



UNIVERSIDAD DE GUAYAQUIL

**FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

**DESARROLLO DE UN CLASIFICADOR DE ACEITUNAS
DE MESA POR SU VARIEDAD**

PROYECTO DE TITULACIÓN

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

AUTOR (ES):

**FAJARDO ZAMBRANO DANIEL ARTURO
TOALOMBO GOYES MILTON FRANKLIN**

TUTOR:

ING. ALFONSO GUIJARRO RODRÍGUEZ MGS.

**GUAYAQUIL – ECUADOR
2019**

 PRESIDENCIA REPÚBLICA DEL ECUADOR			 Plan Nacional de Ciencia, Tecnología, Innovación y Saberes			 senescyt <small>Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación</small>		
REPOSITORIO NACIONAL EN CIENCIAS Y TECNOLOGÍA								
FICHA DE REGISTRO DE TESIS								
TÍTULO: “DESARROLLO DE UN CLASIFICADOR DE ACEITUNAS DE MESA POR SU VARIEDAD ”								
AUTORES: Daniel Fajardo Zambrano Milton Toalombo Goyes				REVISORES: Ing. Erick González Linch, Mgs.				
INSTITUCIÓN: Universidad de Guayaquil				FACULTAD: Ciencias Matemáticas y Físicas				
CARRERA: Ingeniería en Sistemas Computacionales								
FECHA DE PUBLICACIÓN:				Nº DE PÁGS:				
ÁREA TEMÁTICA: BIOTECNOLOGÍA, BIOÉTICA Y BIODERECHO.								
PALABRAS CLAVES: Extracción de Características, Procesamiento digital de Imágenes, Reconocimiento de Patrones, Segmentación de Imágenes, Sistema de Visión Artificial.								
RESUMEN: El consumo de aceitunas se ha incrementado debido a su utilización en el área de la gastronomía, el éxito de su producción se origina en países europeos como; España, Italia, Francia, entre otros, extendiéndose a países latinoamericanos como Argentina y Uruguay. Existe una amplia variedad de aceitunas, siendo el sujeto de estudio de esta investigación las aceitunas de mesa las cuales son verdes, cambiantes y negras, su calidad depende del nivel de madurez al momento de su recolección. Ante tal situación, el presente trabajo tiene como objetivo el desarrollo de un clasificador de aceitunas de mesa por su variedad mediante la metodología del procesamiento digital de imágenes que considera un conjunto de técnicas de coloración, grises, píxeles, entre otros, tomando en cuenta las características importantes del fruto como el color, la forma y textura, las mismas que se obtienen mediante redes neuronales convolucionales, el cual generará un modelo que permitirá detectar las aceitunas mediante la webcam, captando aquellas que se encuentren en óptimas condiciones para ser procesadas ofrezcan una mejor calidad en la preparación de platos gourmet.								
Nº DE REGISTRO (en base de datos):				Nº DE CLASIFICACIÓN				
DIRECCIÓN URL:								
ADJUNTO PDF				SI <input checked="" type="checkbox"/>		NO <input type="checkbox"/>		
CONTACTO CON AUTORES: Daniel Fajardo Zambrano Milton Toalombo Goyes				Teléfono: 0979465674 042966333		Email: daniel.fajardo@ug.edu.ec milton.toalombog@ug.edu.ec		
CONTACTO DE LA INSTITUCIÓN: Universidad de Guayaquil Víctor Manuel Rendón y Baquerizo Moreno				Nombre: Ab. Juan Chávez Atocha Esp.				
				Teléfono: 2307729				
				Email: juan.chaveza@ug.edu.ec				

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de investigación, “DESARROLLO DE UN CLASIFICADOR DE ACEITUNAS DE MESA POR SU VARIEDAD” elaborado por el Sr. Fajardo Zambrano Daniel Arturo y el Sr. Toalombo Goyes Milton Franklin, **Alumnos no titulados** de la Carrera de Ingeniería en Sistemas Computacionales, Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil, previo a la obtención del Título de Ingeniero en Sistemas Computacionales, me permito declarar que luego de haber orientado, estudiado y revisado, la Apruebo en todas sus partes.

Atentamente,

**ING. ALFONSO GUIJARRO RODRÍGUEZ, MGS.
TUTOR**

DEDICATORIA

Este trabajo de titulación está dedicado principalmente a Dios por brindarme la sabiduría necesaria que me ha permitido llegar tan lejos.

A mis padres quienes me han brindado todo su apoyo incondicional todos los días.

Daniel Arturo Fajardo Zambrano

Dedico el presente trabajo de titulación a mi familia por apoyarme en los momentos necesarios y a los instructores que han impartido conocimiento durante el transcurso que he estudiado la carrera de Ingeniería en Sistemas Computacionales.

Milton Franklin Toalombo Goyes

AGRADECIMIENTO

Agradezco principalmente a Dios por darme el conocimiento y fortaleza que me han permitido llegar hasta tan lejos.

A mis padres que han sido un pilar fundamental en mi vida, al brindarme los recursos necesarios e inculcarme valores que me han convertido en una persona de bien.

Agradezco también a mi tutor de titulación Ing. Alfonso Guijarro que, gracias al conocimiento compartido con nosotros, nos ha ayudado para culminar el presente trabajo.

Daniel Arturo Fajardo Zambrano

En primer lugar, agradezco a mis padres por la contribución en mi educación, tanto académica, como de la vida, por su incondicional apoyo. Un especial agradecimiento a mi tutor de titulación Ing. Alfonso Guijarro la cual, con su infinita sabiduría y conocimiento que ha compartido con nosotros, nos ha apoyado para avanzar y culminar el presente trabajo.

Milton Franklin Toalombo Goyes

TRIBUNAL PROYECTO DE TITULACIÓN

Ing. Fausto Cabrera Montes, M.Sc.
DECANO DE LA FACULTAD
CIENCIAS MATEMÁTICAS Y
FÍSICAS

Ing. Gary Reytez Zambrano, Mgs.
DIRECTOR DE LA CARRERA
DE INGENIERÍA EN
SISTEMAS
COMPUTACIONALES

Ing. Alfonso Guijarro Rodríguez,
Mgs.
PROFESOR TUTOR DEL
PROYECTO DE TITULACIÓN

Ing. Erick González Linch,
Mgs.
PROFESOR REVISOR DEL
PROYECTO DE TITULACIÓN

Ab. Juan Chávez Atocha, Esp.
SECRETARIO

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Proyecto de Titulación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la UNIVERSIDAD DE GUAYAQUIL”

Daniel Arturo Fajardo Zambrano

Milton Franklin Toalombo Goyes



UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

**CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

DESARROLLO DE UN CLASIFICADOR DE ACEITUNAS DE
MESA POR SU VARIEDAD

Proyecto de Titulación que se presenta como requisito para optar por el título de
INGENIERO EN SISTEMAS COMPUTACIONALES

Autor: Daniel Arturo Fajardo Zambrano
C.I. 0929118016

Autor: Milton Franklin Toalombo Goyes
C.I. 0921893657

Tutor: Ing. Alfonso Guijarro Rodríguez, Mgs.

Guayaquil, 06 Marzo de 2020

CERTIFICADO DE ACEPTACIÓN DEL TUTOR

En mi calidad de Tutor del proyecto de titulación, nombrado por el Consejo Directivo de la Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil.

CERTIFICO:

Que he analizado el Proyecto de Titulación presentado por los estudiantes Daniel Arturo Fajardo Zambrano y Milton Franklin Toalombo Goyes, como requisito previo para optar por el título de Ingeniero en Sistemas Computacionales cuyo título es:

DESARROLLO DE UN CLASIFICADOR DE ACEITUNAS DE MESA POR SU VARIEDAD

Considero aprobado el trabajo en su totalidad.

Presentado por:

Fajardo Zambrano Daniel Arturo
Toalombo Goyes Milton Franklin

0929118016
0921893657

Tutor: Ing. Alfonso Guijarro Rodríguez, Mgs.

Guayaquil, 06 Marzo de 2020



UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

Autorización para Publicación de Proyecto de Titulación en Formato Digital

1. Identificación del Proyecto de Titulación

Nombre Alumno: Daniel Arturo Fajardo Zambrano	
Dirección: Durán, Cdla. Abel Gilbert Mz C23 V4	
Teléfono: 0979465674	E-mail: daniel.fajardo@ug.edu.ec

Nombre Alumno: Milton Franklin Toalombo Goyes	
Dirección: Sauces 6 Mz 349 V7	
Teléfono: 2966333	E-mail: milton.toalombog@ug.edu.ec

Facultad: Ciencias Matemáticas y Física
Carrera: Ingeniería en Sistemas Computacionales
Proyecto de titulación al que opta:
Profesor guía: Ing. Alfonso Guijarro Rodríguez, Mgs.

Título del Proyecto de titulación:
Desarrollo de un clasificador de aceitunas de mesa por su variedad.

Tema del Proyecto de Titulación:
Aceituna, Clasificador, Python, Tensorflow

2. Autorización de Publicación de Versión Electrónica del Proyecto de Titulación

A través de este medio autorizo a la Biblioteca de la Universidad de Guayaquil y a la Facultad de Ciencias Matemáticas y Físicas a publicar la versión electrónica de este Proyecto de titulación.

Publicación electrónica:

Inmediata	X	Después de 1 año	
-----------	---	------------------	--

Firma Alumno:

Daniel Arturo Fajardo Zambrano

Milton Franklin Toalombo Goyes

3. Forma de envío:

El texto del proyecto de titulación debe ser enviado en formato Word, como archivo .Doc. O .RTF y Puf para PC. Las imágenes que la acompañen pueden ser: .gif, .jpg o .TIFF.

DVDROM

☐

CDROM



X

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR	III
DEDICATORIA	IV
AGRADECIMIENTO	V
DECLARACIÓN EXPRESA	VII
CERTIFICADO DE ACEPTACIÓN DEL TUTOR	IX
ÍNDICE GENERAL	XI
ABREVIATURAS	XIII
ÍNDICE DE CUADROS	XIV
ÍNDICE DE GRÁFICOS	XV
RESUMEN	XVI
ABSTRACT	XVII
INTRODUCCIÓN	1
CAPÍTULO I - EL PROBLEMA	5
PLANTEAMIENTO DEL PROBLEMA	5
Ubicación del Problema en un Contexto	5
Situación Conflicto Nudos Críticos	7
Causas y Consecuencias del Problema	8
Delimitación del Problema	9
Formulación del Problema	9
Evaluación del Problema	9
OBJETIVOS	10
ALCANCES DEL PROBLEMA	11
JUSTIFICACIÓN E IMPORTANCIA	11
METODOLOGÍA DEL PROYECTO	13
CAPÍTULO II - MARCO TEÓRICO	14
ANTECEDENTES DEL ESTUDIO	14
FUNDAMENTACIÓN TEÓRICA	15
ACEITUNAS	15
DEFINICIÓN	15
ORIGEN	15
TIPO ACEITUNA	16
TIPO DE RECOLECCIÓN ACEITUNA	21
CONSERVACIÓN	25
FORMA DE PRESENTACIÓN	26

DEFECTOS QUE AFECTAN LA CALIDAD DE LAS ACEITUNAS	26
NORMATIVAS DE CALIDAD DE LAS ACEITUNAS.....	27
VISIÓN ARTIFICIAL	29
PROCESAMIENTO DIGITAL DE IMÁGENES	29
DEFINICIÓN	29
FORMATO DE IMAGEN	29
FUNDAMENTACIÓN LEGAL	47
PREGUNTA CIENTÍFICA A CONTESTARSE.....	53
VARIABLES DE LA INVESTIGACIÓN	53
DEFINICIONES CONCEPTUALES.....	54
CAPÍTULO III - METODOLOGÍA DE LA INVESTIGACIÓN.....	58
Análisis de Factibilidad	58
Factibilidad Operacional.....	58
Factibilidad Técnica	59
Factibilidad Legal	60
Factibilidad Económica	60
Etapas de la metodología del proyecto	61
Criterios de Validación.....	78
CAPÍTULO IV - RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	79
RESULTADOS	79
CONCLUSIONES.....	82
RECOMENDACIONES	83
BIBLIOGRAFÍA	84
ANEXOS.....	89

ABREVIATURAS

UG	Universidad de Guayaquil
CISC	Carrera de Ingeniería en Sistemas Computacionales
CC.MM.FF	Facultad de Ciencias Matemáticas y Físicas
Msc.	Master
RGB	Rojo Verde Azul
ANN	Red Neuronal Artificial
HSV	Matiz, Saturación, Valor
PCA	Análisis de Componentes Principales
KNN	K Vecinos más cercanos
IDE	Entorno de Desarrollo Integrado
SVM	Máquina de Soporte Vectorial
MLP	Perceptrón multicapa
Gr	Gramos
Cm	Centímetros
Tn	Tonelada
PDI	Procesamiento Digital de Imágenes
FAO	Organización de las Naciones Unidas para la Alimentación y la Agricultura
HSI	Matiz, Saturación, Intensidad
COI	Consejo Oleícola Internacional
RNC	Redes Neuronales Convolucionales
ROC	Característica Operativa del Receptor
RAM	Memoria de Acceso Aleatorio
SO	Sistema Operativo
FFT	Transformada Rápida de Fourier
DFT	Transformada Discreta de Fourier

ÍNDICE DE CUADROS

Cuadro 1. Causas y consecuencias del problema.....	8
Cuadro 2. Delimitación del problema.....	9
Cuadro 3. Variedades de aceitunas.....	16
Cuadro 4. Factores esenciales de calidad (COI, CODEX y Españolas) – Características que debe poseer el producto final	27
Cuadro 5. Formatos de Imagen.....	30
Cuadro 6. Funciones de DFT de la librería numpy*	33
Cuadro 7. Algoritmo de la Función Perdida	45
Cuadro 8. Requerimientos de Ordenadores.....	59
Cuadro 9. Características de celulares	59
Cuadro 10. Software y librerías utilizadas	60
Cuadro 11 Tabla de Presupuesto	61
Cuadro 12. Características del tamaño	72
Cuadro 13. Comparativa configuración de Yolo	74
Cuadro 14. Visualización de los puntos de controles generados	77
Cuadro 15. Usuarios que realizaron juicio de expertos	78

ÍNDICE DE GRÁFICOS

Gráfico 1. Técnicas de recolección de aceitunas	2
Gráfico 2. Aceituna manzanilla	17
Gráfico 3. Aceituna gordal.....	17
Gráfico 4. Aceituna durzal.....	18
Gráfico 5. Aceituna hojiblanca	18
Gráfico 6. Aceituna cacareña.....	19
Gráfico 7. Aceituna verdial.....	19
Gráfico 8. Aceituna aloreña.....	20
Gráfico 9. Aceituna picual	20
Gráfico 10. Uso de las Aceitunas	21
Gráfico 11. Recolección manual de la aceituna caída al suelo	22
Gráfico 12. Utilización de una escalera para la recolección por ordeño	23
Gráfico 13. Vareo para la recolección de aceitunas	23
Gráfico 14. Proceso de recolección mecanizada usando mantor y vibrador en un tractor.....	24
Gráfico 15. Recolección de olivares usando cosechadora tipo Vendimiadora	25
Gráfico 16. Esquema del Procesamiento Digital de Imágenes.....	34
Gráfico 17 Red Multicapa y Monocapa	40
Gráfico 18. Estructura de una Red Neuronal Convolutiva.....	41
Gráfico 19. Función Identidad.....	43
Gráfico 20. Función Escalón binario.....	44
Gráfico 21. Función Logística.....	44
Gráfico 22. Función Tangencial	44
Gráfico 23. Función Softplus	45
Gráfico 24. Función rectificadora	46
Gráfico 25. Ejemplo de Curvas ROC	46
Gráfico 26. Diagrama de Bloques de la metodología	62
Gráfico 27. Etapas de la metodología del proyecto.....	62
Gráfico 28. Obtención de imágenes mediante la línea de comandos	64
Gráfico 29. Imágenes ubicadas en el ordenador por la ejecución del comando anterior	64
Gráfico 30. Escenarios para la toma de imágenes.....	65
Gráfico 31. Imágenes tomadas.....	65
Gráfico 32. Redimensión de imagen gordal	66
Gráfico 33. Redimensión de imagen botija	67
Gráfico 34. Imágenes generadas para el aumento de datos.....	68
Gráfico 35. Etiquetado de imagen	69
Gráfico 36. Etiquetado manual de la aceituna manzanilla	69
Gráfico 37. Visualización de un archivo xml generado	70
Gráfico 38. Variación en la forma y tamaño	72
Gráfico 39. Archivos del Darkflow	73
Gráfico 40. Ejecución del comando del entrenamiento	76
Gráfico 41. Época 26 de fase del entrenamiento del modelo	76
Gráfico 42 Interfaz de usuario del clasificador.....	79
Gráfico 43. Visualización de Información de la configuración.....	80
Gráfico 44 Detección con el checkpoint 125	80
Gráfico 45 Detección con el checkpoint 3250	81
Gráfico 46 Detección con checkpoint 9375	81
Gráfico 47 Carpeta con capturas.....	81



**UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**DESARROLLO DE UN CLASIFICADOR DE ACEITUNAS DE MESA
POR SU VARIEDAD.**

Autores:

Daniel Arturo Fajardo Zambrano

Milton Franklin Toalombo Goyes

Tutor: Ing. Alfonso Guijarro Rodríguez, Mgs.

RESUMEN

El consumo de aceitunas se ha incrementado debido a su utilización en el área de la gastronomía, el éxito de su producción se origina en países europeos como; España, Italia, Francia, entre otros, extendiéndose a países latinoamericanos como Argentina y Uruguay. Existe una amplia variedad de aceitunas, siendo el sujeto de estudio de esta investigación las aceitunas de mesa las cuales son verdes, cambiantes y negras, su calidad depende del nivel de madurez al momento de su recolección. Ante tal situación, el presente trabajo tiene como objetivo el desarrollo de un clasificador de aceitunas de mesa por su variedad mediante la metodología del procesamiento digital de imágenes que considera un conjunto de técnicas de coloración, grises, píxeles, entre otros, tomando en cuenta las características importantes del fruto como el color, la forma y textura, las mismas que se obtienen mediante redes neuronales convolucionales, el cual generará un modelo que permitirá detectar las aceitunas mediante la webcam, captando aquellas que se encuentren en óptimas condiciones para ser procesadas ofrezcan una mejor calidad en la preparación de platos gourmet.

Palabras claves: Extracción de Características, Procesamiento digital de Imágenes, Reconocimiento de Patrones, Segmentación de Imágenes, Sistema de Visión Artificial.



**UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMATICAS Y FISICAS
CARRERA DE INGENIERIA EN SISTEMAS COMPUTACIONALES**

**DEVELOPMENT OF A CLASSIFIER OF TABLE OLIVES FOR ITS
VARIETY.**

Authors:

Daniel Arturo Fajardo Zambrano

Milton Franklin Toalombo Goyes

Tutor: Ing. Alfonso Guijarro Rodríguez, Mgs.

ABSTRACT

The consumption of olives has increased due to its use in the area of gastronomy, the success of its production originates in European countries such as; Spain, Italy, France, among others, extending to Latin American countries such as Argentina and Uruguay. There is a wide variety of olives, being the subject of this research the table olives which are green, changing and black, their quality depends on the level of maturity at the time of collection. Given this situation, the present work aims to develop a table olives sorter by its variety through the methodology of digital image processing that considers a set of coloring techniques, gray, pixels, among others, taking into account the characteristics important fruit such as color, shape and texture, the same that are obtained through convolutional neural networks, which will generate a model that will detect the olives through the webcam, capturing those that are in optimal conditions to be processed offer a better quality in the preparation of gourmet dishes.

Keywords: Artificial Vision System, Digital Image Processing, Feature Extraction, Image Segmentation, Pattern Recognition.

INTRODUCCIÓN

En España, una de las actividades agrícolas de mayor relevancia es el cultivo de olivo, producido principalmente en la cuenca mediterránea, su laboreo se ha extendido en todo el mundo y durante los últimos años es practicada en otros continentes. Dicha expansión se da por la popularidad y el aumento del consumo de productos que derivan por el cultivo siendo estas las aceitunas de mesa y el aceite de oliva. (Ponce et al, 2018).

Las aceitunas de mesa han ganado terreno en el ámbito gastronómico recientemente debido a que estos han generado un contenido en los consumidores al utilizarlo como herramienta en el arte culinario para la elaboración de platos gourmet, para la preparación las personas lo tienden a endulzar lavando repetidamente, y realizando inmersiones de salmuera, para reducir en la medida de lo posible el característico sabor amargo que posee la fruta; Sin embargo, otras tienden aliñar añadiendo diferentes especias como puede ser limón, naranja, laurel, entre otros, para darle un sabor adicional.

Las técnicas en la recolección de aceitunas han cambiado con el pasar de los años, inicialmente se recogían directamente del árbol con la mano, esta técnica se la conoce como el ordeño, el vareo es una técnica realizada por hombres, el cual utilizaban una vara larga con la que golpeaban las ramas de olivo hasta que caían sobre unas mantas de lona para después ser limpiadas, recoger la aceituna del suelo era practicado por las mujeres y se usaban pequeñas cestas. Las aceitunas eran transportadas hasta la almazara con la ayuda de animales. En la actualidad, se utilizan máquinas agrícolas que agilizan el trabajo como el uso de tractores con remolques para el transporte, el vibrador es un brazo que se acopla a las ramas de olivos y las sacude, la barredora es un vehículo pequeño que absorbe y separa las aceitunas de ramas y piedras, la sopladora sopla la aceituna derribada del árbol y el peine el cual usa

una vara mecánica para golpear aceitunas hasta que caigan al suelo. (Ferguson, 2006).

Gráfico 1. Técnicas de recolección de aceitunas



Elaborado por: IMIDRA
Fuente: (Olive, 2005)

El gráfico 1 corresponde a la recolección de aceitunas en años anteriores cuando la aplicación de maquinaria para la recolección recién se usaba. De acuerdo con el estado de madurez del fruto al momento de su recolección, las aceitunas de mesa las podremos clasificar de la siguiente manera: verde, cambiante y negra. Las aceitunas verdes son frutos recogidos en su estado de madurez idóneo, llegando alcanzar su tamaño normal. Las aceitunas de color cambiante son aquellas recogidas antes de su estado de madurez ideal, debido a esto presentan colores como rosa vinoso, castaño o rosáceo. Las aceitunas negras son aquellas recogidas en plena estado de madurez, presenta variación de color como negro rojizo, negro verdoso o violeta oscuro. Este tipo de frutas poseen un

valor nutricional elevado, dado que tiene grandes cantidades ácidos grasos monoinsaturados y fibras, además que aporta vitaminas A y C.

Aguilera et al (2015) proponen un sistema automático de clasificación de lotes de aceitunas utilizando visión por computador con el objetivo de diferenciar las aceitunas según nivel de calidad separando las aceitunas recogidas del suelo y del árbol, utilizando una metodología que contenga histogramas de imágenes obtenidas en diferentes espacios de color y como clasificador una red neuronal artificial ANN obteniendo un coeficiente de éxito de 97.1%.

Montoya, Cortés & Chaves (2014) proponen la construcción de una báscula electrónica que permite clasificar diferentes frutas en base al tamaño, forma y color, utilizando modelo RGB y el clasificador Bayesiano, logrando obtener resultados con 90% de exactitud para la modernización en los supermercados.

Gonzalez & Ortiz (2016) proponen el desarrollo de un sistema de visión por computador para detectar el estado de madurez de los frutos de guayaba aplicando el índice de color y el espacio RGB, CIELAB, YCbCr y HSV, llevando a cabo un análisis por componentes principales (PCA) para reducir datos redundantes, que generaba datos irrelevantes, logrando obtener resultados con un 90% de precisión.

Herrera, Medina & Ortiz (2016) plantean el desarrollo de dos algoritmos de procesamiento de imágenes, uno para para clasificar el café según su estado de maduración utilizando redes neuronales artificiales y el otro para detectar broca desarrollado mediante criterio de binarización para encontrar zonas negras en la imagen similares a los orificios que deja esta plaga, el clasificador tuvo un 97% de efectividad detectando estados de madurez.

Sandoval & Prieto (2009) presentan un sistema que permita clasificar frutos de café en ocho categorías según el estado de maduración usando técnicas de visión artificial, a partir de nueve características en las que se incluye forma, medidas de color y textura del análisis de la imagen del fruto, el desempeño del clasificador bayesiano corresponde a un 96.88% de éxito.

El presente trabajo está conformado por 4 capítulos, los cuales se mencionan a continuación:

Capítulo I denominado el Problema abarcará todo lo referente a la problemática, como pueden ser las causas y consecuencias, los nudos críticos, la delimitación, el alcance, los objetivos generales y específicos que abordan este proyecto, justificación e importancia de por qué es necesario realizarlo y la metodología empleada.

Capítulo II denominado Marco Teórico: que abarcará los antecedentes del estudio, fundamentación teórica donde se explicará lo relacionado a las aceitunas de mesa, y el procesamiento digital de imágenes junto con sus fases, fundamentación legal y las definiciones conceptuales.

Capítulo III denominado Metodología se mencionará la metodología utilizada para el desarrollo del clasificador de aceitunas, para la cual se utiliza el procesamiento digital de imágenes que consta de diversas técnicas orientada a la coloración, contraste, píxeles en las imágenes y que estará conformada por 5 etapas, las cuales son Captura, Pre-procesamiento, Segmentación, Extracción de características y la clasificación.

Capítulo IV denominado Resultados, Conclusiones y Recomendaciones abarcará los resultados del proyecto, futuros trabajos, se incluye un informe en el que se mostrará el porcentaje de efectividad del clasificador, se darán ciertas recomendaciones para los futuros trabajos relacionados y se detallarán las conclusiones obtenidas durante el desarrollo del clasificador.

CAPÍTULO I

EL PROBLEMA

PLANTEAMIENTO DEL PROBLEMA

Ubicación del Problema en un Contexto

La Carrera de Ingeniería de Sistemas Computacionales posee un claustro docente compuesto por profesionales de diferentes disciplinas, siendo los de ingeniería, quienes tributan en mayor proporción a la formación de los ingenieros en sistemas computacionales, dentro de este grupo humano de profesionales se encuentran los docentes investigadores, quienes a través de propuestas innovadoras presentan temas experimentales para llevar a la praxis temas asociados a la ciencia de la computación como es la visión artificial. En esta ocasión se pretende desarrollar un clasificador que bajo escenarios controlados logre determinar las características óptimas para seleccionar las aceitunas de mesa para uso en la gastronomía.

A través de los años, la manera en la que se realiza la recolección de aceitunas se ha visto envuelta en cambios, antiguamente la necesidad de contratar mano de obra era necesaria pero resultaba muy cara, la mano de obra era usada para técnicas como el vareo que consiste en derribar la aceituna haciendo uso de una vara tratando de no generar mucho daño a las ramas, la técnica del ordeño consiste en recoger a mano las aceitunas directamente del árbol con lo que se asegura que la aceituna esté entera y libre de golpes, la técnica de recogida del suelo como su nombre lo indica consiste en recoger del suelo las aceitunas que han caído de forma natural debido a que han llegado a su estado de madurez completa. Actualmente algunas de estas técnicas se han modernizado implementando la ayuda de máquinas, en el caso de vareo se cambia la vara de madera por una de fibra de vidrio, en la técnica de recogida del suelo se aspira o sopla las aceitunas con la finalidad de amontonarlas

para luego ser transportadas, la técnica de la vibración utiliza una pinza impulsada por un tractor que sacude el árbol para que las aceitunas caigan, al caer son recogidas con un paraguas evitando golpes en el suelo.

En la recolección de las aceitunas se debe tener en consideración los diferentes factores, cómo puede ser la especie, la estación, el suelo donde ha sido cultivado, exposición al sol, entre otros. A manera general se realiza la recolección en épocas de octubre donde su fruto ha alcanzado su tamaño ideal y color real. Sin embargo, países mediterráneos de Europa como Francia, Croacia, España lo recolectan a principios de diciembre, en países como Italia, Montenegro y Bulgaria entre mediados de enero y febrero e inclusive algunos países no lo recolectan hasta que los frutos caen de manera natural.

Para una empresa donde su actividad principal es la cosecha y la recolección de aceitunas, es necesario que dicha fruta posea una mejor calidad para cumplir con las expectativas de los consumidores; por lo que es necesario invertir en mano de obra que realice la clasificación de las aceitunas por su madurez, el inconveniente que se presenta es el costo de la mano de obra para la recolección que suele ser alta, debido a que no hay una máquina que optimice dicho proceso, además la clasificación manual posee un margen de incertidumbre elevado, producido por miembros nuevos del personal de recolección, personal con nulo conocimiento o por las repetidas actividades que ejerce el personal que pueden pasar inadvertido al momento de clasificar las frutas para que lleguen a su destino.

La calidad de las aceitunas es importante al momento de comercializarlas, para ello se debe tomar en consideración ciertos aspectos necesarios como el estado, olor, sabor, presencia de patógenos, entre otros, teniendo en cuenta tres categorías establecidas por las normativas CODEX

Alimentarius: Extra, en la que se considera las aceitunas de calidad superior admitiendo hasta un 5% de tolerancias como puede ser frutos manchados, mutilados, arrugados, textura y coloración anormal; Primera o selecta, abarca las aceitunas de buena calidad con un agrado adecuado de madurez admitiendo hasta en un 10% de tolerancia; Segunda o estándar, considera aquellas aceitunas que no han sido clasificados en las otras dos categorías.

El desarrollo del clasificador resulta necesario debido a que permitirá identificar a las aceitunas para que al ser procesadas ofrezcan una mejor calidad reduciendo el margen de error durante su clasificación. Para ello, la metodología empleada es el procesamiento digital de imágenes que abarca un conjunto de técnicas de coloración, grises, píxeles, entre otros, tomando en cuenta las características del fruto como el color, la forma y textura, logrando obtener datos que se analizarán para posteriormente acoplarlo a un clasificador utilizando redes neuronales convolucionales y poder conseguir el resultado idóneo.

Situación Conflicto Nudos Críticos

La ausencia de conocimientos previos resulta un inconveniente al tratar de realizar el clasificador por lo que es necesario investigar conceptos básicos y leer trabajos similares de diferentes autores. La correcta selección del Entorno de Desarrollo Integrado (IDE) es fundamental debido a que se lo utilizará a lo largo del proyecto, sin embargo, la tendencia para los clasificadores es el uso de Tensorflow por lo que es necesario que el IDE permita la instalación de esta librería.

La obtención de muestras de aceitunas para el procesamiento de imágenes del clasificador, dado que en la fase de entrenamiento del clasificador se debe contar con una gran cantidad de imágenes y entre estas deben estar los tres tipos de aceitunas de mesa: verde, cambiante,

y negra; con la finalidad de aumentar el porcentaje de éxito. Para la extracción de características se debe tomar en cuenta el tamaño, color y textura del fruto, la cual existen varias técnicas y métodos como pueden ser: Transformada de Gabor, Descriptores de Fourier, entre otros. Existen diversos clasificadores que ayudan en la organización, como puede ser el Binario, Bayesiano, Support Vector Machines (SVM), multilayer perceptron (MLP), entre otras. Se debe seleccionar una de ellas donde el resultado obtenido presente un margen de error menor, para posteriormente poder desarrollarla con la herramienta Tensorflow.

Causas y Consecuencias del Problema

En el cuadro 1 se presentan algunas causas y consecuencias generadas a partir del problema:

Cuadro 1. Causas y consecuencias del problema
Causas **Consecuencias**

Aceitunas con presencia de parásito	Disminución calidad del producto
Aparición de enfermedades del olivo en su etapa de madurez	Disminución en la producción de aceitunas
Presencia de materias extrañas o impropias de la madurez	Dificultad en la detección de la imagen
Falta o no uso de tecnología	Disminución del tiempo de recolección y clasificación de aceitunas de mesa
No uso de software open source	Terceras personas no podrán rediseñar el clasificador utilizando nuevas herramientas
La fatiga visual por parte del personal encargado	Una clasificación imprecisa afectando a la calidad de las aceitunas
Procesado o conservación errónea de las aceitunas de mesa	Aparición de olores o sabor desagradables
Variaciones climáticas extremas	Muerte de brotes, ramas o la planta completa de olivo

Elaborado por: Daniel Fajardo, Milton Toalombo
Fuente: Información del Proyecto

Delimitación del Problema

En el cuadro 2 se detalla la delimitación del problema

Cuadro 2. Delimitación del problema

Campo	Investigación
Área	Tecnológica - Agricultura
Aspecto	Prototipo clasificador de aceitunas
Tema	Desarrollo de un clasificador de aceitunas de mesa por su variedad

Elaborado por: Daniel Fajardo, Milton Toalombo
Fuente: Información del Proyecto

Formulación del Problema

¿Cómo el desarrollar un clasificador de aceitunas de mesa por su variedad ayuda a los procesos gourmet?

Evaluación del Problema

Se tomará en cuenta los siguientes aspectos:

Delimitado: En tres meses se desarrollará en la ciudad de Guayaquil un clasificador de aceitunas que facilite la identificación de estas por su variedad.

Claro: Desarrollar un clasificador de aceitunas de mesa por su variedad con la finalidad de identificarlas para que al ser procesadas ofrezcan una mejor calidad para la preparación de platos gourmet.

Evidente: Reducir el margen de error en la clasificación de las aceitunas, mediante la utilización de las técnicas de machine learning permitiendo clasificarlas de manera automática.

Concreto: Elaborar un algoritmo que detecte el grado de madurez de las aceitunas de mesa para su clasificación.

Relevante: Mediante técnicas de procesamiento de imágenes se desarrollará el clasificador para aumentar el margen de éxito.

Factible: Es factible, porque a través de un algoritmo apoyado de tecnologías de procesamiento digital de imágenes se elabora el clasificador, logrando optimizar la manera de clasificar las aceitunas de mesa.

OBJETIVOS

OBJETIVO GENERAL

Desarrollar un clasificador de variedades de aceitunas de mesa mediante el reconocimiento de características que presenta el fruto, la cual estas son forma, color y tamaño aplicando herramientas que permitan identificar los patrones dados, para que al ser procesadas ofrezcan una mejor calidad en la preparación de platos gourmet.

OBJETIVOS ESPECÍFICOS

- 1.- Determinar los requerimientos básicos que permiten caracterizar las aceitunas de mesa para preparar escenarios preliminares del clasificador por su variedad en visión por computador.
- 2.- Aplicar técnicas del procesamiento digital de imágenes en el IDE a partir de la base de imágenes de pruebas para disminuir el margen de error del clasificador.
- 3.- Desarrollar los algoritmos que permitan la supresión de materiales u objetos que no pertenezcan al fruto y que permitan reconocer las variedades de aceitunas de mesa por color, forma y textura para determinar de mejor manera sus características y proceder con su clasificación.
- 4.- Aplicar criterios de validación al clasificador para contribuir con los índices de calidad en la selección de las aceitunas de mesa.

ALCANCES DEL PROBLEMA

- Para determinar los requerimientos básicos que permiten caracterizar las aceitunas de mesa es necesario contar con una base de fotografías, equipo que permita realizar el procesamiento, cámara para la adquisición de imágenes, entre otros.
- Para aplicar técnicas del procesamiento digital de imágenes es necesario establecer los formatos de las imágenes previamente recolectadas, aplicar descriptores para su respectiva extracción de características acorde al color, forma y textura que posee las aceitunas de mesa y la selección de un IDE de programación.
- Para desarrollar los algoritmos de supresión de materiales u objetos externos, es necesario aplicar técnicas de reconocimiento de borde, eliminación de contraste, brillo, entre otros. Además, determinar los píxeles de la paleta de colores.
- Para aplicar criterios de validación, es necesario, definir los requerimientos del clasificador, separar imágenes para las pruebas y entrenamiento, preparar instrumentos. Además, para elevar el índice de calidad es necesario determinar cuándo un fruto alcanza la máxima calidad planteada por expertos.

JUSTIFICACIÓN E IMPORTANCIA

Existen 18 tipos de aceituna que pueden usarse tanto en aceite como también aceitunas de mesa, por lo que es importante clasificar los frutos de mejor característica para aumentar la producción de aceitunas. En España, la aceituna se ha convertido en algo esencial para la cocina debido a sus propiedades nutricionales que ayudan a mantener una dieta saludable, es sencillo encontrarlas, tardan en descomponerse, por lo que

además de ser encontrado como aceite, las aceitunas de mesa acompañan bien a una ensalada.

Las aceitunas de mesa se pueden clasificar por su coloración, en la que se encuentran aceitunas verdes siendo estas las recogidas en su maduración óptima, aceitunas de color cambiante cuyo color suele ser violeta, rosáceo o castaño, son frutos recogidos antes de la completa madurez, aceitunas negras, las cuales provienen de frutos que fueron recogidos en plena madurez, también se puede clasificar por sus diversas variedades como es la Manzanilla de Sevilla, la cual es la aceituna de mesa más importada debido a la calidad del fruto; Gordal Sevillana, la cual es reconocida por su peso medio de 12,5 gr; Durzal, debido a que su productividad es alta en tiempo de cosecha, entre otros.

Cabe mencionar que el tamaño medio de las aceitunas es de 1 a 4 cm de longitud y de 0.6 a 2 cm de diámetro, la textura de la aceituna es importante al momento de exportarlas tomando en cuenta tres categorías: Extra, en la que se considera las aceitunas de calidad superior admitiendo hasta un 5% de tolerancias como puede ser frutos manchados, mutilados, arrugados, textura y coloración anormal. Primera o selecta, abarca las aceitunas de buena calidad con un agrado adecuado de madurez admitiendo hasta en un 10% de tolerancia; Segunda o estándar, considera aquellas aceitunas que no han sido clasificados en las otras dos categorías.

El desarrollo del clasificador resulta importante debido a que ayudará a las personas que se dedican a la cosecha y cultivo de aceitunas a reducir el margen de error durante la clasificación.

METODOLOGÍA DEL PROYECTO

Para la metodología se utiliza el Procesamiento Digital de Imágenes (PDI) que estará conformada en 5 fases las cuales son: captura, pre-procesamiento, segmentación, extracción de características y la clasificación utilizando RNC. En cada fase se abarcará los conocimientos respectivos tomando en cuenta las distintas técnicas y algoritmos que existen.

El procesamiento digital de imágenes permite manipular imágenes a través de la computadora según las necesidades de los usuarios como pueden ser: segmentación, filtrado, reconocimiento; teniendo como intención principal el desarrollo o mejora de algoritmos cuya aplicación está destinada al área de la robótica. (López, Pelayo & Forero, 2015).

CAPÍTULO II

MARCO TEÓRICO

ANTECEDENTES DEL ESTUDIO

El avance vertiginoso que presenta la tecnología, ha permitido ofrecer multiples soluciones en los diferentes procesos que asocian lo laboral, como la mano de obra en el campo, etapas de clasificación de expertos, entre otras, además con el surgimiento del aprendizaje automático se ha logrado aumentar la confianza en las personas de permitirle a los sistemas reemplazar el trabajo del hombre debido a los resultados mostrados en escenarios experimentales, donde las tasas de rendimiento en actividades rutinarias han permitido clasificar objetos, reconocimiento y detección de patrones, entre otros, la prueba de esto se soporta con trabajos de otros autores que han podido resolver problemas similares a lo propuesto en este proyecto.

Bonilla & Prieto (2016) mediante un sistema de visión por computador y varios tipos de camara determinan el estado de maduración de frutos de guayaba, para ello utilizaron características del índice de color, trabajaron con cuatro espacios de color RGB, CIELab, YCbCr y HSV además hicieron uso del clasificador bayesiano, k-vecinos y red neuronal de perceptrón multicapa para lograr categorizar dicha fruta en tres fases, inmadura, en maduración y madura con un 90% de éxito al implementar la red neuronal como clasificador.

Silva & Lizcano (2012) proponen usar visión artificial para evaluar el estado de madurez de la piña perolera, para obtener las imágenes se tomaron fotografías diarias a la fruta durante 11 días teniendo un total de

1500 imágenes, de la cual una parte fue usada en el proceso de entrenamiento y lo restante en el proceso de evaluación, trabajaron con dos espacios de color RGB y HSV, para la segmentación utilizan el método de Otsu además del uso del método de clasificación no supervisada por agrupamiento mediante cluster, generando una efectividad del 96,36%.

Marin & Madrigal (2016) proponen: usar máquinas de vectores de soporte y procesamiento de imagen para clasificar la calidad de fresas, para ello tomaron fotografías de algunas fresas para luego sustraerle el fondo y reducir la imagen, después en la segmentación hacen uso del modelo RBG y OHTA para continuar con la extracción de características donde toman en cuenta la forma, madurez y tamaño del fruto para finalmente pasarlas al clasificador SVM, logrando una precisión del 100% en la diferencia entre los 2 tipos y un error del 6,5% determinando el tamaño.

FUNDAMENTACIÓN TEÓRICA

ACEITUNAS

DEFINICIÓN

La aceituna es un pequeño fruto de forma elipsoidal cuyo tamaño puede inferir dependiendo de la variedad. En la etapa de madurez el color de las aceitunas se presenta desde un color rojizo hasta negro y en el caso de las aceitunas de mesa se cosechan verdes hasta llegar a un tono amarillento a rojizo negro. (Cabezas, 2011).

ORIGEN

Cabezas (2011) indica que el olivo es originario de Persia según historiadores teniendo gran importancia en las culturas romanas y árabes. Siendo los romanos aquellos responsables de la expansión del cultivo de olivo debido a que lo esparcieron por todo su imperio, el cultivo de olivo se introdujo en América en el siglo XVI con la llegada de los colonizadores españoles. El siglo XX ha sido testigo de un progresivo incremento a nivel

mundial de la elaboración y el consumo de aceitunas de mesa, hasta llegarse a más del millón de toneladas anuales de la actualidad (Fernandez & Barranco, 1999). Es cultivada en todos los países de la cuenca mediterránea siendo los principales España, Italia, Turquía, Marruecos. (Olive, 2015).

TIPO ACEITUNA

El cultivo de olivo se ha incrementado en los años recientes dando como resultado una gran variedad de especies de dicho fruto destinados a la mesa o elaboración de aceite de oliva, En el cuadro 3 se muestran las principales variedades de aceitunas.

Cuadro 3. Variedades de aceitunas

ACEITUNAS				
Variedad	Nombre Científico	Tipo	Localidad	Forma
Manzanilla de Sevilla	<i>Olea europaea pomiformis</i>	Mesa	Cuenca de Guadalquivir, Sevilla, Badajoz	Globosa
Gordal Sevillana	<i>Olea europaea regalis</i>	Mesa	Sevilla	Elongada
Hojiblanca	<i>Olea europaea arolensis</i>	Mesa / Aceite	Córdoba, Málaga, Sevilla y Granada	Globosa
Dulzal	Dulzal de Carmona	Mesa	Carmona, Arahál y Morón en Sevilla	Globosa
Cacereña	<i>Olea europea</i> var. <i>Sativa</i> u <i>Olea sylvestris</i>	Mesa / Aceite	Cáceres y Salamanca	Esférico de base truncada
Verdial	<i>Olea europea vridula</i>	Mesa / Aceite	Andalucía	Elongada
Aloreña	Aceituna Aloreña de Málaga	Mesa	Málaga	Globosa
Cordobesa	OFPEC Cordobesa	Mesa	Córdoba	Globosa
Lechín	<i>Olea europaea laperrinei</i>	Mesa	Granada / Sevilla.	Globosa
Obregón		Mesa	Sevilla	Globosa

Elaborado por: Daniel Fajardo, Milton Toalombo
Fuente: (Cabezas, 2011)

Manzanilla de Sevilla

Es la aceituna de mesa de mayor exportación globalmente, se caracteriza por tener una forma casi redonda, un tamaño mediano y un hueso pequeño. (véase gráfico 2).

Gráfico 2. Aceituna manzanilla



Elaborado por: Cabezas
Fuente: (Cabezas, 2011)

Gordal Sevillana

Conocida principalmente por su gran tamaño además de tener color verde con pequeñas manchas blancas, al poseer poco aceite su consumo se da en aderezo. (véase gráfico 3).

Gráfico 3. Aceituna gordal



Elaborado por: Cabezas
Fuente: (Cabezas, 2011).

Durzal.

Conocida también como Morona, tiene parecido con la manzanilla, aunque un poco más alargada, es muy usado en aderezos. (véase gráfico 4).

Gráfico 4. Aceituna durzal



Elaborado por: Cabezas
Fuente: (Cabezas, 2011).

Hojiblanca.

Esta variedad de aceituna es de maduración tardía, suele ser de tamaño grande con una coloración violeta a negro, su recolección tiende a ser mecanizada debido a su gran resistencia al desprendimiento. (véase gráfico 5).

Gráfico 5. Aceituna hojiblanca



Elaborado por: Cabezas
Fuente: (Cabezas, 2011).

Cacareña.

Es muy similar a la manzanilla en cuanto al aspecto, siendo un poco más clara, posee un tamaño medio grande con forma esférica, pese a poseer poco aceite este es de gran calidad por lo que esta variedad de aceituna es usada tanto para mesa como aceite. (véase gráfico 6).

Gráfico 6. Aceituna cacareña

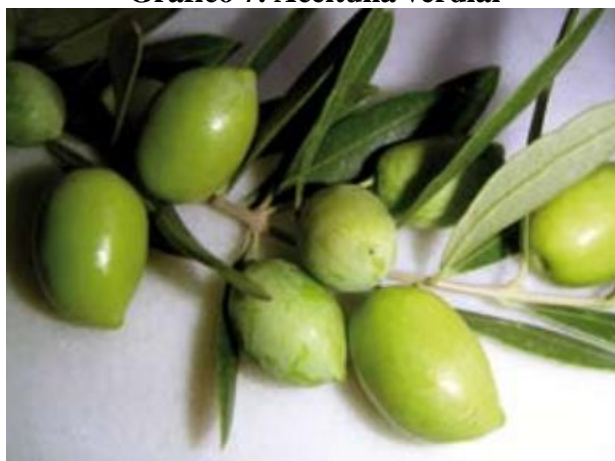


Elaborado por: Cabezas
Fuente: (Cabezas, 2011).

Verdial.

Es consumida para mesa, posee un sabor muy agradable y un tamaño mayor en comparación a otras aceitunas tipo manzanillas. (véase gráfico 7).

Gráfico 7. Aceituna verdial



Elaborado por: Cabezas
Fuente: (Cabezas, 2011).

Aloreña.

Esta variedad de aceituna es redonda con un bajo contenido en oleuropeína, que es el componente amargo de las aceitunas. (véase gráfico 8).

Gráfico 8. Aceituna aloreña



Elaborado por: Cabezas
Fuente: (Cabezas, 2011).

Cabe mencionar que, en cuanto a aceite de oliva, Jaén es la principal región productora, la variedad que produce principalmente es el Olivo Picual (*Olea Europea Sativa*) siendo esta una de las clases de aceitunas muy importante para la producción de aceite de oliva de gran calidad. (véase gráfico 9).

Gráfico 9. Aceituna picual



Elaborado por: Cabezas
Fuente: (Cabezas, 2011).

En cuanto a su grado de madurez, para la Organización de las Naciones Unidas para la Alimentación y la Agricultura FAO (2013) se clasifican de la siguiente manera:

Aceitunas verdes. – Son recogidos durante su periodo de madurez una vez alcanzado su tamaño normal.

Aceitunas de color cambiante. – Son recogidos antes de alcanzar su completa madurez.

Aceitunas negras. – Son recogidos en plena madurez.

En el gráfico 10 se mencionan las aceitunas de mesa y de aceite a través de un diagrama de Venn



TIPO DE RECOLECCIÓN ACEITUNA

La recolección es un factor que influye en la calidad de las aceitunas, pero no siempre es posible recogerlas en su momento óptimo, para ello se dispone de diferentes métodos de recolección, estos pueden ser a mano, con la ayuda de máquinas o golpeando el árbol. (Olive, 2015).

La recolección manual es el sistema predilecto para la cosecha de frutas y verduras de consumo fresco, en las que se pretende obtener una calidad superior. Mientras que la recolección mecanizada es preferida en cultivos con fines industriales en los que se realiza un procedimiento adicional

para obtener un producto final en base a la fruta cosechada. (Zuñiga, Riquelme & González, 2019)

García, Molino, Ruiz, Cano & Pérez. (2017) menciona que los Sistemas Tradicionales son los siguientes:

Recogida del fruto del suelo

Consiste en la recolección manual de las aceitunas que estén caídas al suelo de manera natural; Sin embargo, la calidad de las aceitunas recolectadas por esta forma, tienden a ser malas. (véase gráfico 11)

Gráfico 11. Recolección manual de la aceituna caída al suelo



Elaborado por: García, Molino, Ruíz, Cano & Pérez
Fuente: (García, Molino, Ruíz, Cano & Pérez, 2017)

Ordeño

Un sistema empleado antiguamente para las aceitunas de mesa consiste en coger manualmente las aceitunas del árbol y depositarlas en pequeños recipientes que lleva el operario llamado macaco, pasándola después a cajas para su transporte. Dependiendo del tamaño de los árboles, tienen que utilizar de escaleras. (véase gráfico 12).

Gráfico 12. Utilización de una escalera para la recolección por ordeño



Elaborado por: García, Molino, Ruíz, Cano & Pérez
Fuente: (García, Molino, Ruíz, Cano & Pérez, 2017)

Vareo

Es uno de los métodos más usados para el derribo de las aceitunas del árbol, consiste en golpear las ramas del árbol procurando que el golpe no cause un daño grave a los frutos, sin embargo, golpear la aceituna merma la calidad de esta. (véase gráfico 13).

Gráfico 13. Vareo para la recolección de aceitunas



Elaborado por: García, Molino, Ruíz, Cano & Pérez
Fuente: (García, Molino, Ruíz, Cano & Pérez, 2017)

Los sistemas mecanizados son los siguientes:

Mantor y Vibrador

Un tractor que posee un vibrador y bastidor unidos a los mismos brazos que suben y bajan en conjunto, que despliega las lonas del suelo permitiendo adaptarse a la altura de los troncos. El tractor avanza hacia el olivo con las lonas desplegadas al suelo y la pinza del vibrador abierta, una vez el tronco está en el hueco central, éstas se despliegan por su derecha e izquierda llegando a bajar toda la copa del olivo con la finalidad de sacudir y que los frutos se desprendan del olivo, cayendo hacia las lonas. (Sierra, 2006). (véase gráfico 14).

Gráfico 14. Proceso de recolección mecanizada usando mantor y vibrador en un tractor



Elaborado por: Sierra
Fuente: (Sierra, 2006)

Cosechadora Cabalgante tipo Vendimiadora

Emplean un sistema de derribo basado en sacudidores de copa, proporcionando una recolección mecanizada integral del olivar permitiendo una reducción de coste. (Ribes, Roldán & Garcia, 2012) (véase gráfico 15).

Gráfico 15. Recolección de olivares usando cosechadora tipo Vendimiadora



Elaborado por: Ribes, Roldán & García
Fuente: (Ribes, Roldán & García, 2012)

Mantillas

El uso de espuestas para trasegar las aceitunas desde el pie de los árboles hasta el remolque se ha quedado obsoleto, ahora se usa las mantillas, debido a que administran mayor cantidad de aceitunas y su trasiego es mecanizado, permitiendo dejar un reguero en los olivares en las zonas respectivas. (Sierra, 2006)

CONSERVACIÓN

Es una fase muy delicada debido a que un mínimo descuido puede acelerar la fermentación dando como resultado cambios en el sabor y olor. La conservación de las aceitunas para el consumo es posible mediante los siguientes métodos según Olive (2015).

- Características propias de la elaboración.
- Atmósfera protectora.
- Vacío.
- Adición de conservas.
- Refrigeración.
- Pasteurización.
- Esterilización.

FORMA DE PRESENTACIÓN

De acuerdo con FAO (2013) las aceitunas se presentan de las siguientes formas:

Aceitunas enteras

Aceitunas con o sin pedúnculo que conservan su forma original y no están deshuesadas.

Aceitunas deshuesadas

Aceitunas a las que se ha sacado el hueso y que conservan prácticamente su forma original.

Aceitunas rellenas

Aceitunas deshuesadas, rellenas con uno o más productos adecuados (pimiento, cebolla, almendras, apio, anchoa, aceituna, cáscara de naranja o limón, avellana, alcaparra, etc.) o sus pastas comestibles.

Aceitunas para ensalada

Aceitunas enteras rotas o rotas y deshuesadas, con o sin alcaparras, con material de relleno, cuando predominan en comparación con el conjunto del producto comercializado en esta forma.

Alcaparrado o mezclas

Aceitunas enteras o deshuesadas, generalmente de pequeño tamaño, con alcaparras y con material de relleno o sin él, empacadas con otros productos comestibles en salmuera, tales como piezas de: cebolla, zanahoria, apio, pimiento y otros ingredientes comestibles.

DEFECTOS QUE AFECTAN LA CALIDAD DE LAS ACEITUNAS

Las aceitunas de mesa deben de cumplir unos parámetros mínimos para su comercialización como estar sanas, limpias, el olor y sabor ha de ser normal, sin defectos que comprometan su conservación, sin gérmenes patógenos o sus toxinas, etc. Pero además de estos parámetros, hay otra serie de defectos que pueden presentar, que implican que sean de mayor o menor calidad. (Segovia, Martínez & López, 2016).

Segovia, et al (2016) menciona los defectos a considerar son los siguientes:

- Defectos de textura.
- Manchas y daños superficiales.
- Manchas que afectan a la pulpa.
- Frutos rotos.
- Vacías o con relleno incompleto (en aceitunas rellenas).
- Materias extrañas inocuas.

NORMATIVAS DE CALIDAD DE LAS ACEITUNAS

Los procesos de limpieza y de lavado son importantes para eliminar la suciedad y preparar la fruta para la extracción de aceite, estos procesos consisten en un flujo de aire que elimina los cuerpos extraños más ligeros que las aceitunas principalmente hojas y pequeñas ramas; posteriormente se basan en un proceso de lavado donde los desechos, más pesados, son arrastrados (piedras, pedazos de tierra y barro). (Aguilera et al., 2015).

Se recogen en las Normas del Consejo Oleícola Internacional (COI), CODEX Alimentarius y las Españolas, tanto internas (Reglamentación Técnico-Sanitaria) como para el comercio exterior (Normas de Calidad para la Exportación). Algunas de ellas pretenden asegurar unas características mínimas en el producto terminado para su respectivo consumo. (Fernández & Barranco, 1999)

Cuadro 4. Factores esenciales de calidad (COI, CODEX y Españolas) – Características que debe poseer el producto final

Las aceitunas deben estar:

- Sanas.
- Limpias.
- Exentas de sabor y olor anormales.
- Con la madurez adecuada.
- Exenta de defectos que pueden afectar a su consumo o conservación.
- Sin síntomas de alteración en curso o fermentación anormal.
- Calibradas (Sólo para enteras, deshuesadas, rellenas y mitades).
- De una sola variedad en el mismo envase.
- De color uniforme, excepto aliñadas y de color cambiante.
- Exenta de gérmenes patógenos o de sus toxinas.

Elaborado por: Daniel Fajardo – Milton Toalombo

Fuente: Fernández & Barranco (1999)

De acuerdo con las normativas Codex Alimentarius decretadas por la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO, 2013) establece que las clasificaciones por calidades pueden ser:

Extra, Fantasía o A

Se considerarán comprendidas dentro de esta categoría las aceitunas de calidad superior que posean en grado máximo las características propias de su variedad y su preparación comercial. No obstante, siempre que ello no afecte al buen aspecto del conjunto ni a las características organolépticas de cada fruto, podrán presentar muy ligeros defectos de color, forma o firmeza de pulpa o epidermis. En esta categoría podrán clasificarse las variedades apropiadas de aceitunas enteras, partidas, seccionadas, deshuesadas o rellenas.

Primera o I, Selecta o B

En esta categoría se incluirán las aceitunas de buena calidad, con un grado de madurez adecuado y que presenten las características propias de su variedad y preparación comercial. Siempre que ello no afecte al buen aspecto del conjunto ni a las características organolépticas individuales de cada fruto, podrán presentar ligeros defectos de color, forma, epidermis o firmeza de pulpa. Podrán clasificarse dentro de esta categoría todos los tipos, preparaciones y presentaciones de aceitunas de mesa, salvo las troceadas o las rotas.

Segunda o II, Estándar o C

Comprenderá las aceitunas de mesa que, no pudiendo clasificarse en las dos categorías anteriores, respondan a las condiciones generales definidas para las aceitunas de mesa en esta sección.

La agricultura de precisión consiste en aplicar diferentes tecnologías para el manejo de todos los aspectos agrícolas teniendo como propósito principal mejorar la calidad del cultivo sin afectar la medio ambiente, por lo que se debe tener en cuenta la necesidad real del cultivo y de esta manera aplicarlo solamente donde resulte necesario y se obtenga un

beneficio económico (García, Martínez & García ,2018). Por ello la agricultura de precisión resulta como una de la mejores herramientas utilizadas en los cultivos para incrementar la precisión de la producción y automatizar procesos con la creciente demanda generada por los consumidores.

VISIÓN ARTIFICIAL

Es la ciencia enfocada en procesar imágenes digitales o videos para su comprensión y su posterior análisis, la visión artificial permite también la automatización de tareas de inspección, recalcando que la inspección se realiza sin contacto físico y se puede personalizar de acuerdo con las necesidades del investigador, como tomar en cuenta las diversas piezas que puede captar la imagen reduciendo tiempos en la mayoría de estos procesos. (Alvear, Rosero, Peluffo & Pijal, 2016). Un software muy usado para temas de visión artificial es OpenCV que permite visualizar datos y extraer información de imágenes y videos, está disponible en Linux, Mac y Windows.

PROCESAMIENTO DIGITAL DE IMÁGENES

DEFINICIÓN

El procesamiento digital de imágenes tiene bases en matemáticas y probabilidad, pero resulta fundamental el análisis humano al momento de escoger entre las diferentes técnicas existentes, debido a esto comunmente la elección se la realiza mediante juicios visuales. (Villet, 2005).

FORMATO DE IMAGEN

Actualmente existe una gran variedad formatos de imagen que son representados a través de mapas de bits, todos ellos creados durante las últimas décadas, algunos de estos fueron creados por empresas que desarrollaron programas con la finalidad de trabajar con gráficos como es el caso de los formatos GIF o PCX, otros son resultado de un intento de

normalizar los formatos de imagen como son JPEG, PNG entre otros. En el cuadro 5 Villet (2005) considera los siguientes formatos:

Cuadro 5. Formatos de Imagen	
Formato	Modos de color
BMP (.bmp)	-Modos RGB -Color Indexado -Escala de Grises -Mapa de Bits
GIF (.gif)	-GIF89a: color indexado RGB -Compuserve GIF: mapa de bits, escala de grises, color Indexado -RGB a un canal
PICT (.pct; .pic)	-Color Indexado -Escala de Grises -Mapa de bits sin canales alfa
JPEG (jpg; .jpeg)	-Escala de Grises -RBG -CMYK
PNG (.png)	-Mapa de Bits -Escala de Grises -Color Indexado -RGB
PHOTOSHOP (.psd)	-Todos los modos de color admitidos -Canal alfa y tintas planas -Guías, trazados -Capas de ajuste, de texto, efectos capa -Escala de Grises
TARGA (.tga; .vda; .icb; .vst)	-Color Indexado -RBG 16 y 24 bits (sin canales) -RBG de 32 bits (un canal)

Elaborado por: Daniel Fajardo, Milton Toalombo.

Fuente: (Villet, 2005).

El PDI maneja 4 factores que son utilizados durante cada una de las fases, estos factores son:

- Imágenes RGB.
- Imágenes Indexadas.
- Imágenes en Escala de Grises.
- Imágenes Binaria.

ALGORITMO DE PROCESAMIENTO DIGITAL DE IMÁGENES

Algoritmo sobre operaciones estadísticas

HISTOGRAMA

El histograma es similar a un gráfico de barras en la que se muestran variables que pueden ser continuas o discretas, es necesario agruparlo en clases debido al gran número de datos utilizados. (Avilés, Salavarría & Ricardo, 2019).

Mediante la función `cv2.calcHist()` que proporciona OpenCV, que permite realizar el cálculo de histograma de la siguiente manera:

```
cv2.calcHist(images, channels, mask, histSize, ranges[,  
             hist[, accumulate]])
```

Donde los siguientes parámetros son:

- `images`: imagen de entrada, puede ser a escala de grises o colores.
- `channels`: índice de canal para el cual deseamos calcular el histograma, en una imagen a escala de grises [0], si la imagen es a colores podemos indicar [0], [1], [2] para los canales B, G, R respectivamente.
- `mask`: máscara que define la región sobre la que se desea calcular el histograma, es opcional.
- `histSize`: intensidad máxima [256].
- `ranges`: rango de valores [0, 256].

Algoritmo sobre operaciones espaciales

ESPACIOS DE COLOR

A través de modelos de referencia los espacios de color describen la manera en que se organizan y representan los colores (Cardozo, Méndez, Legal & Vázquez, 2018). Existen algunos espacios de color entre los que se puede mencionar:

ESPACIO RGB

Este submodelo de color contiene los colores que un dispositivo puede generar a partir de la combinación numérica de los colores rojo, verde y azul.

ESPACIO HSI

Es muy utilizado en el procesamiento digital de imágenes, es caracterizado por utilizar el tono saturación e intensidad de estas y suele ser representado como un cono.

El algoritmo que permite realizar la conversión de espacio del color de una imagen es de la siguiente manera:

```
colormath.color_conversions.convert_color(color, target_cs, through_rgb_type=<class'colormath.color_objects.sRGBColor'>, target_illuminant=None, *args, **kwargs)
```

Donde:

- color: una instancia de color para convertir.
- target_cs: la clase de color para convertir.
- through_rgb_type (BaseRGBColor): si durante la conversión entre los espacios de color originales y de destino debe pasar por RGB, esto determina qué tipo de RGB usar.
- target_illuminant (None o str): si durante la conversión de RGB a un espacio de color reflectante desea terminar explícitamente con un cierto iluminante. De lo contrario, se utilizará el iluminante nativo del espacio RGB.

Algoritmo sobre dominio de frecuencias

Transformada de Fourier

Para Gaviria & Ruíz (2016) consideran a la transformada de Fourier como: “Un procedimiento matemático que descompone una función en las frecuencias que le forman de la misma manera que un prisma descompone la luz en los diferentes colores y longitudes de onda.” (pág.

12). La transformada de Fourier puede ser usada para caracterizar la imagen digital a través de etapas fundamentales:

- Convertir la imagen a una escala de grises, evitando así el brillo de imagen y reduciendo el ruido llegando a obtener buenos resultados.
- Ubicar los picos más altos en función del orden central, etiquetándolos de forma ascendente y uno en cada vecindad.
- Usando estadística como es la media, mediana, desviaciones, para obtener las características más importantes de una imagen.
- Clasificar y comparar en base a estándares previamente establecidos.

Cuadro 6. Funciones de DFT de la librería numpy*

TRANSFORMADA DE FOURIER DISCRETA (DFT) (numpy.fft)	
Estándar de la Transformada Rápida de Fourier (FFT)	
fft (a[, n, axis, norm])	Permite calcular una dimensión de DFT
ifft (a[, n, axis, norm])	Permite calcular una dimensión inversa de DFT
fft2 (a[, s, axes, norm])	Permite calcular DFT en 2 dimensiones
ifft2 (a[, s, axes, norm])	Permite calcular DFT inversa en 2 dimensiones
fftn (a[, s, axes, norm])	Permite calcular N dimensiones de DFT
ifftn (a[, s, axes, norm])	Permite calcular la inversa de N dimensiones de DFT
Real de la Transformada Rápida de Fourier (FFT)	
rfft (a[, n, axis, norm])	Permite calcular una dimensión de DFT para una entrada de dato real.
irfft (a[, n, axis, norm])	Permite calcular la inversa de N puntos de DFT para una entrada de dato real.
rfft2 (a[, s, axes, norm])	Permite calcular FFT en 2 dimensiones de una matriz de datos reales.
irfft2 (a[, s, axes, norm])	Permite calcular la inversa de FFT en 2 dimensiones de una matriz de datos reales.
rfftn (a[, s, axes, norm])	Permite calcular N dimensiones de DFT para una entrada de datos reales.
irfftn (a[, s, axes, norm])	Permite calcular la inversa de N dimensiones de FFT para una entrada de datos reales.
Hermitiana de la Transformada Rápida de Fourier (FFT)	
hfft (a[, n, axis, norm])	Permite calcular FFT de una señal que tiene simetría hermitiana, es decir, de un espectro real.
ihfft (a[, n, axis, norm])	Permite calcular la inversa de FFT de una señal que tiene simetría hermitiana.
Rutinas de Ayudas	
fftfreq (n[, d])	Permite devolver las frecuencias de las muestras de DFT.
rfftfreq (n[, d])	Permite devolver las frecuencias de las muestras de DFT (Para utilizarlo con rfft, irfft).
fftshift (x[, axes])	Permite cambiar el componente de frecuencia cero al centro del espectro.
ifftshift (x[, axes])	La inversa de fftshift .

Elaborado por: Daniel Fajardo, Milton Toalombo.

Fuente: Página Oficial SciPy.org

** En el cuadro 6 se presenta los comandos de la Transformada De Fourier Discreta que pertenecen a la librería numpy que fueron obtenidas a través de la página oficial de SciPy.org*

Donde los parámetros son:

- a: array_like: Matriz de entrada, puede ser compleja.

- **s**: secuencia de entradas, opcional. Forma (longitud de cada eje) de la salida (**s** [0] se refiere al eje 0, **s** [1] al eje 1, etc.). Esto corresponde a **n** para **ifft** (**x**, **n**). A lo largo de cada eje, si la forma dada es más pequeña que la de la entrada, la entrada se recorta. Si es más grande, la entrada se rellena con ceros. si no se da **s**, se usa la forma de la entrada a lo largo de los ejes especificados por los ejes. Vea las notas para el problema sobre el relleno de **ifft** zero.
- **axes**: secuencia de ints, opcional. Ejes sobre los cuales calcular la FFT. Si no se da, se usan los dos últimos ejes. Un índice repetido en los ejes significa que la transformación sobre ese eje se realiza varias veces.
- **norm**: {Ninguno, *orto*}, opcional. Por Defecto es ninguno.

En el gráfico 16 muestre el esquema de las fases en las que se descompone el procesamiento digital de imágenes.

Gráfico 16. Esquema del Procesamiento Digital de Imágenes



Elaborado por: Paláez, Vaccaro Guijarro.
Fuente: (Paláez, Vaccaro & Guijarro, 2015)

CAPTURA DE IMAGEN

Enfoque

Es un ajuste que debe realizarse en las cámaras antes de tomar una fotografía con la finalidad de ver los objetos con nitidez, dependiendo de

la definición del detalle en la imagen y la resolución de la cámara. Existen dos modos de enfoque: el automático que usa como base la diferencia de contraste de los objetos, este modo no posee un buen rendimiento en condiciones con poca iluminación y el modo manual donde el fotógrafo debe mover el telémetro adecuadamente hasta encontrarse el enfoque buscado. (Astudillo, 2017).

Profundidad

Hace referencia al espacio por delante y detrás del plano enfocado, lo componen los puntos más cercanos y lejanos, depende de tres factores: longitud focal del lente, grado de apertura del diafragma y distancia cámara sujeto. (Astudillo, 2017).

Resolución

Es la calidad de la imagen que depende de la cantidad de píxeles por pulgada que posee el dispositivo. (Astudillo, 2017).

PRE-PROCESAMIENTO

Esta etapa consiste en procesos como la redimensión, conversión del espacio de color, filtrado, delimitación de regiones, suavizado de imágenes, entre otras.

Píxel

Es el elemento más pequeño que compone a la imagen, combinando los colores básicos rojo, verde y azul el píxel obtiene su tono de color, según Querejeta (2015) tres características son que le permiten distinguir al píxel con facilidad: la forma cuadrada, posición relativa al resto de píxeles en un mapa de bits, profundidad de color expresada en bits.

SEGMENTACIÓN

La segmentación es la etapa más compleja del PDI, consiste en subdividir una imagen en varias partes aplicando métodos que se lo clasifica por umbralización, bordes y regiones, con la finalidad de resaltar las partes de

interés con el resto de la imagen. Al aplicar el método seleccionado va a depender de la naturaleza propia de la imagen al momento de procesar y de los requerimientos que necesita el autor.

Segmentación por umbralización

“La umbralización es uno de los métodos más importantes de la segmentación de imágenes. Se define el umbral como una función que convierte una imagen con diferentes tonalidades en una imagen en blanco y negro.” (Palomino, Concha, 2009, pág. 14).

Segmentación por bordes

Transformada de Hough

“Este método consiste en calcular el gradiente de una imagen, luego crear un campo de acumulación en función de los parámetros de la función que se busca, y posteriormente los máximos en el espacio de acumulación permitiendo detectar bordes de objetos.” (Palomino, Concha, 2009, pág. 14).

$$\rho = x \cos\theta + y \sin\theta \quad (1)$$

Donde ρ corresponde a la longitud normal desde el origen hasta la línea y θ corresponde al ángulo respecto al eje x.

Algoritmo de Canny

“Este algoritmo emplea máscaras de convolución y basado en la primera derivada, las cuales representan aproximaciones en diferencias finitas. Los puntos de contorno son como zonas de píxeles en las que existe un cambio brusco de nivel de gris.” (Rebaza, 2007, pág. 1)

Para calcular los bordes con el método Canny se usa el siguiente algoritmo:

```
canny = cv2.Canny(imagen, umbral_minimo, umbral_maximo)
```

Donde:

- canny: es la imagen resultante. Aparecerán los bordes detectados tras el proceso.
- imagen: es la imagen original.

- umbral_minimo: es el umbral mínimo en la umbralización por histéresis
- umbral_maximo: es el umbral máximo en la umbralización por histéresis

Segmentación por regiones

Crecimiento por regiones

“Es una técnica muy utilizada que consiste en el crecimiento de regiones agrupando pixeles adyacentes que presentan características o propiedades similares añadiendo a los puntos semilla, los puntos vecinos que tengan propiedades similares, como intensidad, textura, color, etc.” (Palomino, Concha, 2009, pág. 14).

División y fusión de regiones

“Consiste en dividir una imagen en un conjunto de regiones arbitrarias disyuntas, posteriormente son fusionadas si tienen propiedades similares como nivel de gris similares, o son divididos si no comparten las mismas propiedades, como variaciones considerables de niveles de gris.” (Palomino, Concha, 2009, pág. 14).

Etiquetado por componentes conexos

El etiquetado por componentes permite analizar los objetos en imágenes binarias, asignando una única etiqueta a cada componente. Este etiquetado está basado en el vecindario de pixeles utilizando dos pixeles p y q conectados entre sí a través de un camino de pixeles, existen 2 tipos de conectividad: Conectividad 4 que es cuando se tiene 4 vecinos y la Conectividad 8 cuando tiene 8 vecinos. (Maldonado, Alza & García).

EXTRACCIÓN DE CARACTERÍSTICAS

Una vez segmentado las imágenes, en esta etapa se lleva a cabo diferentes procesos como es el etiquetado, paleta de colores, para la obtención de sus características.

Etiquetado

“El etiquetado permite diferenciar los objetos entre sí. Hecho esto, es posible obtener fácilmente todo un abanico de propiedades de cada uno de ellos, tales como: perímetro, área, excentricidad, solidez, orientación, entre muchas otras” (Salina, Sánchez, Miranda, Rodríguez & Rosales, 2015, pág. 5).

Paleta de colores

Indica los colores más convenientes a usar en una imagen, Querejeta (2015) menciona que uno de los inconvenientes que se presentaban antiguamente con la paleta de colores era que solo se podía trabajar con 256 colores lo que limitaba a la calidad de las imágenes, esto fue solucionado utilizando colores que predominen en la imagen, por ejemplo, en una imagen donde predomine el verde, utilizara más ese color que rojo o azul.

CLASIFICACIÓN

Técnicas de Machine Learning

Machine learning o conocido como aprendizaje automático en español es un término que denota la forma, en la cual una máquina aumenta su conocimiento y mejora sus capacidades de actuación (performance) en un entorno. De esta manera, el proceso de aprendizaje puede ser visto como un generador de cambios en el sistema que aprende y que pueden ser revocados o ampliados. (Moreno. et al, 1999).

Aprendizaje supervisado

El aprendizaje supervisado se refiere al subconjunto de Machine Learning donde se generan modelos para predecir el resultado de salida en base a ejemplos históricos de esa variable de salida. Los modelos se construyen a partir de los algoritmos de Machine Learning y características o atributos de los datos de entrenamiento para que podamos predecir el valor utilizando otros valores obtenidos a partir de datos de entrada.

Aprendizaje no supervisado

Detección automática de patrones inmersos en los datos y relaciones no conocidas entre ellos para transformarlas en una estructura comprensible

Factores a tomar en cuenta al momento de escoger un clasificador

- Exactitud.
- Rapidez.
- Claridad.
- Tiempo de aprendizaje.

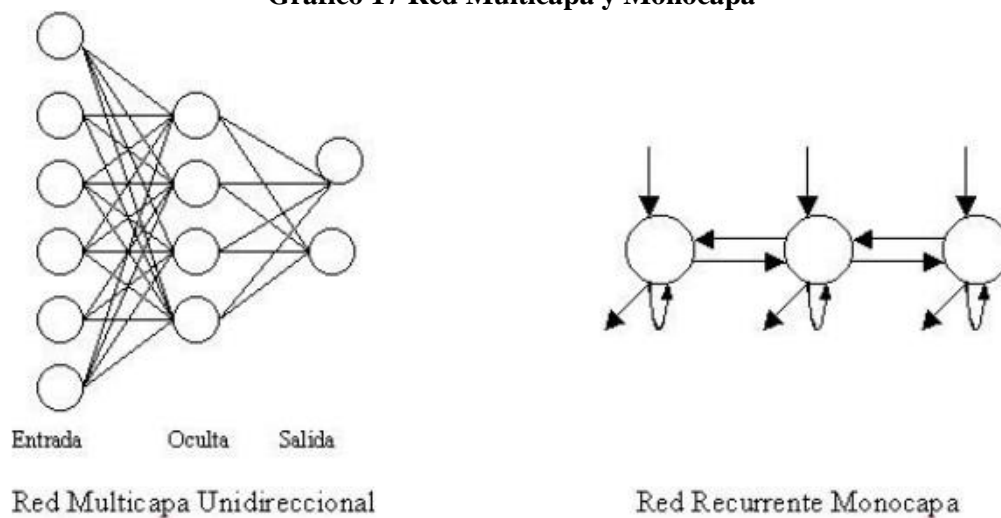
RED NEURONAL ARTIFICIAL

Inspirado en el funcionamiento del sistema nervioso de los animales, las redes neuronales artificiales son un sistema de interconexiones de neuronas que se apoyan entre sí para producir una salida. El modelo de la red neuronal fue desarrollado por Rumelhar y McClelland en 1986, siendo este modelo básico y esencial para construir redes neuronales artificiales.

La arquitectura es la estructura de conexiones de la red, las conexiones sinápticas son unidireccionales por lo que transmiten información en un solo sentido. Las neuronas suelen agruparse en capas, las redes están compuestas de 3 capas en cuya entrada no se modifican los datos de información que reciben y estos son duplicados en las salidas, para luego ser evaluados por las capas ocultas y de salida proporcionando así la respuesta a la red.

Cuando la estructura es completamente conectada, el valor de cada capa de entrada se duplica para luego ser enviado las capas ocultas, el proceso se repite en cada capa oculta produciendo los valores finales en la salida. En el gráfico 17, se muestra cómo funciona la red multicapa y monocapa.

Gráfico 17 Red Multicapa y Monocapa



Elaborado por: Pérez 2018
Fuente: Pérez. (2018)

En el aprendizaje los pesos generados se refinan mediante el algoritmo de backpropagation, cuya finalidad es minimizar el error en los valores de salida comparando los resultados obtenidos con los resultados deseados cuando el aprendizaje sea supervisado y se cuente con las respectivas etiquetas, esos valores se pueden extraer y ser almacenados para no tener que realizar el proceso de entrenamiento y hallarlos nuevamente.

RED NEURONAL CONVOLUCIONAL

La red neuronal convolucional es similar a una red neuronal clásica, su diferencia radica al momento de incrementar la dimensión y pesos, estos pasan de ser números a una matriz de n dimensiones cuyo nombre es conocido como filtros que serán aplicadas en cada convolución con la finalidad de obtener la salida. (Rodríguez, 2017), por lo que es necesario contar con una gran potencia computacional y bastante tiempo para la fase de entrenamiento.

Filtro

Un filtro es una cuadrícula formada por números discretos. Los números en cada celda de la cuadrícula son los pesos del kernel y son aprendidos durante el entrenamiento de la Red Neuronal Convolucional (Díaz, 2018).

Kernel

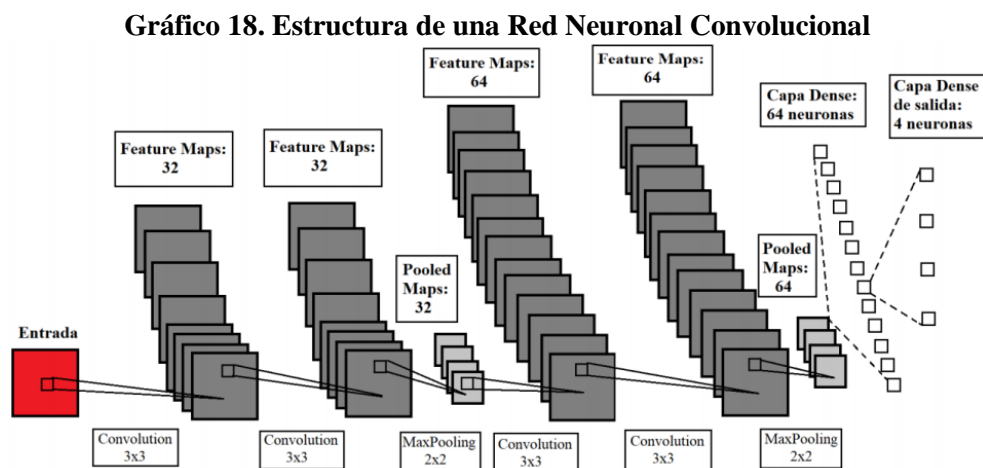
Se le denomina al conjunto de filtros que realizan la función de convolución con una entrada dada para generar una salida que se trata de un mapa de características con el objetivo de ir reduciendo las dimensiones de la entrada, la cual suele ser en un array multidimensional (Díaz, 2018).

Stride paso

Es el intervalo de desplazamiento para la aplicación de filtros sobre las imágenes de entrada. (Moreno, Avilés & Ovalle, 2018)

Estructura

En el gráfico 18, se muestra como está estructurada una red neuronal convolucional.



Elaborado por: Delfín, González, Rojas 2018

Fuente: Delfín, González, Rojas. (2018)

Una red neuronal convolucional está conformada de lo siguiente:

- Entrada que consta de los píxeles de las imágenes registradas, teniendo en consideración el alto y ancho, además si la imagen es en escala de grises se debe multiplicar por 1 canal o por 3 canales si la imagen es a color (Red, Green, Blue).
- Capa Oculta se lleva a cabo operaciones como el pooling para reducir la imagen, otras convoluciones para pasar filtros y detectar ciertos

patrones, con el objetivo de reducir las neuronas que no son importantes al momento de reconocer patrones.

- Salida permite visualizar las variables ordenadas de acuerdo con los datos de entradas, estos datos serán las neuronas.

Tipos de Capas

Convolutacional

Al alimentar una imagen como entrada, esta se convertirá en una matriz de valores de píxeles, cuyos valores pueden variar de 0 a 255 y las dimensiones serán la altura de la imagen, anchura de la imagen y el número de canales. (Ravichandiran, 2019).

Pooling

Esta capa es utilizada para reducir las dimensiones espaciales manteniendo solo los detalles necesarios para reducir la carga computacional, existen diferentes tipos de pooling u operaciones de agrupación, entre los que destaca Max Pooling, el cual solo toma valores máximos del mapa de características dentro de la ventana. (Ravichandiran, 2019).

Sampling

Upsampling

Capa contraria al pooling la cual transforma la matriz de información a una más grande. (Rodríguez, 2017).

Downsampling

Este proceso reduce la frecuencia de muestreo de una señal, generalmente es usado para reducir la velocidad de datos. (Rodríguez, 2017).

Flatternig

Consiste en convertir a la matriz de salida en un vector unidimensional.

Fully Connected

Es la penúltima capa de una red convolucional, esta capa funciona como un filtro transformado en sistema en una red neuronal unidimensional en la que cada neurona estará conectada a cada vóxel de la capa anterior. (Rodríguez, 2017).

Capa Softmax

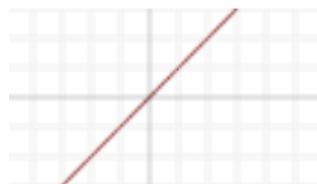
Es la última capa en una red convolucional, transforma los resultados de la capa fully connected en resultados probabilísticos con la finalidad de realizar la clasificación de cada clase con probabilidad. (Rodríguez, 2017).

Función de Activación

Según Antona Cortés (2017) la función de activación se combina con el estado actual de la neurona para conseguir el estado futuro de esta (activada/desactivada), las funciones de activación más comunes son:

Identidad: Es una función de activación muy simple que siempre devuelve como salida su valor de entrada. Su rango es $(-\infty, \infty)$, y es una función monótona. (véase gráfico 19).

Gráfico 19. Función Identidad



$$f(x) = x \quad (2)$$

Elaborado por: Carlos Antona Cortés

Fuente: Antona Cortés, C. (2017)

Escalón binario: Es la función más usada por redes neuronales binarias, ya que no es lineal y es bastante sencilla. El Perceptrón y Hopfield son algunos ejemplos de redes que utilizan esta función. Cuenta con un rango $(0,1)$, y es monótona. (véase gráfico 20).

Gráfico 20. Función Escalón binario



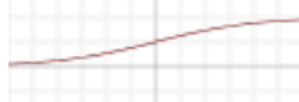
$$f(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases} \quad (3)$$

Elaborado por: Carlos Antona Cortés

Fuente: Antona Cortés, C. (2017)

Logística / Sigmoide / Softstep: Es una de las funciones más utilizadas para modelar redes neuronales. Su rango es continuo en los valores (0,1), es monótona e infinitamente diferenciable. (véase gráfico 21).

Gráfico 21. Función Logística



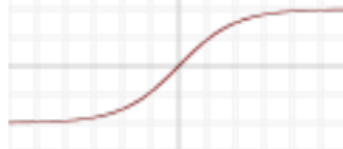
$$f(x) = \frac{1}{1+e^{-x}} \quad (4)$$

Elaborado por: Carlos Antona Cortés

Fuente: Antona Cortés, C. (2017)

Tangente hiperbólica: Esta función es utilizada por redes con salidas continuas. Un ejemplo sería el Perceptrón multicapa con retropropagación, ya que su algoritmo de aprendizaje necesita una función derivable. Cumple con un rango (-1,1), es monótona, y se aproxima a la función identidad en su origen. (véase gráfico 22).

Gráfico 22. Función Tangencial



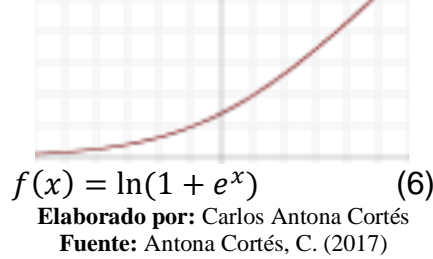
$$f(x) = \tanh(x) = \frac{e^x + e^{-x}}{e^x - e^{-x}} \quad (5)$$

Elaborado por: Carlos Antona Cortés

Fuente: Antona Cortés, C. (2017)

Softplus: Aproximación suavizada de la función de activación rectificadora. (véase gráfico 23).

Gráfico 23. Función Softplus



Función Perdida

Para Antona (2017) es: un parámetro usado para compilar el modelo, en el cuadro 7 keras proporciona las siguientes:

Cuadro 7. Algoritmo de la Función Perdida	
ALGORITMO	DEFINICIÓN
<i>mean_squared_error(y_true, y_pred)</i>	La cual calcula el error cuadrático medio.
<i>mean_absolute_error / mae</i>	Calcula el error medio absoluto.
<i>binary_crossentropy:</i>	Calcula la entropía cruzada en problemas de clasificación binaria.

Elaborado por: Daniel Fajardo, Milton Toalombo.
Fuente: Antona (2017)

Función Relu

La unidad lineal rectificadora (ReLu) es una función de activación que a menudo es usada en redes neuronales, ayuda a resolver inconvenientes generados por la fuga de gradiente y otras activaciones logísticas (), su fórmula es:

$$a_{i,j,k} = \max(z_{i,j,k}, 0) \quad (7)$$

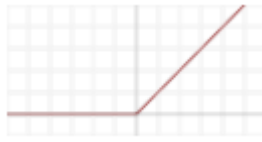
Donde:

$z_{i,j,k}$ = entrada de la función de activación (i,j) en el canal k.

max = permite computar rápido funciones de activación sigmoide o tanh.

En el gráfico 19, se muestra la función ReLU. (véase gráfico 24).

Gráfico 24. Función rectificadora



Elaborado por: Carlos Antona Cortés

Fuente: Antona Cortés, C. (2017)

Como ventaja, esta función acelera la convergencia del algoritmo de gradiente descendente comparando con otras funciones, ReLU puede ser implementada con operaciones simples aplicando un umbral en cero. Como desventajas esta función esta frágil durante el entrenamiento llegando a morir en ciertas ocasiones.

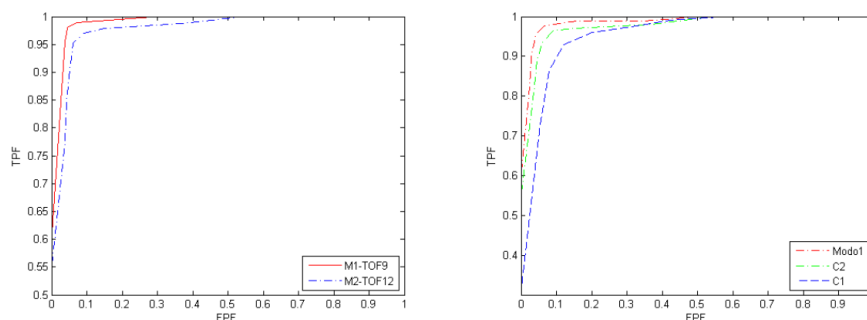
Función Softmax

Esta función permite que la salida de la red neuronal se normalice representado inmediatamente una distribución de probabilidad discreta con valores entre 0 y 1 con respecto a cada una de las clases. (Bonaccorso, 2019).

Curvas ROC

Muestran la tasa de aciertos de una clase contra la tasa de errores de otra, mientras sea más pronunciada la curva, mayor será el acierto al clasificar ambas clases. Su valor se asigna mediante el cálculo de áreas que existen bajo la curva que puede tomar valores desde 0 a 1 indicando la probabilidad de que la clase sea la correcta, si el área bajo la curva es superior a 0.5 tiene una capacidad discriminatoria al azar. (Pazos, 2015). En el gráfico 25 se muestra un ejemplo de la gráfica de la curva ROC.

Gráfico 25. Ejemplo de Curvas ROC



Elaborado por: Giacomantone, Violin, Lorenti y & De Giusti

Fuente: (Giacomantone, Violin, Lorenti y & De Giusti, 2016)

FUNDAMENTACIÓN LEGAL

Constitución de la República del Ecuador 2008

Régimen del buen vivir

Sección primera – Educación

Capítulo primero – inclusión y equidad

Art. 350.- El sistema de educación superior tiene como finalidad la formación académica y profesional con visión científica y humanista; la investigación científica y tecnológica; la innovación, promoción, desarrollo y difusión de los saberes y las culturas; la construcción de soluciones para los problemas del país, en relación con los objetivos del régimen de desarrollo.

Art. 355.- El Estado reconocerá a las universidades y escuelas politécnicas autonomía académica, administrativa, financiera y orgánica, acorde con los objetivos del régimen de desarrollo y los principios establecidos en la Constitución. Se reconoce a las universidades y escuelas politécnicas el derecho a la autonomía, ejercida y comprendida de manera solidaria y responsable. Dicha autonomía garantiza el ejercicio de la libertad académica y el derecho a la búsqueda de la verdad, sin restricciones; el gobierno y gestión de sí mismas, en consonancia con los principios de alternancia, transparencia y los derechos políticos; y la producción de ciencia, tecnología, cultura y arte.

Sus recintos son inviolables, no podrán ser allanados sino en los casos y términos en que pueda serlo el domicilio de una persona. La garantía del orden interno será competencia y responsabilidad de sus autoridades. Cuando se necesite el resguardo de la fuerza pública, la máxima autoridad de la entidad solicitará la asistencia pertinente. La autonomía no exime a las instituciones del sistema de ser fiscalizadas, de la responsabilidad social, rendición de cuentas y participación en la planificación nacional. La Función Ejecutiva no podrá privar de sus rentas o asignaciones presupuestarias, o retardar las transferencias a ninguna

institución del sistema, ni clausurarlas o reorganizarlas de forma total o parcial.

Sección Octava Ciencia, Tecnología, Innovación y Saberes Ancestrales

Art. 385.- El sistema nacional de ciencia, tecnología, innovación y saberes ancestrales, en el marco del respeto al ambiente, la naturaleza, la vida, las culturas y la soberanía, tendrá como finalidad:

- 1.- Generar, adaptar y difundir conocimientos científicos y tecnológicos.
- 2.- Recuperar, fortalecer y potenciar los saberes ancestrales.
- 3.- Desarrollar tecnologías e innovaciones que impulsen la producción nacional, eleven la eficiencia y productividad, mejoren la calidad de vida y contribuyan a la realización del buen vivir.

Art. 386.- El sistema comprenderá programas, políticas, recursos, acciones, e incorporará a instituciones del Estado, universidades y escuelas politécnicas, institutos de investigación públicos y particulares, empresas públicas y privadas, organismos no gubernamentales y personas naturales o jurídicas, en tanto realizan actividades de investigación, desarrollo tecnológico, innovación y aquellas ligadas a los saberes ancestrales.

El Estado, a través del organismo competente, coordinará el sistema, establecerá los objetivos y políticas, de conformidad con el Plan Nacional de Desarrollo, con la participación de los actores que lo conforman.

Art. 387.- Será responsabilidad del Estado:

- 1.- Facilitar e impulsar la incorporación a la sociedad del conocimiento para alcanzar los objetivos del régimen de desarrollo.

2.- Promover la generación y producción de conocimiento, fomentar la investigación científica y tecnológica, y potenciar los saberes ancestrales, para así contribuir a la realización del buen vivir, al sumak kawsay.

3.- Asegurar la difusión y el acceso a los conocimientos científicos y tecnológicos, el usufructo de sus descubrimientos y hallazgos en el marco de lo establecido en la Constitución y la Ley.

4.- Garantizar la libertad de creación e investigación en el marco del respeto a la ética, la naturaleza, el ambiente, y el rescate de los conocimientos ancestrales.

5.- Reconocer la condición de investigador de acuerdo con la Ley.

Art. 388.- El Estado destinará los recursos necesarios para la investigación científica, el desarrollo tecnológico, la innovación, la formación científica, la recuperación y desarrollo de saberes ancestrales y la difusión del conocimiento. Un porcentaje de estos recursos se destinará a financiar proyectos mediante fondos concursables. Las organizaciones que reciban fondos públicos estarán sujetas a la rendición de cuentas y al control estatal respectivo.

Ley Orgánica de Educación Superior (LOES)

Título a ámbito objeto fines y principios del sistema de educación superior

Capítulo II

Art. 8.- Serán Fines de la Educación Superior.- La educación superior tendrá los siguientes fines:

- a) Aportar al desarrollo del pensamiento universal, al despliegue de la producción científica y a la promoción de las transferencias e innovaciones tecnológicas;
- f) Fomentar y ejecutar programas de investigación de carácter científico, tecnológico y pedagógico que coadyuven al

mejoramiento y protección del ambiente y promuevan el desarrollo sustentable nacional;

Art. 13.- Funciones del Sistema de Educación Superior.- Son funciones del Sistema de Educación Superior: literales a, b, d f, ñ

- a) Garantizar el derecho a la educación superior mediante la docencia, la investigación y su vinculación con la sociedad, y asegurar crecientes niveles de calidad, excelencia académica y pertinencia;
- b) Promover la creación, desarrollo, transmisión y difusión de la ciencia, la técnica, la tecnología y la cultura;
- d) Fortalecer el ejercicio y desarrollo de la docencia y la investigación científica en todos los niveles y modalidades del sistema;
- f) Garantizar el respeto a la autonomía universitaria responsable;
- ñ) Brindar niveles óptimos de calidad en la formación y en la investigación.

Título II Autonomía responsable de las universidades y escuelas politécnicas

Capítulo II

Patrimonio y financiamiento de las instituciones de educación superior

Art. 20.- Del Patrimonio y Financiamiento de las instituciones del sistema de educación superior.- En ejercicio de la autonomía responsable, el patrimonio y financiamiento de las instituciones del sistema de educación superior estará constituido por:

- a) Los beneficios obtenidos por su participación en actividades productivas de bienes y servicios, siempre y cuando esa participación no persiga fines de lucro y que sea en beneficio en la institución;

- g) Los recursos provenientes de herencias, legados y donaciones a su favor;
- h) Los fondos autogenerados por cursos, seminarios extracurriculares, programas de posgrado, consultorías, prestación de servicios y similares, en el marco de lo establecido en esta Ley;

Art. 35.- Asignación de recursos para investigación, ciencia y tecnología e innovación.- Las instituciones del Sistema de Educación Superior podrán acceder adicional y preferentemente a los recursos públicos concursables de la pre asignación para investigación, ciencia, tecnología e innovación establecida en la Ley correspondiente.

Capítulo 12

Personal académico

Art. 148.- de la LOES agrega los profesores o profesoras e investigadores o investigadoras que hayan intervenido en una investigación tendrán derecho a participar, individual o colectivamente, de los beneficios que obtenga la institución del Sistema de Educación Superior por la explotación o cesión de derechos sobre las invenciones realizadas en el marco de lo establecido en esta Ley y la de Propiedad Intelectual. Igual derecho y obligaciones tendrán si participan en consultorías u otros servicios externos remunerados. Las modalidades y cuantía de la participación serán establecidas por cada institución del Sistema de Educación Superior en ejercicio de su autonomía responsable.

PLAN NACIONAL DEL BUEN VIVIR – ECUADOR

Estrategias para el 2013 – 2017

Matriz de políticas y lineamientos estratégicos Objetivos

Objetivo 4. Fortalecer las capacidades y potencialidades de la ciudadanía

4.4 Mejorar la calidad de la educación en todos sus niveles y modalidades, para la generación de conocimiento y la formación integral de personas creativas, solidarias, responsables, críticas, participativas y productivas, bajo los principios de igualdad, equidad social y territorialidad

4.4 i.) Asegurar en los programas educativos la inclusión de contenidos y actividades didácticas e informativas que motiven el interés por las ciencias, las tecnologías y la investigación, para la construcción de una sociedad socialista del conocimiento.

4.6 Promover la interacción recíproca entre la educación, el sector productivo y la investigación científica y tecnológica, para la transformación de la matriz productiva y la satisfacción de necesidades

4.6. f) Fortalecer y promocionar la formación técnica y tecnológica en áreas prioritarias y servicios esenciales para la transformación de la matriz productiva, considerando los beneficios del sistema dual de formación

4.6. j) Generar mecanismos de incentivo y acceso a financiamiento de programas y proyectos de investigación científica y desarrollo tecnológico, promoviendo su implementación con criterios de priorización para el desarrollo del país.

Objetivo 8. Consolidar el sistema económico social y solidario, de forma sostenible.

8.1. Invertir los recursos públicos para generar crecimiento económico sostenido y transformaciones estructurales

8.1. d) Articular la inversión del Estado Central con las empresas públicas, las entidades del sistema de seguridad social, las universidades y escuelas politécnicas, la banca pública y otros niveles de gobierno, en el marco de la planificación nacional.

Es importante mencionar estos artículos, debido a que en ellos se manifiestan los aspectos a tener en cuenta al momento de realizar una investigación de carácter tecnológico, la misma que esta sustentada en la LOES y el plan nacional del buen vivir, permitiendo promover el desarrollo sustentable nacional al transformar la matriz productiva, beneficiando con este proyecto a determinadas personas para que puedan identificar variedades de aceitunas, logrando así mejorar la calidad de la materia prima.

PREGUNTA CIENTÍFICA A CONTESTARSE

¿La elaboración de un clasificador de aceitunas de mesa por su variedad utilizando los requerimientos básicos que permiten caracterizarlas ofrecerá una mejor calidad de la materia prima para la preparación de platos?

VARIABLES DE LA INVESTIGACIÓN

Variable Independiente

Clasificador de aceitunas de mesa

Variable Dependiente

Variedad de aceitunas

DEFINICIONES CONCEPTUALES

Imagen

Una imagen puede considerarse como el conjunto de puntos de colores, es decir, una sucesión coherente de puntos que conforman una matriz de información para el uso digital. (Querejeta, 2015).

Imagen Digital

Una imagen digital está conformada por un número finito de elementos y cada uno tiene una localidad y un valor particular. A estos elementos se les llama puntos elementales de la imagen o píxeles, siendo este último el término comúnmente utilizado para denotar la unidad mínima de medida de una imagen digital. (Dobernack, 2013).

Modelo RGB

El modelo RGB es uno de los más utilizados por los sistemas informáticos para crear y reproducir los colores en monitores y pantallas. Está basado en la llamada síntesis aditiva, donde las intensidades de luz relativas al rojo, al verde y al azul son sumadas entre sí para conseguir los distintos colores, incluyendo el negro y el blanco. (Dobernack, 2013).

Píxel

Un píxel es la menor unidad homogénea en color que forma parte de una imagen digital. (Querejeta, 2015).

Aceituna

Es el fruto del olivo que es un árbol cultivado principalmente en los países de la cuenca del Mediterráneo, inicialmente es de color verde y cambia a morado o negro según el estado de madurez, posee un compuesto amargo que debe removerse en gran parte antes del consumo. (Olive, 2015).

Clasificador

Es un sistema que a través de cierta información ingresada por el humano reconoce de objetos, estos pueden ser voces, letras, caras, entre otros. Es una técnica útil que necesita principalmente del reconocimiento de patrones para clasificar objetos. (Flores, Lara, Gutiérrez, De los Cobos & Rincón, 2017).

Python

Es un lenguaje de programación interpretado que destaca por su legibilidad al momento de escribir código, al ser multiparadigma convive de forma nativa con aspectos funcionales, orientados a objetos, entre otros. Cuenta con una amplia biblioteca estándar además de herramientas que permiten resolver tareas útiles como es la lectura de ficheros en distintos formatos o desarrollar interfaces gráficas, entre otras. (Troyano, Cruz, González, Vallejo & Toro, 2018).

Librería

Una librería es una colección de funciones implementadas en un lenguaje de programación que permite su reutilización a través de una interfaz determinada. (Molina & Santamaria, 2017).

Procesamiento Digital de Imágenes

El procesamiento digital de imágenes es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información. (Vela, 2013).

Tensorflow

Desarrollado por Google, permite a investigadores y desarrolladores crear modelos de aprendizaje automático que pueden ser implementados en la mayoría de los dispositivos como ordenadores, laptops, smartphones, raspberry pi, entre otros. (Planche & Andrés, 2019).

Numpy

Es un paquete fundamental para realizar aprendizaje automático en python, entre sus herramientas posee varias clases y subclases que permiten representar matrices y otras funciones de álgebra lineal. (Molina & Santamaria, 2017).

Keras

Es una librería de deep learning escrita en Python capaz de ejecutarse sobre Tensorflow y Theano con la cual permite diseñar modelos propios de redes neuronales. (Cortés, 2017).

OpenCV

Es una biblioteca de visión artificial libre comúnmente utilizada en trabajos donde se requiere de la detección o reconocimiento de objetos, esto se debe a que cuenta con diversas funciones que abarcan áreas reconocimiento facial, calibración de cámaras, visión robótica entre otras. (Bravo, 2018).

IDE

Entorno de Desarrollo Integrado es un software que contiene un conjunto de herramientas para la programación como es la interfaz gráfica, compilador y depurador. (Quiroz, Muñoz, & Noël, 2012).

Algoritmo

Es una secuencia de instrucciones cada una de las cuales representa una tarea bien definida y puede ser llevada a cabo en una cantidad finita de tiempo y con un número finito de recursos computacionales (Bottazzi, Costarelli, D'Elia, et al., 2017).

Ruido

El ruido es causado por las interferencias electrónicas debidas a los circuitos y componentes necesarios para interpretar los datos del sensor y convertirlos en una imagen digital. El ruido puede reducirse con sensores

de alta calidad, comprimiendo la imagen y evitando que la temperatura del sensor aumente. (Campos, Bozal, 2005).

Red Neuronal

Es un sistema de computación compuesto por un gran número de elementos simples, elementos de procesos muy interconectados, los cuales procesan información por medio de su estado dinámico como respuesta a entradas externas. (Matich, 2001).

CAPÍTULO III

METODOLOGÍA DE LA INVESTIGACIÓN

El presente proyecto va dirigido a una investigación cuasiexperimental debido a que permite caracterizar las aceitunas por su variedad mediante descriptores previamente designados, la cual forma parte del requisito a nivel de sistema para el desarrollo del clasificador, por lo tanto, para el avance de este trabajo es seleccionado la metodología PDI.

Análisis de Factibilidad

Este proyecto servirá como apoyo al momento de clasificar aceitunas, con la finalidad de obtener calidad en los envíos de dicho cultivo. Resulta factible, porque a través de algoritmos de visión artificial se realiza el PDI con lo cual se entrena clasificador aumentando su precisión, logrando optimizar la manera de clasificar las aceitunas de mesa por su variedad. Cabe mencionar que con ayuda de los avances de la tecnología permiten agilizar las labores cotidianas en diferentes campos, especialmente en la agricultura donde el riego del cultivo y la recolección del fruto se tiende hacer automático, logrando obtener una calidad en el cultivo, en la producción y en la automatización de procesos. Por ello, la agricultura de precisión resulta como una de la mejores herramientas utilizadas en los cultivos para incrementar la producción y automatizar procesos con la finalidad de satisfacer la demanda de los consumidores.

Factibilidad Operacional

Resulta útil para el personal encargado de la recolección puesto que tendrán una segunda evaluación de la variedad de la aceituna por clasificar, el cumplir con una jornada de trabajo de esta índole puede

causar estrés y fatiga visual, debido a que se debe tomar en cuenta los estándares de calidad definidos por expertos.

Factibilidad Técnica

Para el desarrollo del clasificador se utilizaron ordenadores con gran capacidad computacional cuyas características se detallan en el cuadro 8, la cámara de los celulares con lo que se efectuó la captura de imágenes se encuentran en el cuadro 9 y el software utilizado durante el transcurso del proyecto en el cuadro 10.

Cuadro 8. Requerimientos de Ordenadores

ESPECIFICACIONES TÉCNICAS	LAPTOP DELL INSPIRON 15	ORDENADOR CLON	ORDENADOR CLON
Procesador	Intel Core I7 7700HQ 2.80 Ghz	Intel Core I7 4790 3.6 Ghz	Intel Core I7 6700 3.4Ghz
RAM	8 GB DDR4 2400 Mhz	16 GB DDR3 1600 Mhz	8 GB DDR4 2100 Mhz
Tarjeta Gráfica	Nvidia GTX 1050 4GB	Nvidia GTX 1060 6GB	Nvidia GTX 1060 6GB
SO	Windows 10 Pro	Windows 10 Pro	Windows 10 Pro

Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

Cuadro 9. Características de celulares

ESPECIFICACIONES TÉCNICAS	CELULAR 1	CELULAR 2
Marca	Samsung	Huawei
Nombre	Galaxy J7 Pro	P30 Lite
Modelo	SM J730GM/DS	MAR-LX3A
Resolución de cámara trasera	13 MP, f/1.7, 27mm (wide), AF	Triple, 24 MP (f/1.8, PDAF) + 8MP (ultrawide) + 2 MP (f/2.4, profundidad)

Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

Cuadro 10. Software y librerías utilizadas

SOFTWARE	
Entorno de Trabajo Integrado	PyCharm Community Edition 2019 2.5
Lenguaje de Programación	Python 3.5
OpenCV	Versión 4.1.2
Keras	Versión 2
TensorFlow	Versión 2.0
NumPy	Versión 1.17.4
Elaborado por: Daniel Fajardo, Milton Toalombo	
Fuente: Información del Proyecto	

Factibilidad Legal

En esta etapa se revisará a fondo el proyecto con la finalidad de validar si el software respeta leyes mencionadas en el capítulo II, en caso de incumplir alguna ley se asumirá las consecuencias. La factibilidad legal resulta de mucha importancia debido a que autoriza a los desarrolladores del software a poseer el derecho de autor del código fuente y documentación presentada.

Factibilidad Económica

En esta etapa se toma en cuenta el costo para el desarrollo del software, para la obtención del valor de este proyecto se debe tener en cuenta los recursos y egresos que han sido utilizados en detalle para generar dicho costo.

Cuadro 11 Tabla de Presupuesto

	Descripción	Cantidad	Costo Unitario	Costo Total
Talento Humano	Programador	2	\$1200 (3 meses)	\$2400
	Laptops	1	\$1500	\$1500
	Ordenador	2	\$1400	\$2800
Equipos	Teléfono inteligente	2	\$300	\$600
	Lámpara	2	\$7	\$14
	Pycharm	1	\$0	\$0
Software	Python	1	\$0	\$0
	Libra de botija	2	\$1,50	\$3
Aceitunas	Frasco de manzanilla	2	\$3	\$6
	Impresiones	500	\$0,25	\$125
Suministro	Empaste de la tesis	1	\$50	\$50
	Publicación de paper	1	\$150	\$150
Total				\$7.648

Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

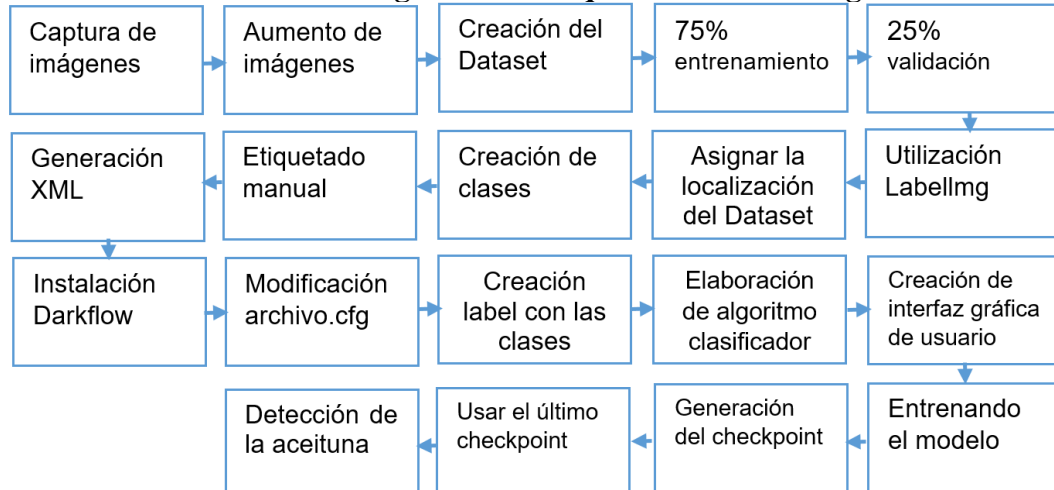
Etapas de la metodología del proyecto

El proyecto utiliza la metodología PDI englobando diversas etapas, desde la captura de imágenes tomadas con la cámara trasera del celular en diferentes ángulos y distancias a través de un escenario preliminar para la generación del dataset, hasta la generación del modelo para el reconocimiento del objeto. Motivo por el cual se toma en consideración ciertos aspectos como el dataset de imágenes, obtención de las características a partir de este y la elección de un clasificador. La metodología PDI aplicada consta de las etapas de captura, preprocesamiento, segmentación, extracción de características y clasificador las cuales se han mencionado en profundidad en el capítulo dos.

Se ha seleccionado las RNC para el desarrollo del clasificador en base a los resultados que han obtenido otros autores al momento de ser comparado con otros, como puede ser el bayesiano o vecinos más próximos presentando una pequeña variación en el porcentaje precisión,

donde RNC llega a obtener una precisión mayor debido a que la cantidad de procesos para llevar a cabo para la clasificación, los cuales disminuyen empleando librerías recientes, como tensorflow, keras entre otros (véase gráfico 26).

Gráfico 26. Diagrama de Bloques de la metodología

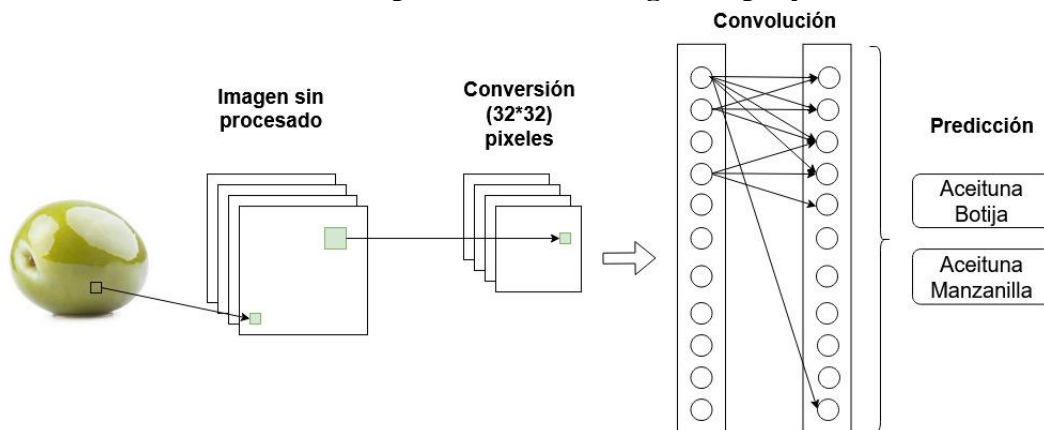


Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

Como se puede observar, en el gráfico 27 representa el esquema de las fases del PDI, tomando el dataset como variables de entradas seguidas del preprocesamiento de esta, iniciando la fase de entrenamiento donde se perpetran las convoluciones, es decir, se realiza un barrido de (2x2) pixeles conocido como maxpooling, de izquierda a derecha con la finalidad de reducir datos irrelevantes al momento de la detección, para finalmente generar del modelo.

Gráfico 27. Etapas de la metodología del proyecto



Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

Captura

En las primeras etapas del desarrollo por efectos prácticos se procede a utilizar web scraping, herramienta utilizada para la extracción de información en diferentes sitios webs, por ello se aplica la herramienta google image download para la obtención de 100 imágenes aproximadamente de las aceitunas botija y manzanilla para el dataset, sin la necesidad de tener que adquirir estas variedades, por lo que se lo considera como una versión preliminar.

Para la instalación se ejecuta el siguiente código en la línea de comando del sistema operativo:

```
C:\Users\Users> pip install google_images_download  
  
C:\Users\Users> googleimagesdownload --keywords  
nombre_objeto --limit 100
```

Este código permite descargar las imágenes que ha nombrado en keywords, seguido de las cantidades a descargar en limit. Cabe mencionar que hay ciertas palabras claves que se pueden añadir al código anterior para asignar el tipo, la resolución, en que sitio web o dominio en concreto desea descargar las imágenes, entre otros.

Una vez ejecutado el código comienza con la descarga de cada imagen, estas quedarán almacenadas en una carpeta que se crea inmediatamente de la efectucción del algoritmo con el nombre downloads en la línea de comandos del sistema operativo se puede visualizar las imágenes que ya han sido descargadas. (véase gráfico 28 y 29).

Gráfico 28. Obtención de imágenes mediante la línea de comandos

```

Administrador: Símbolo del sistema - googleimagesdownload --keywords "Gordal de Sevilla" --limit 100

C:\Users\Sakur\PycharmProjects\ProyectoAceitunas\darkflow-master\new_model_data>googleimagesdownload --keywords
"Gordal de Sevilla" --limit 100

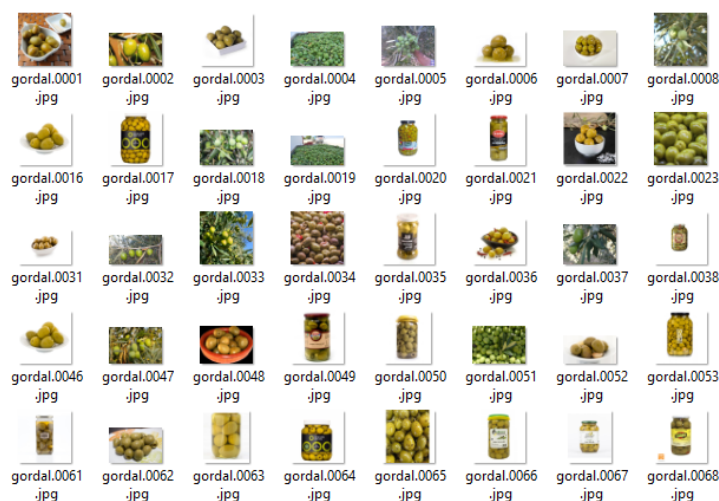
Item no.: 1 --> Item name = Gordal de Sevilla
Evaluating...
Starting Download...
Completed Image ----> 1.Gordal%20de%20Sevilla%201.JPG
Completed Image ----> 2.Gordal%20de%20Sevilla%202.JPG
Completed Image ----> 3.FIGURA-10-Fruto-de-la-variedad-Gordal-Sevillana-con-zofairones-frutos-pequeños.png
Completed Image ----> 4.image.jpg
Completed Image ----> 5.gordal-sevillana.png
Completed Image ----> 6.aceituna_sevillana_calidad_tradicion.jpg
Completed Image ----> 7.Gordal%20de%20Sevilla%203.JPG
Completed Image ----> 8.71d524c86c97341fbb9025601b5b39ceb407f337.jpg
Completed Image ----> 9.Gordal_Sevillana_02.jpg
Completed Image ----> 10.aceituna_gordal_verdeo.jpg
Completed Image ----> 11.olivo-gordal-sevillana-c-1.jpg
Completed Image ----> 12.aceituna_sevillana534.jpg
Completed Image ----> 13.Gordal%28de%82Sevilla%21.JPG
Completed Image ----> 14.51d725c86c97341ybb95b39ceb40498.jpg
Completed Image ----> 15.Gordal_Sevillana4_06.jpg

```

Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

Gráfico 29. Imágenes ubicadas en el ordenador por la ejecución del comando anterior



Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

Durante las últimas pruebas llevadas a cabo utilizando la versión preliminar, los resultados obtenidos no se acercaron a las expectativas esperadas, llegando a dar una precisión menor al 50%, motivo por el cual se vuelve a generar el dataset, por ello se importa las aceitunas manzanillas de España y al no encontrar un vendedor para las aceitunas gordal, se optó por utilizar otra variedad la cual es botija que fue importada de Perú.

Los escenarios de captura se fabricaron con hojas A4 simulando una forma cuadrada, además de usar lámparas con luz blanca para obtener la

iluminación en las fotos tal como se visualiza el gráfico 30. Por ello se emplean dos Smartphone que son utilizados para realizar las tomas desde diferentes ángulos y distancias, posteriormente aplicar diferentes técnicas para generar aumentos de imágenes que se mencionan en la siguiente etapa, llegando a obtener más de 1000 imágenes aproximadamente para cada variedad. (véase gráfico 31).

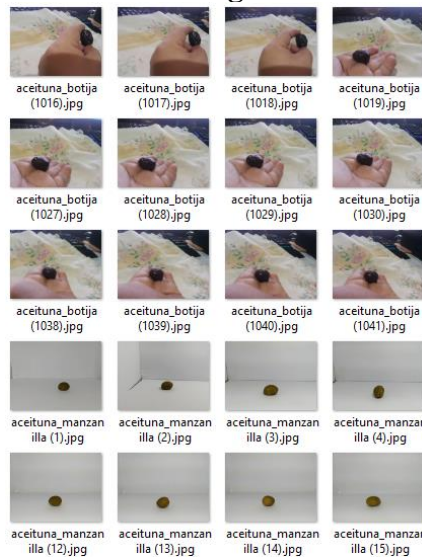
Gráfico 30. Escenarios para la toma de imágenes



Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

Gráfico 31. Imágenes tomadas



Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

Pre-Procesamiento

Para obtener una versión preliminar se procedió a utilizar un dataset de las 100 imágenes obtenidas para la variedad de gordal y manzanilla

mediante el web scraping, las resoluciones de las imágenes eran variadas, por ello se utilizó un algoritmo que permite redimensionar todas las imágenes a una escala de (32*32) pixeles logrando optimizar el tiempo que puede tardar en entrenar el modelo. (véase gráfico 32).

```
tamaño=32
dim= (tamaño, tamaño)
imagen = cv2.resize(imagen, dim, cv2. INTER_AREA)
```

Donde:

tamaño = pixeles a redimensionar

dim = matriz que toma los valores de tamaño

imagen = variable que almacena el cambio

Gráfico 32. Redimensión de imagen gordal



gordal.0001.jpg



gordal001.jpg

Propiedad	Valor	Propiedad	Valor
Imagen		Imagen	
Id. de imagen		Id. de imagen	
Dimensiones	800 x 800	Dimensiones	32 x 32
Ancho	800 pixeles	Ancho	32 pixeles
Alto	800 pixeles	Alto	32 pixeles
Resolución horizontal	96 ppp	Resolución horizontal	96 ppp
Resolución vertical	96 ppp	Resolución vertical	96 ppp
Profundidad en bits	24	Profundidad en bits	24

Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

A través de consultas realizadas a expertos mencionan que el dataset debe tener al menos 100 imágenes, caso contrario será impreciso el clasificador, el 75% de estas serán usadas para el entrenamiento con la finalidad de obtener un mayor reconocimiento al momento de la detección, y el 25% restante para validación, motivo por el cual se procede a utilizar ciertos comandos que permiten aplicar filtros como la rotación, reducción de ruido, aumento y disminución de brillo, entre otras a las imágenes previamente obtenidas para guardarlas como nuevas,

esto se puede lograr mediante el siguiente algoritmo. (véase gráfico 33 y 34).

```
Datagen = ImageDataGenerator(
    rescale = 1./255,
    rotation_range = 30,
    width_shift_range = 0.2,
    height_shift_range = 0.2,
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True,
    fill_mode = 'nearest'.
    Brightness_range = [0.2,0.8]
)
```

Donde

ImageDataGenerator = Librería de keras para aumento de datos

rescale = Cambiar la resolución de la imagen

rotation_range = rango de rotación

width_shift_range = rango de desplazamiento en anchura

height_shift_range = rango de desplazamiento en altura

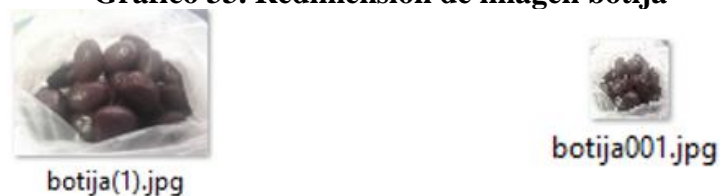
shear_range = rango de recorte

zoom_range = zoom aplicado a la imagen

horizontal_flip = volteo horizontal

fill_mode = modo relleno

Gráfico 33. Redimensión de imagen botija

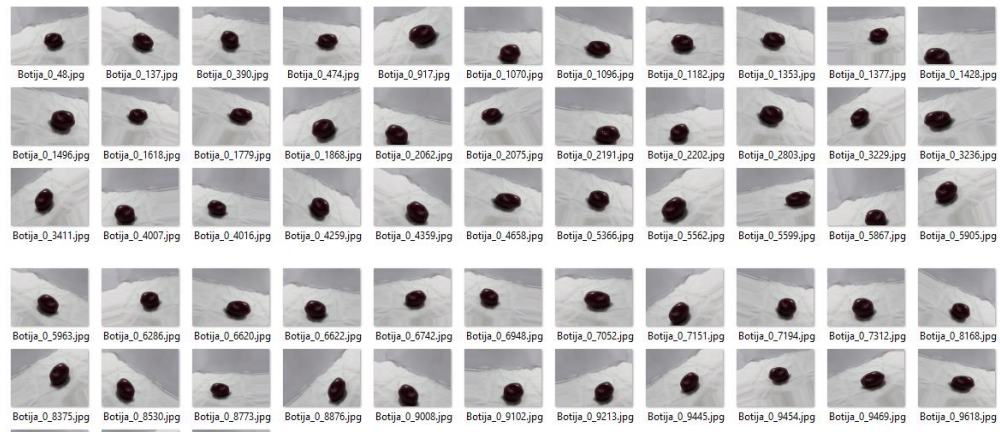


Propiedad	Valor	Propiedad	Valor
Imagen		Imagen	
Id. de imagen		Id. de imagen	
Dimensiones	4096 x 3072	Dimensiones	32 x 32
Ancho	4096 píxeles	Ancho	32 píxeles
Alto	3072 píxeles	Alto	32 píxeles
Resolución horizontal	96 ppp	Resolución horizontal	96 ppp
Resolución vertical	96 ppp	Resolución vertical	96 ppp
Profundidad en bits	24	Profundidad en bits	24

Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

Gráfico 34. Imágenes generadas para el aumento de datos



Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

Segmentación

El etiquetado en formato xml permite referenciar objetos marcados en la imagen, guardando información del tamaño de resolución de esta, además de la ruta y ubicación de los objetos en el plano X Y. Para la detección de objetos resulta fundamental el etiquetado, el cual requiere mayor tiempo debido a que se realiza de manera manual, para el proyecto aproximadamente cuatro horas fueron necesarias para realizar el etiquetado de cada aceituna dando origen a que se genere archivos xml por cada imagen, la cual se usará para generar el modelo.

Una vez obtenido el dataset de imágenes de las variedades de las aceitunas, se procede a realizar la etiquetación manual de cada una de ellas, utilizando la herramienta llamada labelImg ubicada en el repositorio de github <https://github.com/tzutalin/labelImg>, en ella se explica cómo realizar la instalación mediante líneas de comando de los sistemas operativos como Windows, Linux, macOS, además que permite realizar etiquetados generando archivos de extensión xml con el comando `Python labelImg.py`. (véase gráfico 35).

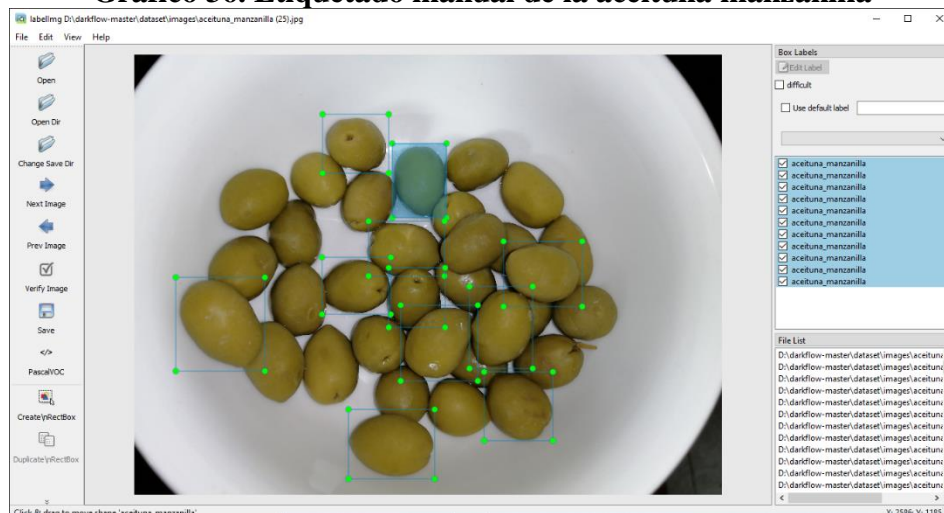
Gráfico 35. Etiquetado de imagen



Elaborado por: Daniel Fajardo, Milton Toalombo
Fuente: Información del Proyecto

Para comenzar a utilizar Labellmg, lo primero que se debe hacer es indicar la carpeta donde está almacenado el dataset de las imágenes para que esta herramienta reconozca las imágenes y posteriormente indicar en que carpeta desea almacenar los archivos xml generados una vez haya sido etiquetada la imagen, al momento de etiquetar una aceituna en concreto de la imagen debe escribir el nombre de la clase, tal como se puede visualizar en el gráfico 36.

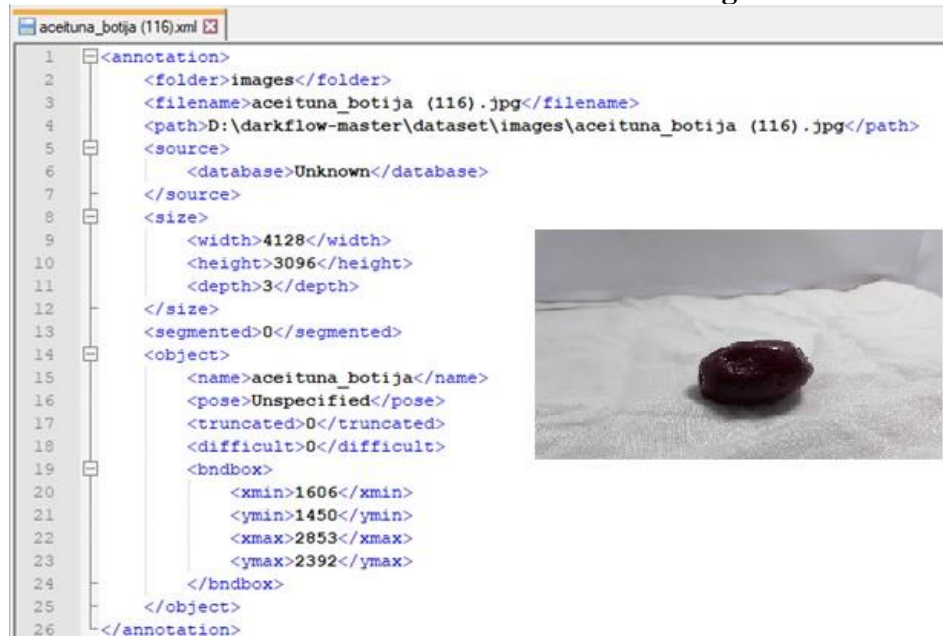
Gráfico 36. Etiquetado manual de la aceituna manzanilla



Elaborado por: Daniel Fajardo, Milton Toalombo
Fuente: Información del Proyecto

El archivo xml generado a partir de labellmg contiene información como la ruta de la imagen original, el nombre de la clase, las coordenadas x, y, la altura y anchura de cada rotulado que haya sido colocado en la imagen. Como se puede visualizar en el gráfico 37 solo hay una etiqueta debido a que había una aceituna en dicha imagen.

Gráfico 37. Visualización de un archivo xml generado



Elaborado por: Daniel Fajardo, Milton Toalombo
Fuente: Información del Proyecto

Extracción de Características

Una de las ventajas que ofrece el uso de RNC es que no consideran la fase de extracción de características de manera manual a cada una de las imágenes que se posee en el dataset, si no que el proceso es automatizado permitiendo reconocer ciertos patrones y rasgos que tengan en común entre todas las imágenes que la abarcan, es por ello, que debe tener aproximadamente mil imágenes tomadas desde diferentes ángulos, distancias, aplicando desenfoque de movimiento, modificando brillo, iluminación, entre otros, además de diferentes escenarios o backgrounds para que este no sea una característica propia de lo que se desea detectar.

Cabe mencionar que expertos en el tema proponen realizar el dataset en base al modelo de escala de grises permitiendo otorgar facilidad al momento de la detección, reduciendo las cantidades de características impropias de lo que se desea detectar como puede ser el escenario, entre otros.

Entre las características más habituales que fueron usadas para la detección y reconocimiento de las variedades de las aceitunas se tomaron en consideración las siguientes:

- El color es una de las principales características para detectar las variedades existentes de las aceitunas, además del estado de madurez propio del fruto, debido a que cada color es un punto en un espacio de color determinado, y la diferencia entre dos colores se considera como la distancia entre ellos.

Existen variedades donde el color del fruto maduro no es verde y negra, sino que puede ser amarillo o marrón, para la variedad botija y manzanilla utilizada en este trabajo, el color del epicardio varía en la gamma del negro y verde oscuro para las primeras etapas respectivamente, y cuando están maduros para el momento de su recolección el color tiende a llegar a ser morado o violeta y verde claro. Para la caracterización del color fueron utilizados dos modelos de color las cuales son el RGB y HSI, cada uno de estos modelos describe al color aplicando la media de componente.

- La forma es otra propiedad importante al momento de detectarlas; desde la obtención de las aceitunas se presentaba con una textura firme, robusta, además de la forma ovalada como una elipse casi perfecta que se mantenía en un buen estado debido a que estaba envasado. Sin embargo, durante los transcurso de los días estas cualidades fueron variando como la textura con pequeños rasgos de arrugas, se tornaron blandas, y la forma ovalada que poseía, ahora era achatada, tal como se ve en el gráfico 38.
- En cuanto al tamaño puede variar según la variedad, en las primeras etapas del fruto es pequeño y va aumentando ligeramente hasta llegar a la plena madurez. Cuando la maduración ha llegado a la última fase, se decliva causando que el tamaño disminuya a medida que el fruto se va secando, en cuanto a las aceitunas de mesa procesadas, no maduran, pero al ser dejadas al ambiente

durante unas cuantas horas se comienzan a secar y disminuye su tamaño. Tal como se ve en el gráfico 38.

Gráfico 38. Variación en la forma y tamaño



Elaborado por: Daniel Fajardo, Milton Toalombo
Fuente: Información del Proyecto

Se toma en cuenta características que corresponden con el tamaño, como puede ser:

Cuadro 12. Características del tamaño

	Tamaño	Fórmula Elipse
1	Área	$a \times b \times \pi$
2	Perímetro	$2\pi \sqrt{\frac{a^2 + b^2}{2}}$
3	Longitud	$4 \int_0^a \sqrt{1 + \frac{b^2 x^2}{a^2 (b^2 - x^2)}} dx$
4	Excentricidad	$2 \tan^{-1} \left(\sqrt{\frac{a-b}{a+b}} \right)$

Elaborado por: Daniel Fajardo, Milton Toalombo
Fuente: Información del Proyecto

Clasificador

Una vez etiquetadas las imágenes del dataset, se procede a entrenar el modelo, para ello se utiliza una versión modificada de YOLO (You Only Look Once) llamada Darkflow ubicada en uno de los repositorios de github para la descarga <https://github.com/thtrieu/darkflow>; Esta herramienta trae

modelos que han sido pre-entrenados con 80 clases propias como son mesa, silla, computador, persona, entre otros con su respectiva configuración, además incluye un test que se puede ejecutar luego de la instalación de Darkflow para probar la eficacia de reconocimiento al momento de utilizarlo ya sea a través de capturas en imágenes, videos o mediante una webcam. (véase gráfico 39).

Gráfico 39. Archivos del Darkflow

abagshaw Merge pull request #657 from cclauss/patch-1	
cfg	Fixed yolo-4c cfg to have 4 classes and added new labels flag
darkflow	import tensorflow as tf for line 16
sample_img	Clean up wording
test	Back to travis
.coveragerc	back to pytest-cov
.gitignore	Add tests for training
.travis.yml	Get rid of newline
LICENSE	license
README.md	Revert demo image switch
demo.gif	Add
flow	move flow code to cli.py
labels.txt	update README
preview.png	++ preview.png
setup.py	Added support for macOS

Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

El proceso de instalación de darkflow se realiza mediante línea de comandos, ubicando la carpeta descargada recientemente, se deben ejecutar los siguientes.

```
python setup.p build_ext -inplace
pip install -e .
pip install .
```

En este proyecto se utiliza dos variedades de aceitunas, botija y manzanilla, por ello se utiliza la configuración de tiny yolo ubicada en /cfg/ tiny-yolo.cfg para entrenamiento del modelo, la cual permite agilizar el tiempo de entrenamiento, debido a la cantidad de clases a utilizar, esta configuración permite realizar un entrenamiento hasta con 80 clases.

En el Cuadro 13, se puede visualizar los cambios realizados en la configuración de tiny yolo, que corresponden al pre-procesado de los datos de entrada y el archivo de configuración, además contiene información sobre las capas de convoluciones, y finalmente la utilización de maxpool para la reducción de datos innecesarios que para este proyecto no fueron modificadas.

Cuadro 13. Comparativa configuración de Yolo

Valores por defecto	Cambios aplicados
batch=64	batch=64
subdivisions=8	subdivisions=8
width=416	width=416
height=416	height=416
channels=3	channels=3
momentum=0.9	momentum=0.9
decay=0.0005	decay=0.0005
angle=0	angle=0
saturation = 1.5	saturation = 1.5
exposure = 1.5	exposure = 1.5
hue=.1	hue=.1
learning_rate=0.001	learning_rate=0.001
max_batches = 120000	max_batches = 40100
policy=steps	policy=steps
steps=-1,100,80000,100000	steps=-1,100,20000,30000
scales=.1,10,.1,.1	scales=.1,10,.1,.1
[convolutional]	[convolutional]
size=1	size=1
stride=1	stride=1
pad=1	pad=1
filters=425	filters=35
activation=linear	activation=linear
[region]	[region]
anchors = 0.738768,0.874946, 2.42204,2.65704, 4.30971,7.04493, 10.246,4.59428, 12.6868,11.8741	anchors = 1.08,1.19, 3.42,4.41, 6.63,11.38, 9.42,5.11, 16.62,10.52
bias_match=1	bias_match=1
classes=80	classes=2
coords=4	coords=4
num=5	num=5
softmax=1	softmax=1
jitter=.2	jitter=.2
rescore=1	rescore=1

Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

Al tener menos clases se ve afectada la variable de filtros mostrada en la tabla, por ello, se modifica la cantidad de clases que se van a utilizar para generar el modelo, y se aplica una fórmula matemática mostrada al final

del párrafo para el filtro. Si este último cambio no se efectúa, ocurrirá algún error al realizar el entrenamiento.

$$5(class + 5) \quad (8)$$

Donde

`class` = cantidad de clases usadas para generar el modelo

Luego de la modificación y guardado del archivo tiny yolo se procede a iniciar con el entrenamiento con la línea de comandos, para ello se debe dirigir a la carpeta de darflow y ejecutar lo siguiente:

```
python flow --model cfg/tiny-yolo-voc-bot-man.cfg --  
load bin/tiny-yolo-voc.weights --train --annotation  
dataset/annotations --dataset dataset/images --gpu 0.6  
--epoch 300
```

Donde:

model = archivo de configuración de tiny yolo.

load = archivo correspondiente a los pesos de tiny yolo.

train = carpeta que contiene los xml generados del etiquetado.

dataset = ubicación de la carpeta que contiene el dataset.

gpu = capacidad en porcentaje asignada del gpu para el entrenamiento.

epoch = número de épocas a realizar.

Para el trabajo se ha utilizado un ordenador que posee una tarjeta gráfica Nvidia GeForce GTX 1060 6GB, que permite realizar un entrenamiento más rápido debido al poder computacional que posee, dependiendo del modelo el valor asignado al GPU puede variar. En el caso que no se posea una GPU, bastaría con eliminarla del comando y el entrenamiento se realizaría usando el CPU, pero con esto el tiempo de entrenamiento aumentará.

Una vez ejecutado se carga la configuración del modelo, el peso, las anotaciones donde estarán las etiquetas de cada imagen, el dataset para posteriormente analizar si se encuentra algún inconveniente antes del

entrenamiento. En el caso que no encuentre alguna anomalía, comenzará el entrenamiento tal como se observa en el gráfico 40.

Gráfico 40. Ejecución del comando del entrenamiento

```

[+] Símbolo del sistema
WARNING:tensorflow:From C:\Python 3.7\lib\site-packages\tensorflow\python\training\rmsprop.py:119: calling Ones.__init__ (from tensorflow.python.ops.init_ops) with dtype
pe is deprecated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
2020-02-09 21:53:25.936810: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AV
X2
2020-02-09 21:53:25.947528: I tensorflow/stream_executor/platform/default/dso_loader.cc:42] Successfully opened dynamic library nvcuda.dll
2020-02-09 21:53:26.000588: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1640] Found device 0 with properties:
name: GeForce GTX 1060 6GB major: 6 minor: 1 memoryClockRate(GHz): 1.8475
pciBusID: 0000:01:00:0
2020-02-09 21:53:26.017866: I tensorflow/stream_executor/platform/default/dlopen_checker_stub.cc:25] GPU libraries are statically linked, skip dlopen check.
2020-02-09 21:53:26.031027: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1763] Adding visible gpu devices: 0
2020-02-09 21:53:28.446227: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1181] Device interconnect StreamExecutor with strength 1 edge matrix:
2020-02-09 21:53:28.451801: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1187] 0
2020-02-09 21:53:28.456659: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1200] 0: N
2020-02-09 21:53:28.463240: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1326] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 3686
MB memory) -> physical GPU (device: 0, name: GeForce GTX 1060 6GB, pci bus id: 0000:01:00:0, compute capability: 6.1)
Finished in 9.059234380722046s

Enter training ...

cfg/tiny-yolo-voc-bot-man.cfg parsing dataset/annotations
Parsing for ['aceituna_botija', 'aceituna_manzanilla']
[=====]100%  aceituna_manzanilla (999).xml
Statistics:
aceituna_botija: 2941
aceituna_manzanilla: 1680
Dataset size: 2098
Dataset of 2098 instance(s)
Training statistics:
  Learning rate : 1e-05
  Batch size    : 16
  Epoch number  : 500
  Backup every  : 2000
2020-02-09 21:54:07.200055: W tensorflow/core/common_runtime/bfc_allocator.cc:237] Allocator (GPU_0_bfc) ran out of memory trying to allocate 2.22GiB with freed_by_cou
nt=0. The caller indicates that this is not a failure, but may mean that there could be performance gains if more memory were available.
2020-02-09 21:54:07.270159: W tensorflow/core/common_runtime/bfc_allocator.cc:237] Allocator (GPU_0_bfc) ran out of memory trying to allocate 2.32GiB with freed_by_cou
nt=0. The caller indicates that this is not a failure, but may mean that there could be performance gains if more memory were available.
2020-02-09 21:54:07.402551: W tensorflow/core/common_runtime/bfc_allocator.cc:237] Allocator (GPU_0_bfc) ran out of memory trying to allocate 2.26GiB with freed_by_cou
nt=0. The caller indicates that this is not a failure, but may mean that there could be performance gains if more memory were available.
2020-02-09 21:54:07.500026: W tensorflow/core/common_runtime/bfc_allocator.cc:237] Allocator (GPU_0_bfc) ran out of memory trying to allocate 2.22GiB with freed_by_cou
nt=0. The caller indicates that this is not a failure, but may mean that there could be performance gains if more memory were available.
```

Elaborado por: Daniel Fajardo, Milton Toalombo
Fuente: Información del Proyecto

Posteriormente se comienza a generar un checkpoint equivalente a los 125 pasos, la cantidad de pasos para la generación del checkpoint varia en base a la cantidad de imágenes en el dataset. En el gráfico 41, se puede visualizar el último checkpoint generado llegando a la época 26.

Gráfico 41. Época 26 de fase del entrenamiento del modelo

```

[+] Símbolo del sistema - python flow --model cfg/tiny-yolo-voc-bot-man.cfg --load 3500 --train --annotation dataset/annotations --dataset dataset/images --gpu 0.6
step 5373 - loss 1.1044862270355225 - moving ave loss 1.6453110049032
step 5374 - loss 1.7131285667419434 - moving ave loss 1.6520927618870744
step 5375 - loss 1.2198951244354248 - moving ave loss 1.6088729974219094
Checkpoint at step 5375
step 5376 - loss 1.0137399435043335 - moving ave loss 1.5493596920301518
step 5377 - loss 2.4009366035461426 - moving ave loss 1.6345173831817508
step 5378 - loss 1.1922214031219482 - moving ave loss 1.5902877851757706
step 5379 - loss 1.0741546154022217 - moving ave loss 1.5386744681984157
step 5380 - loss 1.3429961204528809 - moving ave loss 1.5191066334238623
step 5381 - loss 1.9115841388702393 - moving ave loss 1.5583543839685001
step 5382 - loss 1.2984325885772705 - moving ave loss 1.5323622044293772
step 5383 - loss 1.1336276531219482 - moving ave loss 1.4924887492986345
step 5384 - loss 2.8158674240112305 - moving ave loss 1.6248266167690941
step 5385 - loss 0.8214164972305298 - moving ave loss 1.5444856048159576
step 5386 - loss 1.7534457445144653 - moving ave loss 1.5653816187858085
step 5387 - loss 1.612928867340008 - moving ave loss 1.5701363436412366
step 5388 - loss 0.906352162361145 - moving ave loss 1.5037579255132276
step 5389 - loss 1.8569573163986206 - moving ave loss 1.5390778646017669
step 5390 - loss 2.13840389251709 - moving ave loss 1.5990104673932994
step 5391 - loss 1.3577485004533691 - moving ave loss 1.5748842714993065
step 5392 - loss 1.1921008825302124 - moving ave loss 1.536605932602397
step 5393 - loss 1.1266191005706787 - moving ave loss 1.4956072493992252
step 5394 - loss 0.8783203363418579 - moving ave loss 1.4338785580934885
step 5395 - loss 2.709989547729492 - moving ave loss 1.5614896570570889
step 5396 - loss 1.681792140007019 - moving ave loss 1.5735199053520819
step 5397 - loss 1.7821910381317139 - moving ave loss 1.5943870186300453
step 5398 - loss 1.3044244050979614 - moving ave loss 1.5653907572768369
step 5399 - loss 1.8017017841339111 - moving ave loss 1.5890218599625443
step 5400 - loss 3.849623918533325 - moving ave loss 1.8150820658196223
step 5401 - loss 2.3944766521453857 - moving ave loss 1.8730215244521986
step 5402 - loss 1.4581314325332642 - moving ave loss 1.8315325152603052
step 5403 - loss 1.498982548713684 - moving ave loss 1.798277518605643
step 5404 - loss 2.161947250366211 - moving ave loss 1.8346444917817
step 5405 - loss 1.4602012634277344 - moving ave loss 1.7972001689463033
step 5406 - loss 1.213772654333862 - moving ave loss 1.7388574175050118
step 5407 - loss 0.7584524750709534 - moving ave loss 1.640816923261606
step 5408 - loss 1.5477900505065918 - moving ave loss 1.6315142359861048
step 5409 - loss 1.0570855140686035 - moving ave loss 1.5740713637943546
step 5410 - loss 1.0953478813171387 - moving ave loss 1.5261990155466332
step 5411 - loss 1.3962669372558594 - moving ave loss 1.5132058077175559
step 5412 - loss 1.5020203590393066 - moving ave loss 1.512087262849731
```

Elaborado por: Daniel Fajardo, Milton Toalombo
Fuente: Información del Proyecto

Inicialmente con el dataset de imágenes descargadas mediante el web scraping, se utilizó para corroborar la efectividad del funcionamiento del yolo con esas imágenes se realizó la fase de entrenamiento, cuya duración fue de 4 horas aproximadamente logrando obtener resultados insatisfactorios. Motivo por el cual se ha realizado otro dataset de imágenes que fue explicado en el apartado de captura, por ello, se vuelve a realizar el mismo procedimiento con la finalidad de obtener resultados óptimos.

Cabe mencionar que con el nuevo dataset de imágenes se espera que la duración del entrenamiento fuera mayor debido a la cantidad de imágenes que poseía a diferencia del primer dataset, sin embargo, al completar 125 pasos del entrenamiento se generaba un checkpoint, hasta llegar a las 300 épocas. Para ello, se debe dejar la máquina encendida entrenando el modelo, por lo menos una semana y media. En el cuadro 14 se visualiza los puntos de control generados en la carpeta ckpt desde la primera época.

Cuadro 14. Visualización de los puntos de controles generados	
Punto de Control	Validación
1125	
3000	
8500	

Elaborado por: Daniel Fajardo, Milton Toalombo
Fuente: Información del Proyecto

El último checkpoint generado es el que se utilizará para la detección de aceitunas junto con `tiny_yolo.cfg`, siendo estos necesarios al momento de ejecutar futuras detecciones, la cual puede realizarse mediante imágenes o a través de una webcam, para luego de ser procesada, mostrar en un cuadro la ubicación de la aceituna indicando si su variedad es botija o manzanilla. Cabe mencionar, que a ciertas distancias no reconocerá la aceituna, esto puede solucionarse incluyendo un dataset que simule estas distancias. La red convolucional no ha logrado mostrar los índices o factores de madurez en el resumen mostrado por consola al momento de ejecución, sin embargo, otros análisis implementados con diferentes frutas han utilizado técnicas que se podrían acoplar como una mejora para este trabajo.

Criterios de Validación

Juicios de Expertos

Para la validación del proyecto se utilizó el instrumento de juicio de expertos con la finalidad de realizar las pruebas de funcionalidad y porcentaje de validación del software desarrollado, adicional que los expertos que realicen la validación correspondiente puedan ofrecer valorización para este proyecto y que las técnicas implementadas sean las adecuadas. (Véase cuadro 15). En la sección de anexos se encuentra la ficha de juicios de expertos junto con la constancia de este.

Cuadro 15. Usuarios que realizaron juicio de expertos

Usuarios	Cargos
Ing. Miguel Botto Tobar, MSC.	Ing. en sistemas computacionales

Elaborado por: Daniel Fajardo, Milton Toalombo

Fuente: Información del Proyecto

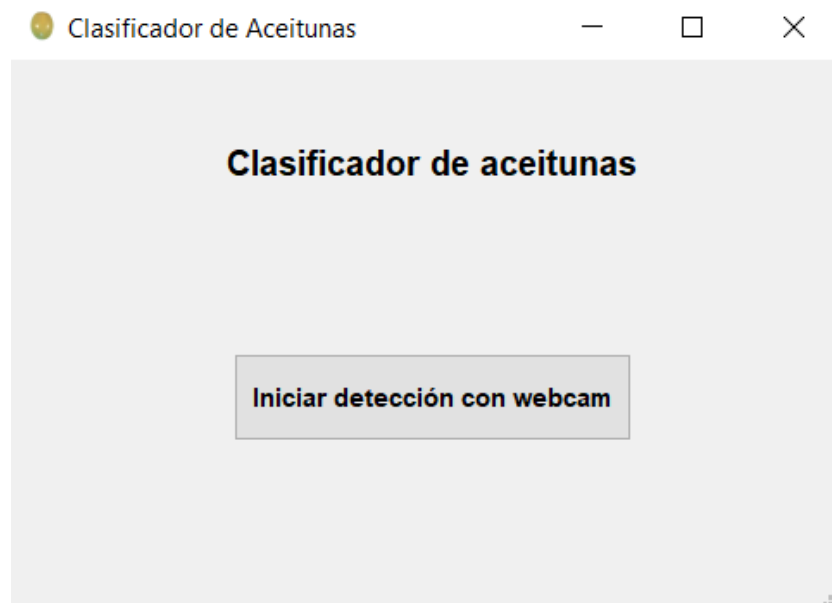
CAPÍTULO IV

RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

RESULTADOS

Se elaboró una interfaz gráfica con una opción, para realizar la detección de aceitunas utilizando una camara web, en el gráfico 42 se muestra dicha interfaz.

Gráfico 42 Interfaz de usuario del clasificador



Elaborado por: Daniel Fajardo, Milton Toalombo

Al momento de iniciar con la detección, en la consola de python se muestra la información de la configuración y se carga el modelo anteriormente generado tal como se muestra en el gráfico 43.

Gráfico 43. Visualización de Información de la configuración

```
Source | Train? | Layer description | Output size
-----+-----+-----+-----
WARNING:tensorflow:From D:\darkflow-master\darkflow\net\ops\baseop.py:84: T

| | input | | (? , 416, 416, 3)
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 416, 416, 16)
Load | Yep! | maxp 2x2p0_2 | | (? , 208, 208, 16)
WARNING:tensorflow:From D:\darkflow-master\darkflow\net\ops\simple.py:106:

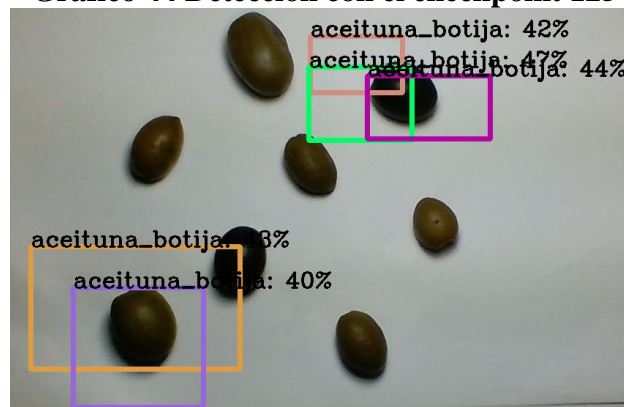
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 208, 208, 32)
Load | Yep! | maxp 2x2p0_2 | | (? , 104, 104, 32)
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 104, 104, 64)
Load | Yep! | maxp 2x2p0_2 | | (? , 52, 52, 64)
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 52, 52, 128)
Load | Yep! | maxp 2x2p0_2 | | (? , 26, 26, 128)
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 26, 26, 256)
Load | Yep! | maxp 2x2p0_2 | | (? , 13, 13, 256)
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 13, 13, 512)
Load | Yep! | maxp 2x2p0_1 | | (? , 13, 13, 512)
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 13, 13, 1024)
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 13, 13, 1024)
WARNING:tensorflow:From D:\darkflow-master\darkflow\net\build.py:132: The n

2020-02-11 23:54:26.681405: I tensorflow/core/platform/cpu_feature_guard.cc
Init | Yep! | conv 1x1p0_1 linear | | (? , 13, 13, 35)
-----+-----+-----+-----
GPU mode with 0.6 usage
WARNING:tensorflow:From C:\Python 3.7\lib\site-packages\tensorflow\python\t
Instructions for updating:
Use standard file APIs to check for files with this prefix.
Loading from ./ckpt/tiny-yolo-voc-bot-man-3500
Finished in 4.4797444343566895s
```

Elaborado por: Daniel Fajardo, Milton Toalombo

En el gráfico 44 se muestra la detección de la camara web utilizando el checkpoint 125 como modelo, mostrando erróneamente la ubicación de la aceituna.

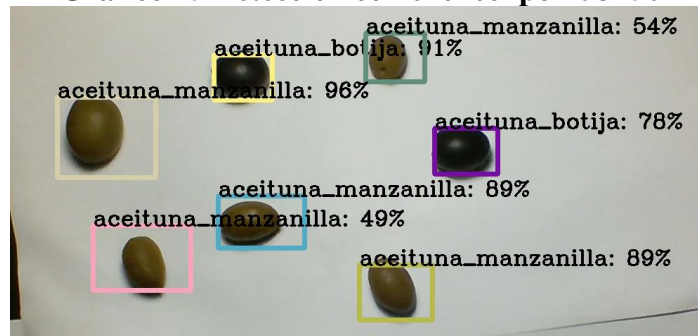
Gráfico 44 Detección con el checkpoint 125



Elaborado por: Daniel Fajardo, Milton Toalombo

Después de varias horas de entrenamiento se realizó nuevamente una prueba utilizando el checkpoint 3250, presentado resultados aceptables con un incremento de precisión en la detección de las aceitunas como se visualiza en el gráfico 45.

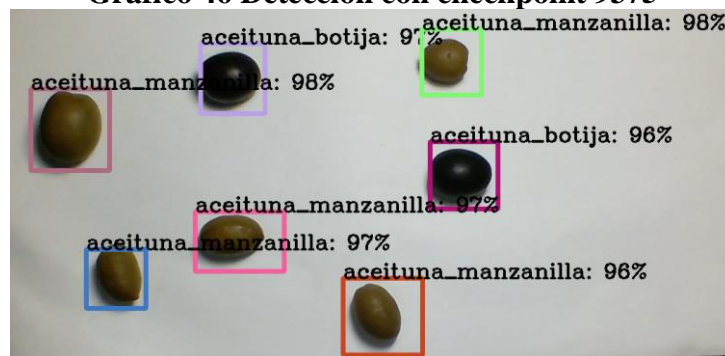
Gráfico 45 Detección con el checkpoint 3250



Elaborado por: Daniel Fajardo, Milton Toalombo

Llegando al checkpoint 9375 se decidió realizar otra prueba obteniendo resultados formidables durante la detección. (véase gráfico 46).

Gráfico 46 Detección con checkpoint 9375



Elaborado por: Daniel Fajardo, Milton Toalombo

Pese a obtener dichos resultados se tomó la decisión de continuar con el entramiento, con la finalidad de reducir ciertos errores en la detección que dicho checkpoint mostraba. Además, se añadió la función de capturar el frame mientras se ejecuta la detección, para ello se debe presionar la tecla s, el frame será almacenado como una imagen en formato .jpg, con ello la imagen se guardará en la ruta ../darkflow-master/capturas/, en caso de que no exista la carpeta, esta será creada por defecto. (véase gráfico 47).

Gráfico 47 Carpeta con capturas

Este equipo > Disco local (D:) > darkflow-master > capturas



Elaborado por: Daniel Fajardo, Milton Toalombo

CONCLUSIONES

La cantidad de imagenes que posee el dataset es fundamental para evitar resultados imprecisos, por lo que se usó un dataset de 1000 imágenes aproximadamente por variedad, cada una con su respectiva etiqueta generada con la herramienta LabelImg.

La generación de los checkpoints durante el entrenamiento ayudó a realizar diferentes pruebas en las que se corroboró que la precisión aumentaba a medida que este continuaba, después de varios días de entrenamiento, se tomo el checkpoint donde el margen de error era menor. Se concluye que la utilización de técnicas que permiten generar cantidades de datos para el aumento del dataset es próspero ya que ayuda en obtener resultados favorables.

Al momento de entrenar el modelo, este adquiere conocimiento para la detección de las aceitunas, a partir de la información guardada en las anotaciones, mismas que fueron generadas durante el proceso de etiquetado, se toma en cuenta que la fase de entrenamiento puede demorar bastante, debido a la cantidad de imágenes y la resolución que estas posean en el dataset, además de las especificaciones técnicas del ordenador donde se haya ejecutado para la generación del modelo.

Luego de un entrenamiento de 36 horas para la generación del modelo se completaron 25 épocas, permitiendo que el reconocimiento sea más preciso, aunque tiende a tener ciertas falencias como es el caso de la detección a una distancia considerable, o muy cercana.

RECOMENDACIONES

Para crear un modelo desde cero es necesario contar con más de 1000 de imágenes en el dataset, en caso de tener escasez de estas, se puede implementar técnicas de aumento de datos.

Se puede agregar más variedades de aceitunas, para que su detección sea efectiva, se recomienda realizar la captura personalmente en lugar de obtener imágenes de internet. De esa misma forma se puede lograr la detección de diferentes frutas.

En caso de no tener un ordenador con GPU Nvidia, se recomienda la utilización de un entorno de máquina virtual en la nube como puede ser google colab.

BIBLIOGRAFÍA

Ponce, J. M., Aquino, A., Segura, F., Millán, B., & Andújar, J. M. (2018). Estimación automatizada del peso y calibre de aceitunas mediante análisis de imagen. *Actas de las XXXIX Jornadas de Automática, Badajoz*, 958 -966.

Ferguson, L. (2006). Tendencias actuales en la recolección mecánica de las aceitunas. *Grasas y aceites*, 57(1), 9-15. doi: 10.3989/gya.2008.v59.i3.509

Aguilera, D. A., Moreno, O. C., Gila, D. M., García, J. G., & Ortega, J. G. (2015). Clasificación automática de lotes de aceitunas en almazara mediante visión por computador. *Actas de las XXXVI Jornadas de Automática, Bilbao*, 1(1), 59-65.

Montoya Holguin, C., Cortés Osorio, J. A., & Chaves Osorio, J. A. (2014). Sistema automático de reconocimiento de frutas basado en visión por computador. *Ingeniare. Revista chilena de ingeniería*, 22(4), 504-516. doi: 10.4067/S0718-33052014000400006

Bonilla-González, J. P., & Prieto-Ortiz, F. A. (2016). Determinación del estado de maduración de frutos de feijoa mediante un sistema de visión por computador utilizando información de color. *Revista de Investigación, Desarrollo e Innovación*, 7(1), 111-126. doi: 10.19053/20278306.v7.n1.2016.5603

Martínez Llano, G. E., Ortiz, M., Manuel, S., Herrera Pérez, J. C., Berrio Pérez, J. S., & Beleño Sáenz, K. D. J. (2016). Clasificación de los frutos de café según su estado de maduración y detección de la broca mediante técnicas de procesamiento de imágenes. *Revista Prospectiva*, 4(1), 15-22. doi: 10.15665/rp.v14i1.640.

Sandoval, Z., & Prieto, F. (2009). Procesamiento de imágenes para la clasificación de café cereza. *Revista Prospectiva*, 7(1), 67-73.

Marin, N. S., & Madrigal, C. A. (2016). Clasificación de calidad de fresas usando procesamiento de imágenes y máquinas de vectores de soporte. *VIII Congreso Internacional de Computación y Telecomunicaciones. Universidad Inca Garcilaso de la Vega*. 1(1), 157-163.

López, A. J., Pelayo, M. P., & Forero, Á. R. (2015). Enseñanza del Procesamiento de Imagenes en Ingeniería usando Python. *Revista Versión Abierta Español–Portugués*, 3(4), 179-186.

Olive, P. T. (2015). Alteraciones y pérdida de la calidad en aceituna de mesa y aceite de oliva. *IMIDRA Instituto Madrileño de Investigación y desarrollo rural, agrario y alimentario*, 1-37.

Silva, L. A., & Lizcano, S. (2012). Evaluación del estado de maduración de la piña en su variedad perolera mediante técnicas de visión artificial. *Revista Iteckne*, 9(1), 31-41.

Cabezas, J. E. (2011). La aceituna de mesa: nociones sobre sus características, elaboración y cualidades. *Aceituna de Mesa Fundación para el Fomento y Promoción*. 1(1), 1-156.

CODEX STAN 66-1981, FAO (2013). Food and Agriculture Organization. 1(1). 1-16.

Garrido Fernández, A., & Romero Barranco, C. (1999). Calidad de las aceitunas de mesa. *Grasas y Aceites*. 50(3), 225-230, doi: 10.3989.

Zúñiga Acevedo, R. A., Riquelme Sanhueza, J., & González López, B. B. (2019). Evaluación de cosechadoras cabalgantes en el cultivo del maqui (*Aristotelia chilensis* (Mol.) Stuntz) (Doctoral dissertation, Universidad de Talca (Chile). Escuela de Agronomía.). 1(1), 1-34.

Sierra, J. G. (2006). Últimas novedades en la recolección mecanizada de la aceituna. *Revista Vida Rural*. 1(238), 72-75.

Ribes, J. G., Roldán, G. B., & García, S. C. (2012). El futuro del olivar tradicional y de su recolección. *Revista Vida Rural*, 1(345), 30-38

Fernández Segovia, I., García Martínez, E. M., & Fuentes López, A. (2016). Parámetros de calidad de aceitunas de mesa. *Universitat Politècnica de València, Escuela Técnica Superior de Ingeniería Agronómica y del Medio Natural*. 1(1), 1-7

Molina L., Jimenez B., Ruiz F. Garcia F., Cano J., Pérez J., Técnicas de cultivo: Recolección mecanizada del olivar. *Consejería de Agricultura y Pesca*. 2(1), 1-84

Garcia, D. C., Martínez, G. Á., & Garcia, M. E. (2018). Raspberry Pi y Arduino: semilleros en innovación tecnológica para la agricultura de precisión. *Informática y Sistemas: Revista de Tecnologías de la Informática y las Comunicaciones*, 2(1), 74-82, doi: 10.33936/isrtic.v2i1.1134

Alvear, V., Rosero, P., Peluffo, D., & Pijal, J. (2016). Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura. *Enfoque UTE, Vol 8(Nº1: Edición Especial INCISCOS)*, 244-256, doi: 10.29019.

Villet, J. R. M. (2005). Procesamiento digital de Imágenes. *Facultad de Ingeniería Universidad Autónoma de San Luis Potosí*, 1(1), 1-100.

Avilés Pincay, A. I., Salavarría, B., & Ricardo, J. (2019). Elaborar un Clasificado para el Proceso Digital de Imágenes. *Estado del Arte (Doctoral dissertation, Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas. Carrera de Ingeniería en Sistemas Computacionales)*, 1(1), 1-242.

Cardozo, R., Méndez, Á., Legal Ayala, H., & Vázquez Noguera, J. L. (2018). Mejora de imágenes a color utilizando un enfoque morfológico multiescala. In XXIV Congreso Argentino de Ciencias de la Computación. 1(1), 381-390

Gaviria, N. E. A., & Ruíz, J. A. O. (2016). Análisis comparativo de descriptores para la clasificación de telas utilizando imágenes. Universidad Tecnológica de Pereira. Facultad de Ingenierías Eléctrica, Electrónica, Física, y Ciencias de la Computación. Ingeniería Electrónica. 1(1), 1-38.

Astudillo Loyola, M. P. (2017). Estandarización de parámetros de la cámara fotográfica y encuadre de la imagen para fotogrametría facial. Universidad de Chile. Facultad de odontología. Departamento del niño y ortopedia dentomaxilar. 1(1), 1-64.

Maldonado, A. P. B., Alza, L. C. L., & García, R. J. H. (2015). Segmentación y conteo de las líneas de la nariz del feto en imágenes ecográficas de las 11-13+ 6 semanas de gestación. *Revista Ingeniería*, 20(1), 65-78.

Matich, D. J. (2001). Redes Neuronales: Conceptos básicos y aplicaciones. *Universidad Tecnológica Nacional, México*. 1(1), 1-55.

Peláez, J. I., Vaccaro, G., & Guijarro, A. (2015). Un Modelo para la Categorización de Hormigones Mediante Procesamiento Digital de Imágenes. In 19th World Multiconference on Systemics, Cybernetics-CISCI, 13(1), 15-18.

Salinas, J. J. J., Sanchez, L. O. S., Miranda, R. C., Rodriguez, J. M. O., & Rosales, H. G. (2015). Algoritmo de procesamiento digital de imágenes para la detección y evaluación de heridas de pie diabético. *Difusión@ Revista en Ingeniería y Tecnología, UAZ*, 8(1), 1-6.

Palomino, N. L. S., & Concha, U. N. R. (2009). Técnicas de segmentación en procesamiento digital de imágenes. *Revista de investigación de Sistemas e Informática*, 6(2), 9-16.

Rebaza, J. V. (2007). Detección de bordes mediante el algoritmo de Canny. *Escuela Académico Profesional de Informática. Universidad Nacional de Trujillo*. 4(1), 1-8.

Querejeta Simbeni, Pedro (2015). Procesamiento digital de imágenes. Universidad Nacional del Sur, Avda. Alem 1253, B8000CPB Bahía Blanca, Argentina, 1(1), 1-5.

Dobernack, N. A. (2013). Implementación de un sistema de detección de señales de tráfico mediante visión artificial basado en FPGA. *Universidad de Sevilla, Sevilla, España*. 2(1), 1-255.

Vela, F. (2013). Transformada de Fourier en procesamiento digital de Imágenes. Funciones de Variable Compleja. *Universidad Nacional del Sur, Avda. Alem 1253, B8000CPB Bahía Blanca, Argentina*.1(1), 1-4

Rodriguez Castello, D. (2017). Extracción de cráneo en imágenes de resonancia magnética del cerebro utilizando una red neuronal convolucional 3D. (Bachelor's thesis, Universitat Politècnica de Catalunya). 1(1), 1-38.

Pazos Mandiá, J. C. (2015). Optimización de curvas ROC para perceptrones multicapa mediante técnicas evolutivas (Bachelor's thesis).1(1).1-63
doi: 10016/23211

Díaz Badra, R. (2018). Clasificador de vestimenta basado en redes neuronales. Universidad de las Palmas de Gran Canaria. 1(1), 1-56.

Delfin, C. M., González, C. J., & Rojas, J. O. (2018). Clasificación de manzanas utilizando visión artificial y redes neuronales artificiales. *Ingeniería y Región*, 1(20), 52-57. doi: 10.25054/22161325.1917

Ravichandiran S. (2019). Hands-On Reinforcement Learning with Python. Birmingham, U.K.: Packt. 1(1), 156-160, 1260618.

Planche B., Andres E. (2019). Hands-On Computer Vision with Tenseroflow 2. Birmingham, U.K.: Packt. 1(1), 68-69, 2290819.

Bonaccorso G. (2019). Mastering Machine Learning Algorithms. Birmingham, U.K.: Packt. 1(1) ,1240518.

Antona Cortés, C. (2017). Herramientas modernas en redes neuronales: la librería Keras (Bachelor's thesis). Escuela Politécnica Superior. Universidad Autónoma de Madrid. 1(1), 1-60.

Rashcka S., Mirjalili V. (2019). Python Machine Learning Second Edition. Birmingham, U.K.: Packt. 2(4), 449-450.

Giacomantone, J., Violini, M. L., Lorenti, L., & De Giusti, A. E. (2016). Supresión de segundo plano en imágenes de tiempo de vuelo. In XXII Congreso Argentino de Ciencias de la Computación (CACIC). 1(1), 1064-1073.

Moreno, R. J., Avilés, O., & Ovalle, D. M. (2018). Red neuronal convolucional para discriminar herramientas en robótica asistencial. *Revista Visión electrónica Universidad Distrital Francisco José de Caldas*, 12(2). 208-214, doi: 10.14483/22484728.13996

Cruz, J. F., Velázquez, P. L., Andrade, M. A. G., Silva, S. G. D. L. C., & Garcia E. A. R. (2017). A classifier system using smoothgraph coloring. *Revista de Matemática: Teoría y Aplicaciones*, 24(1), 129-156. doi: 10.15517/RMTA.V24I1.27795

Troyano, J. A., Cruz, F., González, M., Vallejo, C. G., & Toro, M. (2018). Introducción a la Programación con Python, Computación Interactiva y Aprendizaje Significativo. *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática*, 3(1), 223-230.

Campos, A. C., & Fernández-Bozal, J. (2005). La imagen digital. *Revista Española Ortodoncia*, 35(3), 255-266.

Bottazzi, C., Costarelli, S., D'Elia, J., Dalcin, L., Galizzi, D., Olivera, J., ... & Storti, M. (2017). Algoritmos y Estructuras de Datos. Info-FICH sistema de información. Santa Fe. Argentina. 1(1), 1-306.

Liu Y. (2019). Python Machine Learning By Example Second Edition. Birmingham, U.K.: Packt. 2(1), 40-41, 1270219

Quiroz, P., Muñoz, R., & Noël, R. (2012). Desarrollo de un lenguaje de programación y entorno de desarrollo que facilite la programación de robots LEGO mindstorms. In *XVII Congreso Internacional de Informática Educativa*, 1(1), 50-53.

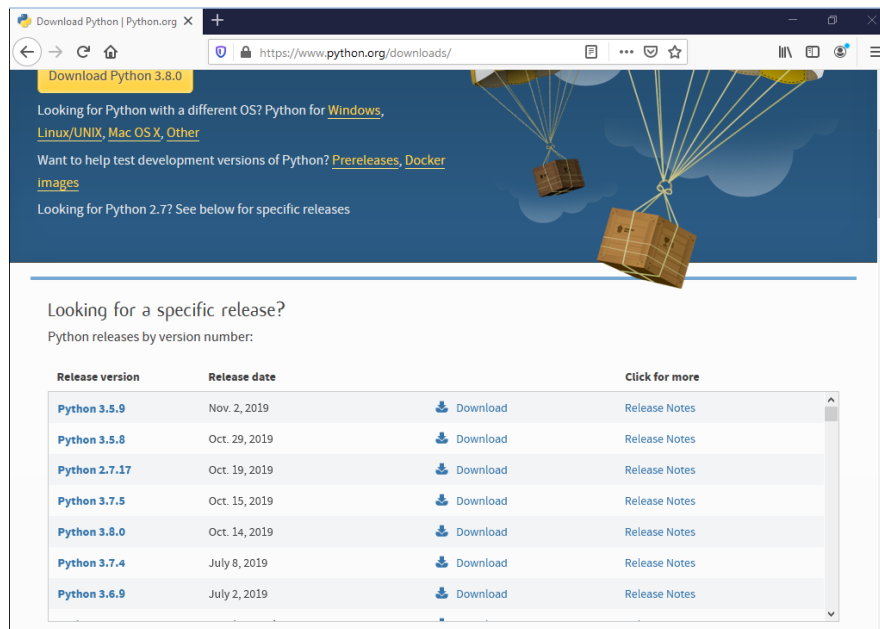
Jiménez Bravo, R. (2018). Sistema de seguimiento de objetos usando OpenCv, ArUco y Filtro de Kalman extendido. (Trabajo Fin de Grado Inédito). Universidad de Sevilla, Sevilla. 1(1), 1-83.

ANEXOS

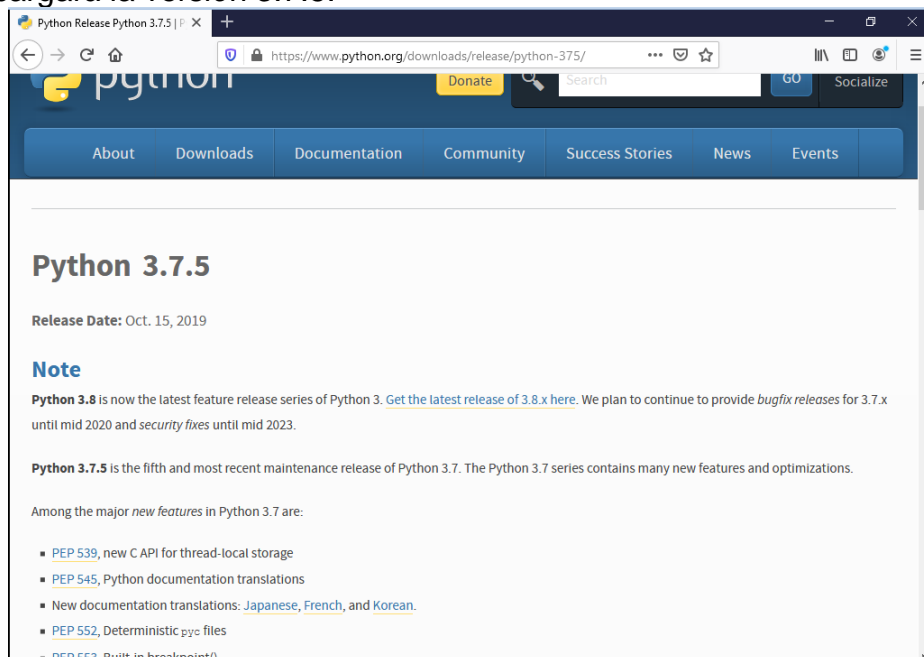
Manual de Usuario

Instalación de Python

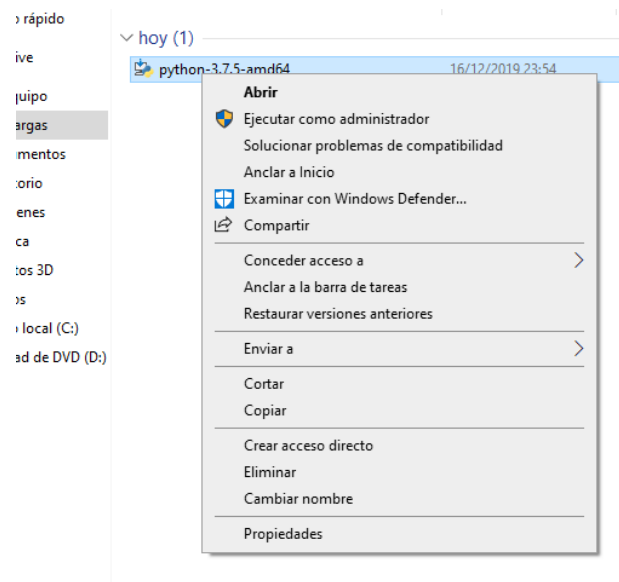
1) Como primer paso se debe ir a la sección de descargas de la página oficial de Python.



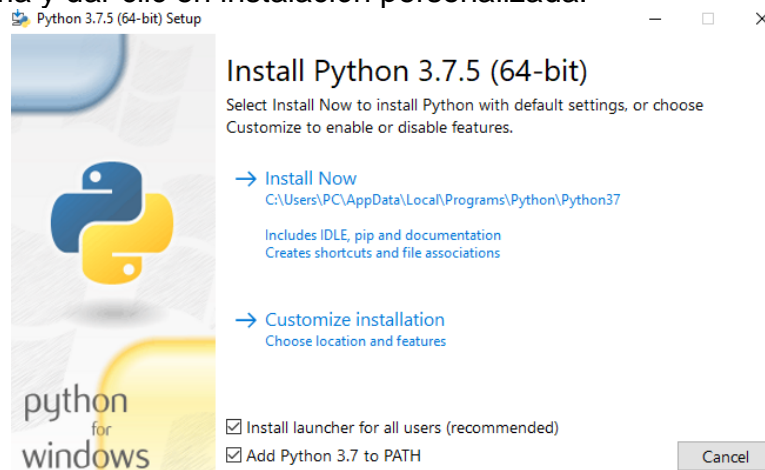
2) Se selecciona la versión a utilizar en el proyecto, para este caso se descargará la versión 3.7.5.



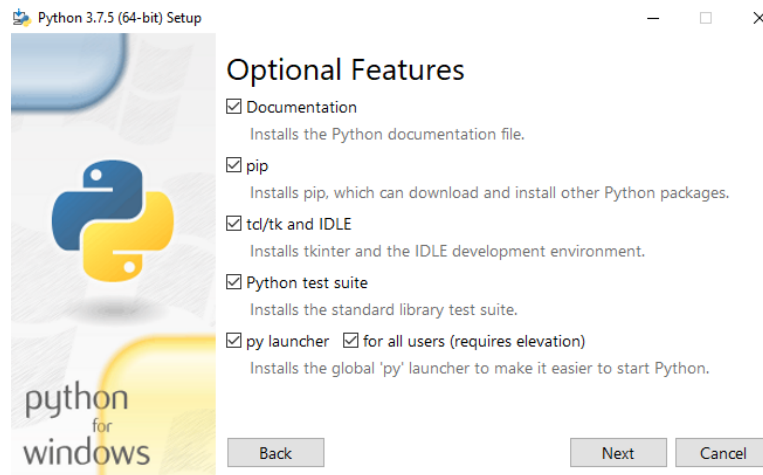
3) Se ejecuta el archivo de instalación como administrador .



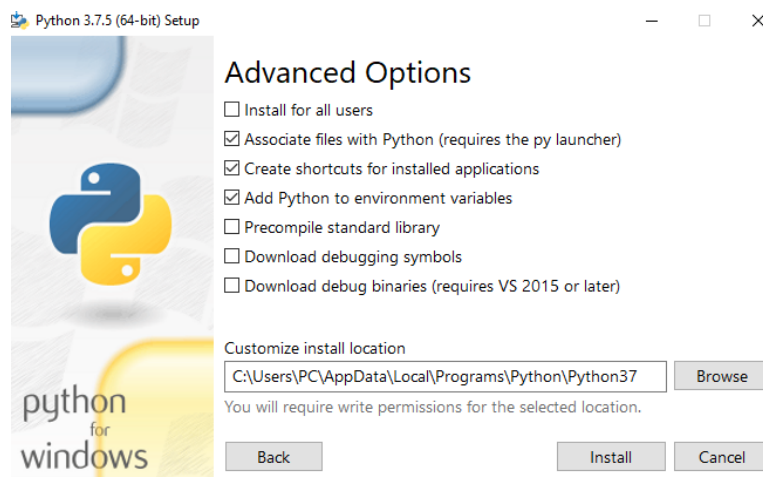
4) Se verifica que esté seleccionada la casilla PATH para añadir la ruta del sistema y dar clic en instalación personalizada.



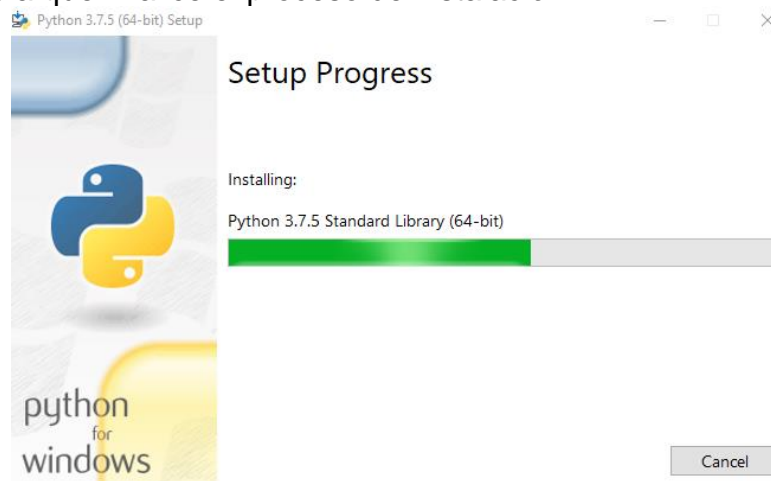
5) Verificar que la siguiente ventana tenga marcadas todas las opciones y continuar.



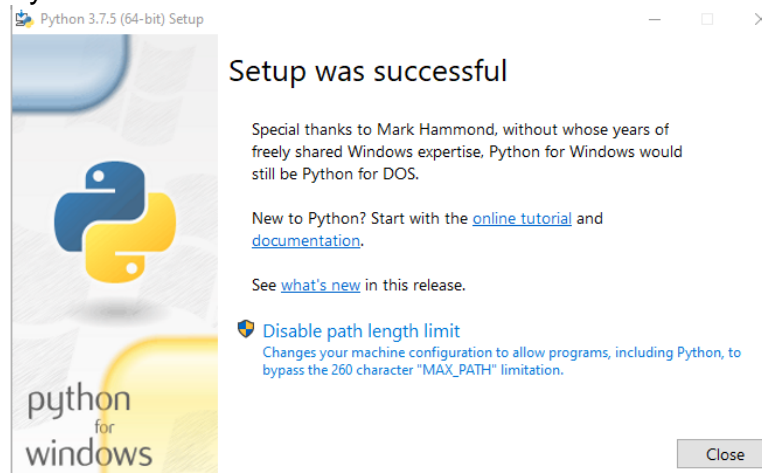
6) Seleccione la ruta de instalación, para este proyecto se dejó la ruta por defecto.



7) Espere a que finalice el proceso de instalación.



8) Al final la instalación aparecerá un mensaje, dar clic en Disable path length limit y cerrar.



9) Ir a CMD y escribir Python para verificar que esté instalado

```
Símbolo del sistema - python
Microsoft Windows [Versión 10.0.18363.592]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

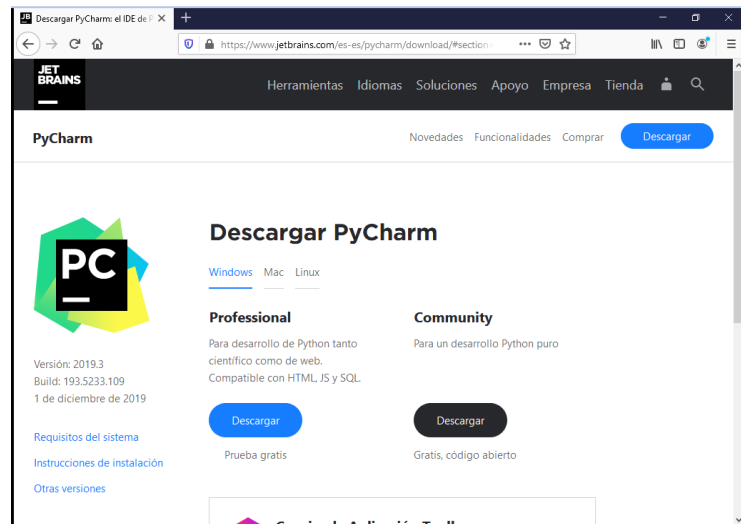
C:\Users\dafz>python
Python 3.7.5 (tags/v3.7.5:5c02a39a0b, Oct 15 2019, 00:11:34) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Instalación de PyCharm

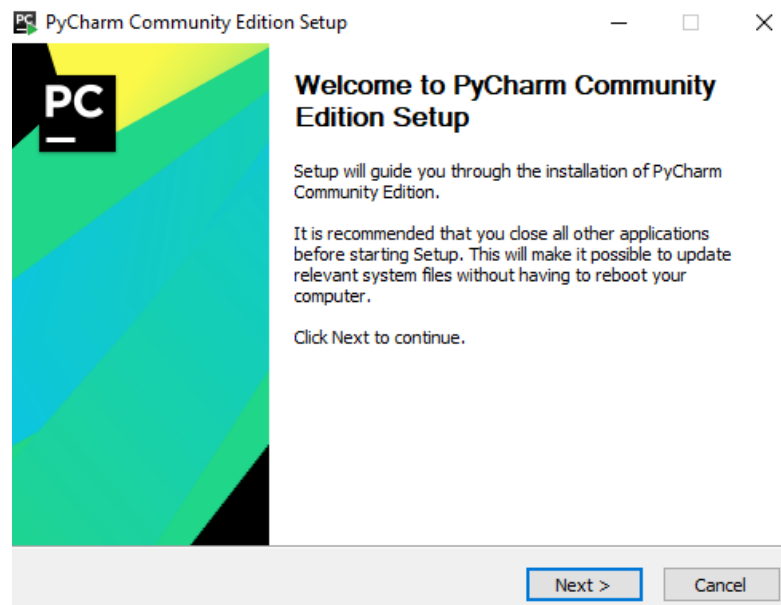
1) Para el proceso de instalación necesitaremos el driver de IDE de Pycharm, para ello, se debe ir a la oficial de Pycharm para descargarlo



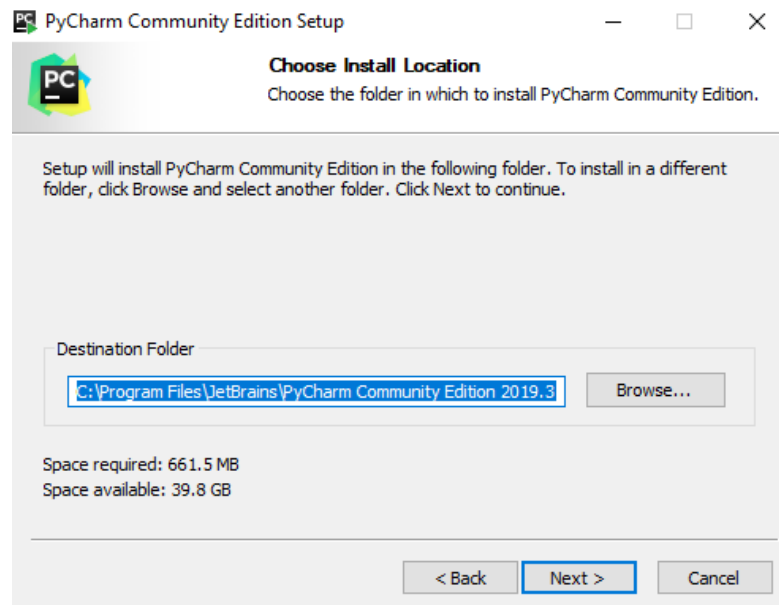
- 2) Hay 2 versiones, para este proyecto se ha seleccionado la versión Community.



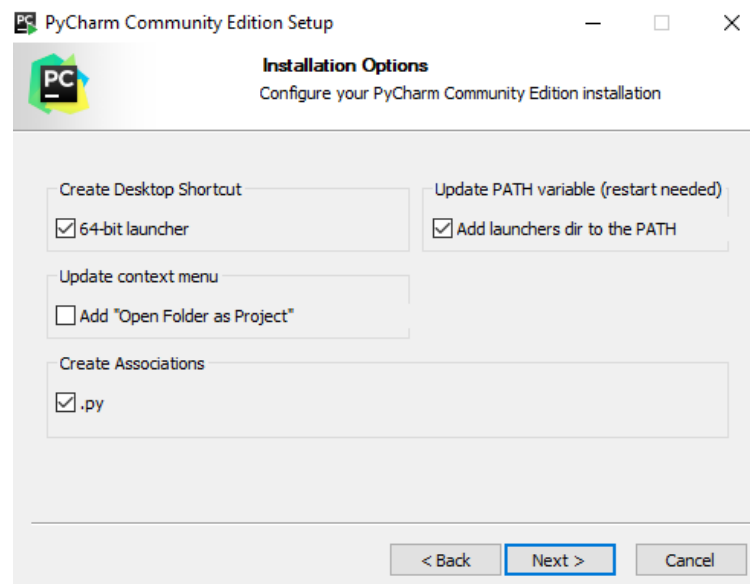
- 3) Una vez descargado, se debe ejecutar la aplicación .exe para comenzar con el proceso de instalación.
- 4) Aparecerá una ventana del proceso de instalación de Pycharm la cual se debe dar click en Next



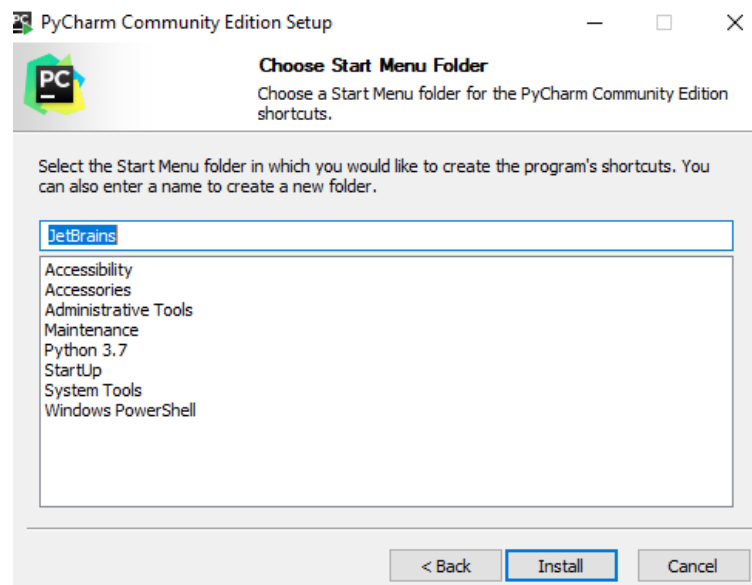
- 5) Se indicará el directorio donde se desea instalar Pycharm, para este proyecto se ha dejado la ruta por defecto, posteriormente se da click en Next.



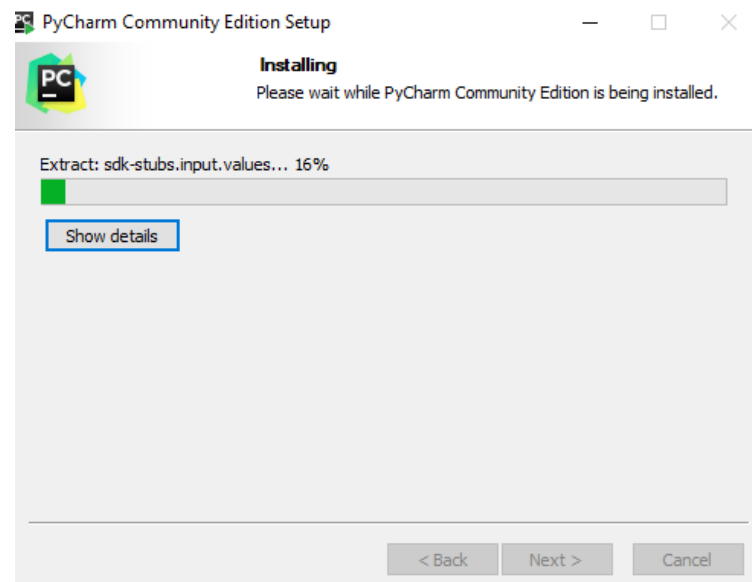
- 6) Se debe activar la casilla .py para asociar la extensión con PyCharm y Add launcher to Path que añadirá PyCharm como variable de entorno.



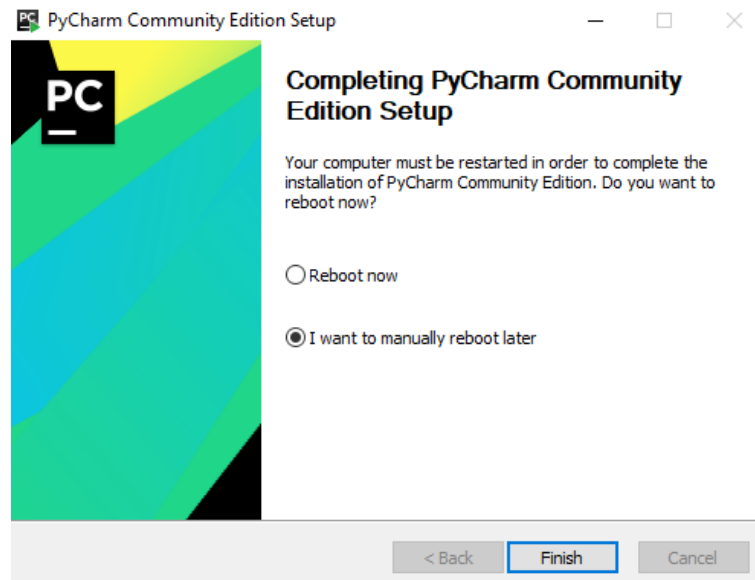
7) Dar clic en Install para que comience la instalación.



8) Esperar a que se complete la instalación.



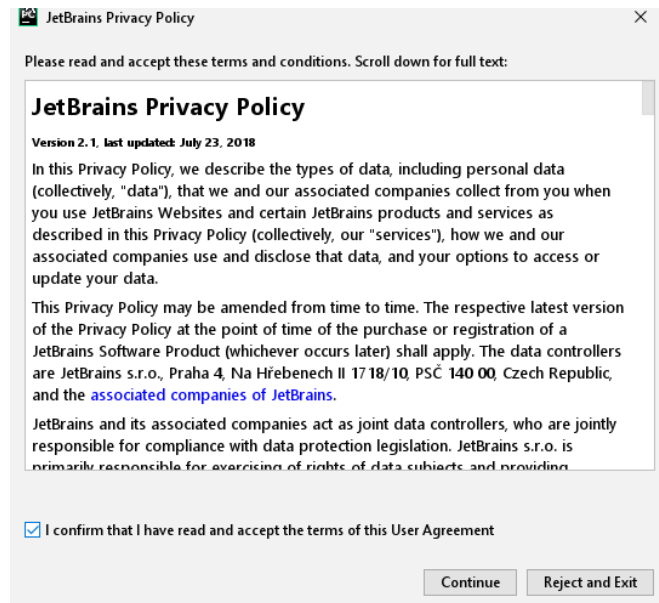
- 9) Seleccionar Reboot now y dar clic en Finish para finalizar la instalación y se reinicie el ordenador.



- 10) Una vez reiniciado en ordenador, abrir PyCharm para realizar la configuración inicial.



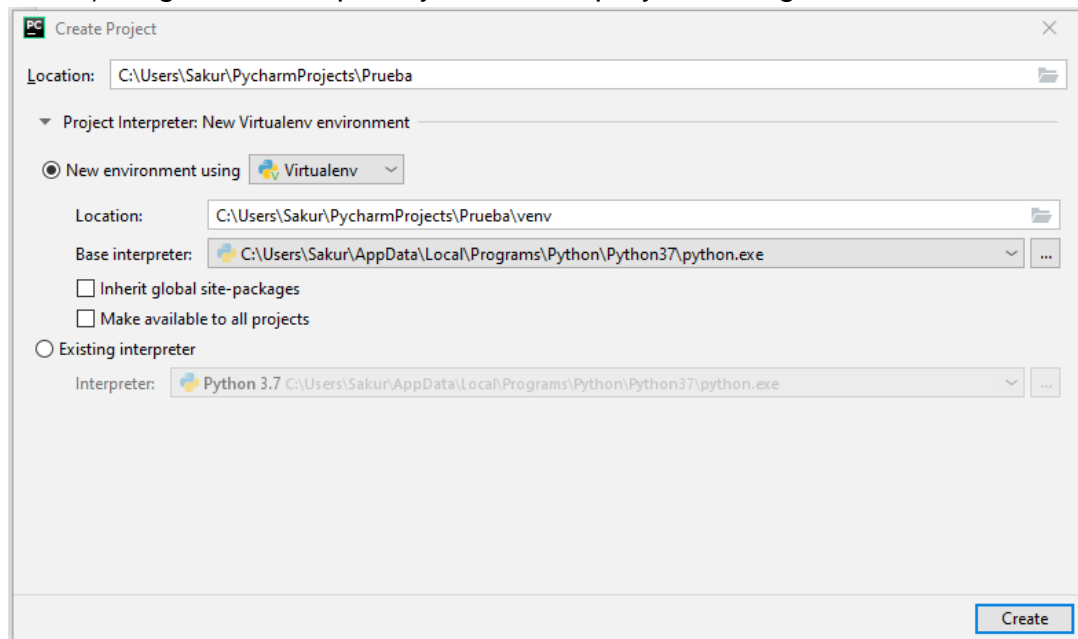
11) Aceptar los términos y condiciones



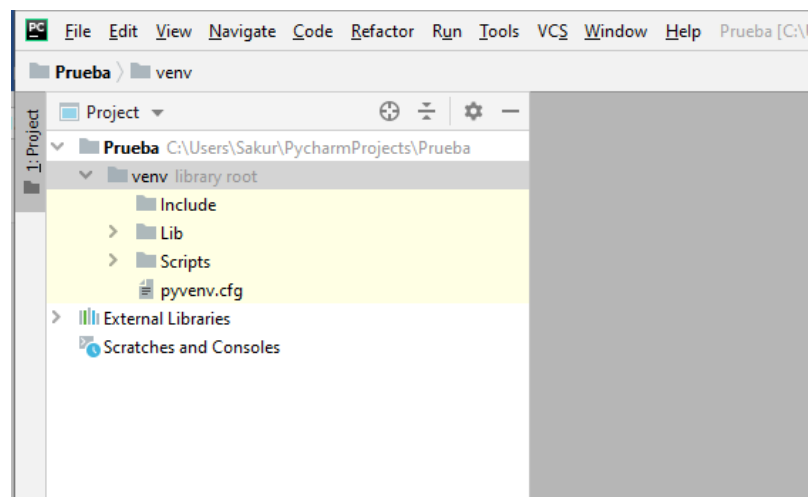
12) Clic en crear un nuevo proyecto.



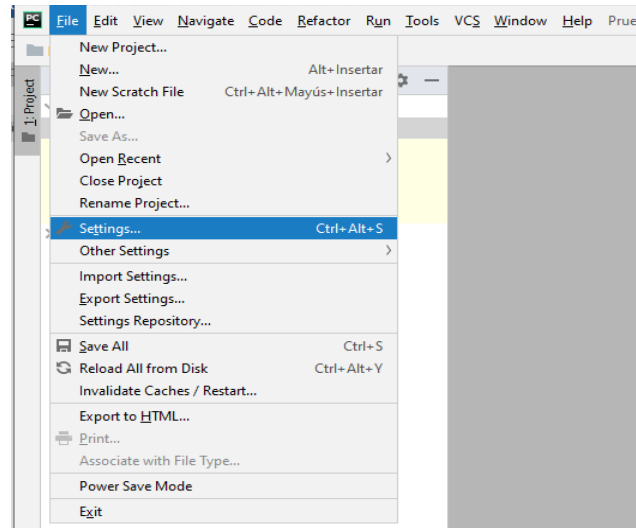
13) Asignar un intérprete y nombre al proyecto, luego dar clic a Create.



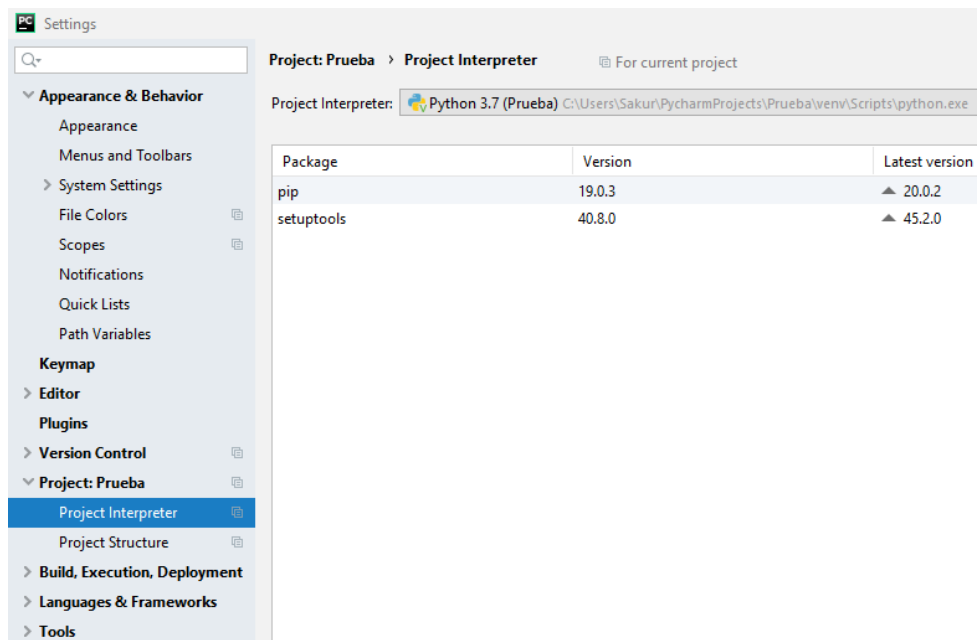
14) Instalar librerías, se pueden instalar por comandos cmd o utilizando PyCharm.



15) Dar clic en Archivos y luego en Configuración



16) Dar clic en Proyecto, aparecerán dos opciones, elegir interprete del proyecto, se mostrará un listado de los paquetes que se encuentran instalados, en esta misma ventana se puede instalar nuevos paquetes haciendo clic en el signo de suma.

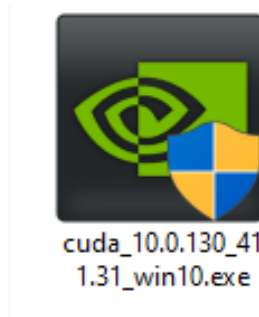


En la tabla mostrada a continuación se listan las librerías, paquetes usados para el desarrollo del proyecto

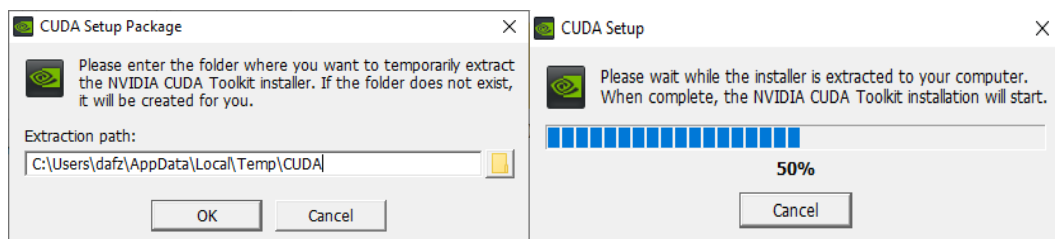
Paquete	Versión
Clic	7.0
Cython	0.29.14
Kera	2.31
Keras-Application	1.0.8
Keras-Preprocessing	1.1.0
Markdown	3.1.1
Pillow	7.0.0
PyQt5	5.13.0
PyQt5-sip	12.7.1
PyQt5-stub	5.13.1.4
PyQtDesigner	5.10.1
PyYAML	5.3
Werkzeug	0.16.0
absi-py	0.9.0
astor	0.8.1
atomicwrites	1.3.0
cachetools	4.0.0
chardet	3.0.4
colorama	0.43
cycler	0.10.0
darkflow	1.0.0
gast	0.3.2
google-image-download	2.8.0
google-pasta	0.1.8
grpcio	1.26.0
h5py	2.10.0
importlib-metadata	1.4.0
joblib	0.14.1
kiwisolver	1.1.0
lxml	4.4.2
matplotlib	3.1.2
more-itertools	8.1.0
numpy	1.18.1
oauthlib	3.1.0
opencv-contrib-python	4.1.2.30
opencv-python	4.2.0
opt-einsum	3.1.0
Pip	20.0.1
protobuf	3.11.2
py	1.8.1
pyparsing	2.4.6
pyqt5-tools	5.13.0.1.5
python-dateutil	2.8.1
python-dotenv	0.10.5
scikit-learn	0.22.1
scipy	1.4.1
selenium	3.141.0
setuptools	41.2.0
six	1.13.0
tensorboard	1.14.0
tensorflow	1.14.0
tensorflow-estimator	1.14.0
tensorflow-gpu	1.14.0
termcolor	1.1.0
urllib3	1.25.8
wheel	0.33.6
wrapt	1.11.2
zipp	1.0.0

Instalación de NVIDIA CUDA

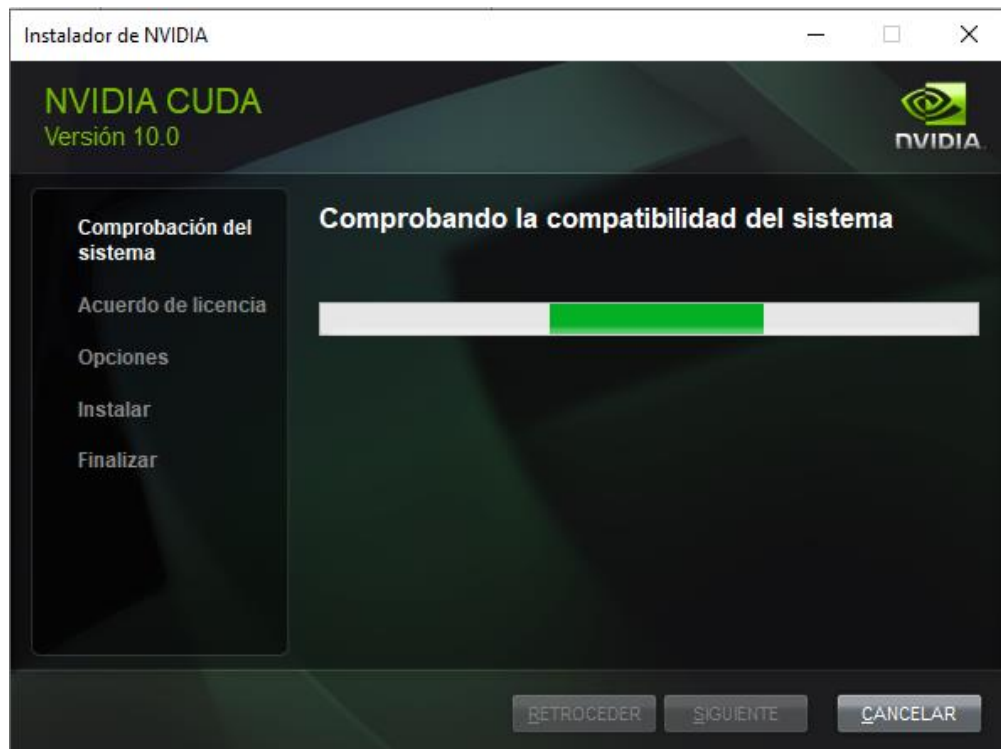
Como se ha mencionado anteriormente, Tensorflow permite usar el GPU para procesamiento en caso de que se disponga, el hardware utilizado en los ordenadores posee una NVIDIA GTX 1060 6GB, pero para que esta tarjeta gráfica sea reconocida para el procesamiento es necesario instalar CUDA que se encuentra en la página de NVIDIA para los desarrolladores, la versión utilizada en el proyecto es la 10.0.130.



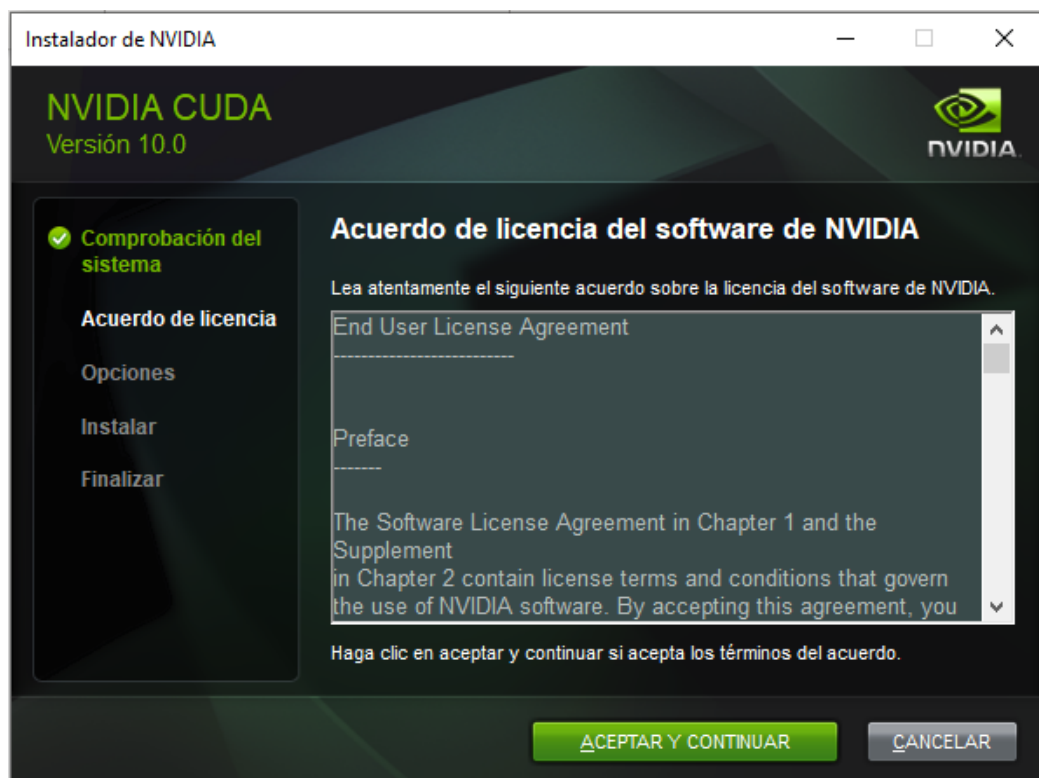
Al instalar CUDA se creará una carpeta temporal con un aviso similar al del gráfico 47 con los archivos necesarios para la instalación, que luego se borrará automáticamente.



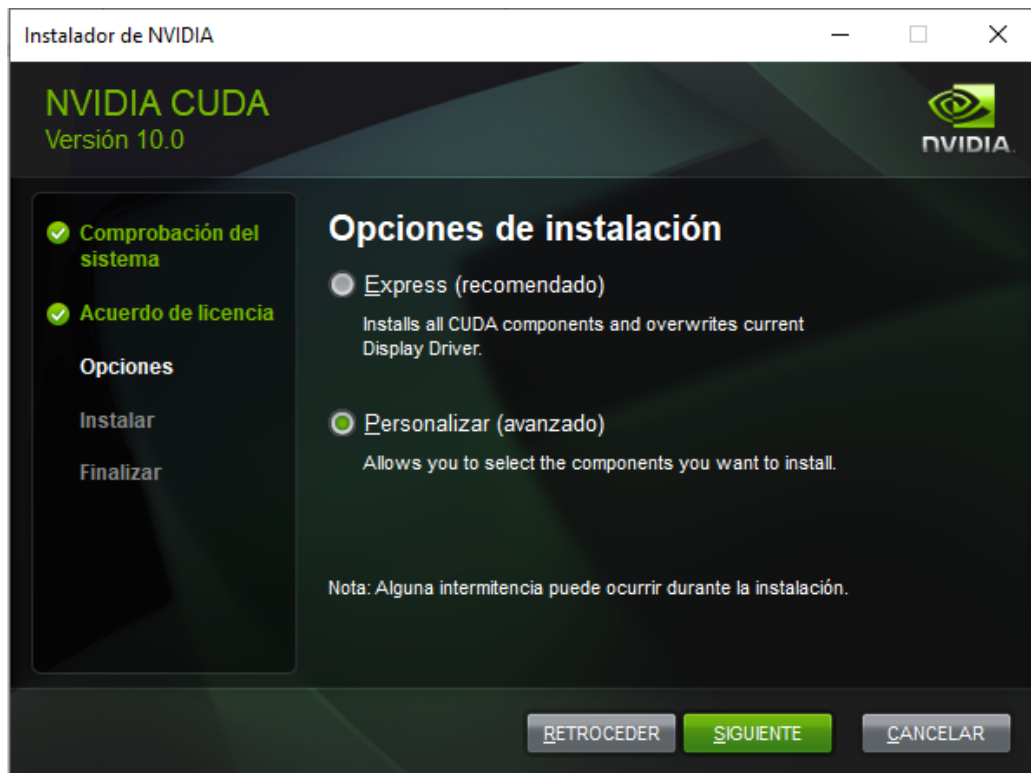
Luego de preparar la carpeta temporal se abrirá el instalador de CUDA que comprobará si el ordenador cumple con los requisitos como se muestra en el siguiente gráfico.



En caso de cumplir con los requisitos se debe aceptar el acuerdo de licencia para continuar con la instalación.



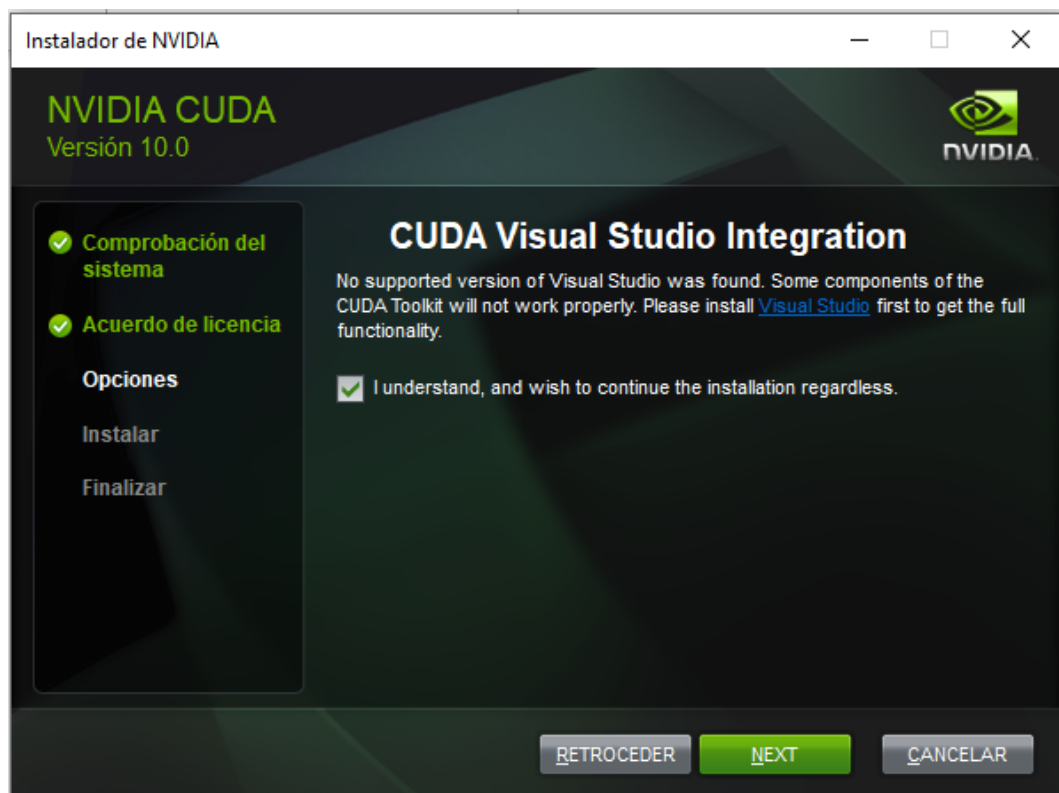
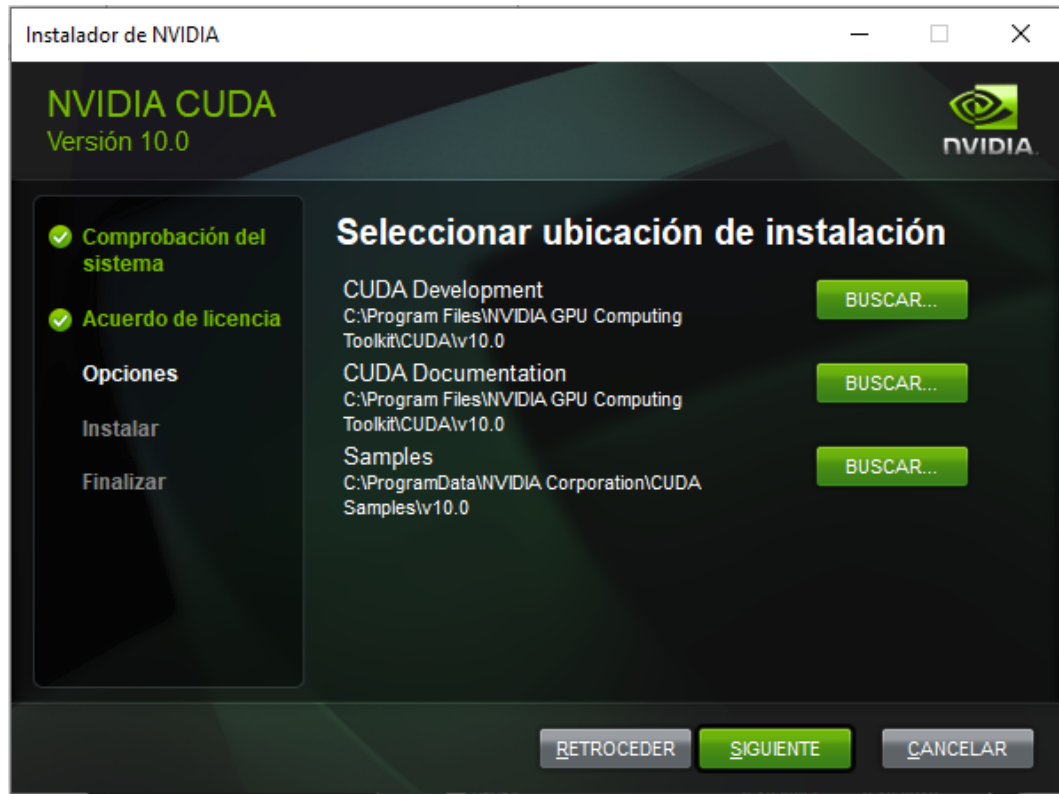
En opciones de instalación se seleccionó personalizar debido a que el ordenador ya poseía el Display Driver.



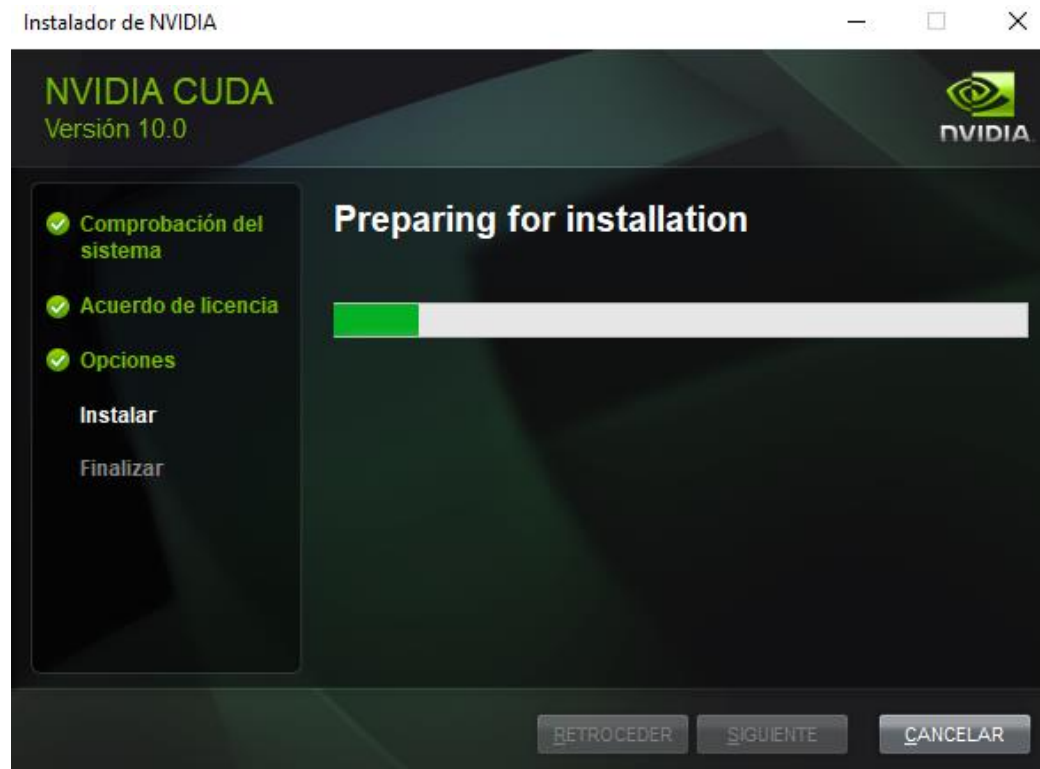
Al poseer los otros drivers, solo se dejó seleccionada la opción de CUDA



En cuanto a la ubicación de la instalación se dejó por defecto las rutas.



Comenzará la instalación, cuyo tiempo se verá afectado dependiendo de las especificaciones del ordenador.



FICHA DE JUICIOS DE EXPERTOS

DATOS GENERALES

APELLIDOS Y NOMBRES DEL EXPERTO	CARGO O INSTITUCIÓN	NOMBRE DEL INSTRUMENTO	AUTOR DEL INSTRUMENTO
Botto Tobar Miguel	Docente en la Universidad de Guayaquil	Clasificador de aceitunas de mesa	Milton Toalombo, Daniel Fajardo
TÍTULO	Desarrollo de un clasificador de aceitunas de mesa por su variedad		

ASPECTOS DE VALIDACIÓN

INDICADOR	CRITERIO	DEFICIENTE 0-20				REGULAR 21-40				BUENA 41- 60				MUY BUENA 61- 80				EXCELENTE 81 - 100			
		5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
CLARIDAD	Están formulados con lenguaje apropiado que facilita su comprensión.																X				
OBJETIVIDAD	Están expresados en conductas observables, medibles															X					
ACTUALIDAD	Esta acorde a los aportes recientes en la disciplina de estudio																X				
SUFICIENCIA	Son suficientes la cantidad y calidad de ítems presentados en el instrumento																X				
INTENCIONALIDAD	Es adecuado para valorar la variable seleccionada																X				
CONSISTENCIA	Está basado en aspectos teóricos y científicos																	X			
METODOLOGÍA	El instrumento se relaciona con el método planteado en el proyecto																	X			
APLICABILIDAD	El instrumento es de fácil aplicación																	X			

CONSTANCIA DE JUICIO DE EXPERTO

Nombre del experto: Ing. Miguel Botto Tobar, MSC.

Por medio de la presente, hago constancia que se realizó la revisión de la ejecución del proyecto por medio de un prototipo para el reconocimiento de los patrones a través del lenguaje de programación Python, realizado por los Sres. Daniel Arturo Fajardo Zambrano y Milton Franklin Toalombo Goyes estudiantes egresados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, cuyo trabajo de titulación es: **“Desarrollo de un clasificador de aceitunas por su variedad”**.

Una vez indicadas las correcciones pertinentes del caso considero que dicho proyecto es válido para su aplicación.

Ing. Miguel Botto Tobar, MSC.
C.I. 1204824328

Desarrollo de un Clasificador de Aceitunas de Mesa por su Variedad

Daniel Fajardo Zambrano¹, Milton Toalombo Goyes¹

¹ Facultad Ciencias, Matemáticas y Física, Universidad de Guayaquil, Victor Manuel Rendón y Baquerizo Moreno, Guayaquil, Ecuador
{daniel.fajardoz, milton.toalombog}@ug.edu.ec

Resumen. El consumo de aceitunas se ha incrementado debido a su utilización en el área de la gastronomía, el éxito de su producción se origina en países europeos como; España, Italia, Francia, entre otros, extendiéndose a países latinoamericanos como Argentina y Uruguay. Existe una amplia variedad de aceitunas, siendo el sujeto de estudio de esta investigación las aceitunas de mesa las cuales son verdes, cambiantes y negras, su calidad depende del nivel de madurez al momento de su recolección. Ante tal situación el presente trabajo tiene como objetivo el desarrollo de un clasificador de variedades de aceitunas de mesa tomando en consideración las características de forma, color y textura para que al ser procesadas ofrezcan una mejor calidad en la preparación de platos gourmet.

Palabras Clave: Extracción de Características, Procesamiento Digital de Imágenes, Reconocimiento de Patrones, Segmentación de Imágenes, Sistema de Visión Artificial.

1 Introducción

En España, una de las actividades agrícolas de mayor relevancia es el cultivo de olivo, producido principalmente en la cuenca mediterránea, su laboreo se ha extendido en todo el mundo y durante los últimos años es practicada en otros continentes. Dicha expansión se da por la popularidad y el aumento del consumo de productos que derivan por el cultivo siendo estas las aceitunas de mesa y el aceite de oliva. [1]

Las aceitunas de mesa han ganado terreno en el ámbito gastronómico recientemente debido a que estos han generado un contento en los consumidores al utilizarlo como herramienta en el arte culinario para la elaboración de platos gourmet, para la preparación las personas lo tienden a endulzar lavando repetidamente, y realizando inmersiones de salmuera, para reducir en la medida de lo posible el característico sabor amargo que posee la fruta; Sin embargo, otras tienden aliñar añadiendo diferentes especias como puede ser limón, naranja, laurel, entre otros, para darle un sabor adicional.

Las técnicas en la recolección de aceitunas han cambiado con el pasar de los años, inicialmente se recogían directamente del árbol con la mano, esta técnica se la conoce como el ordeño, el vareo es una técnica realizada por hombres el cual utilizaban una vara larga con la que golpeaban las ramas de olivo hasta que caían sobre unas mantas de lona para después ser limpiadas, recoger la aceituna del suelo era practicado por las mujeres y se usaban pequeñas cestas. Las aceitunas eran transportadas hasta la almazara con la ayuda de animales. En la actualidad, se utilizan máquinas agrícolas que agilizan el trabajo como el uso de tractores con remolques para el transporte, el vibrador es un brazo que se acopla a las ramas de olivos y las sacude, la barredora es un vehículo pequeño que absorbe y separa las aceitunas de ramas y piedras, la sopladora sopla la aceituna derribada del árbol y el peine el cual usa una vara mecánica para golpear aceitunas hasta que caigan al suelo. [2]

De acuerdo con el estado de madurez del fruto al momento de su recolección, las aceitunas de mesa las podremos clasificar de la siguiente manera: verde, cambiante y negra. Las aceitunas verdes son frutos recogidos en su estado de madurez idóneo, llegando alcanzar su tamaño normal. Las aceitunas de color cambiante son aquellas recogidas antes de su estado de madurez ideal, debido a esto presentan colores como rosa vinoso, castaño o rosáceo. Las aceitunas negras son aquellas recogidas en plena estado de madurez, presenta variación de color como negro rojizo, negro verdoso o violeta oscuro. Este tipo de frutas poseen un valor nutricional elevado, dado que tiene grandes cantidades ácidos grasos monoinsaturados y fibras, además que aporta vitaminas A y C.

Aguilera *et al* [3] proponen un sistema automático de clasificación de lotes de aceitunas utilizando visión por computador con el objetivo de diferenciar las aceitunas según nivel de calidad separando las aceitunas recogidas del suelo y del árbol, utilizando una metodología que contenga histogramas de imágenes obtenidas en diferentes espacios de color y como clasificador una red neuronal artificial ANN obteniendo un coeficiente de éxito de 97.1%.

La construcción de una báscula electrónica que permite clasificar diferentes frutas en base al tamaño, forma y color, utilizando modelo RGB y el clasificador Bayesiano, logrando obtener resultados con 90% de exactitud para la modernización en los supermercados. [4]

En [5] proponen el desarrollo de un sistema de visión por computador para detectar el estado de madurez de los frutos de guayaba aplicando el índice de color y el espacio RGB, CIELAB, YCbCr y HSV, llevando a cabo un análisis por componentes principales (PCA) para reducir datos redundantes, que generaba datos irrelevantes, logrando obtener resultados con un 90% de precisión.

[6] plantean el desarrollo de dos algoritmos de procesamiento de imágenes, uno para clasificar el café según su estado de maduración utilizando redes neuronales artificiales y el otro para detectar broca desarrollado mediante criterio de binarización para encontrar zonas negras en la imagen similares a los orificios que deja esta plaga, el clasificador tuvo un 97% de efectividad detectando estados de madurez.

[7] presentan un sistema que permita clasificar frutos de café en ocho categorías según el estado de maduración usando técnicas de visión artificial, a partir de nueve características en las que se incluye forma, medidas de color y textura del análisis de la imagen del fruto, el desempeño del clasificador bayesiano corresponde a un 96.88% de éxito.

Este proyecto utilizará los paquetes de Python tensorflow y keras junto con el algoritmo YOLO (You Only Look Once), que se han vuelto muy populares para la detección de objetos. El dataset consta de 2800 imágenes, donde cada clase tiene 1400 imágenes aproximadamente,

2 Materiales y Métodos

Se utilizó la metodología Procesamiento Digital de Imágenes (PDI) englobando diversas etapas, desde la captura de imágenes tomadas con la cámara trasera del celular en diferentes ángulos y distancias a través de un escenario preliminar para la generación del dataset, hasta la generación del modelo para el reconocimiento del objeto. Motivo por el cual, se toma en consideración ciertos aspectos como el dataset de imágenes, obtención de las características a partir de este y la elección de un clasificador. La metodología PDI aplicada consta de las etapas de captura, preprocesamiento, segmentación, extracción de características y clasificador las cuales se han mencionado en profundidad en el capítulo dos.

Se ha seleccionado las RNC para el desarrollo del clasificador en base a los resultados que han obtenido otros autores al momento de ser comparado con otros, como puede ser el Bayesiano o Vecinos más próximos presentando una pequeña variación en el porcentaje precisión, donde RNC llega a obtener una precisión mayor debido a que la cantidad de procesos para llevar a cabo para la clasificación, los cuales disminuyen empleando librerías recientes, como tensorflow, keras entre otros.

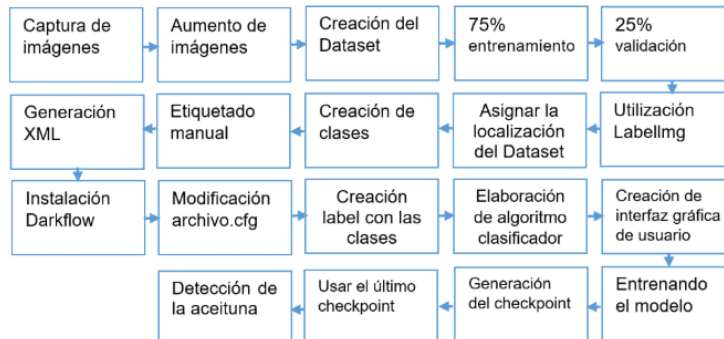


Figura 2. 1 Diagrama de Bloques de la metodología

Captura

Los escenarios de captura se fabricaron con hojas A4 simulando una forma cuadrada, además de usar lámparas con luz blanca para obtener la iluminación en las fotos. Por ello se emplean dos Smartphone que son utilizados para realizar las tomas desde diferentes ángulos y distancias, posteriormente aplicar diferentes técnicas para generar aumentos de imágenes que se mencionan en la siguiente etapa, llegando a obtener más de 1000 imágenes aproximadamente para cada variedad.



Figura 2. 2 Escenarios para la toma de imágenes

Pre-Procesamiento

A través de consultas realizadas a expertos mencionan que el dataset debe tener al menos 100 imágenes, caso contrario será impreciso el clasificador, el 75% de estas serán usadas para el entrenamiento con la finalidad de obtener un mayor reconocimiento al momento de la detección, y el 25% restante para validación, motivo por el cual se procede a utilizar ciertos comandos que permiten aplicar filtros como la rotación, reducción de ruido, aumento y disminución de brillo, entre otras a las imágenes previamente obtenidas para guardarlas como nuevas.

Algoritmo 1 Permite aplicar filtros como la rotación, reducción de ruido, aumento y disminución de brillo, entre otras a las imágenes previamente obtenidas para guardarlas como nuevas, esto se puede lograr mediante el siguiente algoritmo.

```
Datagen = ImageDataGenerator(
    rescale = 1./255,
    rotation_range = 30,
    width_shift_range = 0.2,
    height_shift_range = 0.2,
    shear_range = 0.2,
    zoom_range = 0.2,
    horizontal_flip = True,
    fill_mode = 'nearest'.
    Brightness_range = [0.2,0.8]
)
```

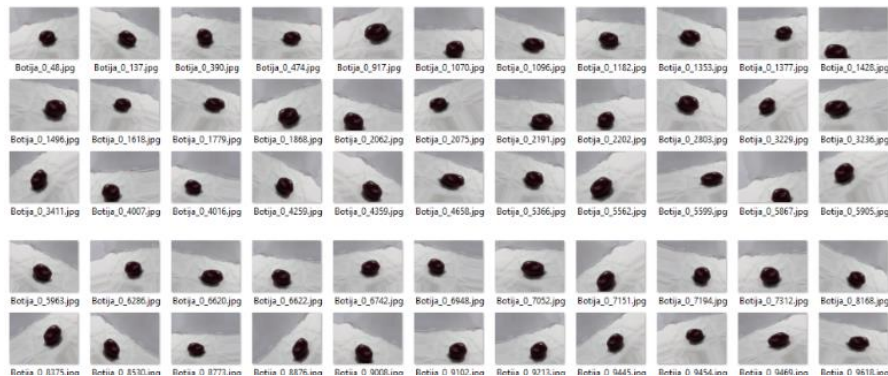


Figura 2. 3 Imágenes generadas para el aumento de datos

Segmentación

El etiquetado en formato xml permite referenciar objetos marcados en la imagen, guardando información del tamaño de resolución de esta, además de la ruta y ubicación de los objetos en el plano X Y. Para la detección de objetos resulta fundamental el etiquetado, el cual requiere mayor tiempo debido a que se realiza de manera manual, para el proyecto aproximadamente cuatro horas fueron necesarias para realizar el etiquetado de cada aceituna dando origen a que se genere archivos xml por cada imagen, la cual se usará para generar el modelo.

Una vez obtenido el dataset de imágenes de las variedades de las aceitunas, se procede a realizar la etiquetación manual de cada una de ellas, utilizando la herramienta llamada labellmg ubicada en el repositorio de github

<https://github.com/tzutalin/labelImg>, en ella se explica cómo realizar la instalación mediante líneas de comando de los sistemas operativos como Windows, Linux, macOS, además que permite realizar etiquetados generando archivos de extensión xml con el comando Python labelImg.py.

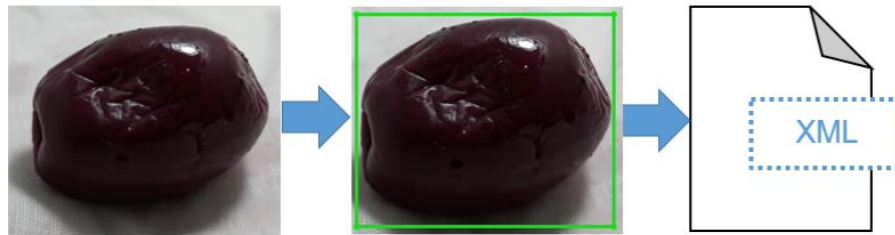


Figura 2. 4. Etiquetado de imagen

Antes de comenzar a realizar el etiquetado con LabelImg, lo primero que se debe hacer es indicar la carpeta donde está almacenado el dataset de las imágenes para que esta herramienta reconozca las imágenes y posteriormente indicar en que carpeta desea almacenar los archivos xml generados una vez que haya sido etiquetada la imagen, al momento de etiquetar una aceituna en concreto de la imagen debe escribir el nombre de la clase.

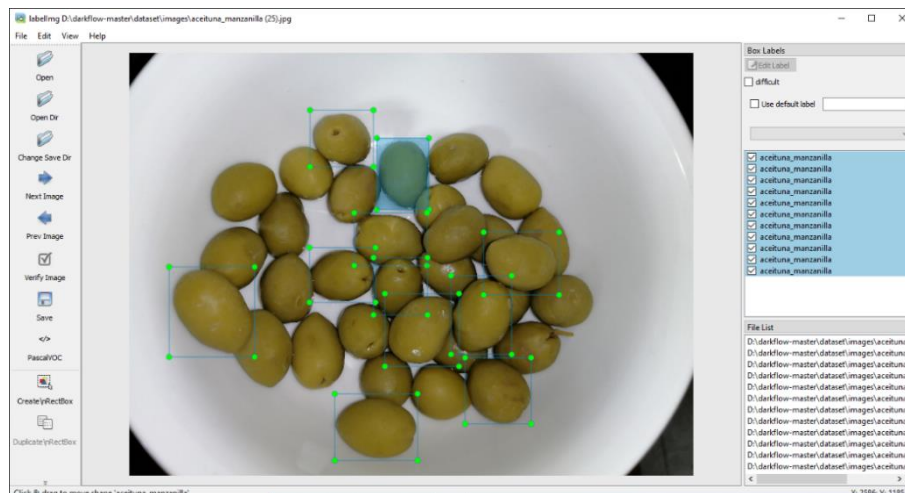


Figura 2. 5 Etiquetado manual de la aceituna manzanilla

Extracción de Características

Las características usadas para la detección y reconocimiento de las variedades de las aceitunas son las siguientes:

- El color es una de las principales características para detectar las variedades existentes de las aceitunas, existen variedades donde el color del fruto maduro no es verde y negra, sino que puede ser amarillo o marrón, para la variedad botija y manzanilla utilizada en este trabajo, el color del epicardio varía en la gamma del

negro y verde oscuro para las primeras etapas respectivamente, y cuando están maduros para el momento de su recolección el color tiende a llegar a ser morado o violeta y verde claro. Para la caracterización del color fueron utilizados dos modelos de color las cuales son el RGB y HSI, cada uno de estos modelos describe al color aplicando la media de componente.

- La forma es otra propiedad importante al momento de detectarlas; desde la obtención de las aceitunas se presentaba con una textura firme, robusta, además de la forma ovalada como una elipse casi perfecta que se mantenía en un buen estado debido a que estaba envasado. Sin embargo, durante los transcurso de los días estas cualidades fueron variando como la textura con pequeños rasgos de arrugas, se tornaron blandas y la forma ovalada que poseía, ahora era achatada.

- En cuanto al tamaño puede variar según la variedad, en las primeras etapas del fruto es pequeño y va aumentando ligeramente hasta llegar a la plena madurez. Cuando la maduración ha llegado a la última fase, se decliva causando que el tamaño disminuya a medida que el fruto se va secando, en cuanto a las aceitunas de mesa procesadas, no maduran, pero al ser dejadas al ambiente durante unas cuantas horas se comienzan a secar y disminuye su tamaño.



Figura 2. 6 Variación en la forma y tamaño

Clasificador

En este trabajo se utiliza dos variedades de aceitunas, botija y manzanilla, cada una corresponde a una clase, la cantidad de clases se utiliza la configuración de tiny yolo ubicada /cfg/ tiny-yolo.cfg para entrenamiento del modelo, esta permite agilizar el tiempo de entrenamiento, debido a la cantidad de clases a utilizar, esta configuración permite realizar un entrenamiento hasta con 80 clases.

Los cambios realizados en la configuración de tiny yolo, que corresponden al pre-procesado de los datos de entrada, el archivo de configuración además contiene información sobre las capas de convoluciones, la utilización de maxpool para la reducción de datos innecesarios que para este proyecto no fueron modificadas.

Algoritmo 2 Configuración de YOLO para la generación del modelo.

```

batch=64
subdivisions=8
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1
learning_rate=0.001
max_batches = 40100
policy=steps
steps=-1,100,20000,30000
scales=.1,10,.1,.1
[convolutional]
size=1
stride=1
pad=1
filters=35
activation=linear
[region]
anchors = 1.08,1.19, 3.42,4.41, 6.63,11.38,
9.42,5.11, 16.62,10.52
bias_match=1
classes=2
coords=4
num=5
softmax=1
jitter=.2
rescore=1

```

Al tener menos clases se ve afectada la variable de filtros mostrada en la tabla, por ello se modifica la cantidad de clases que se van a utilizar para generar el modelo, y se aplica una fórmula matemática mostrada al final del párrafo para el filtro. Si este último cambio no se efectúa, ocurrirá algún error al realizar el entrenamiento.

$$5 (class + 5) \quad (1)$$

Donde:

class = cantidad de clases usadas para generar el modelo

Algoritmo 3 Se procede a iniciar con el entrenamiento con la línea de comandos, para ello se debe dirigir a la carpeta de darflow y ejecutar lo siguiente:

```
python flow --model cfg/tiny-yolo-voc-bot-man.cfg --load bin/tiny-yolo-voc.weights --train --annotation dataset/annotations --dataset dataset/images --gpu 0.7 --epoch 300
```

Para el trabajo se ha utilizado un ordenador que posee una tarjeta gráfica Nvidia GeForce GTX 1060 6GB, que permite realizar un entrenamiento más rápido debido al poder computacional que posee, dependiendo del modelo el valor asignado al GPU puede variar. En el caso que no se posea una GPU, bastaría con eliminarla del comando y el entrenamiento se realizaría usando el CPU, con esto el tiempo de entrenamiento aumentará.

3 Resultados

Cada checkpoint se generó al pasar aproximadamente una hora y treinta minutos, para llegar a las 300 épocas se debe dejar el ordenador encendido, entrenando el modelo por lo menos una semana y media. En la tabla 4.1 se visualiza los puntos de control generados en la carpeta ckpt desde la primera época.


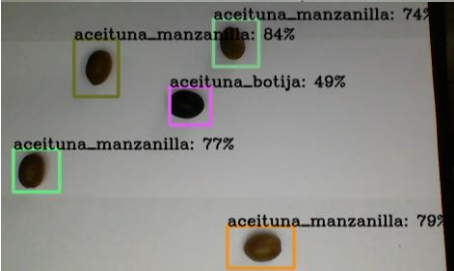

Punto de Control	Validación
1125	 <p> aceituna_botija: 45% aceituna_botija: 49% aceituna_manzanilla: 64% aceituna_manzanilla: 46% </p>
3000	 <p> aceituna_manzanilla: 74% aceituna_manzanilla: 84% aceituna_botija: 49% aceituna_manzanilla: 77% </p>
8500	 <p> aceituna_manzanilla: 79% aceituna_manzanilla: 86% aceituna_manzanilla: 82% aceituna_manzanilla: 89% aceituna_manzanilla: 91% aceituna_manzanilla: 93% </p>

Tabla 4. 1 Visualización de los puntos de controles generados

El último checkpoint generado es el que se utilizará para la detección de aceitunas junto con `tiny_yolo.cfg`, siendo estos necesarios al momento de ejecutar futuras detecciones, la cual puede realizarse mediante imágenes o a través de una webcam, para luego de ser procesada, mostrar en un cuadro la ubicación de la aceituna indicando si su variedad es botija o manzanilla. Cabe recalcar que a ciertas distancias no reconocerá la aceituna, esto puede solucionarse incluyendo un dataset que simule estas distancias.

4 Conclusiones

La cantidad de imágenes que posee el dataset es fundamental para evitar resultados imprecisos, por lo que se usó un dataset de 1000 imágenes aproximadamente por variedad, cada una con su respectiva etiqueta generada con la herramienta LabelImg.

La generación de los checkpoints durante el entrenamiento ayudó a realizar diferentes pruebas en las que se corroboró que la precisión aumentaba a medida que este continuaba, después de varios días de entrenamiento, se tomó el checkpoint donde el margen de error era menor.

Se concluye que la utilización de técnicas que permiten generar cantidades de datos para el aumento del dataset es próspero ya que ayuda en obtener resultados favorables.

Luego de un entrenamiento de 36 horas para la generación del modelo se completaron 25 épocas, permitiendo que el reconocimiento sea más preciso, aunque tiende a tener ciertas falencias como es el caso de la detección a una distancia considerable, o muy cercana.

Referencias

- [1] Ponce, J. M., Aquino, A., Segura, F., Millán, B., & Andújar, J. M. “*Estimación automatizada del peso y calibre de aceitunas mediante análisis de imagen*”. *Actas de las XXXIX Jornadas de Automática, Badajoz*, Vol. 1, No. 1, pp. 958 -966. (2018)
- [2] Ferguson, L. “*Tendencias actuales en la recolección mecánica de las aceitunas*”. *Grasas y aceites*, Vol. 57, No.1, pp. 9-15. doi: 10.3989/gya.2008.v59.i3.509. (2006)
- [3] Aguilera, D. A., Moreno, O. C., Gila, D. M., García, J. G., & Ortega, J. G. “*Clasificación automática de lotes de aceitunas en almazara mediante visión por computador*”. *Actas de las XXXVI Jornadas de Automática, Bilbao*, Vol. 1, No. 1, pp. 59-65. (2015)
- [4] Montoya Holguin, C., Cortés Osorio, J. A., & Chaves Osorio, J. A. “*Sistema automático de reconocimiento de frutas basado en visión por computador*”. *Ingeniare. Revista chilena de ingeniería*, Vol. 22, No. 4, pp. 504-516. doi: 10.4067/S0718-33052014000400006. (2014)
- [5] Bonilla-González, J. P., & Prieto-Ortiz, F. A. “*Determinación del estado de maduración de frutos de feijoa mediante un sistema de visión por computador utilizando información de color*”. *Revista de Investigación, Desarrollo e Innovación*, Vol. 7, No. 1, pp. 111-126. doi: 10.19053/20278306.v7.n1.2016.5603. (2016)
- [6] Martínez Llano, G. E., Ortiz, M., Manuel, S., Herrera Pérez, J. C., Berrio Pérez, J. S., & Beleño Sáenz, K. D. J. “*Clasificación de los frutos de café según su estado de maduración y detección de la broca mediante técnicas de procesamiento de imágenes*”. *Revista Prospectiva*, Vol. 7, No 1, pp 15-22 doi: 10.15665/rp.v14i1.640. (2016)
- [7] Sandoval, Z., & Prieto, F. “*Procesamiento de imágenes para la clasificación de café cereza*”. *Revista Prospectiva*, Vol. 7, No. 1, pp. 67-73. (2009)