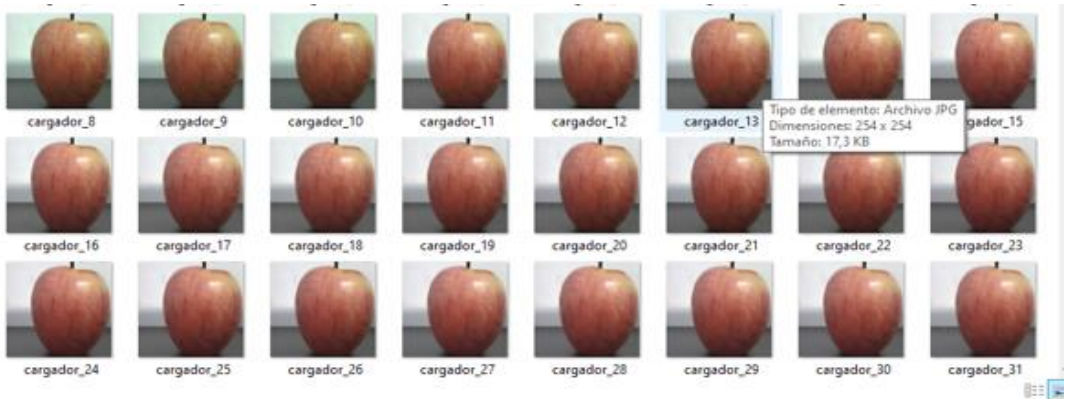


Figura 32. Parte de la selección de dataset para manzanas buenas.

3.2.9.3 SELECCIÓN DE MANZANAS MALAS

En esta parte se escoge las manzanas que no se han conservado en buen estado (ver figura 33), es decir aquellas manzanas que han sido maltratadas en post cosecha, presentan golpes o que ya estén en estado de podredumbre, si presentan algún tipo de plaga. Con toda esta información se logra un mejor entrenamiento para este tipo de clasificación.

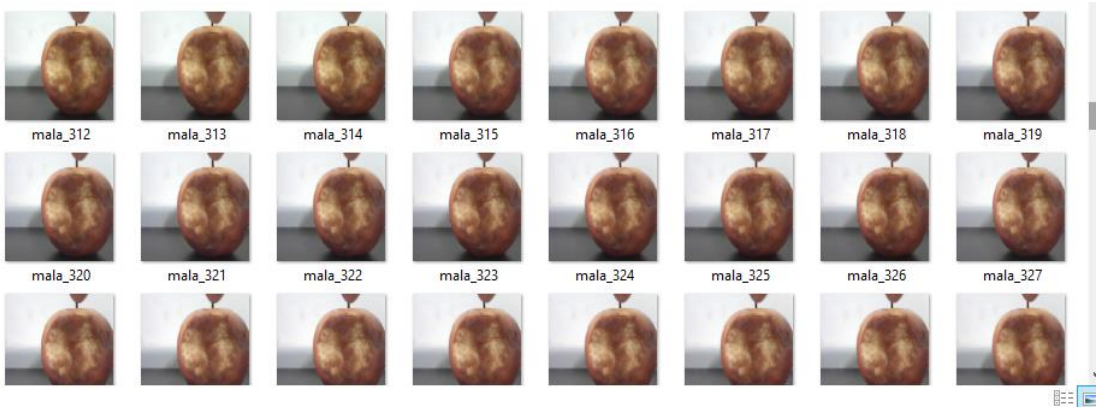


Figura 33. Parte de la selección de dataset para manzanas dañadas.

3.2.9.4 SELECCIÓN DE AMBIENTE SIN MANZANA

Para un correcto funcionamiento del sistema de clasificación se optó por agregar un tercer tipo de clasificación, este corresponde a cuando no existe manzana alguna que analizar (ver figura 34), para ello se toma capturas de cuando no existe la fruta en la banda y esos datos son guardados y enviados para el entrenamiento, de esta manera se logra estabilizar los resultados de predicción de la red neuronal entrenada.

Figura 34. Ambiente sin fruta para el entrenamiento.

3.2.9.5 DISEÑO DEL PROGRAMA PARA EL DATASEET

Python junto a su biblioteca de OpenCv permite múltiples usos en lo que se refiere a visión computarizada. Para el presente trabajo se hace uso de las funciones de captura en tiempo real, la cual permite hacer capturas de lo que la cámara está mostrando. A parte se puede dimensionar las imágenes, de tal manera que coincidan con las dimensiones óptimas que va a utilizar la aplicación web TEACHEABLE MACHINE para el enteramiento, estas dimensiones deben ser: 224x224px como también 200x200px.

3.2.10 DISEÑO CON PROTEUS

Para un mejor desarrollo del proyecto se decidió realizar una simulación en la herramienta de software PROTEUS sobre todo para ver el funcionamiento virtualizado del servomotor (ver figura 35).

Así poder detectar cual falla en la sección de clasificación y poder corregirla antes de la implementación, esto se realiza con el objetivo de evitar dañar algún componente antes de subir la programación a la placa controladora.

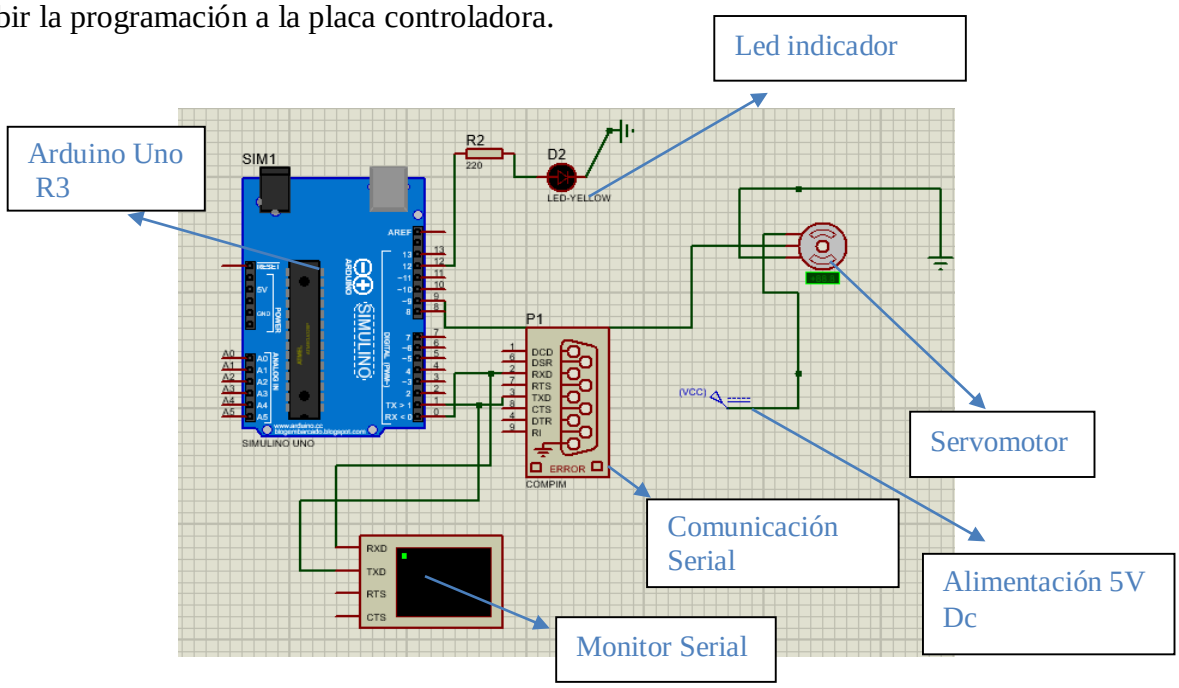


Figura 35. Diseño con Proteus del sistema

3.2.11 CONFIGURACIÓN DE TEACHABLE MACHINE

3.2.11.1 SELECCIÓN DEL TIPO DE PROYECTO

Teachable Machine provee hasta el momento de 3 tipos de proyectos, estos son los siguientes:

A. PROYECTO IMAGEN

Tiene como objetivo entrenar redes neuronales basadas dataseet de imágenes como también lo puede hacer usando la aplicación de la cámara web, lo cual facilita una breve adquisición de datos para el entrenamiento, se escoge Proyecto de imagen (Ver figura 36) para fines del sistema de clasificación.

Figura 36. Vista selección Proyecto tipo imagen de Teachable Machine

En esta opción se permite realizar el entrenamiento de un modelo de acuerdo a las clases que se añadan (ver Figura 37), en dichas clases se pueden subir las imágenes del dataseet que se han creado a partir de capturas de manzanas.

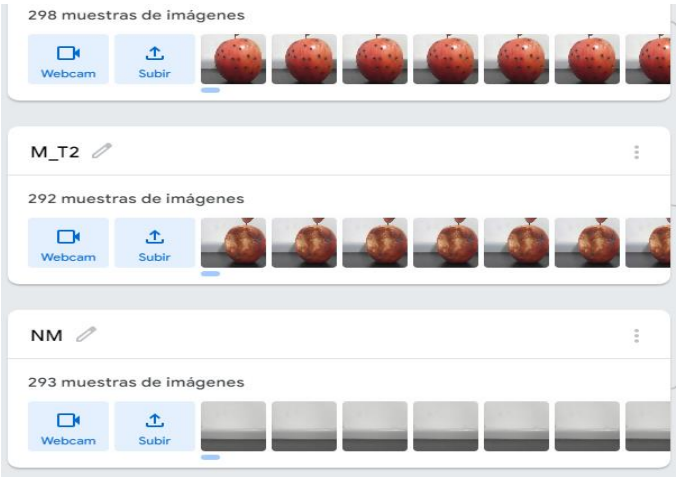


Figura 37. Vista de clases de proyecto imagen.

El dataset (ver figura 39) usado en el presente proyecto se lo realizo a partir de capturas de manzanas con defecto y sin defecto a través de un código de python de tal forma que dichas imágenes que se usaran para el entrenamiento tengan las dimensiones y corrección de coloro adecuados para hacer un correcto entrenamiento.

Figura 39. Vista dataset generado a través de python.

3.2.11.2 SELECCIÓN DEL MODELO

Se selecciona Modelo de imagen estándar (ver figura 40), ya que permite subir los archivos del dataseet con las características con las cuales se realizó las capturas para el entrenamiento.

Figura 40. Vista selección Modelo de imagen estándar

3.2.11.3 AÑADIR CLASES

En este apartado la aplicación web permite añadir muestras de imágenes de acuerdo a la clase requerida (ver figura 41). Las muestras pueden ser imágenes de archivos locales como también se puede usar la aplicación de la cámara web para grabar un video corto donde se muestre el objeto con la clase a entrenar.

Figura 41. Vista del apartado para añadir clases.

3.2.11.4 PREPARACIÓN Y CONFIGURACIÓN DEL MODELO

Aquí se logra configurar como se quiere el entrenamiento del modelo, como el número de épocas, tamaño del lote y la tasa de aprendizaje (ver figura 42).

De acuerdo a los resultados obtenidos en el entrenamiento se puede ir variando los valores para mejorar el porcentaje de predicción de la red neuronal.

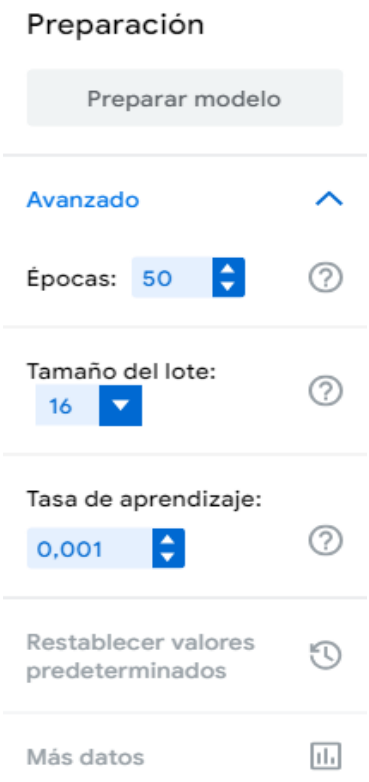


Figura 42. Vista de configuración del modelo.

3.2.11.5 VISTA PREVIA DEL MODELO

En vista previa (ver figura 43) se puede apreciar los resultados obtenidos del entrenamiento y permite probar la red neuronal entrenada para que tan aceptable es el nivel de predicción del modelo.

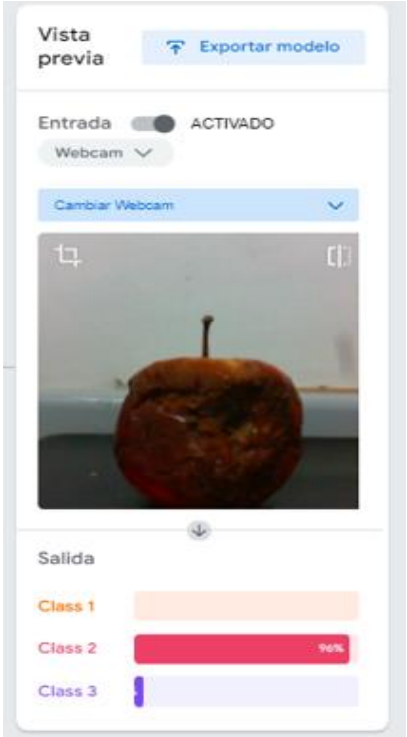


Figura 43. Vista previa de un modelo entrenado.

3.2.11.6 EXPORTACIÓN DEL MODELO

La aplicación web de Teachable Machine al finalizar el entrenamiento permite exportar el modelo entrenado para ser usado en cualquier tipo de proyecto.

Existen 3 tipos de formatos generales de exportación los cuales se escogerá la opción de TensorFlow:

Tensorflow. - En esta opción da la posibilidad de descargar el modelo entrenado y convertirlo en una versión compatible ya sea para usar con KERAS (ver figura 44) o un modelo simple de TensorFlow, debido a las ventajas que posee Keras se opta por descargar este modelo a mas que es compatible con las bibliotecas de python que se usan en el sistema.

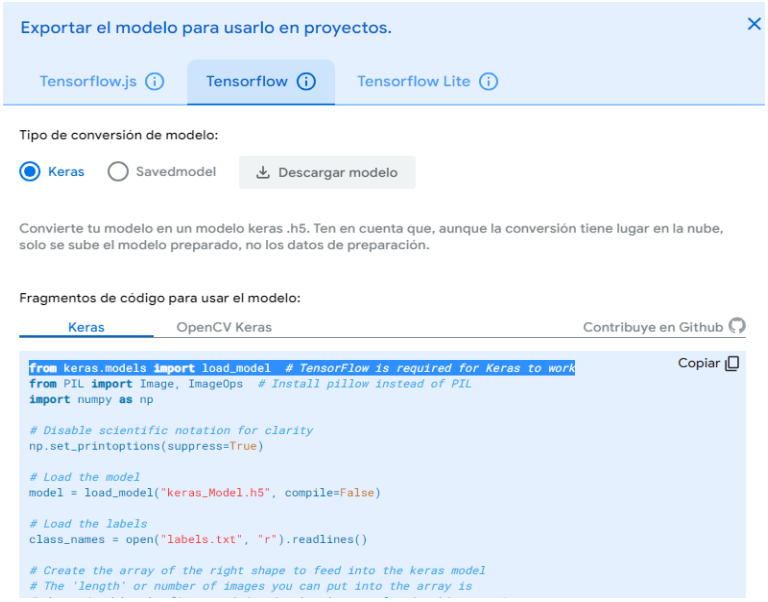


Figura 44. Selección para descargar modelo compatible con KERAS.

3.2.12 DISEÑO DE GUI EN QT DESIGNER

Para el diseño en QT Designer se lo realiza a través de la plantilla Windows (ver figura 45).

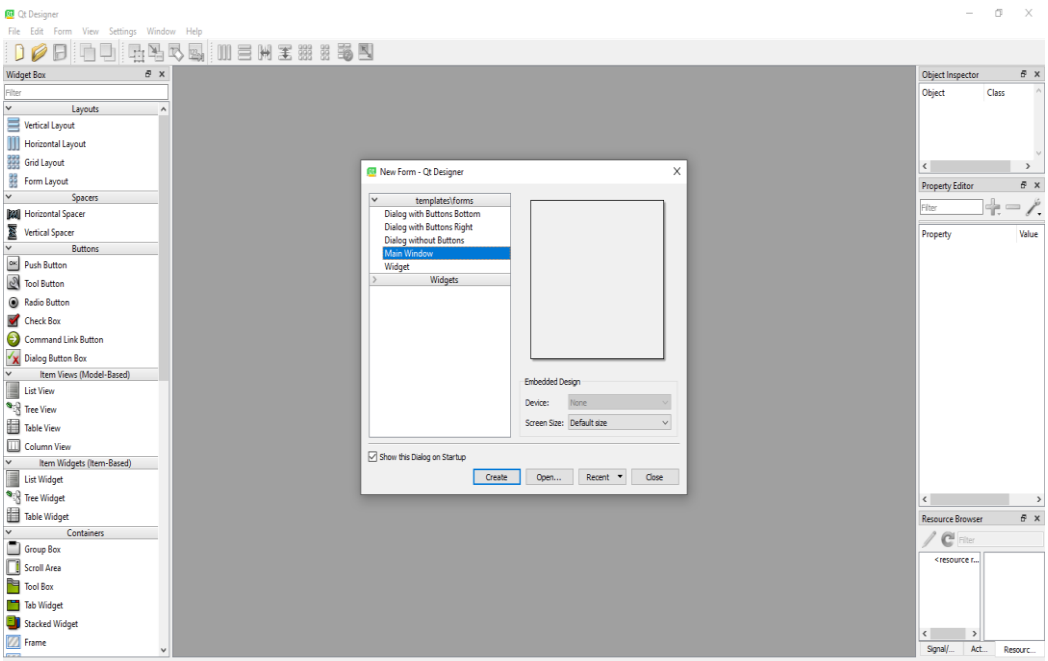


Figura 45. Selección de la plantilla Window.

Se procede a colocar el frame o marco donde se colocará el video captura de la cámara en tiempo real de dimensiones 250 x 160 (ver figura 46).

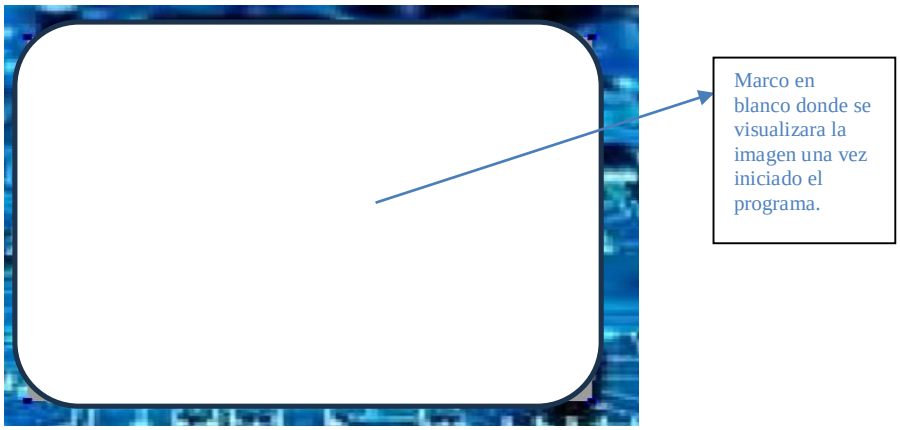


Figura 46. Vista del Frame de captura de video.

Se coloca las etiquetas que permitirán la visualización de los parámetros del proyecto los cuales son FPS, Confidence Score, y Clase (ver figura 47).

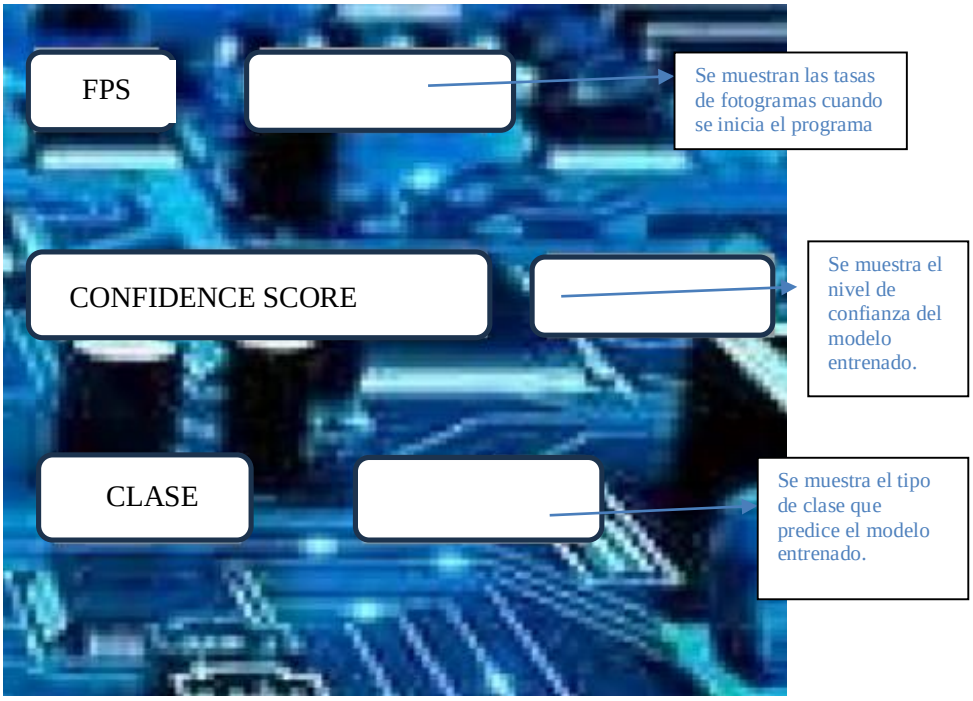


Figura 47. Etiquetas principales del modelo entrenado.

Se procede a colocar los botones que indicaran el inicio y el final del proceso de clasificación (ver figura 48).

Figura 48. Botones principales del GUI.

Se cambia la configuración de parámetros para los botones de tal manera que mejore su apariencia en la presentación dentro del marco principal para la interfaz gráfica (ver figura 49).

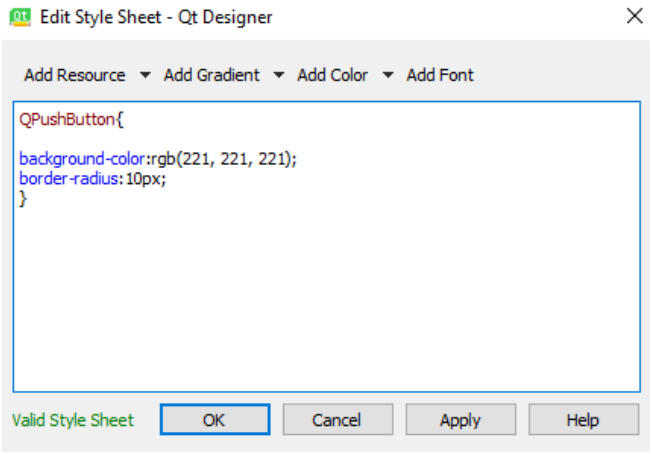


Figura 49. Configuración de parámetros.

Para poder agregar recursos extras es necesario agregar el archivo de recurso al explorador de la plataforma, así poder utilizar estos recursos en la plataforma python (ver figura 50).

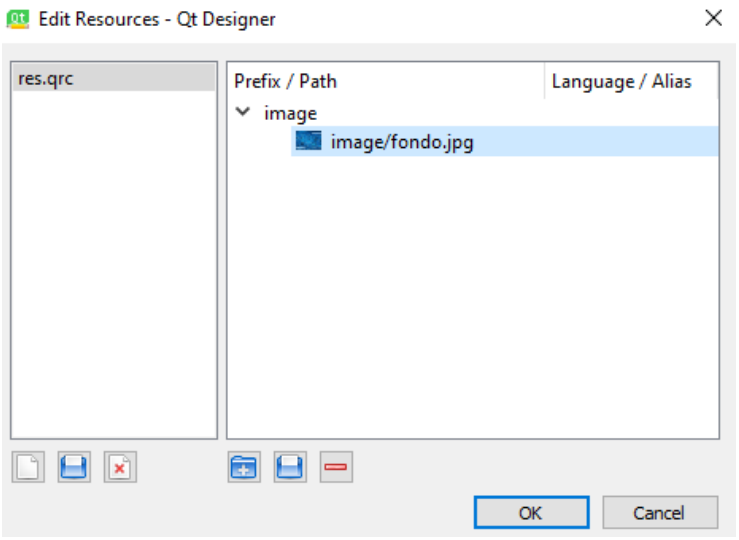


Figura 50. Vista de agregar recursos.

Una vez habiendo terminado el diseño del GUI para el proyecto se exporta el código (ver figura 51) para uso en la plataforma Python

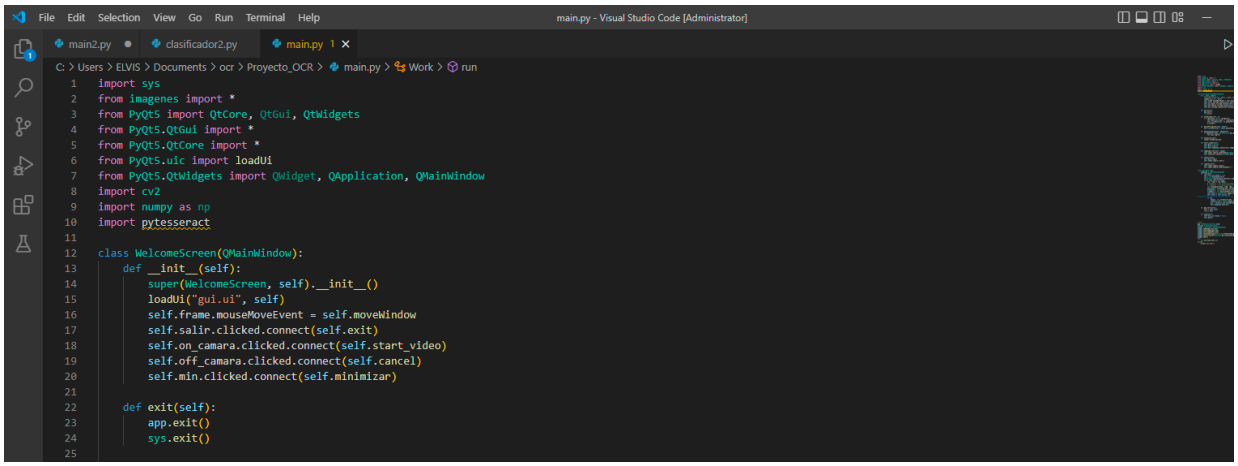


Figura 51. Parte del código del GUI en Python.

3.3 ESTUDIO DE FACTIBILIDAD

3.3.1 FACTIBILIDAD TÉCNICA

Para la banda transportadora se usa un motorreductor reciclado de una camilla de hospital, el cual conectado a un sistema de poleas logra realizar el movimiento de tal manera que desplace a la banda, la estructura de la banda se diseñó de acuerdo a la necesidad del proyecto, para esto se especifica el uso solo el desplazamiento de la fruta, no es activado por algún controlador y esto facilita su uso.

3.3.2 ANÁLISIS DE COSTOS DEL PROYECTO

En los equipos físicos se evaluó las mejores opciones de compra para el presente proyecto de acuerdo a las especificaciones requeridas, esta información se la puede ver en la tabla 6.

Tabla 6. Costos de Equipos

CANT.	DESCRIPCIÓN	COSTO $
1	MOTOR REDUCTOR AC	60,00
1	PLACA ARDUINO R3	15,00
1	LAPTOP THINKPAD T420	400,00
1	SERVOMOTOR MG995	25,00
1	CAMARA WEB	35,00
1	PROTOTIPO DE ETAPA DE CLASIFICACIÓN	200,00
TOTAL $		**1135**

Para las licencias de software que constan en la tabla 7 no se requirió realizar compra de alguna debido a que el software usado en el presente proyecto es de uso educativo y gratuito.

Tabla 7. Costos de Licencias de Software

CANT.	DESCRIPCIÓN	COSTO $
1	LICENCIA TECHEABLE MACHINE	0,00
1	LICENCIA PYTHON	0,00
1	LICENCIA ARDUINO	0,00
TOTAL $		**0,00**

Adicional a los elementos ya mencionados se adjunta en la tabla 8 los elementos adicionales que se usaron en el presente proyecto, dichos elementos no necesitaron un estudio en sus especificaciones ya que son generales y no importaba la calidad. Con un total de $51 quedan detallado los gastos por estos elementos.

Tabla 8. Costos de otros Elementos

CANT.	DESCRIPCIÓN	COSTO $
1	PENDRIVE KINGSTON 8GB	10,00
	CABLE ELÉCTRICO #12	25,00
	CANALETAS	6,00
	1,5 metros CABLE USB TIPO B	10,00
TOTAL $		**51,00**

El costo total del proyecto sumando: costos de equipo, costos de licencias de software, costos de otros elementos y el costo de mano de obra se suma un total de $1486 cuyos rubros se detallan en la tabla 9.

Tabla 9. Costos Finales del proyecto

DESCRIPCIÓN	COSTO $
COSTOS DE EQUIPOS	1135,00
COSTOS DE LICENCIAS DE SOFTWARE	0,00
COSTOS DE OTROS ELEMENTOS	51,00
COSTO DE MANO DE OBRA	300,00
TOTAL $	**1486,00**

CAPÍTULO IV

4.1 PRUEBAS Y RESULTADOS

4.1.1 ENTRENAMIENTO DEL MODELO

Para el entrenamiento se procede a subir los dataseet de las respectivas clases: manzanas buenas, manzanas malas, y estado vacío para que el sistema detecte cuando no existe la presencia de alguna fruta en el ambiente (ver figura 52).

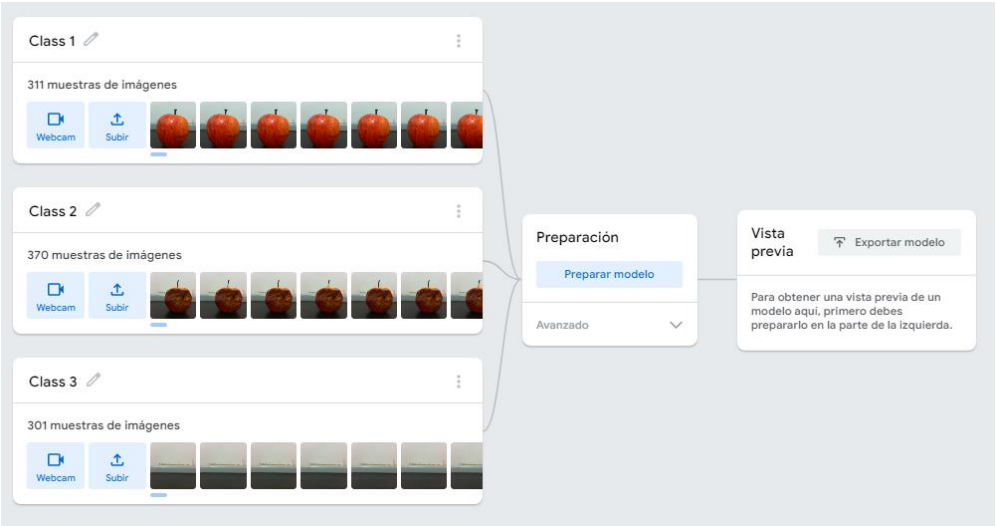


Figura 52. Muestras de imágenes para las clases.

Con una configuración de 50 épocas, un tamaño de lote aproximado de 16 y una taza de aprendizaje de 0,001 se observa como en 30 segundos obtenemos un entrenamiento optimo con 26 de 50 épocas, es decir aproximadamente 1 minuto se obtuvo el entrenamiento total con la herramienta Teachable Machine (ver figura 53).

Figura 53. Preparación del modelo.

La precisión por clase se calcula usando muestras de pruebas como se observa en la tabla 10.

Tabla 10. Precisión por clase

PRECISION POR CLASE		
CLASS	ACCURACY	#SAMPLES
Class 1	1.00	47
Class 2	1.00	56
Class 3	1.00	46

Una matriz de confusiones resume el grado de precisión del modelo. Se puede usar este modelo para determinar que clases resultan confusas para el modelo.

El eje y (muestras) representa la clase de muestras. El eje x (predicción) representa la clase a la que el modelo asigna las muestras después del aprendizaje. Por lo tanto, si la clase de una muestra es "pera", pero su predicción es "sandia", quiere decir que, después de aprenderse los datos de entrenamiento, el modelo determina incorrectamente que una muestra de "pera" pertenece a "sandia". Por lo general esto significa que las dos clases comparten características que el modelo detecta y que una muestra concreta de "pera" se parece más a las muestras de "sandia" (ver figura 54).

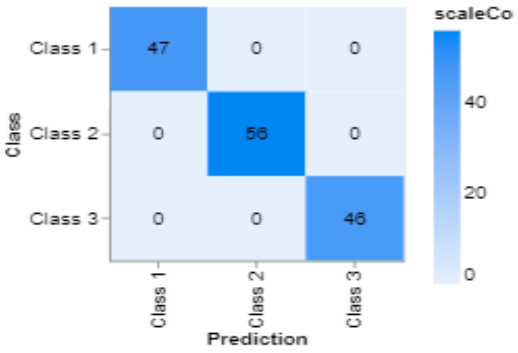


Figura 54. Matriz de confusión del modelo entrenado.

La precisión es el porcentaje de clasificaciones que un modelo acierta durante la preparación. Si el modelo clasifica correctamente 70 muestras de 100, la precisión se calcula dividiendo 70 entre 100, que da como resultado 0,7.

Si las predicciones del modelo son perfectas, entonces la precisión es igual a 1, caso contrario tendremos valores menores a 1.

En el enteramiento representa que en 2 épocas se alcanzó una precisión por época de 0,926 (ver figura 55)

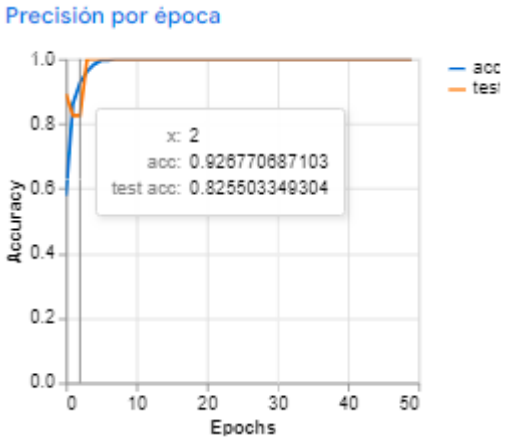


Figura 55. Precisión en época 2.

En el enteramiento representa que en 10 épocas se alcanzó una precisión por época de 0,999 casi cerca del 1 (ver figura 56).

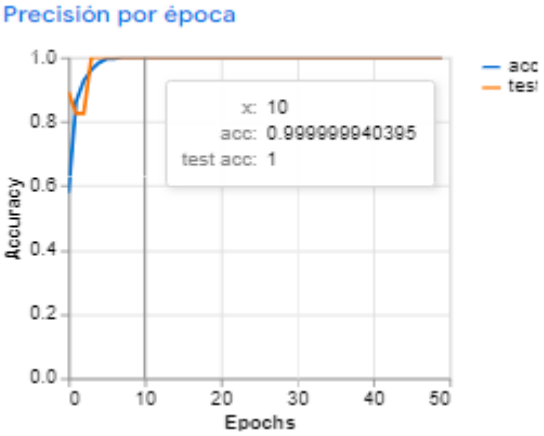


Figura 56. Precisión en época 10.

La pérdida se encarga de medir el nivel de aprendizaje de un modelo en el momento de predecir la clasificación correcta para un conjunto demuestras determinado. Si dichas predicciones ejecutadas por el modelo son perfectas, la perdida será igual a 0; caso contrario, tiende ser superior a 0.

Como ejemplo de lo que mide, podemos definir dos modelos: A y B. El modelo A predice la clasificación correcta de una muestra, no obstante, el nivel de confianza en esa predicción solo es del 60%.

El modelo B también predice la clasificación correcta de la misma muestra; pero su nivel de confianza en esa predicción es del 90%.

Ambos modelos son igual de precisos, pero el valor de pérdida del modelo B es inferior.

En el caso del entrenamiento con las 3 clases; Manzana buena, Manzana Mala y ambiente sin manzana se obtuvieron los siguientes resultados:

En el momento que el entrenamiento se encuentra en la época 5 se observa que se tiene una pérdida de prueba de 0.119372 mientras que en la pérdida total tenemos 0.126528 (ver figura 57).

Figura 57. Perdida en época 5.

En el momento que el entrenamiento se encuentra en la época 10 se observa que se tiene una pérdida de prueba de 0.03349128 mientras que en la pérdida total tenemos 0.031240, en este caso se aproxima más a 0, casi cerca de ser nula la perdida (ver figura 58).

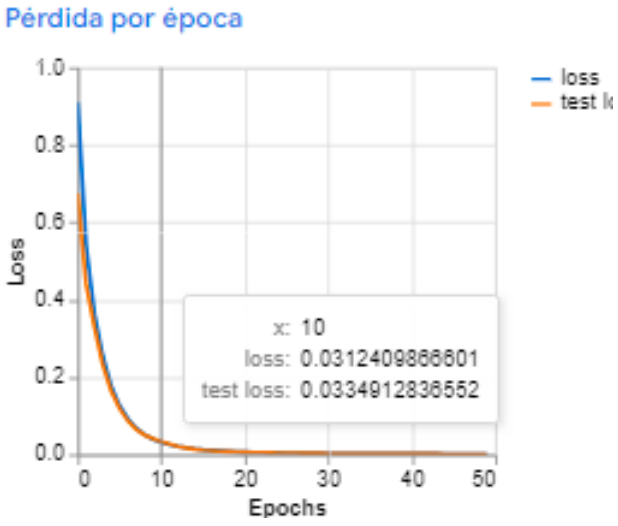


Figura 58. Perdida en época 10.

En el momento que el entrenamiento se encuentra en la época 49 se observa que se tiene una pérdida de prueba de 0.000836 mientras que en la pérdida total tenemos 0.00083630, en este caso la pérdida es prácticamente 0 (ver figura 59).

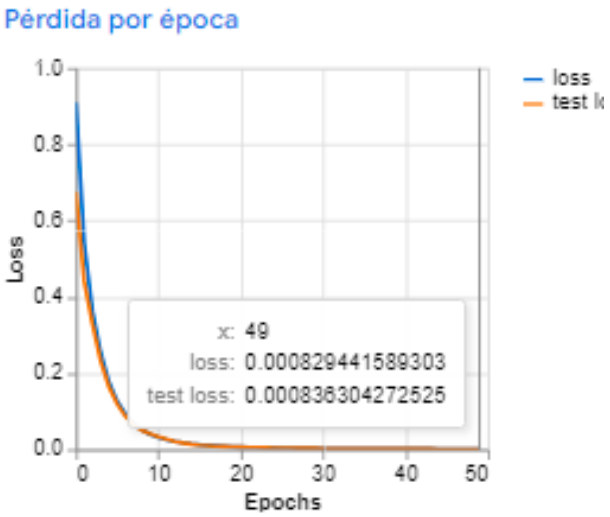


Figura 59. Perdida en época 49.

A continuación, se muestra la vista previa del modelo entrenado ejecutando la predicción en tiempo real, donde se observa una predicción perfecta para el caso 3 el cual es “estado vacío” o “clase 3” (ver figura 60).

Figura 60. Vista previa del modelo en clase 3.

En el caso de “manzana mala” o “clase 2” tenemos una predicción del 96% en el cual modelo está seguro de que se proyecta pertenece a la clase mencionada (ver figura 61).

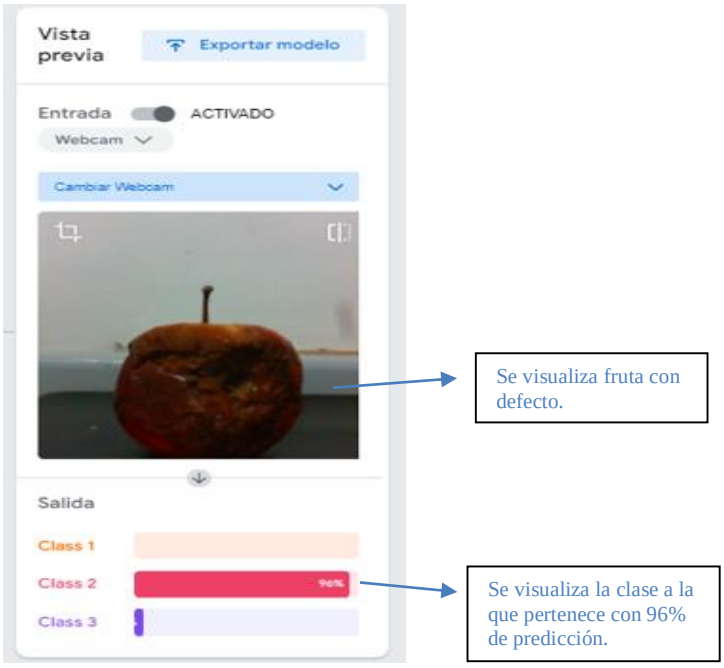


Figura 61. Vista previa del modelo en clase 2.

En el caso de "manzana buena" o "clase 1" tenemos una predicción del 95% en el cual modelo está seguro de que se proyecta pertenece a la clase mencionada (ver figura 62).

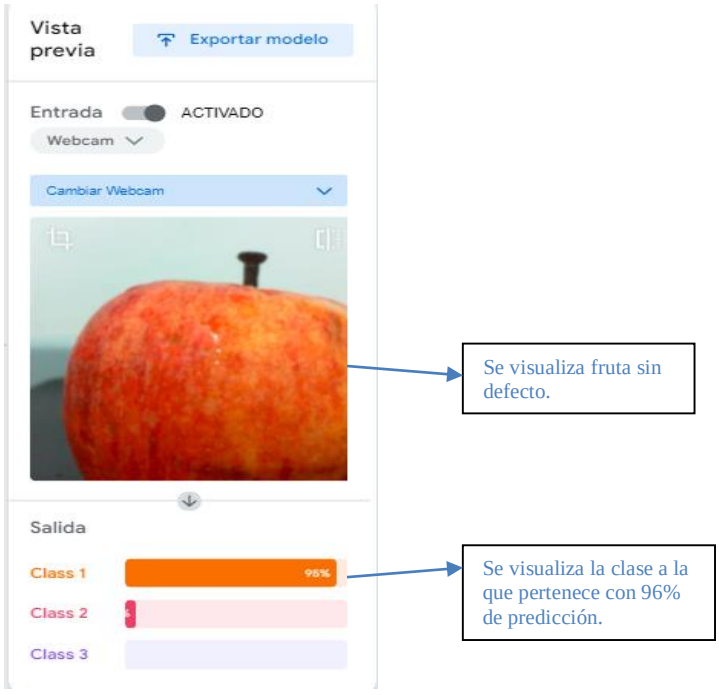


Figura 62. Vista previa del modelo en clase 1.

Realizado el entrenamiento, las estadísticas, y predicciones en la vista previa del modelo con resultados óptimos se puede exportar el modelo para utilizarlo en el sistema de visión artificial para corroborar estos datos y asegurarse de que el modelo está en perfectas condiciones con el sistema.

4.1.2 PRUEBA DEL MODELO CON EL SISTEMA DE VISIÓN ARTIFICIAL

El sistema diseñado en python haciendo uso de las bibliotecas de Tensorflow para utilizar el modelo entrenado en Teachable Machine y Opencv para realizar la captura de video en tiempo real de tal forma que se pueda predecir el tipo de clase a través del modelo entrenado en tiempo real con lo que se esa proyectando en la cámara.

Para el caso de manzana buena en la prueba se obtuvo un 99.5% con la fruta que se estaba usando en ese momento. Siento un resultado óptimo utilizando el sistema de Visión Computarizada (Ver figura 63).

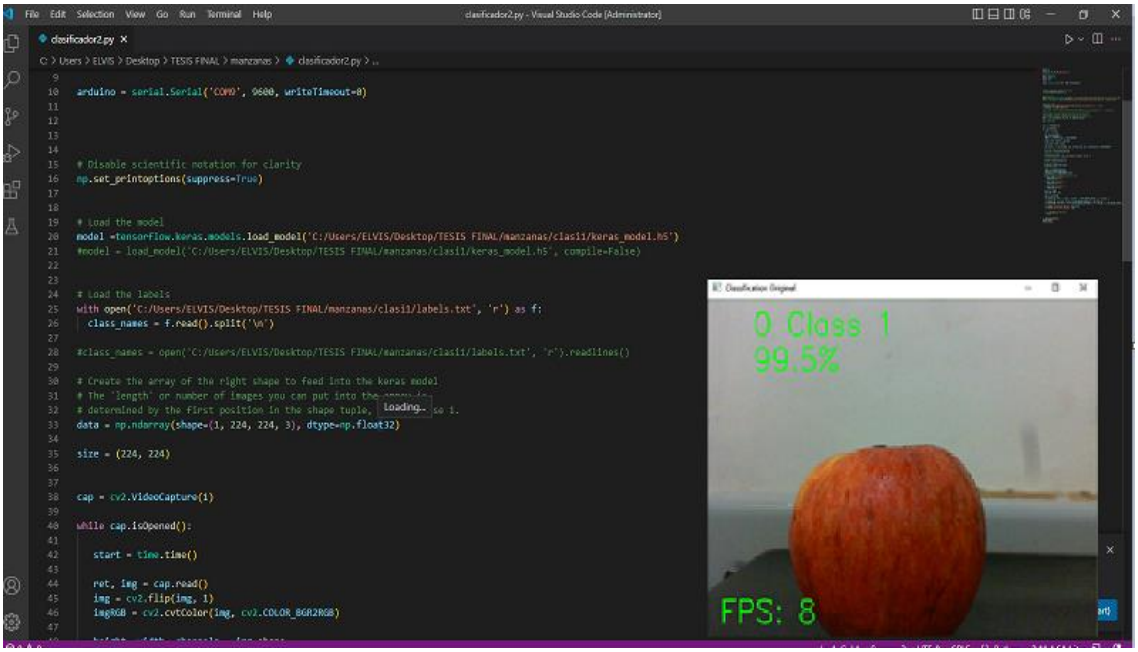


Figura 63. Sistema de visión computarizada manzana buena.

Para el caso de manzana mala se obtuvo un 99,97 % con la fruta usada en ese instante, siendo así un resultado casi perfecto para esta clase en el sistema de Visión Computarizada (Ver figura 64).

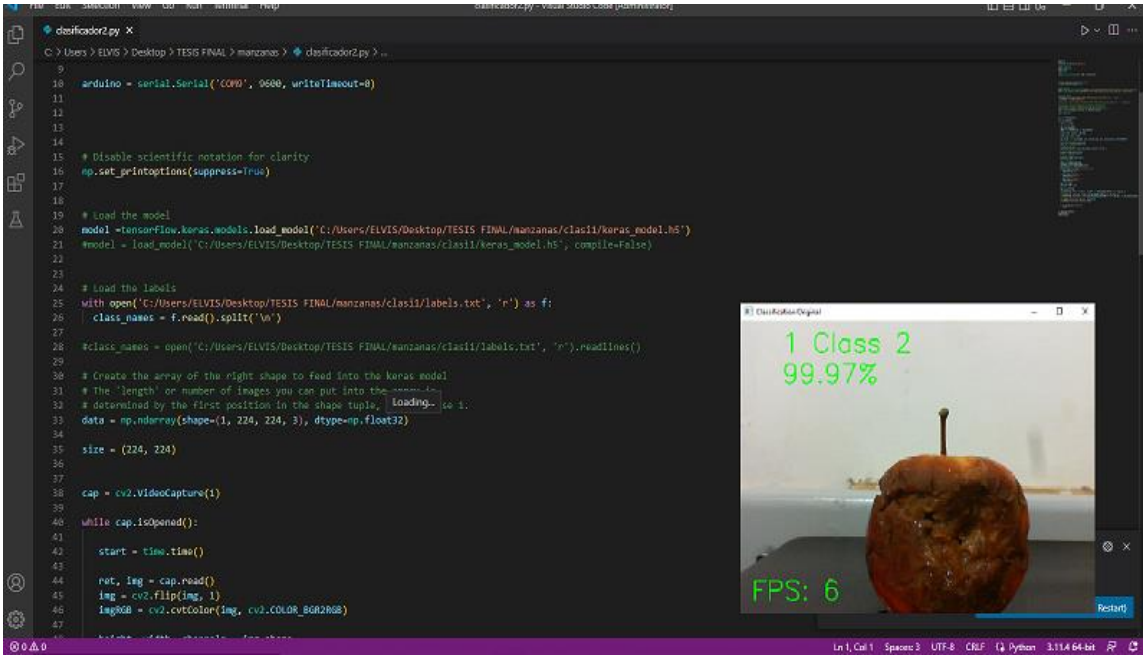


Figura 64. Sistema de visión computarizada manzana mala.

Para el caso de estado vacío se obtuvo un 99,98 % con el ambiente en ese instante sin la presencia de alguna fruta, siendo así un resultado perfecto para esta clase en el sistema de Visión Computarizada (ver figura 65).

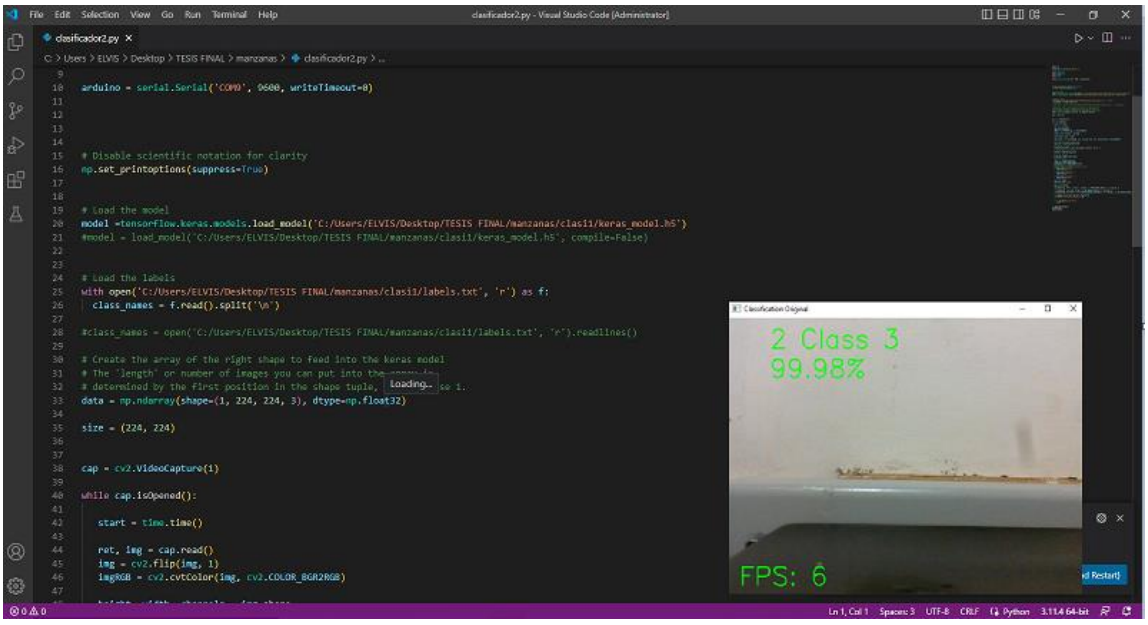


Figura 65. Sistema de visión computarizada ambiente sin manzana.

4.1.3 PRUEBA DEL SISTEMA CON EL CONTROLADOR ELECTRÓNICO DE GIRO DEL MOTOR

Aplicando comunicación serial se logró comunicar el controlador con el sistema de visión artificial.

Para el caso de manzana buena se observa que el sistema envía como dato esto indica que el motor no va a girar 90 grados según los parámetros de configuración para el sistema de clasificación. Se obtiene un 99.5 % de predicción (ver figura 66).

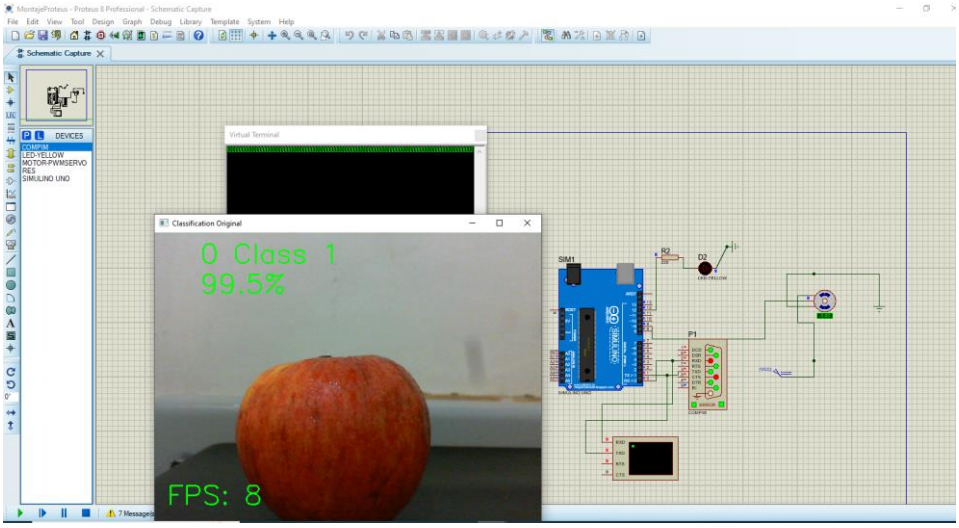


Figura 66. Etapa de clasificación simulada manzana buena.

Para el caso de manzana mala se observa que el sistema envía como dato 0 esto indica que el motor va a girar 90 grados según los parámetros de configuración para el sistema de clasificación. Se obtiene un 99.7 % de predicción (ver figura 67).

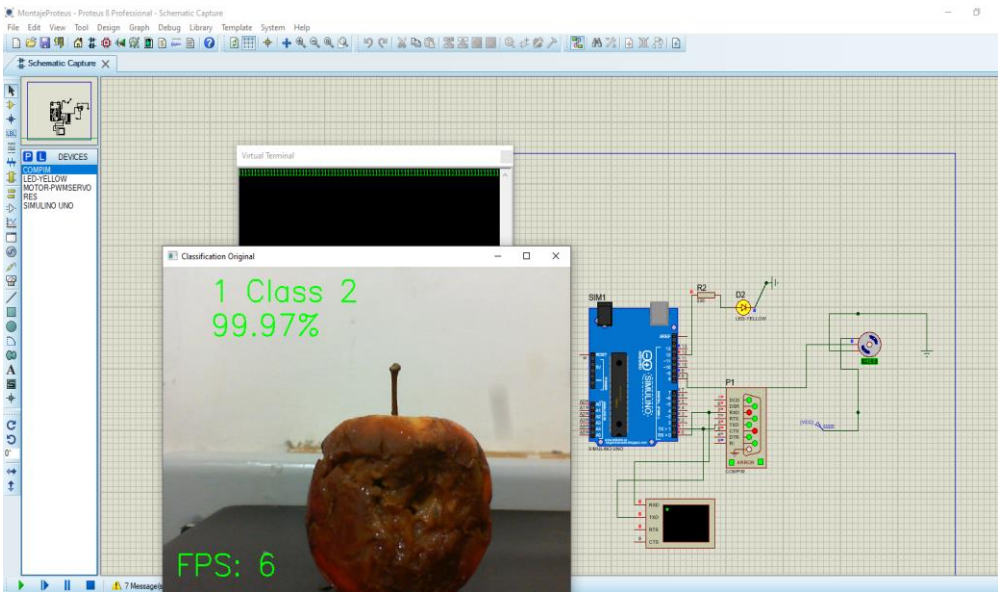


Figura 67. Etapa de clasificación simulada manzana mala.

Para el caso estado vacío mala se observa que el sistema envía como dato 0 esto indica que el motor no va a girar 90 grados según los parámetros de configuración para el sistema de clasificación. Se obtiene un 99.7 % de predicción (ver figura 68).

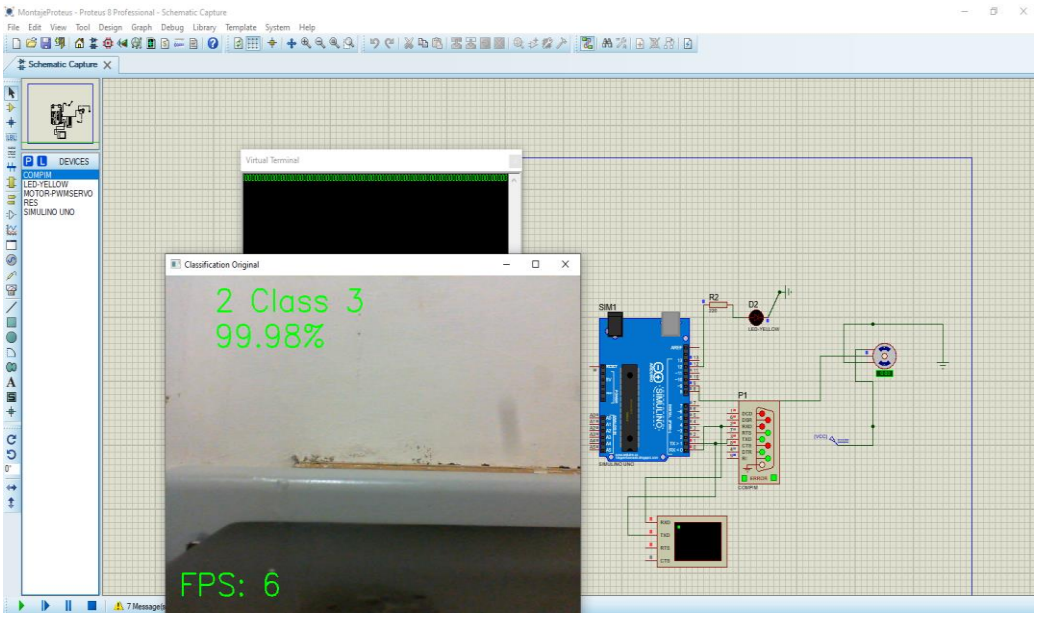


Figura 68. Etapa de clasificación simulada estado vacío.

4.1.4 DISEÑO FINAL CON LA INTERFAZ DE USUARIO

El Diseño Final nos muestra la Interfaz Gráfica de Usuario en ejecución conectada al sistema de análisis y clasificación (ver figura 69).

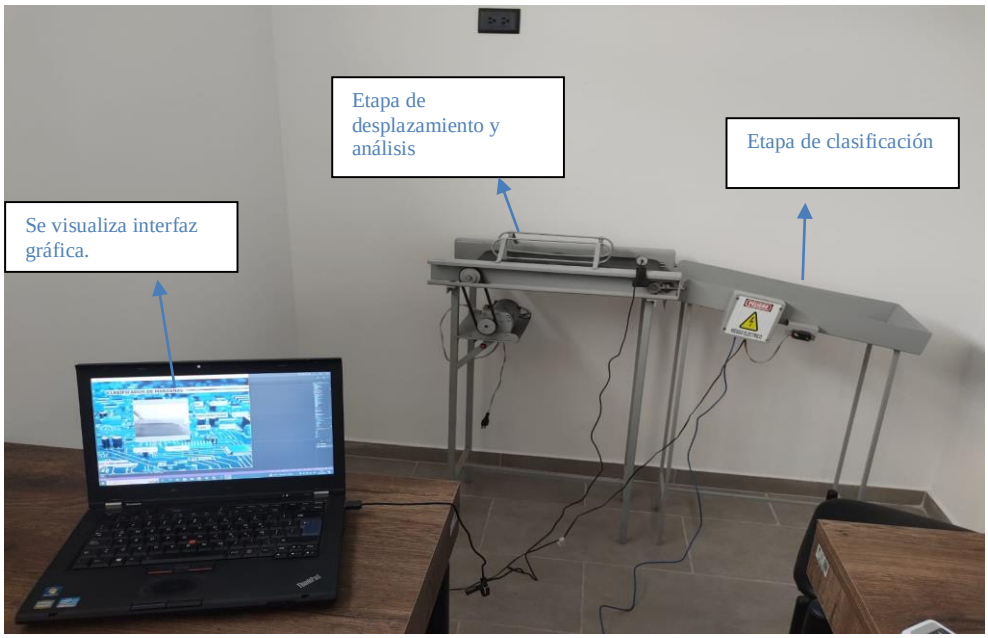


Figura 69. Vista 1 ambiente sin manzana.

Se realiza la primera clasificación que es el estado vacío donde el sistema indica que no existe ninguna fruta, obteniendo 0,99 en Confidence Score (ver figura 70).

Figura 70. Vista 2 ambiente sin manzana.

Para esta clasificación el sistema no debe accionar el servomotor (ver figura 71).

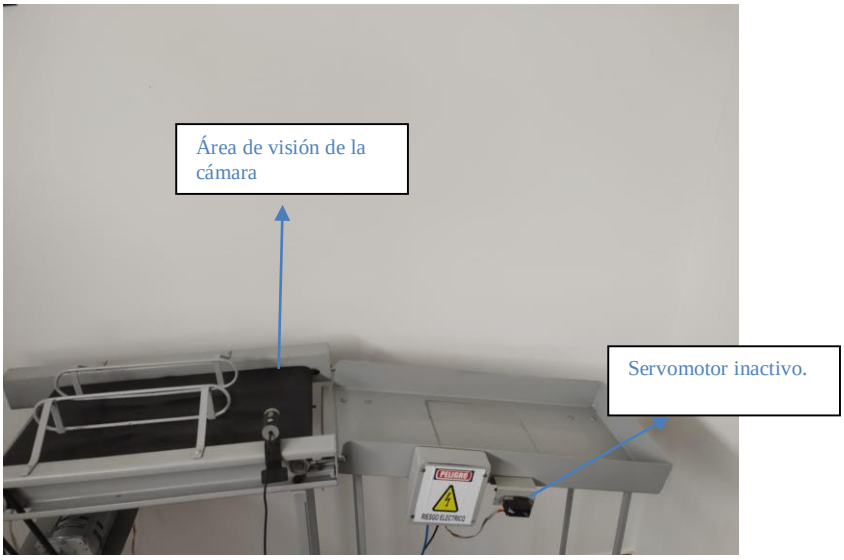


Figura 71. Vista 3 ambiente sin manzana.

Se procede a enviar una manzana en estado bueno, el sistema al analizarlo obtiene 0.97 en Confidence Score, siendo un resultado positivo y alto para el desempeño del modelo entrenado (ver figura 72 y 73).

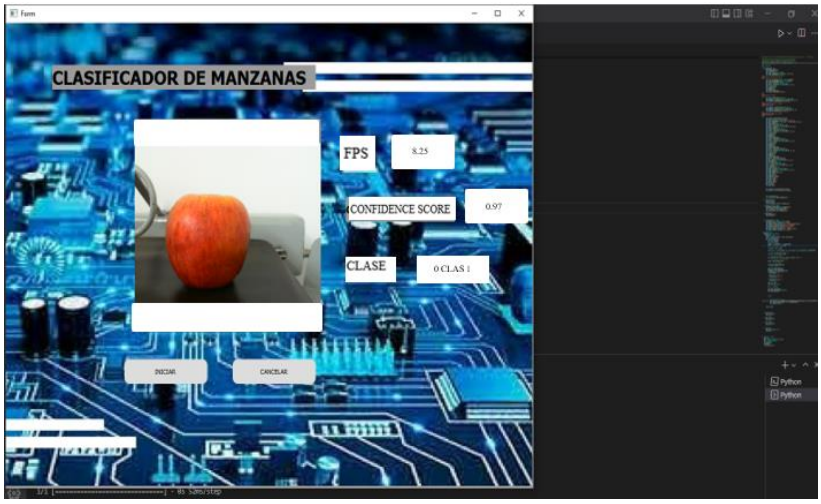


Figura 72. Vista 1 clasificación Manzana buena.

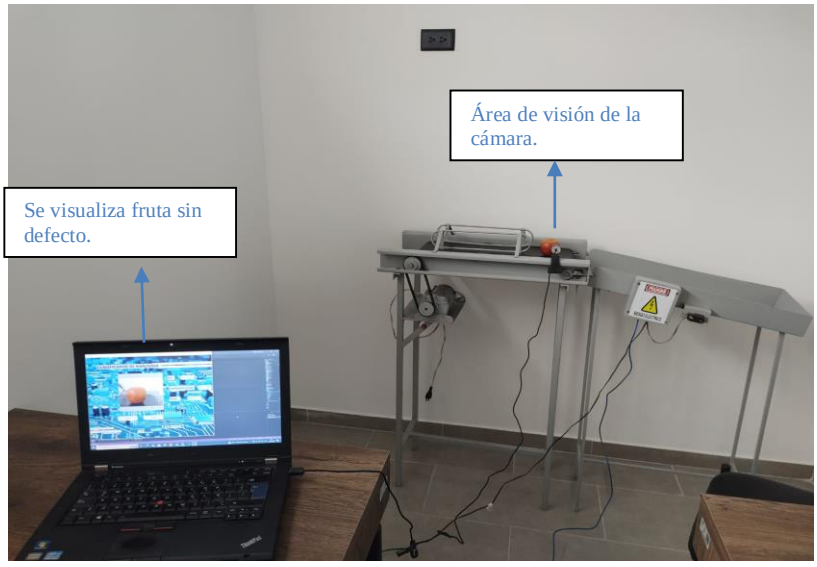


Figura 73. Vista 2 clasificación Manzana buena.

Para este tipo de clasificación el servomotor no se acciona y la fruta procede a terminar su recorrido al final de la clasificadora donde estarán las manzanas buenas (ver figura 74).

Figura 74. Etapa de clasificación manzana Buena.

Se procede a enviar una manzana en mal estado donde el sistema de clasificación nos arroja un 0.99 para Confidence Score, siendo un resultado muy alto y acertado para el modelo entrenado. Posteriormente el servomotor se acciona y la manzana en mal estado pasa al lugar donde quedaran las manzanas con esa clasificación (ver figura75 y figura 76).

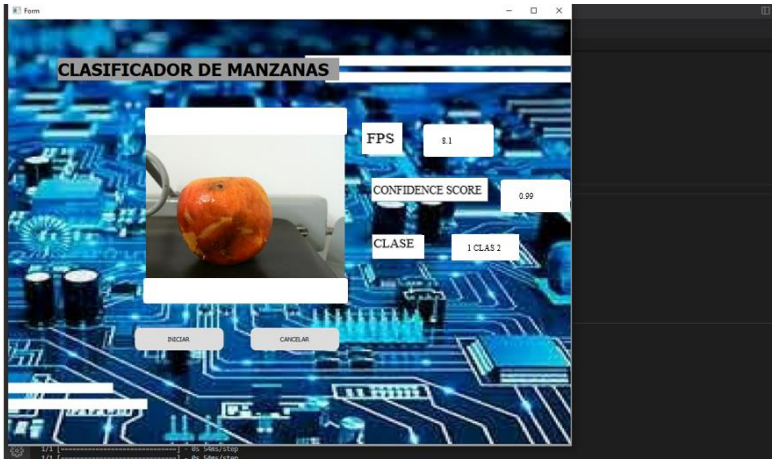


Figura 75. Vista 1 clasificación manzana con defecto.

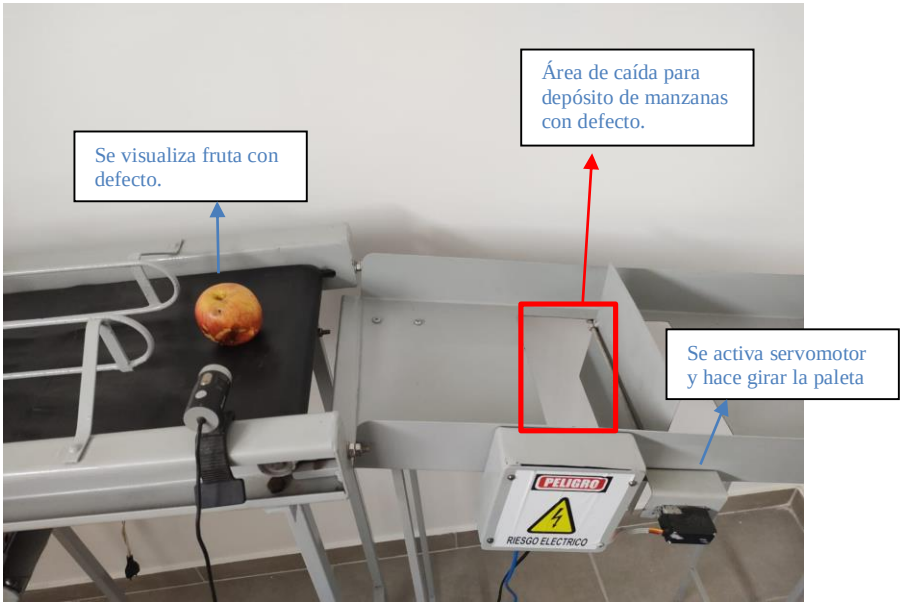


Figura 76. Vista 2 clasificaciones manzana con defecto.

4.2 CONCLUSIONES Y RECOMENDACIONES

4.2.1 CONCLUSIONES

- Se entrenó una red neuronal artificial utilizando librerías de TensorFlow y KERAS en la aplicación web Teachable Machine para lograr que clasifique manzanas según su defecto sean estas buenas o malas de tal manera que el modelo se pudo descargar y usarlo en el sistema. La configuración del entrenamiento fue establecida para 50 épocas, un tamaño de lote aproximado de 16 y una taza de aprendizaje de 0,001 en donde se obtuvieron los siguientes resultados: precisión por época al final del entrenamiento de 0,999 casi cerca del 1, perdida por época al final del entrenamiento de 0.00083630 casi nula o aproximadamente 0, y las predicciones para cada clase superan el 98%.
- Se implementó un sistema de visión artificial en Python mediante el módulo OpenCV para la adquisición de imágenes en el momento que la manzana sea desplazada por la banda transportadora, y así el sistema pueda analizar estas capturas de video en tiempo real con el modelo entrenado fusionado al sistema.
- Se utilizó la placa arduino UNO R3 como controlador electrónico con las librerías necesarias para controlar el giro de un servomotor cuyo eje está conectado a una paleta logrando el sistema físico para la clasificación de manzanas. Para esto se hizo la simulación de la etapa de clasificación en el software Proteus con los elementos simulados, en donde se obtuvo tiempos de respuesta positivos por parte de la placa controladora arduino conectada al servo. Con estos resultados óptimos en la simulación se procedió a implementar en físico.
- Se diseñó una interfaz de usuario de nivel básico mediante la herramienta de software Qtdesigner cuyas librerías fueron compatibles con el sistema de visión artificial desarrollado en Python para el monitoreo del sistema. En donde se puede visualizar las capturas donde se transporta la fruta a través de la banda en tiempo real para ser analizado en el sistema, se presenta los valores de FPS, Confidence Score y el Tipo de clase.

4.2.2 RECOMENDACIONES

- Para la selección del DataSet es preferible utilizar manzanas de acuerdo a las normas internacionales para la inspección de manzanas que se establecen principalmente a través comisiones como la OECD.
- Es mejor usar la aplicación web de Teachable Machine debido a que proporciona el modelo entrenado en TensorFlow junto con la biblioteca Keras la cual facilita el aprendizaje y no demora más de 2 min máximo. Teachable machine hace uso de servidores dedicados para el entrenamiento haciendo más factible que el uso del ordenador personal para dicha carga de trabajo.
- Se debe añadir una clase con las muestras de imágenes en donde el ambiente este vacío, esto ayuda a que el modelo pueda detectar cuando no existe ninguna fruta y mejore la precisión de las demás clases.
- Para el desplazamiento de la fruta es preferible usar una banda de dimensiones ligeras y montar una baranda para que la fruta no pierda el rumbo ya que puede darse el caso que por el movimiento de la banda tienda a perder el rumbo debido a la forma casi esférica de la fruta.
- Se debe alimentar a la placa controladora arduino con un adaptador de 12 V dc y para la comunicación serial un cable USB tipo “A” a tipo “B”.

4.3 BIBLIOGRAFÍAS

[1]] Valarezo Beltrón, Carlos Oswaldo. (2020). Caracterización de fincas productoras de limón (citrus aurantifolia) en Portoviejo, Ecuador. Revista de Investigación e Innovación Agropecuaria y de Recursos Naturales, 7(1), 88-94. Recuperado en 26 de julio de 2022, de http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S2409-16182020000100012&lng=es&tlng=es.

[2] Pacheco Patiño, J. M. (2021). PROCESOS DE CALIDAD EMPLEADOS EN LA PLANTA INDUSTRIAL DE LA ASOCIACIÓN DE PRODUCTORES DE FRUTAS DE MANABÍ (Bachelor's thesis, Jipijapa. UNESUM).

[3] Urgilés, R. M. I. (2016). DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO Y MANIPULACIÓN DE FRUTAS UTILIZANDO VISIÓN ARTIFICIAL Y BRAZO ROBÓTICO INDUSTRIAL. Edu.ec. http://www.dspace.espol.edu.ec/xmlui/bitstream/handle/123456789/37302/D-103441.pdf?sequence=-1&isAllowed=y

[4] Tigrero, A., Andrés, V., Oñate, P., & Josué, E. (s/f). Diseño e implementación de un módulo didáctico de inspección y clasificación de frutas usando visión artificial por medio de imágenes fuera del espectro visible para el laboratorio de mecatrónica de la universidad de las fuerzas armadas ESPE extensión Latacunga. Edu.ec. Recuperado el 26 de julio de 2022, de http://repositorio.espe.edu.ec/bitstream/21000/14945/1/T-ESPEL-MEC-0146.pdf

[5] Juárez, C. (2021, 18 mayo). Visión artificial para mejorar la calidad. The Food Tech. https://thefoodtech.com/seguridad-alimentaria/vision-artificial-para-mejorar-la-calidad/#%3A%7E%3Atext%3D%C2%BFQu%C3%A9%20es%20la%20visi%C3%B3n%20artificial%2Cfrutos%20y%20otros%20productos%20agr%C3%ADcolas

[6] Membrives, P. (2019, 19 noviembre). Sistema de control de calidad a través de visión artificial. Nunsys. https://www.nunsys.com/nunsys-desarrolla-un-innovador-sistema-de-control-de-calidad-por-vision-artificial/

[7] Arteaga, G. (2020, 26 octubre). Investigación bibliográfica -Cómo llevar a cabo una. TestSiteForMe. https://www.testsiteforme.com/investigacion-bibliografica/

[8] Velázquez, A. (2020, 21 enero). ¿Qué es la Investigación Exploratoria? QuestionPro. https://www.questionpro.com/blog/es/investigacion-exploratoria/

[9] Fuentes, E. F. (s/f). El diagnostico social. Wordpress.com. Recuperado el 26 de julio de 2022, de https://modulosocioterritorial.files.wordpress.com/2009/09/diagnc3b3stico-social.pdf

[10] Mitrović, M. (s/f). Upc.edu. Recuperado el 29 de octubre de 2023, de https://upcommons.upc.edu/bitstream/handle/2099/15578/12.pdf

[11] (S/f-b). Recuperado el 1 de marzo de 2023, de http://file:///C:/Users/ELVIS/Downloads/Dialnet-ClasificacionDeManzanasUtilizandoVisionArtificialY-7059317.pdf

[12] (S/f). Oecd.org. Recuperado el 29 de octubre de 2023, de http://www.oecd.org/agriculture/fruit-vegetables/

[13] (S/f). Edu.ec. Recuperado el 1 de marzo de 2023, de https://upec.edu.ec/subsitios/citt/images/Proyectos%20Investigaci%C3%B3n/2015/3_vision_artif/10_art_ci.pdf

[14] Díaz-Ramírez, J. (2021). Aprendizaje Automático y Aprendizaje Profundo. Ingeniare. Revista Chilena de Ingeniería, 29(2), 180–181. https://doi.org/10.4067/s0718-33052021000200180

[15] Redes Neuronales Convolucionales en Profundidad - DataSmarts Español. (2018, diciembre 1). DataSmarts Español; DataSmarts. https://datasmarts.net/es/redes-neuronales-convolucionales-en-profundidad/

[16] (S/f). Researchgate.net. Recuperado el 22 de marzo de 2024, de https://www.researchgate.net/figure/Esquema-conectividad-dispersa-5_fig4_341194410

[17] Neuronales, R., Salas, A.-R., & Salas, R. (s/f). Redes Neuronales Artificiales. Cloudfront.net. Recuperado el 1 de marzo de 2023, de https://d1wqtxts1xzle7.cloudfront.net/50358783/Redes_Neuronales_Artificiales-libre.pdf?1479332869=&response-content-disposition=inline%3B+filename%3DRedes_Neuronales_Artificiales.pdf&Expires=1677707876&Signature=HH0~NG4WJfMOFipQtf8SW41q0nWVDQUfVfw6VvpAdIWVv-M~3F7dj~ibjNTwHkpazkhBcxmfmj4XTw40sZFjhXmN6l0BF1kGiU~Ww634w4id6eXDNwZF6OStKMKNmRLzFxSFULt5qyUnB49hsCK0uQ-6ttIxbOufcIbrNJlby6RpoUFXI1qKgFHwY6IQlwGVkZHVWnRAnirq-YLjtVe2oxryjS9LkG6YYgiUAcXRhrcXvrrbojwHAGSiJdY6tmw26f62Jk-bG4O2JW-upSZ4prK4fsF-JPLRmobci4ozOYvVZaro2zBuO9N6rI~dL8QHb14qFFZ-vpjgRtlyrY9fig__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[18] ¿Qué es una red neuronal en Deep Learning? (2022, agosto 10). KeepCoding Bootcamps. https://keepcoding.io/blog/red-neuronal-en-deep-learning/

[19] (S/f-b). Ieee.org. Recuperado el 4 de noviembre de 2023, de https://ieeexplore.ieee.org/abstract/document/8993089

[20] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 4510–4520.

[21] Camacho-Poveda, É. C., & Ruge-Ruge, Í. A. (2015). Diseño de un microprocesador de propósito educativo. Ingenio Magno, 6, 86–99. http://revistas.ustatunja.edu.co/index.php/ingeniomagno/article/view/1096

[22] el sitio web http://usuarios. lycos. es/sfriswolker, B. en. (s/f). INTRODUCCIÓN AL MICROCONTROLADOR. Cloudfront.net. Recuperado el 1 de marzo de 2023, de https://d1wqtxts1xzle7.cloudfront.net/39407044/micro-libre.pdf?1445752291=&response-content-disposition=inline%3B+filename%3DMicro.pdf&Expires=1677658404&Signature=ebd15wEqHSXQpJXzZzoNS6i6fOkKfhBWLGdB5kh-eUGy8-zw5J0iPtymSTGYUOMl1d6y4Yf-VKdyKNNepYh0oZtJYXoFoG088T8BZxSXkrsTSlxz7ymBRy1MpSzk2m8bVOl-Vqik7ovIEWnBYiplxc~Y4k3Z4iRaIoDyE7d8UBP8ZHEA6XvYwSO9NT897vOF7HTBfVCM0UdpgyYGf2GyM3EbnLCHjpDlxpSSERbIOU-huvZEt4hhMmYMfTgcM8bbx6I1GwZmpC3Im7mNoHP2ntF~9OHM3Tf~obTQyc8XzV-uO1Yyo3lWE5z2eAA43FT4k5ahDx0O4RYOu8hwD7udbQ__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[23] Oqueña, Q., & Ciro, E. (s/f). El Hombre y la Máquina. Redalyc.org. Recuperado el 1 de marzo de 2023, de https://www.redalyc.org/pdf/478/47812406007.pdf

[24] (S/f-c). Ieee.org. Recuperado el 5 de noviembre de 2023, de https://ieeexplore.ieee.org/abstract/document/1300702

[25] (S/f-c). Ieee.org. Recuperado el 5 de noviembre de 2023, de https://ieeexplore.ieee.org/abstract/document/8786911

[26] Alexander, C. J., Andrés, C. J., & de Segundo Orden, Q. E. A. G. D. E. S. S. M. P. Y. F. A. (s/f). Scientia Et Technica. Redalyc.org. Recuperado el 5 de noviembre de 2023, de https://www.redalyc.org/pdf/849/84921327003.pdf

[27] Luna, G., & Galo, N. (1992). Sistema didáctico para comunicación serial entre dos computadores personales. QUITO/EPN/1992.

[28] Reyes Andrade, E. M., & Verástegui Gálvez, N. (2020). Importación de cámaras full HD para la elaboración y comercialización de simuladores para el entrenamiento de la cirugía laparoscópica en la ciudad de Lima. Universidad Tecnológica del Perú.

[29] Hernández, A. (1969). Bandas transportadoras. https://repositorio.sena.edu.co/handle/11404/4032

[30] Mendoza, M. L. (2020, julio 16). Qué es un lenguaje de programación. Openwebinars.net. https://openwebinars.net/blog/que-es-un-lenguaje-de-programacion/

[31] Albornoz, M. C. (2014). Diseño de interfaz gráfica de usuario. XVI Workshop de Investigadores en Ciencias de la Computación.

[32] Completo, N. (s/f). Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial. Redalyc.org. Recuperado el 5 de noviembre de 2023, de https://www.redalyc.org/pdf/925/92541011.pdf

[33] Vista de Clasificación de manzanas utilizando visión artificial y redes neuronales artificiales. (s/f). Edu.co. Recuperado el 1 de marzo de 2023, de https://journalusco.edu.co/index.php/iregion/article/view/1917/3869

[33] (S/f-c). Researchgate.net. Recuperado el 1 de marzo de 2023, de https://www.researchgate.net/profile/Juan-Damato/publication/265928339_Procesamiento_de_imagenes_para_la_clasificacion_masiva_de_frutos_basado_en_el_color/links/54bbe5ba0cf29e0cb04be316/Procesamiento-de-imagenes-para-la-clasificacion-masiva-de-frutos-basado-en-el-color.pdf

[34] Castillo, D. F. N., & Mora, O. A. C. (2014). Diseño de un prototipo de una máquina automática de empacado de café. Universidad de La Salle.

[35] Reyes, P., Quiriarte, H., Tello-Mijares, S., & Linares, J. (s/f). Automatización de microscopio digital mediante servomotor para autoenfoque. Edu.mx. Recuperado el 12 de noviembre de 2023, de http://revistacid.itslerdo.edu.mx/coninci2017/07%20Automatizaci%C3%B3n%20de%20microscopio%20digital%20mediante%20servomotor.pdf

[36] Duk, M., Bereziuk, O. V., Lemeshev, M. S., & Volodymyr Bogachuk, V. V. (2018). Means for measuring relative humidity of municipal solid wastes based on the microcontroller Arduino UNO R3. En R. S. Romaniuk & M. Linczuk (Eds.), Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2018. SPIE.

[37] Sequeira Ugalde, M. I., & González Arias, J. D. (2020). Propuesta de implementación de un Framework de automatización de pruebas para Icost en componentes Intel de Costa Rica. https://repositorio.una.ac.cr/handle/11056/21924

[38] Iván García S, & Víctor, C. S. (2015). La visión artificial y los campos de aplicación. Tierra Infinita, 1(1), 98–108. https://doi.org/10.32645/26028131.76

[39] Fernandez, A. (2013). Python 3 al descubierto - 2a ed. Alfaomega Grupo Editor.

[40] Arévalo, V. M., González, J., & Ambrosio, G. (s/f). LA LIBRERÍA DE VISIÓN ARTIFICIAL OPENCV APLICACIÓN A LA DOCENCIA E INVESTIGACIÓN. Uma.es. Recuperado el 12 de noviembre de 2023, de http://mapir.isa.uma.es/varevalo/drafts/arevalo2004lva1.pdf

[41] Pang, B., Nijkamp, E., & Wu, Y. N. (2020). Deep learning with TensorFlow: A review. Journal of Educational and Behavioral Statistics: A Quarterly Publication Sponsored by the American Educational Research Association and

the American Statistical Association, 45(2), 227–248. https://doi.org/10.3102/1076998619872761

[42] Lugaresi, C., Tang, J., Nash, H., Mcclanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W.-T., Hua, W., Georg, M., & Grundmann, M. (s/f). MediaPipe: A framework for perceiving and processing reality. Squarespace.com. Recuperado el 12 de noviembre de 2023, de https://static1.squarespace.com/static/5c3f69e1cc8fedbc039ea739/t/5e130ff310a69061a71cbd7c/1578307584840/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf

[43] Torres, J. (2018). DEEP LEARNING Introducci—n pr‡ctica con Keras. Lulu.com.

[44] Breijo, E. G. (2012). Compilador C CCS y Simulador Proteus para Microcontroladores PIC. Marcombo.

[45] 含思. (2016). Program transmission method of PLC equipments' remote monitoring software based on virtual serial port. Computer science and application, 06(03), 110–118. https://doi.org/10.12677/csa.2016.63014

[46] Peña, C. (2020). Arduino IDE: Domina la programación y controla la placa. RedUsers.

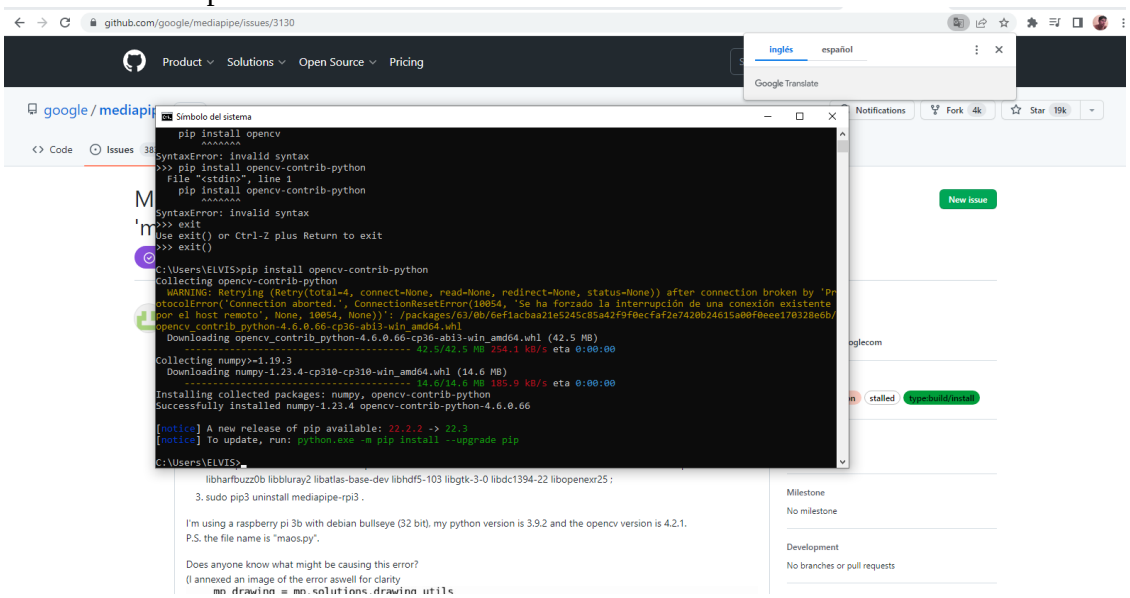
[47] Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A., & Chen, A. (2020). Teachable machine: Approachable web-based tool for exploring machine learning classification. Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems.

[48] Jović, N., & Grego, I. (s/f). Repository of Open Educational Resources for Laboratory Support in Engineering and Natural Science PROCESS OF REVIEW AND PUBLISHING OF OPEN EDUCATIONAL RESOURCES WITHIN RELAB REPOSITORY / form F1 / For the purpose of publishing open educational resources (OER) in the RELAB repository, I am submitting for review a mini-video lesson entitled Controlling Arduino board via Pyserial library and serial port -RC circuit example and the following associated metadata (attached below). Kg.ac.rs. Recuperado el 12 de noviembre de 2023, de https://relab.kg.ac.rs/dist/uploads/2ac3bdf6.pdf

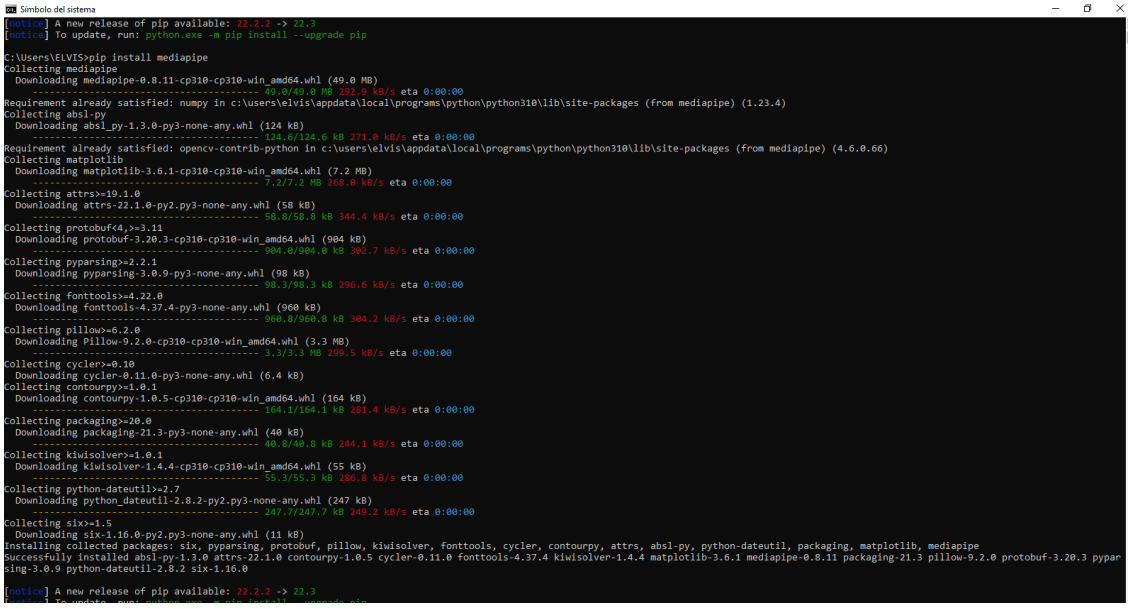
[49] Willman, J. M. (2022). Creating GUIs with Qt Designer. En Beginning PyQt (pp. 217–258). Apress.

4.4 ANEXOS

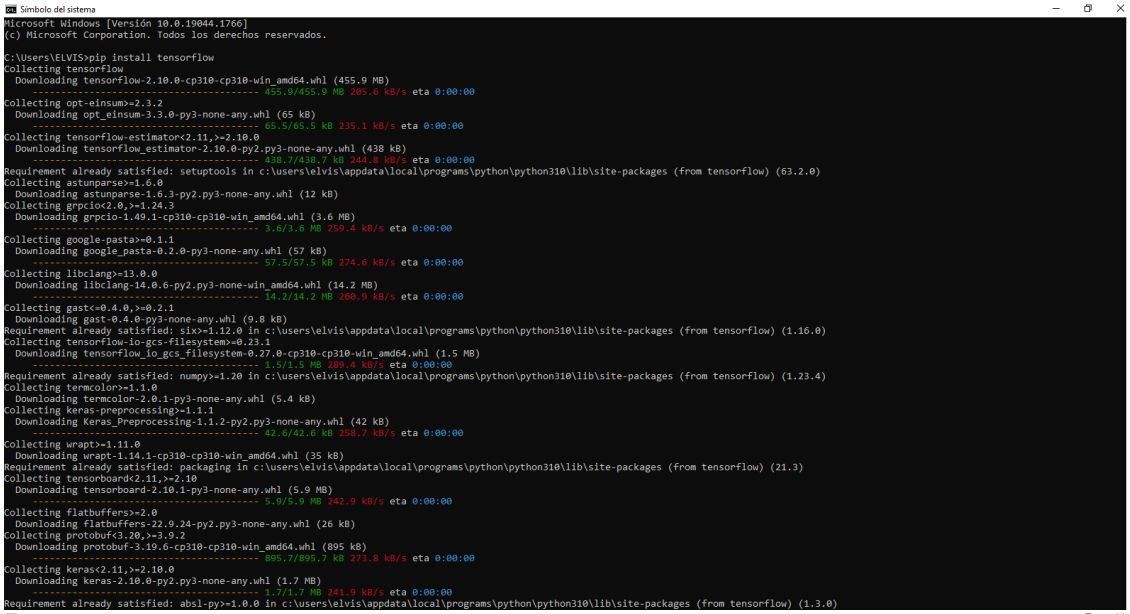
Instalación de opencv

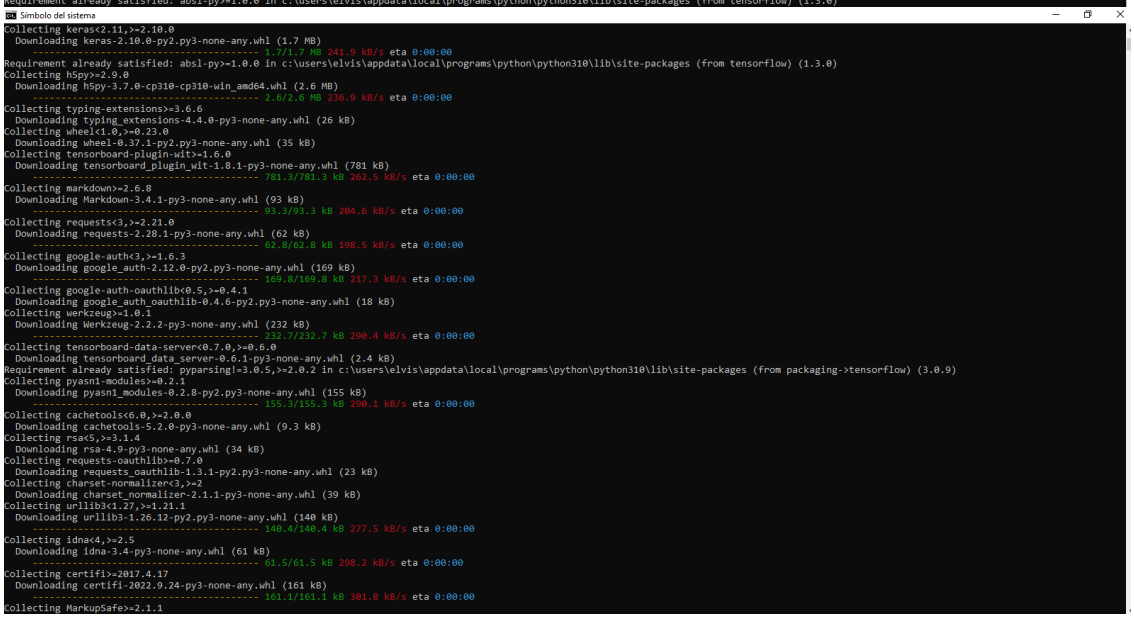


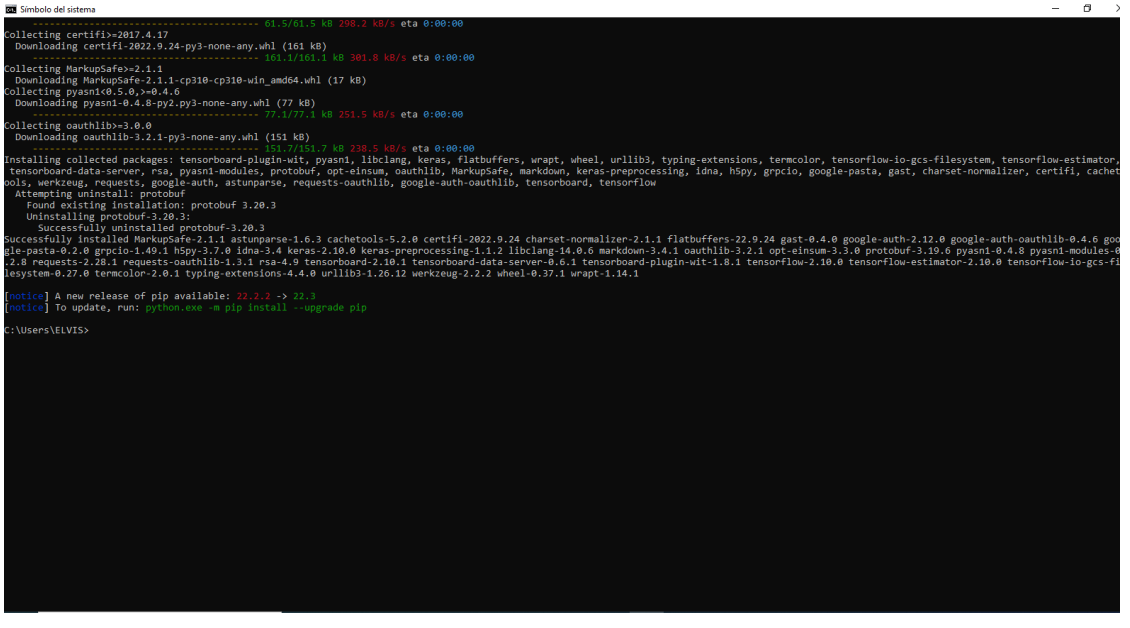
Instalación módulo mediapipe



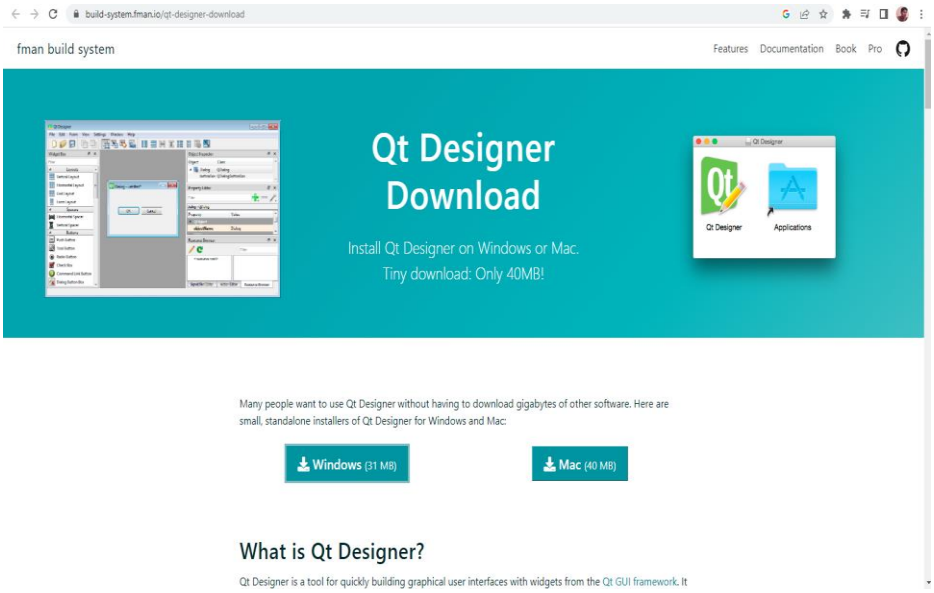
Instalación tensorflow

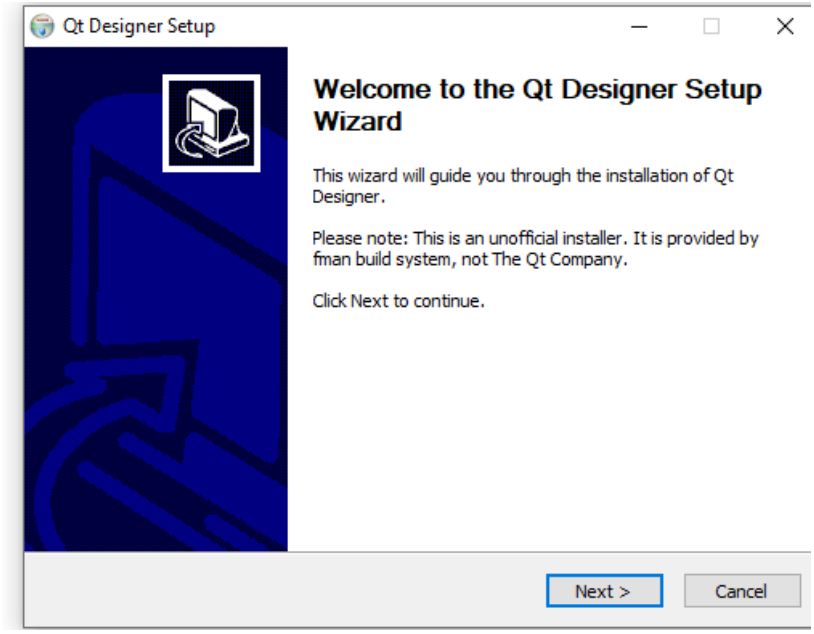


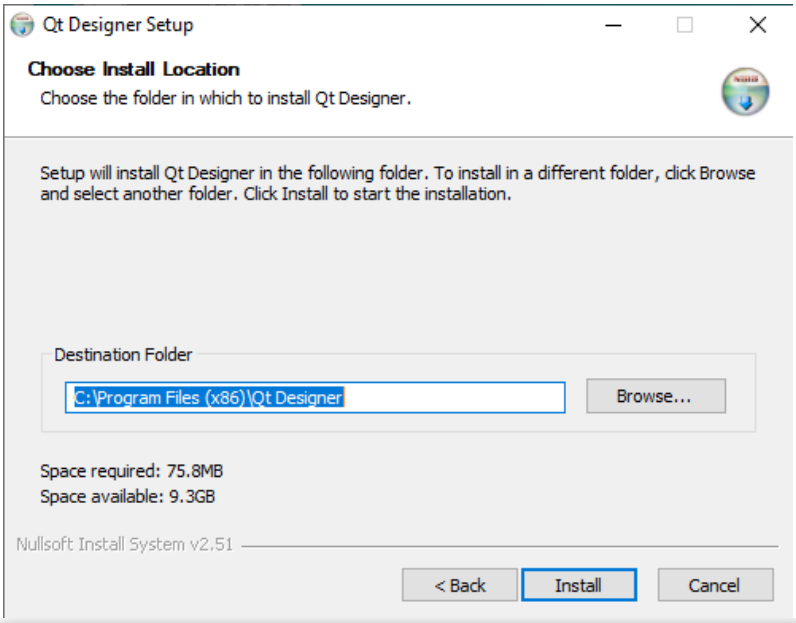


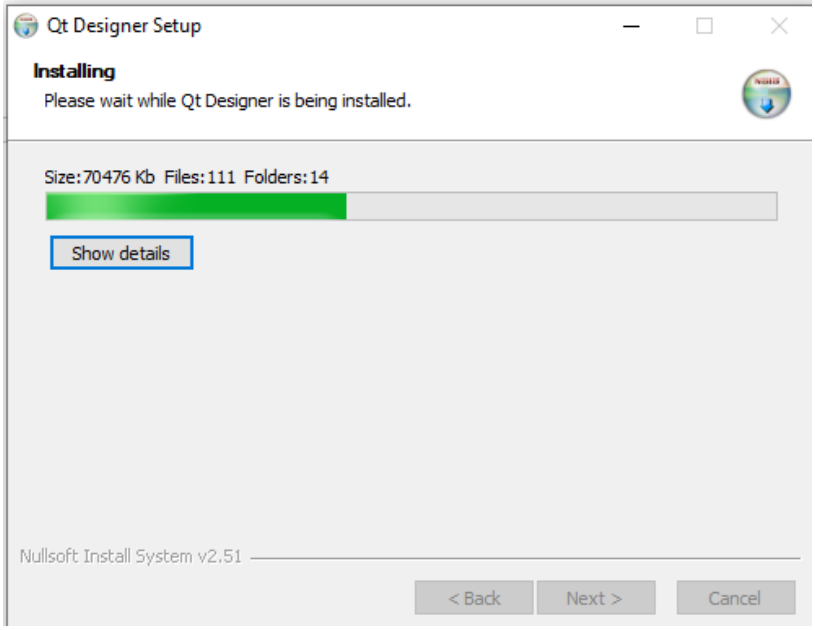


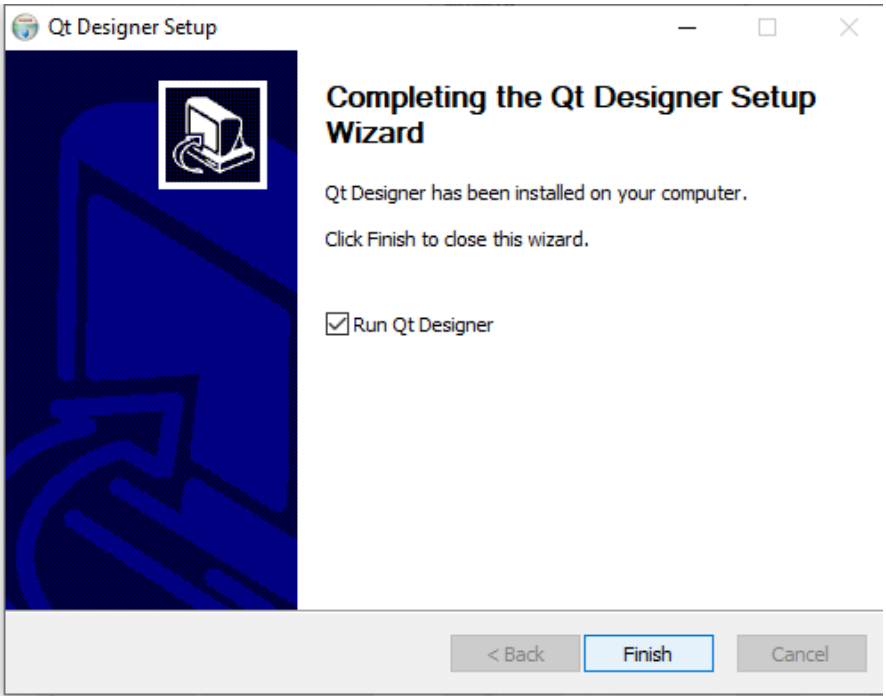
Instalación QT desginer

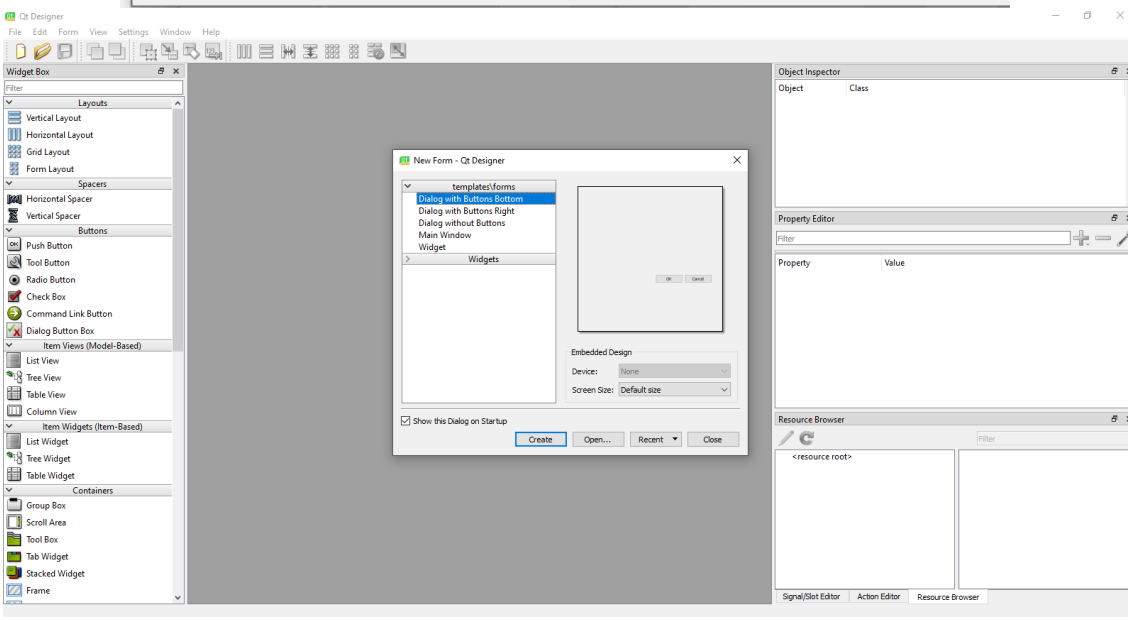





Qt Designer Setup
Choose Install Location
Choose the folder in which to install Qt Designer.
Setup will install Qt Designer in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
C:\Program Files (x86)\Qt Designer
Browse...
Space required: 75.8MB
Space available: 9.3GB
Nullsoft Install System v2.51
< Back
Install
Cancel


Qt Designer Setup
Installing
Please wait while Qt Designer is being installed.
Size: 70476 Kb Files: 111 Folders: 14
Show details
Nullsoft Install System v2.51
< Back
Next >
Cancel


Qt Designer Setup
Completing the Qt Designer Setup Wizard
Qt Designer has been installed on your computer.
Click Finish to close this wizard.
Run Qt Designer
< Back
Finish
Cancel


Qt Designer
File Edit Form View Settings Window Help
Widget Box
Layouts
Vertical Layout
Horizontal Layout
Grid Layout
Form Layout
Spacers
Horizontal Spacer
Vertical Spacer
Buttons
Push Button
Tool Button
Radio Button
Check Box
Command Link Button
Dialog Button Box
Item Views (Model-Based)
List View
Tree View
Table View
Column View
Item Widgets (Item-Based)
List Widget
Tree Widget
Table Widget
Containers
Group Box
Scroll Area
Tool Box
Tab Widget
Stacked Widget
Frame
New Form - Qt Designer
templates\forms
Dialog with Buttons Bottom
Dialog with Buttons Right
Dialog without Buttons
Main Window
Widget
Widgets
Embedded Design
Device:
Screen Size:
Default size
Show this Dialog on Startup
Create
Open...
Recent
Close
Object Inspector
Object
Class
Property Editor
Property
Value
Resource Browser
<resource root>
Signal/Slot Editor
Action Editor
Resource Browser

Instalación de Pytq5