

UNIVERSIDAD PRIVADA ANTENOR ORREGO

ESCUELA DE POSTGRADO



**“ESPECTROSCOPIA CON INFRARROJO Y TECNICAS DE MACHINE
LEARNING Y DEEP LEARNING PARA LA DETECCIÓN Y CLASIFICACIÓN
DE FRUTAS PARA LA AGROINDUSTRIA. CASO: ARÁNDANOS -
EMPRESA TalSA -2018 ”**

TESIS

**PARA OBTENER EL GRADO ACADÉMICO DE MAESTRO
EN INGENIERÍA DE SOFTWARE**

AUTOR:

Br. WALTER AURELIO LAZO AGUIRRE

ASESOR:

Dr. TEOBALDO HERNÁN SAGASTEGUI CHIGNE

Fecha de sustentación: 2019-07-05

Trujillo, 2019

ACREDITACIÓN

El **Dr. Teobaldo Hernán Sagástegui Chigne**, que suscribe, asesor de la Tesis con Título **“ESPECTROSCOPIA CON INFRARROJO Y TECNICAS DE MACHINE LEARNING Y DEEP LEARNING PARA LA DETECCIÓN Y CLASIFICACIÓN DE FRUTAS PARA LA AGROINDUSTRIA. CASO: ARÁNDANOS - EMPRESA TalSA -2018”**, desarrollado por el Br. Walter Aurelio Lazo Aguirre, acredita haber realizado las observaciones y recomendaciones pertinentes, encontrándose expedita para su revisión por parte de los señores miembros del Jurado Evaluador.

Trujillo, 01 de Julio del 2019.

El Asesor:

Dr. Teobaldo Hernán Sagástegui Chigne

El Autor:

Br. Walter Aurelio Lazo Aguirre

PRESENTACIÓN

Señores Miembros del Jurado:

Cumpliendo con los requerimientos estipulados en el reglamento de Grados y Títulos de la Escuela de Postgrado de la Universidad Privada Antenor Orrego, para obtener el grado de Maestro en Ingeniería de Software, pongo a vuestra disposición la presente tesis titulada: **ESPECTROSCOPIA CON INFRARROJO Y TECNICAS DE MACHINE LEARNING Y DEEP LEARNING PARA LA DETECCIÓN Y CLASIFICACIÓN DE FRUTAS PARA LA AGROINDUSTRIA. CASO: ARÁNDANOS - EMPRESA TalSA -2018**

Gracias.

Trujillo, 01 de julio del 2019

Br. Walter Aurelio Lazo Aguirre

Dedicado:

A Dios y la Virgen,
les debo todo lo que he logrado
en la vida.

A mi hija Lucero Melissa Elizabeth
mi preciado tesoro.

AGRADECIMIENTOS

Un agradecimiento muy especial a mi gran amigo y asesor Teobaldo Hernán Sagástegui Chigne por su valioso apoyo que me permite culminar con éxito este trabajo de investigación, y por sus orientaciones y consejos que demuestra el gran ser humano que es.

RESUMEN

ESPECTROSCOPIA CON INFRARROJO Y TECNICAS DE MACHINE LEARNING Y DEEP LEARNING PARA LA DETECCIÓN Y CLASIFICACIÓN DE FRUTAS PARA LA AGROINDUSTRIA. CASO: ARÁNDANOS - EMPRESA TalSA -2018

Las empresas comercializadoras de frutas, tienen la necesidad de optimizar la selección y clasificación de las frutas que comercializan, específicamente aquellos productos que se enviarán al extranjero donde se exige altos índices de calidad.

Necesitan asegurar que el proceso selección y clasificación de frutas se realice con precisión para obtener un producto de alta calidad que satisfaga las exigencias de los clientes. Existen equipos que pueden realizar este trabajo de clasificación, pero son muy costosos para su adquisición al igual que su mantenimiento.

Por lo expuesto anteriormente, la presente tesis desarrolla una propuesta basada en la aplicación de espectroscopia con infrarrojo cercano, Machine Learning y Deep Learning para la detección y clasificación de frutas para la agroindustria. Específicamente, se toma como caso de estudio, la clasificación de arándanos en la empresa TalSA. con la finalidad de establecer una herramienta alternativa para poder hacer la clasificación de arándanos, que permita reducir el costo, el tiempo y optimizar el proceso de detección y clasificación de Frutas.

Para lograr lo antes indicado, se revisó el estado del arte de machine learning, deep learning, y deep neural networks, se revisó el uso de espectroscopia en el reconocimiento de frutas y sus propiedades, se analizó el proceso selección y clasificación de arándanos en la empresa TalSA. Luego, utilizando un equipo de espectroscopia infrarrojo cercano, NIR, se obtuvo el espectro de las muestras de arándanos. Con estos espectros, digitalizados, se realizó el entrenamiento de una red neuronal utilizando lenguaje de programación Python y la plataforma de Keras con TensorFlow. Luego, de realizado el entrenamiento, utilizó la red neuronal para realizar la etapa de testing, con los espectros NIR de nuevas muestras de arándanos, encontrándose que se pueden clasificar los arándanos con una exactitud del 92%, lo que permite confirmar la hipótesis de trabajo.

Por el Br. Walter Aurelio Lazo Aguirre

Palabras clave:

Machine learning, Deep learning, Deep neural networks, Espectroscopia, NIR, FTIR, Reconocimiento de frutas, Arándanos, Red neuronal

ABSTRACT

INFRARED SPECTROSCOPY AND MACHINE LEARNING TECHNIQUES AND DEEP LEARNING FOR THE DETECTION AND CLASSIFICATION OF FRUITS FOR AGRIBUSINESS. CASE: ARANDANS - COMPANY TalSA -2018

The companies marketing fruit, have the need to optimize the selection and classification of the fruits they market, specifically those products that will be sent abroad where high quality indexes are required.

They need to ensure that the fruit selection and sorting process is carried out with precision to obtain a high quality product that meets the demands of the customers. There are teams that can perform this classification work, but they are very expensive for their acquisition as well as their maintenance.

For the above, this thesis develops a proposal based on the application of near infrared spectroscopy, Machine Learning and Deep Learning for the detection and classification of fruits for the agroindustry. Specifically, the classification of blueberries in TalSA is taken as a case study in order to establish an alternative tool to make the classification of blueberries, which allows to reduce the cost, time and optimize the process of detection and classification of Fruits.

To achieve the above, the state of the art of machine learning, deep learning, and deep neural networks was reviewed, the use of spectroscopy in the recognition of fruits and their properties was reviewed, the process of selection and classification of blueberries in the TalSA Company was analyzed. Then, using a near infrared spectroscopy equipment, NIR, the spectrum of the blueberry samples was obtained. With these spectra, digitized, the training of a neural network was carried out using the Python programming language and the Keras platform with TensorFlow. Then, after training, it was used the neural network to carry out the testing stage, with the NIR spectra of new blueberry samples, finding that blueberries can be classified with an accuracy of 92%, which confirms the working hypothesis .

By Br. Walter Aurelio Lazo Aguirre

Keywords:

Machine learning, Deep learning, Deep neural networks, Spectroscopy, NIR, FTIR, Fruit recognition, Cranberries, Neural network

ÍNDICE

ACREDITACIÓN	2
PRESENTACIÓN	3
AGRADECIMIENTOS	5
RESUMEN	6
ABSTRACT	7
ÍNDICE DE FIGURAS.....	10
CAPÍTULO I: INTRODUCCION.....	12
1.1. FOCALIZACIÓN DEL PROYECTO	12
1.2. EL PROBLEMA.....	13
1.2.1. REALIDAD PROBLEMÁTICA.....	13
1.2.2. ANÁLISIS DEL PROBLEMA.....	14
1.3. ENUNCIADO DEL PROBLEMA	15
1.4. ALCANCE	15
1.5. ANTECEDENTES	15
1.6. JUSTIFICACIÓN	16
1.7. VIABILIDAD	17
1.8. APORTE	17
1.9. HIPÓTESIS.....	18
1.9.1. VARIABLES.....	18
1.10. OBJETIVOS.....	18
1.10.1. OBJETIVO GENERAL.....	18
1.10.2. OBJETIVOS ESPECÍFICOS.....	18
CAPÍTULO II: MARCO TEÓRICO.....	20
2.1. ARÁNDANOS.....	20
2.1.1. BENEFICIOS PARA LA SALUD.....	20
2.1.2. DATOS DE EXPORTACIÓN DE ARÁNDANO EN EL PERÚ 2016.....	20
2.1.3. PRINCIPALES MERCADOS DE LOS ARÁNDANOS.....	21
2.2. ESPECTROSCOPIA POR INFRARROJO CERCANO.....	24
2.3. MACHINE LEARNING.....	25
2.4. DEEP LEARNING	25
2.5. RED NEURONAL	26
2.5.1. Ventajas de las redes neuronales.....	27
2.6. ELEMENTOS BÁSICOS QUE COMPONEN UNA RED NEURONAL.....	29
2.6.1. Función de entrada	29
2.6.2. Función de activación	31
2.6.3. Función de salida.....	33
2.7. NIVELES O CAPAS DE UNA RED NEURONAL	33
2.8. MECANISMOS DE APRENDIZAJE	34
2.8.1. Aprendizaje supervisado.....	35
2.8.2. Aprendizaje no supervisado.....	37
2.9. ELECCIÓN DEL CONJUNTO DE PESOS	37
2.10. VALIDACIÓN DE LA RED NEURONAL	38
2.11. TOPOLOGÍA DE LAS REDES NEURONALES	38
2.11.1. Redes monocapa	38
2.11.2. Redes multicapa.....	39
2.12. CONEXIÓN ENTRE NEURONAS	39
2.13. REDES BACKPROPAGATION – PROPAGACIÓN HACIA ATRÁS.....	39
2.14. REDES NEURONALES CONVOLUCIONALES	39

2.15.	EL PERCEPTRÓN	41
2.16.	APLICACIÓN DE LAS TÉCNICAS DE ESPECTROSCOPIA NIR Y MADUREZ DE FRUTAS.....	41
2.17.	PROCESAMIENTO DE IMÁGENES DE FRUTAS CON ESPECTROSCOPIA NIR.....	41
2.18.	TECNOLOGÍA DE ESPECTROSCOPIA INFRARROJO CERCANO NIR.....	42
2.19.	CLASIFICACIÓN DE FRUTAS UTILIZANDO DEEP NEURAL NETWORK.....	42
2.20.	KERAS	42
2.21.	TENSORFLOW.....	43
2.22.	KERAS-PANDAS.....	44
2.23.	DATASET	45
2.24.	SCIKIT-LEARN.....	45
CAPÍTULO III: MATERIAL Y MÉTODOS		47
3.1.	MATERIAL Y PROCEDIMIENTO:.....	47
3.1.1.	MATERIAL.....	47
3.1.2.	PROCEDIMIENTOS.....	47
3.1.2.1.	DISEÑO DE TECNICAS DE RECOLECCION DE LA INFORMACION	47
3.1.2.2.	TECNICAS E INSTRUMENTOS.....	48
3.1.3.	METODOLOGÍA	48
3.1.4.	OPERACIONALIZACIÓN DE VARIABLES	49
CAPÍTULO IV: RESULTADOS.....		51
4.1.	REVISIÓN DEL ESTADO DEL ARTE DE MACHINE LEARNING, DEEP LEARNING, Y DEEP NEURAL NETWORKS.	51
4.2.	EL USO DE ESPECTROSCOPIA EN EL RECONOCIMIENTO DE FRUTAS Y SUS PROPIEDADES.....	54
4.3.	ANÁLISIS EL PROCESO SELECCIÓN Y CLASIFICACIÓN DE ARÁNDANOS EN LA EMPRESA TALSA.....	56
4.4.	PROPUESTA DE UNA APLICACIÓN UTILIZANDO ESPECTROSCOPIA NIR Y DEEP LEARNING	57
4.5.	USO DE LA APLICACIÓN PROPUESTA EN LA DETECCIÓN Y CLASIFICACIÓN DE ARÁNDANOS DE LA EMPRESA TALSA...	69
4.6.	ANÁLISIS ESTADÍSTICO E INTERPRETACIÓN DE LOS RESULTADOS.....	70
CAPÍTULO V: DISCUSIÓN		72
CONCLUSIONES.....		78
RECOMENDACIONES.....		80
REFERENCIAS BIBLIOGRÁFICAS		81
ANEXOS		84

ÍNDICE DE FIGURAS

FIG. 1 : ÁMBITO DE LA ESPECIALIDAD DE INGENIERÍA DE SOFTWARE	12
FIG. 2 : PRINCIPALES MERCADOS DE ARÁNDANOS: ESTADOS UNIDOS, PAÍSES BAJOS, GRAN BRETAÑA, ESPAÑA, SINGAPUR, CANADÁ, CHINA	21
FIG. 3 : PRINCIPALES EMPRESAS EXPORTADORAS DE ARÁNDANOS EN LA LIBERTAD 2016	22
FIG. 4 : PROCESO ARÁNDANO FRESCO	22
FIG. 5 : LÍNEA DE PROCESO DE ARÁNDANO FRESCO EN TALSÁ	23
FIG. 6 : ACONDICIONAMIENTO Y SELECCIÓN DE ARÁNDANO EN LÍNEAS DE PROCESO	23
FIG. 7 : ESPECTRO INFRARROJO CERCANO.....	24
FIG. 8 : CÁMARAS TÉRMOGRÁFICAS, MULTIESPECTRALES E HIPERESPECTRALES CENTRADAS EN LA BANDA ESPECTRAL DE 0,7 A 1,7 MICRAS O INFRARROJO CERCANO (NIR, NEAR INFRARED).....	24
FIG. 9 : LOS MODELOS DE REDES NEURONALES EN DEEP LEARNING	26
FIG. 10 : EJEMPLO DE UNA RED NEURONAL TOTALMENTE CONECTADA	29
FIG. 11 : EJEMPLO DE UNA NEURONA CON 2 ENTRADAS Y 1 SALIDA.....	30
FIG. 12 : FUNCIÓN DE ACTIVACIÓN LINEAL.	31
FIG. 13 : FUNCIÓN DE ACTIVACIÓN SIGMOIDEA.....	32
FIG. 14 :FUNCIÓN DE ACTIVACIÓN TANGENTE HIPERBÓLICA.....	32
FIG. 15 : INFLUENCIA DE LA SALIDA DE LA NEURONA N_j EN LA ENTRADA DE LA NEURONA N_i	35
FIG. 16 :EQUIPO DE ESPECTROSCOPIA NIR - HEADWALL PHTONICS	57
FIG. 17 : EQUIPO DE ESPECTROSCOPIA NIR CONECTADO A LA COMPUTADORA	58
FIG. 18 : EQUIPO DE ESPECTROSCOPIA NIR Y LA COMPUTADORA EN FUNCIONAMIENTO.	58
FIG. 19 : PROCESO DE EMISIÓN DE LAS RADIACIONES Y CAPTURA DE ESPECTROS NIR	59
FIG. 20 : PROCESO DE CAPTURA DE ESPECTROS NIR Y DIGITALIZACIÓN	59
FIG. 21 : DATOS DE ESPECTROS DIGITALIZADOS, CORRESPONDIENTE A UN ARÁNDANO ..	60
FIG. 22 : DATOS OBTENIDOS DE ESPECTROS NIR DIGITALIZADOS	61
FIG. 23 : DATOS OBTENIDOS DE ESPECTROS NIR PARA ENTRENAMIENTO	63
FIG. 24 : ESQUEMA DE UNA RED NEURONAL	64
FIG. 25 : ALGUNAS FUNCIONES DE ACTIVACIÓN.....	65
FIG. 26 : EXACTITUD DE LAS MEDICIONES EN LA ETAPA DE ENTRENAMIENTO.....	73

CAPÍTULO I

INTRODUCCION

CAPÍTULO I: INTRODUCCION

1.1. FOCALIZACIÓN DEL PROYECTO

ÁMBITO DE LA DISCIPLINA DE INGENIERIA DE SOFTWARE

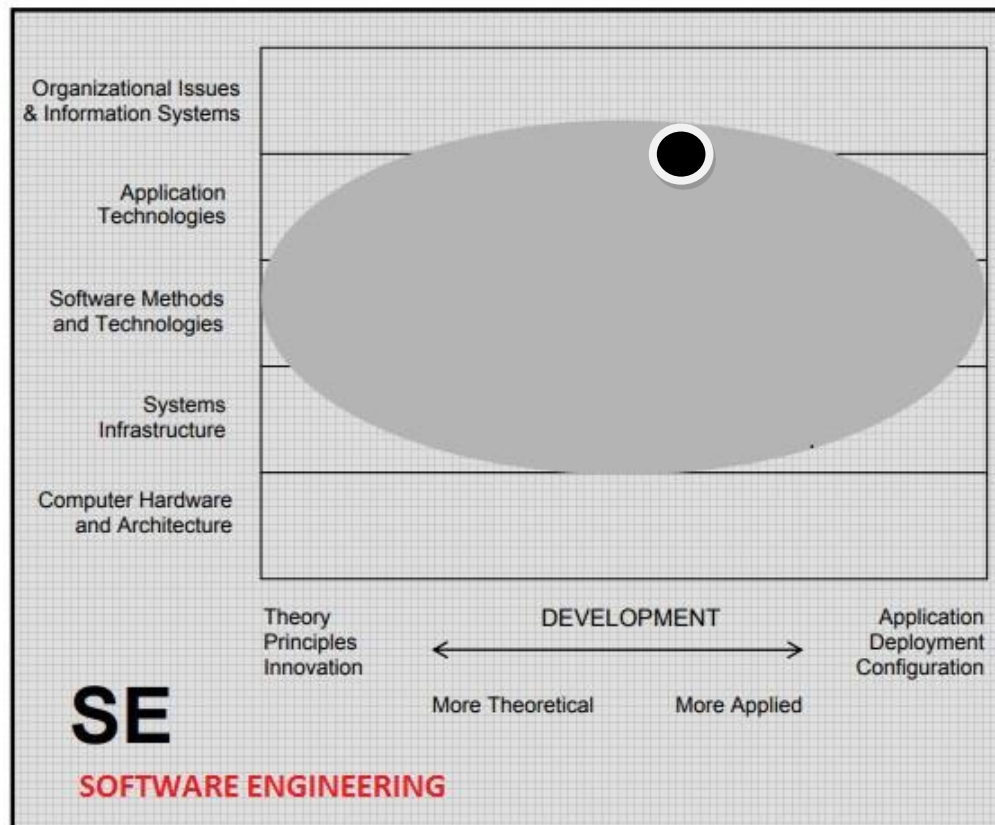


Fig. 1 : Ámbito de la especialidad de Ingeniería de Software

Fuente: (ACM, 2008)

Ingeniería de Software

Esta disciplina se ocupa del diseño e implementación de software complejo, con estándares de calidad, de una manera, confiable y eficiente.

Se basa en los conocimientos de las ciencias de la computación, ingeniería y matemáticas para diseñar, definir y organizar diversos aspectos de un producto software, siguiendo el ciclo de vida del software, incluyendo temas de costo del proceso de desarrollo, aplicando estándares de calidad, con amplio dominio de lenguajes de programación, y, orientado a la aplicación y uso de alta tecnología. (ACM, 2014)

1.2. EL PROBLEMA

1.2.1. Realidad Problemática

El arándano es una fruta cuyo consumo viene aumentado cada vez más debido su alta capacidad antioxidante y diferentes propiedades beneficiosas para la salud. Los Estados Unidos de América es el principal exportador de arándanos, seguido de Canadá.

En los últimos años, los países del hemisferio sur, han aumentado la exportación de arándanos al hemisferio norte aprovechando las diferencias estacionales en la producción. Pero debido a la larga distancia del envío transoceánico, se requiere que los arándanos frescos, en el país de origen cumplan con tener una alta calidad, para que puedan cumplir con los estándares de calidad a su llegada en el país destino (Leiva, 2013).

Tal S.A. es una empresa agroindustrial dedicada al cultivo de frutas, vegetales y a la elaboración de productos frescos y en conservas: espárragos blancos y verdes, arándanos y paltas, cumpliendo con los estándares de calidad definidas por sus clientes.

La empresa TalSA, es uno de los principales exportadores de arándanos frescos, del Perú. Tiene la necesidad de optimizar la selección y clasificación de las frutas que comercializa, específicamente aquellos productos que se enviarán al extranjero donde se exige altos índices de calidad.

El Grupo TalSA necesita asegurar que el proceso selección y clasificación de arándanos se realice con precisión para obtener un producto de alta calidad que satisfaga las exigencias de los clientes.

Existen equipos que pueden realizar este trabajo de clasificación, pero son muy costosos, sus precios están en el orden de 1'250,000 Euros y darle mantenimiento tiene también un alto costo

Para dar soporte al proceso anteriormente mencionado sobre la producción de arándano, se hace necesario abordar el problema de la detección y clasificación de arándanos frescos en la línea de procesos. Esto se realiza en la etapa de acondicionamiento de arándano en líneas de proceso

En tal sentido, este documento de investigación realiza una propuesta alternativa para la detección y clasificación de arándanos frescos en la línea de procesos.

1.2.2. Análisis del Problema

La clasificación de los arándanos en la empresa TalSA se realiza en forma manual. Desde que se cosecha en el campo hasta el envasado es un proceso manual, en el que la calidad de los frutos depende de la forma como se manipula y la capacidad que tienen los trabajadores para reconocer un fruto que es de calidad y diferenciarlo de un fruto que tiene defectos.

Uno de los principales problemas es que se requiere de un trabajador con experiencia en estas labores. Asimismo, este trabajador puede realizar sus labores de manera eficiente un determinado tiempo, luego del cual su eficiencia disminuye.

Los problemas encontrados en la etapa de acondicionamiento de arándano en las líneas de proceso, en la selección y clasificación de arándanos para su envasado y posterior comercialización, en la empresa TalSA son los siguientes:

- Empleo de muchos trabajadores en el proceso de selección y clasificación de los arándanos, lo que origina un sobre costo en mano de obra
- Se requiere que los trabajadores tengan experiencia para reconocer y diferenciar los frutos de calidad y los frutos defectuosos para poder realizar la clasificación.
- Un trabajador, aunque cuente con mucha experiencia, puede realizar sus labores de manera eficiente un determinado tiempo, luego del cual su eficiencia disminuye.
- Existen equipos que pueden realizar este trabajo de clasificación, pero son muy costosos, sus precios están en el orden de 1'250,000 Euros y darle mantenimiento tiene también un alto costo

Una propuesta de valor sería desarrollar una aplicación utilizando el procesamiento de imágenes por medio de espectroscopia infrarrojo cercano, Machine Learning y Deep Learning para la Detección y Clasificación de Arándanos en la empresa TalSA.

El procedimiento que se estudia permitirá identificar las características de los arándanos, de manera no invasiva, a fin de diferenciar frutos en buenas condiciones de aquellos que no lo están, permitiendo separarlos y clasificarlos

a fin de seleccionar los productos de buena calidad para su envasado y comercialización.

1.3. ENUNCIADO DEL PROBLEMA

“¿En qué medida el uso de espectroscopia con infrarrojo y técnicas de Machine Learning y Deep Learning permitirán realizar la detección y clasificación de arándanos en la empresa TalSA?”

1.4. ALCANCE

El alcance del proyecto corresponde a la etapa de acondicionamiento de arándano en las líneas de proceso, en el área de selección y clasificación de Arándanos en la empresa TalSA.

1.5. ANTECEDENTES

Las siguientes investigaciones forman parte de los antecedentes:

1.5.1 (Sladojevic et.al, 2016) en su trabajo “Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. Computational Intelligence and Neuroscience”, presentan los resultados obtenidos utilizando una nueva generación de redes neuronales convolucionales (CNN) para el reconocimiento de imágenes. El principal aporte al trabajo de investigación es un modelo de reconocimiento de enfermedades de las plantas, que se basa en el tratamiento de imágenes de las hojas, mediante el uso de redes convolucionales profundas (CNN).

1.5.2 (Zhang et.al, 2014), en su trabajo, “Fruit classification using computer vision and feedforward neural network” tienen por objetivo el identificar y reconocer frutas con precisión. Utilizan un método de clasificación híbrido basado en el algoritmo de la colonia artificial de abejas caótica (FSCABC) y la red neuronal de alimentación directa (FNN). Su principal aporte al trabajo de investigación es que los resultados experimentales demostraron que el FSCABC – FNN fueron precisos y eficientes para diferenciar y reconocer las frutas.

1.5.3 (Chang L. et.al. 2015), en su trabajo, “DeepFood: Deep Learning-based Food Image Recognition for Computer-aided Dietary Assessment”, tiene como objetivo proponer un método basado en el aprendizaje profundo para enfrentar el problema del reconocimiento de imágenes de alimentos. Su principal aporte es que establecen algoritmos basados en redes neuronales convolucionales

(CNN) con optimizaciones propuestas, como un modelo y una técnica de convolución optimizados.

1.5.4 (Qian Yu et.al, 2016), en su trabajo “Deep Learning Based Food recognition” presentan un método de visión por computadora que tiene por objetivo reconocer y evaluar de manera eficiente la composición de los alimentos.

El principal aporte al trabajo de investigación es que permite el reconocimiento de alimentos con un método basado en redes neuronales convolucionales (CNN).

1.5.5 (Nogales, J., 2016) en su trabajo de tesis doctoral, “Estudio del estado de madurez y la aptitud enológica en uva mediante análisis de imagen hiperespectral”, utiliza el análisis de imagen hiperespectral en la región del infrarrojo cercano para la caracterización de uvas y el desarrollo de diferentes metodologías que permitan evaluar su estado de maduración y aptitud enológica. Su principal aporte al trabajo de investigación es que ha comprobado que el análisis de imagen hiperespectral en el infrarrojo cercano permite diferenciar diferentes variedades de uva consiguiendo resultados similares a los obtenidos mediante procedimientos químicos.

1.6. JUSTIFICACIÓN

La investigación está orientada a desarrollar una mejora del proceso de detección y Clasificación de Frutas para la Agroindustria. Se utiliza como caso de estudio, la clasificación de arándanos en la empresa TalSA.

Justificación Técnica

Permite estudiar la utilización de tecnologías de espectroscopia con infrarrojo cercano y Deep Neural Networks en la Detección y Clasificación de Frutas para la Agroindustria. Caso: clasificación de arándanos en la empresa TalSA.

Justificación Social

Este proyecto de investigación permitirá reducir el costo, el tiempo de clasificación y optimizar el proceso de detección y clasificación de Frutas.

Justificación Académica

El desarrollo del proyecto permitirá Analizar de manera sistémica el estado del arte de Machine Learning y Deep Learning, y de tecnologías de espectroscopia, y plantear una propuesta de aplicación de tecnologías de espectroscopia con infrarrojo cercano

y Deep Neural Networks en la Detección y Clasificación de Frutas para la Agroindustria.

1.7. VIABILIDAD

Este proyecto se desarrolla tomando como premisa que la empresa TalSA, es uno de los principales exportadores de arándanos frescos, del Perú. Tiene la necesidad de optimizar la selección y clasificación de las frutas que comercializa, específicamente aquellos productos que se enviarán al extranjero donde se exige altos índices de calidad. La empresa necesita asegurar que el proceso de selección y clasificación de arándanos se realice con precisión para obtener un producto de alta calidad que satisfaga las exigencias de los clientes.

Para el caso de esta investigación se utilizarán arándanos, de la etapa de acondicionamiento, en las líneas de proceso; en el área de selección y clasificación de Arándanos en la empresa TalSA

1.8. APORTE

La investigación ha realizado los siguientes aportes:

a. Para con la Empresa:

- Tener una propuesta para la clasificación de arándanos como alternativa a la clasificación realizada manualmente y a los equipos que son altamente costosos

b. Para con la universidad:

- Realizar una investigación multidisciplinaria al hacer uso de espectroscopia infrarrojo cercano (NIR) aplicada a la Ingeniería de software.

c. Para con el Tesista:

- Aplicar los conocimientos recibidos durante el desarrollo de los estudios de la maestría en Ingeniería de software
- Usar el área de inteligencia artificial, específicamente, machine learning y deep learning.

1.9. HIPÓTESIS

H1: El uso de espectroscopia con infrarrojo y técnicas de Machine Learning y Deep Learning permitirán realizar la Detección y Clasificación de arándanos de forma automática en la empresa TalSA.

1.9.1. VARIABLES

Variable independiente.

El uso de espectroscopia con infrarrojo y técnicas de Machine Learning y Deep Learning

Variable dependiente.

Detección y Clasificación de arándanos en forma automática en la empresa TalSA.

1.10. OBJETIVOS

1.10.1. OBJETIVO GENERAL

Utilizar el procesamiento de imágenes de arándanos por medio de espectroscopia infrarrojo cercano, Machine Learning y Deep Learning para la Detección y Clasificación de Arándanos en la empresa TalSA.

1.10.2. OBJETIVOS ESPECÍFICOS

- Revisar el estado del arte de Machine Learning, Deep Learning, y Deep Neural Networks.
- Revisar el uso de espectroscopia en el reconocimiento de frutas y sus propiedades
- Analizar el proceso selección y clasificación de arándanos en la empresa TalSA
- Proponer una aplicación utilizando detección y procesamiento de imágenes por medio de espectroscopia con infrarrojo cercano y las Deep Neural Networks para la Detección y Clasificación de Arándanos en la empresa TalSA.
- Utilizar la aplicación propuesta en la detección y clasificación de arándanos en la empresa TalSA.
- Realizar el análisis estadístico e interpretar los resultados

CAPÍTULO II

MARCO TEÓRICO

CAPÍTULO II: MARCO TEÓRICO

En este capítulo se desarrolla un acercamiento a los modelos y conceptos sobre los que se sustenta el proyecto.

2.1. Arándanos

- Nombre científico: *Vaccinium corymbosum* L
- Nombre común: Blueberry, mirtilo, blaubeere, anabias, arándano.
- Familia: Ericaceae
- Considerado dentro del grupo de las frutas finas o berries, con alto contenido de antioxidantes y apreciada por sus propiedades nutracéuticas.

El arándano es una baya originaria de América del Norte, donde crece en forma silvestre.

Los arándanos silvestres han servido como alimento para osos y numerosas especies de Norteamérica. Antes de la llegada del hombre blanco a América, estos frutos eran consumidos por los aborígenes.

Actualmente es la 4º fruta más exportada del mundo y su comportamiento es creciente para los próximos cinco años.

2.1.1. BENEFICIOS PARA LA SALUD

- Poderoso antioxidante.
- Ayuda a prevenir enfermedades como el cáncer y las enfermedades cardiovasculares.
- Tiene poder antiadherente, que hace que las bacterias dañinas para nuestro organismo no se adhieran a las paredes del aparato gastrointestinal.
- Contiene vitamina P, utilizada en problemas de circulación y en afecciones vasculares del ojo.
- Contiene una sustancia conocida como antocianinas que ayudan a fortalecer el colágeno.
- Contiene flavonoides que potencian la memoria y mejoran el aprendizaje y otras funciones cognitivas.

2.1.2. DATOS DE EXPORTACIÓN DE ARÁNDANO EN EL PERÚ 2016

- Perú, es el 5º exportador mundial de arándanos
- El Perú tiene a junio 2017 unas 3,800 hectáreas plantadas de arándanos.
- La expectativa de exportación para el 2017 es de 40 mil toneladas contra 29 mil de la campaña anterior.
- Perú superó a Argentina en exportaciones del hemisferio sur: 18% contra 12%

- Perú tiene el mayor porcentaje de crecimiento anual en áreas de arándanos de América del Sur.
- En los próximos 5 años Perú será el principal exportador de arándanos frescos del Hemisferio Sur
- La región con más superficie cosechada en el Perú es La Libertad
- Región con mayor producción en toneladas, La Libertad.
- Mayor Rendimiento por hectárea: La Libertad.
- Mejores precios en chacra: La Libertad, Ica, Lima.
- Perú exporta 70% a EE.UU.

2.1.3. PRINCIPALES MERCADOS DE LOS ARÁNDANOS

Los principales mercados se encuentran en el hemisferio norte: Norteamérica, Europa y Asia.



Fig. 2 : Principales mercados de arándanos: Estados Unidos, Países Bajos, Gran Bretaña, España, Singapur, Canadá, China

Fuente: Empresa TalSA

PRINCIPALES EMPRESAS EXPORTADORAS 2016 – LA LIBERTAD

Empresas	Año 2016	Hectareas
Camposol	44%	1,466
Grupo Rocio	40%	1,100
Tal S.A.	25%	600
Hortifrut-Tal	15%	500
Agricola Santa Azul	4%	280
Hass Peru S.A.	3%	200
Otros	9%	>80

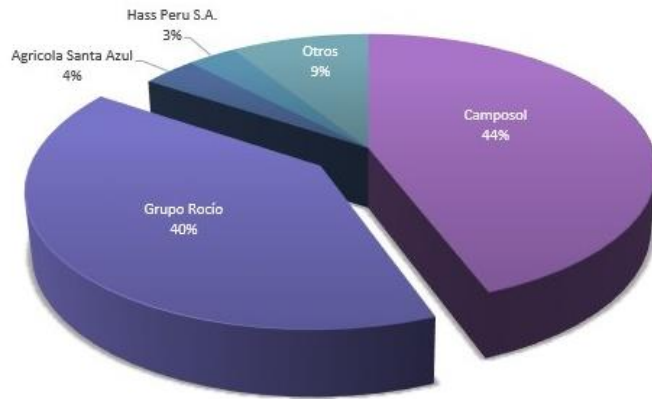


Fig. 3 : Principales empresas exportadoras de arándanos en La Libertad 2016

Fuente: Empresa TalSA

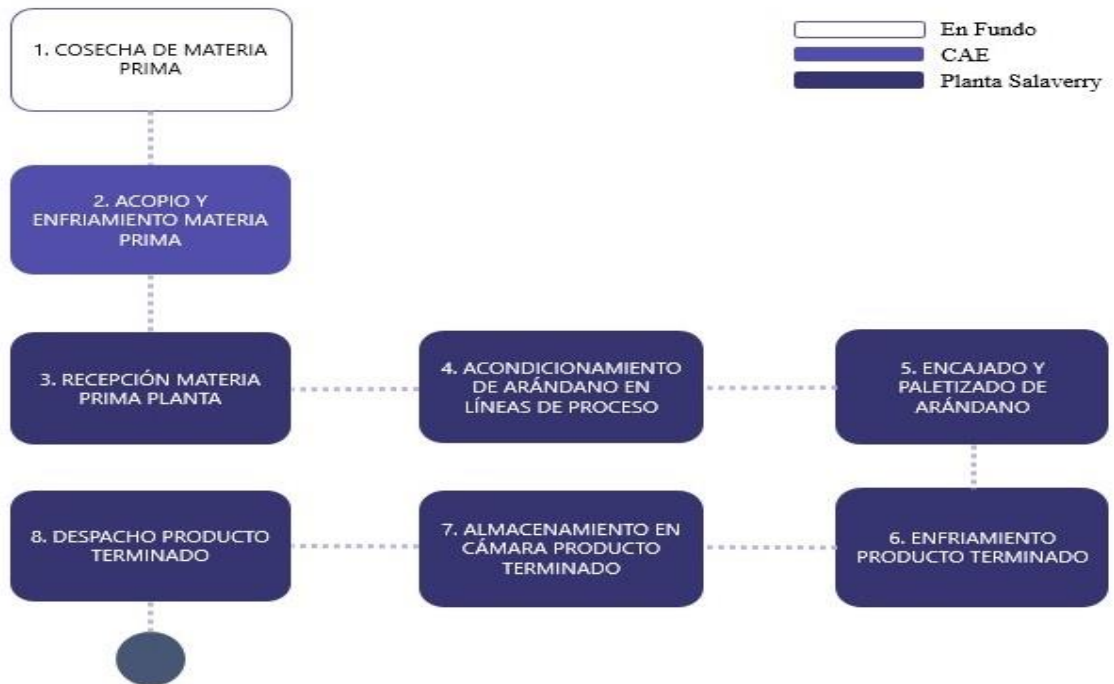


Fig. 4 : Proceso arándano fresco

Fuente: Empresa TalSA



Fig. 5 : Línea de proceso de arándano fresco en TalSA

Fuente: Empresa TalSA



Fig. 6 : Acondicionamiento y selección de Arándano en líneas de proceso

Fuente: Empresa TalSA

2.2. Espectroscopia por Infrarrojo Cercano

Para enfocar la formación de la base de datos de las imágenes proponemos el uso de métodos de detección y procesamiento de imágenes del arándano utilizando espectroscopia infrarrojo cercano. La región espectral del infrarrojo cercano (NIR) se extiende desde el extremo de las longitudes más altas del visible (alrededor de 780 nm) hasta los 3000 nm ($13\,000\text{ cm}^{-1}$ hasta 3300 cm^{-1}). Las bandas de absorción en esta zona son sobretonos o combinaciones de las bandas vibracionales de tensión que se producen en la región de 3000 a 1700 cm^{-1} . La instrumentación utilizada en la región del IR (Infrarrojo) cercano es semejante a la que se emplea para la espectroscopia de absorción ultravioleta/visible. Como fuentes se utilizan las lámparas de tungsteno, y por lo general, las celdas son de cuarzo o sílice fundida como las que se utilizan en el intervalo de 200 a 770 μm .

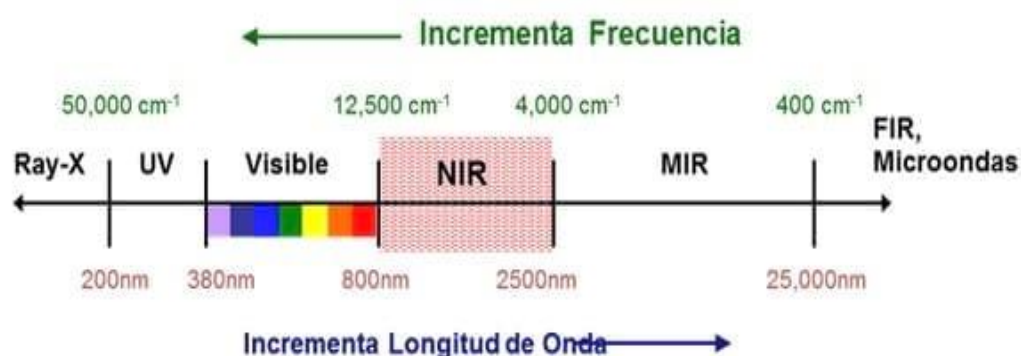


Fig. 7 : Espectro infrarrojo cercano

Fuente: <https://www.engormix.com/balanceados/articulos/tecnologia-nir-sus-usos-t32534.htm>



Fig. 8 : Cámaras térmográficas, multiespectrales e hiperspectrales centradas en la banda espectral de 0,7 a 1,7 micras o Infrarrojo Cercano (NIR, Near Infrared)

Fuente: <http://www.grupoalava.com/ingenieros/actualidad/sistemas-de-imagen-para-el-infrarrojo-cercano-nir-para-diversas-aplicaciones/>

2.3. Machine Learning

El aprendizaje automático (Shalev-Shwartz, 2014) es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan. De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de una información suministrada en forma de ejemplos.

Dependiendo del tipo de salida que se produzca y de cómo se aborde el tratamiento de los ejemplos, los diferentes algoritmos de aprendizaje automático se pueden agrupar en:

Aprendizaje supervisado, pueden aplicar lo que se ha aprendido en el pasado a nuevos datos, prediciendo el valor de una etiqueta de salida, conocidos otros atributos de entrada. A partir de datos cuya etiqueta se conoce se induce una relación entre dicha etiqueta y otra serie de atributos. Esas relaciones sirven para realizar la predicción en datos cuya etiqueta es desconocida. Esta forma de trabajar se desarrolla en dos fases.

1. Entrenamiento (construcción de un modelo usando un subconjunto de datos con etiqueta conocida)
2. Prueba (prueba del modelo sobre el resto de los datos).

Para este método, los algoritmos que se pueden implementar son los siguientes:

- K vecinos más próximos (K-nearest neighbors)
- Redes neuronales artificiales (Artificial neural net Works)
- Máquinas de vectores de soporte (Support vector machines)
- Clasificador Bayesiano ingenuo (Naïve Bayes classifier)
- Árboles de decisión (Decision trees)
- Regresión logística (Logistic regression)

2.4. Deep Learning

Precup, D. (2017). Una red neuronal es un algoritmo que imita el funcionamiento de las neuronas y de las conexiones que hay entre ellas y son entrenadas para que tengan la capacidad de desempeñar una tarea. Se dice que una red neuronal aprende mediante el entrenamiento porque no hay una programación explícita para realizar una tarea, sino que la red se programa sola a partir de ejemplos. Las redes neuronales son el mayor exponente del llamado machine learning o aprendizaje automático.

Las redes neuronales pueden aprender a clasificar y a imitar el comportamiento de sistemas complejos. Si se quiere que aprenda a diferenciar entre manzanas y naranjas sólo se le tiene que proporcionar unos cuantos ejemplares de ambas frutas e indicarle, a la vez, si se trata de una manzana o de una naranja. Una vez entrenada la red neuronal sabrá si está ante una manzana o una naranja. Lo sabrá, aunque las manzanas y naranjas no sean las que se le enseñaron durante el entrenamiento ya que las redes neuronales no memorizan, sino que generalizan. Esa es la clave del aprendizaje de las máquinas.

El estudio de las redes neuronales ha evolucionado hasta lo que hoy se llama Deep Learning.

Deep Learning es una serie de algoritmos emparentados con las redes neuronales que tienen la misma finalidad y un rendimiento mayor que otras formas de Machine Learning. La mayor diferencia es la capacidad de abstracción.

Los algoritmos Deep Learning son capaces de realizar una abstracción semejante por sí mismos, sin necesidad de que alguien la diseñe previamente. Por esta razón se dice que el Deep Learning no sólo es capaz de aprender, sino que, además, puede encontrar significado

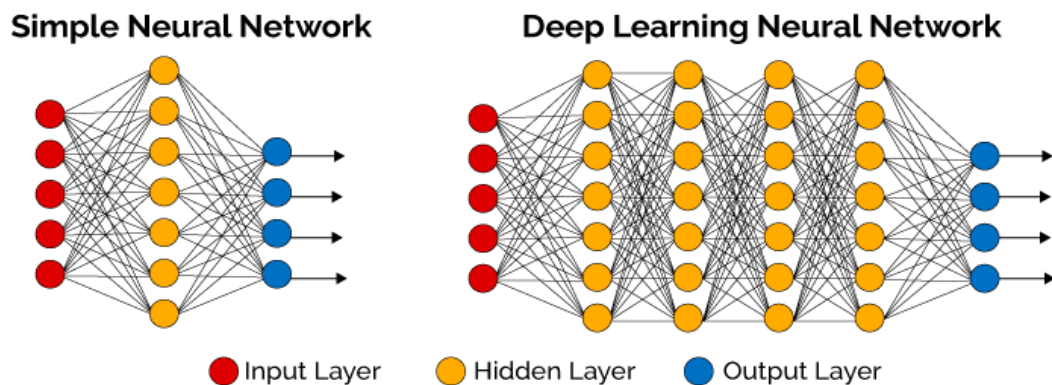


Fig. 9 : Los modelos de redes neuronales en Deep Learning

Fuente: <https://www.ackstorm.com/deep-learning-clasificacion-imagenes/>

2.5. Red neuronal

Es una forma de computación, inspirada en modelos biológicos, que hace uso de modelos matemático compuestos por un gran número de elementos procesales organizados en niveles.

Se pueden representar como 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.

2.5.1. Ventajas de las redes neuronales.

Basados en su conformación y a sus fundamentos, presentan muchas características semejantes a las del cerebro. En este sentido, son capaces de aprender de la experiencia, de generalizar casos anteriores, de abstraer características esenciales de información que puede ser irrelevante, etc. Esto le origina muchas ventajas y que esta tecnología se esté aplicando en múltiples áreas. Entre estas ventajas se tienen:

- **Aprendizaje Adaptativo.**

Es la capacidad de poder aprender a efectuar ciertas tareas en base a un entrenamiento o en base a una experiencia inicial. Esto significa que, aprenden a realizar determinadas tareas cuando son sometidas a un proceso de entrenamiento utilizando un conjunto de casos que sirven como ejemplos. Las redes neuronales al ser sometidas a entrenamiento, aprenden a reconocer y diferenciar patrones. Por esto es que no se requiere de modelos iniciales y tampoco es necesario especificar funciones de distribución de probabilidad. Las redes neuronales son adaptables porque poseen una capacidad de autoajuste de sus componentes. Son dinámicos, porque constantemente se encuentran cambiando para adaptarse a nuevas condiciones.

En el proceso de aprendizaje, se ajustan los pesos de sus enlaces, para que obtengan determinados resultados. Una red neuronal no necesita un algoritmo para resolver un problema, porque puede generar su propia distribución de pesos de sus enlaces al momento del aprendizaje.

- **Auto organización.**

Utilizando una etapa de aprendizaje, puede crear su propia organización o representación de la información que recibe.

Utilizan su capacidad de aprendizaje adaptativo para auto organizar la información que reciben durante la etapa de aprendizaje.

El aprendizaje es la modificación de cada elemento y, la auto organización implica la modificación de la red neuronal completa en la búsqueda de cumplir un objetivo específico. Auto organizan la información, si son utilizadas para reconocer algunos patrones. Un ejemplo común, es la red

backpropagation, que creará su propia representación característica, para reconocer ciertos patrones.

La auto organización origina la generalización, permitiéndoles responder apropiadamente cuando se le proporcionan datos o situaciones nuevas.

El sistema está preparado para generalizar la entrada y, obtener así, una respuesta, aún si la información de entrada no es muy clara o está incompleta.

- **Tolerancia a fallos.**

Si una red neuronal es destruida parcialmente, algunas de sus capacidades se pueden retener, aún si el daño es grande.

No sufre una caída repentina en el caso que se produzca un fallo en un número de neuronas no muy grande, a pesar que el comportamiento del sistema se pueda ver influenciado, En la tolerancia a fallos, se debe considerar:

- a) Existe una tolerancia a fallos respecto a los datos. En este caso una red neuronal, aún puede aprender a reconocer patrones distorsionados o incompletos.
- b) Una red neuronal, pueden continuar realizando su función, aunque con alguna degradación, a pesar que ésta se destruya en parte.

El tener su información distribuida en los enlaces entre neuronas, con algún grado de redundancia en su almacenamiento es lo que brinda la tolerancia a fallos. Es decir, la información es almacenada en forma no localizada.

- **Operación en tiempo real.**

Se requiere y existe hardware especial que permite que las redes neuronales puedan trabajar en paralelo, adaptándose muy bien a esto por su implementación que también es en paralelo, requiriendo de cambio en los pesos de los enlaces sea mínimo.

- **Fácil inserción dentro de la tecnología existente.**

Se puede integrar las redes neuronales en sistemas existentes introduciéndolas en un chip de tal modo que se pueda mejorar la ejecución de ciertas tareas.

Si se entrena una red neuronal para realizar una única tarea, bien definida; utilizando las herramientas de cómputo adecuadas, esta red puede ser

rápidamente entrenada, validada e implementada en un dispositivo de hardware. (Andrade, 2013).

2.6. Elementos básicos que componen una red neuronal.

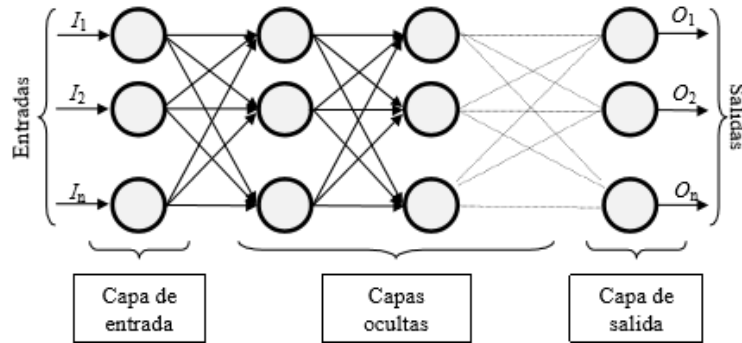


Fig. 10 : Ejemplo de una red neuronal totalmente conectada

Fuente: (Soriano et.al , 2010)

En la figura se tiene una red neuronal, constituida por neuronas interconectadas y arregladas en tres capas. Los datos ingresan por medio de la capa de entrada, pasan a través de la capa oculta y salen por la capa de salida. La capa oculta puede estar constituida por varias capas.

2.6.1. Función de entrada

La neurona trata muchos valores de entrada como si fueran uno solo; esto se denomina entrada global. El uso de una función de entrada, la cual se calcula a partir del vector entrada, permite combinar las entradas (in_{i1} , in_{i2} , ...) dentro de lo que sería la entrada global, gin_i .

Así, la función de entrada puede describirse en los siguientes términos:

$$\text{input}_i = (in_{i1} \cdot w_{i1}) * (in_{i2} \cdot w_{i2}) * \dots (in_{in} \cdot w_{in})$$

donde: * representa al operador apropiado (máximo, sumatoria, producto, etc.), n representa el número de entradas a la neurona N_i , y w_i representa el respectivo peso.

Cada uno de los valores de entrada se multiplican por los pesos asignados a la neurona. Los pesos cambian la medida de la influencia que tienen cada uno de estos valores de entrada. Esto significa que, un valor de entrada grande puede tener una pequeña influencia, si los pesos son los suficientemente pequeños.

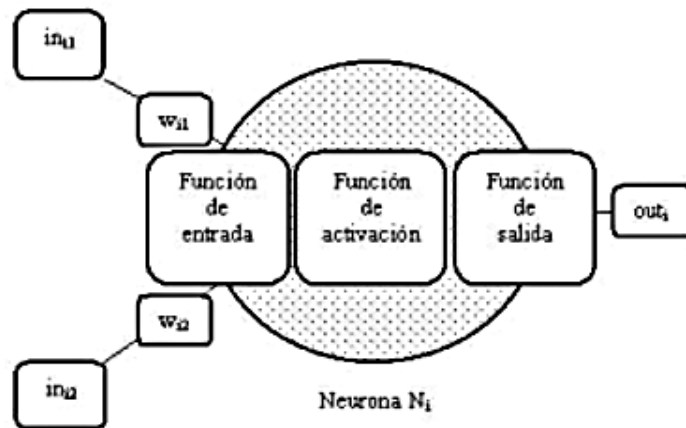


Fig. 11 : Ejemplo de una neurona con 2 entradas y 1 salida.

Fuente:(Soriano et.al , 2010)

La terminología utilizada se explica a continuación:

in_{i1} = Entrada N° 1 para neurona N_i

w_{i1} = peso de in_{i1}

in_{i2} = Entrada N° 2 para neurona N_i

w_{i2} = peso de in_{i2}

out_i = salida de neurona N_i .

Vector de entrada = reunión de todas las entradas $in_i = (in_{i1}, in_{i2}, \dots, in_{in})$.

Entre las funciones de entrada más utilizadas y conocidas se tiene:

- 1) Sumatoria de entradas: suma de todas las entradas a la neurona, multiplicadas por sus respectivos pesos.

$$\sum_j (n_{ij} w_{ij}), \text{ con } j = 1, 2, \dots, n$$

- 2) Producto de entradas: producto de todas las entradas a la neurona, multiplicados por sus pesos respectivos.

$$\prod_j (n_{ij} w_{ij}), \text{ con } j = 1, 2, \dots, n$$

- 3) Máximo de entradas: se considera el valor de entrada más fuerte, multiplicado por su respectivo peso.

$$\text{Max}_j (n_{ij} w_{ij}) \text{ con } j = 1, 2, \dots, n$$

2.6.2. Función de activación

La neurona biológica, tiene un “estado de activación”, pudiendo estar activa (excitada) o inactiva (no excitada). De manera análoga, las neuronas artificiales tienen diferentes estados de activación; algunas solamente dos, pero otras pueden tomar cualquier valor dentro de un grupo establecido.

La función activación calcula el estado de actividad de una neurona; transformando la entrada global en un valor (estado) de activación, en el rango de (0 a 1) o de (-1 a 1). Una neurona puede estar inactiva (0 o -1) o activa (1). La función activación, es una función de la entrada global (g_{in_i}) menos el umbral (Θ_i).

Entre las funciones de activación más comunes, se tiene:

1) Función lineal:

$$f(x) = \begin{cases} -1 & x \leq -1/a \\ a \cdot x & -1/a < x < 1/a \\ 1 & x \geq 1/a \end{cases}$$

Donde: $X = g_{in_i} - \Theta_i$, y $a > 0$.

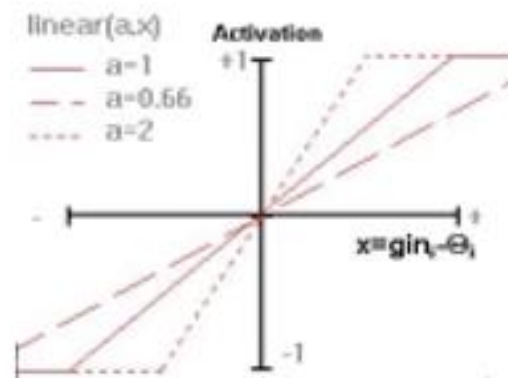


Fig. 12 : Función lineal.

Fuente (Andrade, 2013)

Los valores de salida con esta función de activación serán: $a \cdot (g_{in_i} - \Theta_i)$, cuando $(g_{in_i} - \Theta_i)$ se encuentra en el rango $(-1/a, 1/a)$. Fuera de este rango la salida será 1 o -1, respectivamente.

Si $a = 1$ entonces se tiene que salida = entrada.

2) Función sigmoidea:

$$f(x) = \frac{1}{1 + e^{-g x}}, \text{ con } x = g i n_i - \Theta_i.$$

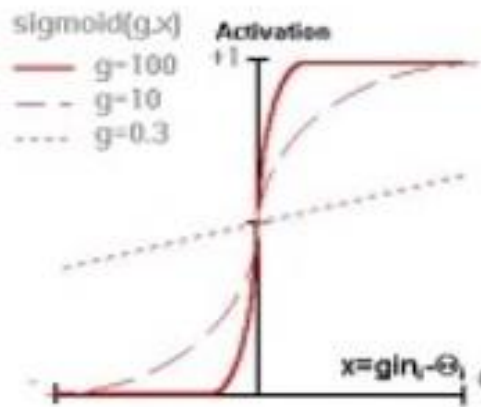


Fig. 13 : Función sigmoidea

Fuente (Andrade 2013)

Los valores que produce esta función se encuentran en el rango de 0 a 1. Si se modifica g , se afecta la pendiente de la función.

3) Función tangente hiperbólica:

$$f(x) = \frac{e^{g x} - e^{-g x}}{e^{g x} + e^{-g x}}, \text{ con } x = g i n_i - \Theta_i.$$

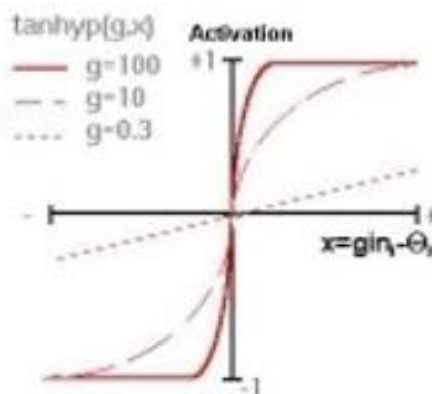


Fig. 14 :Función tangente hiperbólica

Fuente (Andrade, 2013)

Las salidas de la función tangente hiperbólica están en el rango de -1 a 1. Si se modifica g , se afecta la pendiente de la función.

Estas funciones de activación se utilizan porque se requiere tener una elevada potencia cuando se comienza a trabajar. Cuando se llega a otra capa, nuevamente se requiere incrementar la potencia.

2.6.3. Función de salida

Es el componente final de una neurona. El valor que se obtiene, es la salida de la neurona i (out_i); de esta forma, esta función establece el valor que se transfiere a las otras neuronas. Si el valor obtenido es menor que un umbral establecido, no se produce, ninguna salida para la neurona siguiente. No se permite cualquier valor como entrada para una neurona, por consiguiente los valores de salida están en el rango $[0, 1]$ o $[-1, 1]$. Podrían ser valores binarios $\{0, 1\}$ o $\{-1, 1\}$.

Las funciones de salida más conocidas son:

Ninguna: Denominada función identidad. Su característica es: Salida = entrada

$$\text{Binaria: } \begin{cases} 1 & \text{si } act_i \geq \xi_i \\ 0 & \text{de lo contrario} \end{cases}, \text{ donde } \xi_i \text{ es el umbral.}$$

2.7. Niveles o capas de una red neuronal

Las neuronas se distribuyen en niveles o capas, con un número determinado en cada una. Considerando esto, se tienen tres tipos de capas:

- Capa de entrada: recibe directamente la información que viene de fuera de la red.
- Capas ocultas: Son capas internas de la red. No interactúan con el entorno exterior. Pueden ser entre cero y un número grande. Las neuronas de estas capas se pueden interconectar de formas distintas, originándose las topologías de redes neuronales.
- Capas de salidas: Son aquellas que se encargan de transferir la información desde la red hacia el exterior.

Una red es totalmente conectada si todas las salidas desde un nivel llegan a todos y cada uno de los nodos del nivel siguiente.

2.8. Mecanismos de aprendizaje

Los datos de entrada se procesan a través de la red neuronal para lograr una salida. Las redes neuronales pueden hacer generalizaciones a partir de un conjunto de ejemplos.

Una red neuronal aprende a calcular la salida correcta para cada conjunto de entrada utilizando el conjunto de ejemplos. A esta etapa de aprendizaje se denomina: entrenamiento o acondicionamiento.

El conjunto de datos de ejemplos se denomina conjunto de datos de entrenamiento. Si durante el aprendizaje no cambian ni la topología de la red ni las funciones, pero los pesos de cada enlace si pueden cambiar; el aprendizaje se denomina adaptación de los pesos. En el aprendizaje la red neuronal modifica sus pesos como respuesta a los datos de entrada.

Los cambios que se pueden dar son la destrucción, modificación y creación de enlaces entre neuronas. La creación implica que su peso pasa a un valor diferente de cero. La destrucción ocurre cuando el peso pasa a ser cero.

Los pesos de los enlaces se van modificando durante el aprendizaje. Dicho proceso termina, es decir la red ya ha aprendido, cuando los pesos se mantienen estables.

Es importante saber cómo se modifican los pesos, cuáles son los criterios para que cambie el valor de un enlace, si se quiere que la red aprenda nueva información. Se distinguen dos métodos de aprendizaje:

a- Aprendizaje supervisado.

b- Aprendizaje no supervisado.

También se debe considerar si la red puede aprender durante su funcionamiento o si la red debe estar desconectada, hasta que termine dicho proceso.

El primer caso se conoce como aprendizaje on line, y en el segundo caso se conoce como aprendizaje off line.

En el aprendizaje off line, se tiene una fase de aprendizaje o entrenamiento y una fase de operación o funcionamiento.

Se debe contar con datos para el entrenamiento y datos para realizar la prueba o test, para la fase de operación o funcionamiento. Una vez que termina la etapa de entrenamiento de la red, los pesos permanecerán fijos o estables.

Una fórmula para determinar los cambios en los pesos es la siguiente:

Peso Nuevo = Peso Anterior + Cambio de Peso

Matemáticamente: $w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)$

Aquí, t se refiere a la etapa de aprendizaje, $w_{ij}(t+1)$ se refiere al nuevo peso y $w_{ij}(t)$ hace referencia al peso anterior.

2.8.1. Aprendizaje supervisado

Su característica es que el proceso de aprendizaje se lleva a cabo con un entrenamiento controlado por un agente externo o supervisor, que es quien establece cual es la respuesta que la red debe producir conociendo los datos de entrada. Si la salida de la red no es la esperada, se debe modificar los pesos de los enlaces para que la salida se acerque a lo esperado.

El aprendizaje supervisado se puede dar de tres formas:

- 1) Aprendizaje por corrección de error.
- 2) Aprendizaje por refuerzo.
- 3) Aprendizaje estocástico.

Aprendizaje por corrección de error. Se tiene que ajustar los pesos de los enlaces considerando la diferencia entre los valores esperados y los valores obtenidos, siendo esto el grado de error que se produce en la salida. Como ejemplo se tiene el aprendizaje del Perceptron.

La regla que se aplica consiste en calcular la desviación o el error δ , de la salida de cada neurona en la capa de salida. Este valor se utiliza para modificar los pesos de cada enlace de la neurona precedente.

Para el Perceptron, la modificación de los pesos se efectúa así:

$$\Delta w_{ij} = \sigma * out_j * (a_{qi} - out_i)$$

a_{qi} es la salida esperada de la neurona de salida N_i ,

$\delta_i = (a_{qi} - out_i)$ es la desviación u error, de la neurona N_i

σ es el aprendizaje.

N_j (out_j) afecta a la entrada global y a la activación en la salida de la neurona N_i . Se produce un “efecto en cadena”.



Fig. 15 : Influencia de la salida de la neurona N_j en la entrada de la neurona N_i .

Fuente: (Soriano et.al , 2010)

También se tiene la regla de aprendizaje Delta o regla del mínimo error cuadrado (LMS Error: Least Mean Squared Error), que también utiliza la desviación a la salida objetivo, pero considerando todas las neuronas predecesoras que tiene la neurona de salida. Con esto se puede calcular el error global que ocurre en todo el proceso de entrenamiento de la red. A mayor información del error, más rápido se aprende. El error calculado (δ) es repartido entre las neuronas predecesoras, en forma equitativa.

Otra regla es la del aprendizaje de propagación hacia atrás o de backpropagation, también conocido como regla LMS multicapa. Esta es una generalización de la regla de aprendizaje Delta. Esta es la primera regla de aprendizaje que ha permitido modificar los pesos en los enlaces en la capa oculta.

Aprendizaje por refuerzo. Es un aprendizaje supervisado, más lento que el anterior. Se, que se basa en no tener un ejemplo completo del comportamiento esperado. No se indica exactamente la salida esperada en la etapa de entrenamiento, para una determinada entrada.

El supervisor establece con una señal de refuerzo si la salida obtenida es la salida esperada (éxito = +1 o fracaso = -1), y en base a esto, se reajustan los pesos usando criterios de probabilidades.

La función del supervisor es opinar sobre la respuesta de la red. En el caso de supervisión por corrección del error, la función del supervisor es indicar la respuesta que se debe obtener.

Aprendizaje estocástico. Consiste en realizar cambios aleatorios en los pesos de los enlaces de la red. Luego se evalúa su efecto considerando la meta establecida y distribuciones de probabilidad.

El grado de estabilidad de la red, corresponde a una situación en la que los pesos de los enlaces permiten que su funcionamiento sea el que más se acerca al resultado esperado. El aprendizaje consiste en cambiar los pesos en forma aleatoria y calcular la energía de la red. Si la energía es menor después del cambio, significa que el comportamiento de la red se aproxima al esperado y, se acepta el cambio; Si la energía no es menor, entonces el cambio se acepta en base a una distribución de probabilidades.

2.8.2. Aprendizaje no supervisado

También denominado aprendizaje autosupervisado, ajusta los pesos de los enlaces entre las neuronas, sin injerencia de factores externos. La red no recibe información del entorno que le diga si la salida obtenida es correcta o no.

Estas redes, se encargan por si solas, de determinar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos de entrada. La estructura de las redes y el algoritmo de aprendizaje utilizado, permiten hacer la interpretación de la salida obtenida.

La salida obtenida establece el grado de familiaridad o similitud entre la información de entrada y los resultados.

También se puede establecer categorías o clusterización. En este caso la red indica a que categoría pertenece la información de entrada. La red es la encargada de encontrar las categorías apropiadas en base a las correlaciones de los datos. Se consideran dos tipos de algoritmos para el aprendizaje:

- a) Aprendizaje hebbiano.
- b) Aprendizaje competitivo y comparativo.

Aprendizaje hebbiano.

Trata de medir la familiaridad o características de los datos de entrada.

Se fundamenta en el hecho de que si dos neuronas N_i y N_j tienen igual estado al mismo tiempo (ambas activas o ambas inactivas), se incrementa el peso del enlace entre ellas. Las entradas y salidas posibles son: $\{-1, 1\}$ o $\{0, 1\}$ (neuronas binarias). Esta regla de aprendizaje de Hebb se originó a partir de la neurona biológica clásica, que solo puede tener dos estados: activa o inactiva.

Aprendizaje competitivo y comparativo.

Su objetivo es la clusterización o clasificación de los datos de entrada. Se caracteriza porque si se determina un patrón nuevo, que pertenece a una clase conocida, entonces al incluir este nuevo patrón a dicha clase, será parte de su representación. Si el patrón de entrada no pertenece a ninguna clase reconocida, la estructura de la red y sus pesos, se reajustarán para reconocer la nueva clase.

2.9. Elección del conjunto de pesos

Previo al inicio del proceso de entrenamiento se debe establecer el estado inicial, esto es, establecer un conjunto inicial de pesos, para todos los enlaces entre las neuronas.

Para esto se tienen varios criterios. Uno de ellos es, para cada enlace darle un peso aleatorio. Estos deben estar en un intervalo determinado. Este intervalo es del tipo $[-n, n]$, siendo n , un número natural positivo. Los pesos no se encuentran restringidos a ese intervalo, durante el entrenamiento.

Detención del proceso de aprendizaje

Se requiere tener una condición de detención, para saber cuando terminará el proceso de aprendizaje. El entrenamiento termina cuando el error ha alcanzado un mínimo o cuando el error obtenido está por debajo de un determinado umbral, para cada ejemplo dado.

En general, las herramientas de las redes neuronales muestran estos errores por medio de gráficos; los mismos que sirven solamente para tener una idea desarrollo del proceso.

Otra condición para detener el aprendizaje puede ser cuando hayan sido completados cierto número de ciclos y/o pasos de entrenamiento.

Al alcanzar la condición de término del entrenamiento, los pesos permanecerán invariables. Se ha logrado transformar los datos de entrada a los de salida.

Los ejemplos de entrenamiento que se ingresan a la red neuronal deben darse aleatoriamente.

2.10. Validación de la red neuronal

Luego de concluir el entrenamiento, lo que sigue es comprobar que la red neuronal pueda resolver nuevos problemas, del tipo para los que ha sido entrenada. Para esto se requiere de otro conjunto de datos, llamado conjunto de validación o testeo.

Cada ejemplo del conjunto de evaluación tiene los datos de entrada, conociéndose su solución. La solución calculada para cada ejemplo de validación se compara con la solución conocida, para determinar si la salida de la red neuronal es correcta.

2.11. Topología de las redes neuronales

La topología o arquitectura de una red neuronal indica como se organizan las neuronas en capas o agrupaciones de neuronas.

Los parámetros básicos de una red neuronal son el número de capas, el número de neuronas por capa, el grado de conectividad y el tipo de enlaces entre neuronas.

2.11.1. Redes monocapa

La red está constituida por una sola capa y las neuronas establecen conexiones entre ellas. Se utilizan para tareas relacionadas de autoasociación, para restaurar información de entrada que se encuentra incompleta o distorsionada.

2.11.2. Redes multicapa

Las neuronas se encuentran agrupadas en varios niveles o capas.

Normalmente, todas las neuronas de una capa reciben señales de entrada desde otra capa anterior, y envían señales de salida a una capa posterior. (conexiones hacia adelante o feedforward).

También es posible conectar la salida de las neuronas de capas posteriores a la entrada de capas anteriores; a éstas se denomina conexiones hacia atrás o feedback. Estas dos posibilidades permiten establecer dos tipos de redes multicapas: redes con conexiones hacia adelante o redes feedforward, y redes con conexiones tanto hacia adelante como hacia atrás o redes feedforward/feedback.

2.12. Conexión entre neuronas

La conectividad se refiere a la forma en que las salidas de las neuronas están encauzadas para ser entradas de otras neuronas. La salida de un nodo puede ser una entrada de otro, o incluso ser una entrada de sí mismo (conexión autorrecurrente).

Una red de conexión hacia adelante es aquella en la que ninguna salida de las neuronas es entrada de neuronas del mismo nivel o de niveles precedentes.

Las redes de propagación hacia atrás son aquellas en las que sus salidas pueden ser conectadas como entradas de neuronas de niveles previos o del mismo nivel.

2.13. Redes Backpropagation – propagación hacia atrás

Se denominan así porque el error es propagado hacia atrás, es decir el error se propaga desde la capa de salida hacia atrás. Con esto se consigue que durante el entrenamiento se modifiquen los pesos de los enlaces de las neuronas de las capas ocultas.

Este cambio de pesos influye sobre la entrada global, en la activación y en la salida de una neurona.

Se debe considerar la variación de la función de activación al variar los pesos.

Esto se llama sensibilidad de la función activación, de acuerdo al cambio en los pesos. (Soriano, A., et al, J., 2010)

2.14. Redes neuronales convolucionales

Es un tipo de red neuronal en la que las neuronas corresponden a campos receptivos de una manera análoga a las neuronas en la corteza visual primaria del cerebro. Es una variante de un perceptrón multicapa. Son muy efectivas para aplicaciones de visión artificial, clasificación y segmentación de imágenes, etc.

Las redes neuronales convolucionales están formadas por múltiples capas de filtros convolucionales de una o más dimensiones. Luego de cada capa, se agrega una función para realizar un mapeo causal no-lineal.

En redes de clasificación, primero se tiene la fase de extracción de características, compuesta de neuronas convolucionales y de reducción de muestreo. Al final de la red se encuentran neuronas de perceptrón para la clasificación final a partir de las características extraídas.

La fase de extracción se compone de capas alternas de neuronas convolucionales y neuronas de reducción de muestreo.

Neuronas convolucionales

Según Gómez (2018), en la fase de extracción de características, el perceptrón se reemplaza por procesadores en matriz que realizan una operación sobre los datos de imagen 2D que pasan por ellas. La salida de cada neurona convolucional se calcula como:

$$Y_j = g \left(b_j + \sum_i K_{ij} \otimes Y_i \right)$$

El operador de convolución tiene el efecto de filtrar la imagen de entrada con un núcleo previamente entrenado. Con esto se consigue transformar algunas características que se vuelven más dominantes en la imagen de salida al tener estas un valor numérico más alto asignados a los pixeles que las representan.

Neuronas de Reducción de Muestreo

Según Gómez (2018), sostiene que las redes neuronales tienen tolerancia a pequeñas perturbaciones en los datos de entrada. La reducción de muestreo ocurre dentro de una red neuronal convolucional.

El resultado que se obtiene es que el tamaño de los datos se reduce por un factor igual al tamaño de la ventana de muestra sobre la cual se opera.

Neuronas de Clasificación

Luego de las fases de extracción se llega a la fase de clasificación. Para esto, se han depurado los datos, para obtener un conjunto de características únicas para los datos de entrada, y la labor de esta última fase es clasificar estas características con alguna etiqueta, de acuerdo a los objetivos de entrenamiento.

En esta fase las neuronas funcionan como un perceptrón multicapas, donde la salida de cada una se calcula de esta forma:

$$y_j = g \left(b_j + \sum_i w_{ij} \cdot y_i \right)$$

Y_j es la salida de una neurona j que se calcula por medio de la combinación lineal de las salidas Y_i de las neuronas en la capa anterior cada una de ellas multiplicadas con un peso W_{ij} que le corresponde a su enlace. Esto se suma a una influencia b_j y luego se pasa por una función de activación $g(\cdot)$ no-lineal. (Gómez, 2018).

2.15. El perceptrón

Es la red de neuronas más sencilla, compuesta solamente por una capa de neuronas de entrada y otra capa de neuronas de salida.

Una neurona sola y aislada no es útil, pero se vuelve valiosa si asocia a otras neuronas.

El perceptrón se puede ver como una unidad de cálculo que intenta modelar el comportamiento de una neurona natural. Es un algoritmo que puede generar un criterio para seleccionar un grupo, entre un grupo de elementos.

El perceptrón puede ser entrenado para realizar alguna tarea.

Su capacidad y funcionalidad se genera cuando se relacionan con más para generar una red. (Andrade, T. 2013).

2.16. Aplicación de las técnicas de Espectroscopia NIR y madurez de frutas.

Según (Lafuente R., 2015), se puede aplicar las técnicas de Espectroscopia de Infrarrojo Cercano y Análisis de Imágenes Multiespectrales, para utilizarlas como herramientas que permitan determinar parámetros de calidad de las frutas. Para esto se debe realizar una selección de variables con el fin de proponer modelos de calibración para ser aplicados en las líneas de clasificación de frutas de las empresas del rubro.

2.17. Procesamiento de imágenes de frutas con espectroscopia NIR.

Consiste en utilizar la espectroscopia de infrarrojo cercano (NIR) y el análisis de imágenes multiespectrales; para control de calidad de frutas. Al ser métodos no destructivos, evitan el gasto en fruta destruida y posibles errores en selección de muestra a analizar.

2.18. Tecnología de Espectroscopia Infrarrojo cercano NIR

La Tecnología de Infrarrojo cercano, está basada en una combinación de espectros y métodos matemáticos. La luz NIR es reflejada sobre la muestra y se modifica sutilmente de acuerdo a la muestra.

La modificación espectral es convertida en información mostrando la composición de la muestra.

2.19. Clasificación de frutas utilizando Deep Neural Network

Para clasificar una fruta determinada con una red neuronal es necesario extraer características que definan las frutas. Estas características pueden ser el color, la forma, el tamaño, etc. Representar las frutas mediante estas características es una forma de abstracción que debe ser diseñada para que se entrene la red neuronal.

2.20. Keras

Keras es una API de redes neuronales de alto nivel, de código abierto, escrita en Python y capaz de ejecutarse sobre TensorFlow, Microsoft Cognitive Toolkit o Theano. Está especialmente diseñada para posibilitar la experimentación en corto poco tiempo con redes de Aprendizaje Profundo.

Fue desarrollado con un enfoque en permitir la experimentación rápida. Permite pasar de la idea al resultado con el menor retraso posible, facilitando el proceso de investigación.

Keras proporciona una biblioteca de aprendizaje profundo que permite la creación de prototipos de forma fácil y rápida (a través de la facilidad de uso, la modularidad y la extensibilidad). Además, admite redes neuronales convolucionales y redes neuronales recurrentes, así como combinaciones de las dos.

2.20.1. Principios rectores

2.20.1.1. **La facilidad de uso.** Keras es una API diseñada para seres humanos, no máquinas. Pone la experiencia del usuario al frente y al centro. Keras sigue las mejores prácticas para reducir la carga cognitiva: ofrece API consistentes y simples, minimiza el número de acciones de usuario requeridas para los casos de uso comunes y proporciona comentarios claros y procesables en caso de error del usuario.

2.20.1.2. **Modularidad.** Un modelo se entiende como una secuencia o un gráfico de módulos independientes, totalmente configurables, que se

pueden conectar con la menor cantidad de restricciones posible. En particular, las capas neuronales, las funciones de costo, los optimizadores, los esquemas de inicialización, las funciones de activación y los esquemas de regularización son módulos independientes que puede combinar para crear nuevos modelos.

2.20.1.3. **Fácil extensibilidad.** Los nuevos módulos son fáciles de agregar como nuevas clases y funciones, y los módulos existentes brindan amplios ejemplos. Poder crear fácilmente nuevos módulos permite una expresividad total, lo que hace que Keras sea adecuado para la investigación avanzada.

2.20.1.4. **Trabajar con Python.** No hay archivos de configuración de modelos separados en un formato declarativo. Los modelos se describen en el código Python, que es compacto, más fácil de depurar y permite la extensibilidad.

2.20.2. Keras es compatible con varios motores de back-end y no se encierra en un solo ecosistema

Los modelos Keras se pueden desarrollar con una gama de diferentes backends de aprendizaje profundo. Es importante destacar que cualquier modelo Keras que solo aproveche las capas incorporadas será portátil en todos estos backends: puede entrenar un modelo con un backend y cargarlo con otro (por ejemplo, para la implementación). Los backends disponibles incluyen:

- El backend TensorFlow (de Google)
- El backend CNTK (de Microsoft)
- El backend Theano

Keras tiene una adopción más fuerte tanto en la industria como en la comunidad de investigación que cualquier otro marco de aprendizaje profundo, excepto TensorFlow; y la API de Keras es la interfaz oficial de TensorFlow, a través del módulo `tf.keras`. (Keras, s.f.).

2.21. TensorFlow

TensorFlow es una plataforma de código abierto para el aprendizaje automático. Cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos

que permite impulsar el estado del arte en Machine Learning y poder construir e implementar fácilmente aplicaciones con tecnología Machine Learning.

La arquitectura flexible de TensorFlow, ofrece una API, que sirve a principiantes y expertos para desarrollar aplicaciones e implementar el cálculo a una o más CPU o GPU en equipos de escritorio, web, servidores, dispositivos móviles y en la nube. Está disponible para investigación y producción. (TensorFlow, s.f.)

Se basa en un sistema de redes neuronales. Esto significa que puede relacionar varios datos en red simultáneamente, de la misma forma que lo hace el cerebro humano. Por ejemplo, puede reconocer varias palabras del alfabeto porque relaciona las letras y fonemas. Otro caso es el de imágenes y textos que se pueden relacionar entre sí rápidamente gracias a la capacidad de asociación del sistema de redes neuronales.

TensorFlow es la plataforma, open source, de Aprendizaje Profundo más importante del mundo, desarrollado por Google para satisfacer las necesidades de sistemas capaces de construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones, análogos al aprendizaje y razonamiento usados por los humanos. Su flexibilidad y gran comunidad de desarrolladores lo ha posicionado como la herramienta líder en el sector del Deep Learning.

Google liberó TensorFlow para extender esta tecnología y su presencia en el desarrollo de aplicaciones en distintos dispositivos y plataformas. Esto ha permitido que varias empresas utilicen el software o se han basado en el mismo para desarrollar aplicaciones diversas.

2.22. keras-pandas

keras-pandas permite a los usuarios crear y repetir rápidamente modelos de aprendizaje profundo.

Obtener datos con formato y en Keras puede ser tedioso, lento y difícil. Keras-pandas supera estos problemas al proporcionar automáticamente

Transformaciones de datos: se obtiene los datos de las variables X e Y limpias, transformadas y formateadas correctamente.

Canalización de datos: entrada de Keras con formato correcto, capas ocultas y de salida para comenzar a iterar rápidamente

Una 'entrada', sin la molestia de preocuparse por las formas de entrada o los tipos de datos.

Una capa de salida, formateada correctamente para el tipo de variable de respuesta proporcionada

Con estos recursos, es posible construir rápidamente e iterar sobre modelos de aprendizaje profundo, y centrarse en las partes del modelado que disfruta. (Keras-Pandas, s.f.).

2.23. Dataset

Introducido en TensorFlow 1.3, la API Dataset es el método estándar para cargar datos en modelos TensorFlow. Un conjunto de datos es una secuencia de elementos, que se componen de componentes de `tf.Tensor`.

Para usar esto con Keras, se hace un conjunto de datos de elementos del formulario (lote de entrada, lote de salida). A partir de ahí, se crea un iterador de un disparo y un nodo gráfico correspondiente a su método `get_next ()`. Luego, sus componentes se proporcionan a la capa de entrada de la red y al método `Model.compile ()`, respectivamente.

2.24. Scikit-learn

Es la biblioteca de código abierto de Python que implementa una amplia variedad de algoritmos de aprendizaje automático, preprocesamiento, validación cruzada y visualización con la ayuda de una interfaz unificada

CAPÍTULO III

MATERIAL Y MÉTODO

CAPÍTULO III: MATERIAL Y MÉTODOS

En este capítulo se plantean las técnicas, métodos e instrumentos utilizados para evaluar la propuesta.

3.1. MATERIAL Y PROCEDIMIENTO:

3.1.1. MATERIAL

Población

Todo el universo de arándanos en proceso de producción de la Empresa TalSA, en el período correspondiente entre noviembre del 2018 a marzo del 2019.

Muestra

Se elegirá una muestra de 1000 arándanos aleatoriamente y posteriormente se obtendrá sus espectros NIR para formar la base de datos para el aprendizaje y entrenamiento del sistema.

Para las pruebas de clasificación automática, se tomarán uno o varios grupos experimentales de arándanos.

Unidad de análisis

Proceso de clasificación de Arándanos.

3.1.2. PROCEDIMIENTOS

3.1.2.1. DISEÑO DE TECNICAS DE RECOLECCION DE LA INFORMACION

- Tipo de investigación: Cuantitativa
- Diseño de investigación: Descriptiva
- Modelo: Analítico
- Técnica:
 - Análisis de la literatura.
 - Encuestas
 - Análisis de resultados

3.1.2.2. TECNICAS E INSTRUMENTOS

3.1.2.2.1. Técnicas e Instrumentos de Recolección de datos

Técnicas:

- Encuestas
- Toma de imágenes espectrales con equipo NIR
- Entrevistas a juicio de expertos
- Observación

Instrumentos

- Guía de entrevista
- Registro de espectros obtenidos
- Registro de pruebas

3.1.2.2.2. Técnicas e Instrumentos de procesamiento y análisis de datos

Los datos relacionados serán mostrados en gráficos, el análisis de datos y la contrastación de la hipótesis se propone realizar mediante pruebas estadísticas como chi-cuadrado para determinar si se cumple la hipótesis nula o alternativa.

3.1.3. METODOLOGÍA

Se explica la metodología a seguir, conformada por las siguientes fases:

1.- Formación de la Base de Datos

Captura de imágenes espectrales de arándanos, utilizando equipo NIR, digitalización y formación de la base de datos.

2.- Elaboración de la propuesta

Entrenamiento del Sistema de aprendizaje/clasificación de arándanos, utilizando diferentes algoritmos. Se selecciona el más eficiente.

3.- Aplicación de la propuesta

Pruebas de la clasificación automática sobre los grupos experimental de arándanos.

4.- Obtención y Discusión de resultados

5.- Conclusiones y recomendaciones

3.1.4. OPERACIONALIZACIÓN DE VARIABLES

VARIABLES	Definición Conceptual	Indicadores	Tipo	Técnica	Instrumento
V. Independiente El uso de espectroscopia con infrarrojo y técnicas de Machine Learning y Deep Learning	Sistema que soporta un conjunto de especificaciones de funcionalidades de aprendizaje de una base de datos de espectros NIR de arándanos, en una versión de prototipo.	% Precisión en la detección de arándanos durante la etapa de entrenamiento. (madurez, forma, tipo)	Cuantitativo Cuantitativo	Observación Encuestas	Guía de observación Cuestionario Matriz de análisis
V. Dependiente Detección y Clasificación de arándanos en forma automática en la empresa TalSA.	Predicción sobre características de los arándanos (madurez y) y la clasificación de arándanos de diferentes tipos.	Exactitud: % de arándanos clasificados correctamente (madurez, forma, tipo)	Cuantitativo Cuantitativo	observación Encuestas	Registro anecdótico Cuestionario Matriz de análisis

CAPÍTULO IV

RESULTADOS

CAPÍTULO IV: RESULTADOS

A continuación, se desarrolla cada uno de los objetivos específicos propuestos en esta investigación.

4.1. Revisión del estado del arte de Machine Learning, Deep Learning, y Deep Neural Networks.

El campo es demasiado variado, depende mucho del área de aplicación, tal como: computer vision, procesamiento del lenguaje natural, reconocimiento de voz, etc.

El Aprendizaje automático o aprendizaje de máquinas (Machine Learning); es una disciplina de la inteligencia artificial que se dedica a desarrollar técnicas que permitan a un programa aprender sobre la experiencia. Lo hace a través de algoritmos, mejora de herramientas, recopilación y análisis de información.

Las plataformas de Machine Learning tienen muchas aplicaciones como, por ejemplo; el refinamiento y personalización de la interacción entre máquina y usuario y hacer predicciones de comportamiento y clasificación en marketing como la optimización la entrega de publicidad pagada y contenido a las audiencias seleccionadas.

Otra aplicación es la Lingüística Computacional o la Generación Natural del Lenguaje; que es la colocación de las palabras en un orden específico tal que transmita información y además emoción como un arte. La escritura requiere no solo de conocimiento de las reglas del lenguaje, sino también de creatividad.

La Lingüística Computacional es el campo que mezcla lingüística, informática e inteligencia artificial, pues se encarga de perfeccionar las interacciones entre las máquinas y los lenguajes humanos. Es un tema muy complejo y para solucionar este problema, la Generación Natural del Lenguaje ha empezado a aplicar sus avances en la generación de reportes y análisis de mercado.

Su objetivo es desarrollar sistemas capaces de entender el lenguaje humano y generarlo. Para lograrlo, utiliza análisis de texto, estadística y Machine Learning para entender la estructura de oraciones, intención, significado, estilo, etc.

Sus aplicaciones más conocidas son la traducción automática de idiomas, sistemas de recuperación de información y producción automática de textos.

Las Plataformas de Aprendizaje Profundo (Deep Learning) es una de las áreas de mayor crecimiento en inteligencia artificial. El Deep Learning utiliza algoritmos multinivel para crear diferentes niveles de abstracción de información. Algunas aplicaciones de esta son el reconocimiento de imágenes y voz; predicción de tendencias digitales y perfeccionamiento de procesos.

A continuación, se enumeran algunos tipos de aplicaciones por áreas

Aplicaciones en medicina y Psicología

Se está trabajando en un sistema que trata identificar personas con tendencia al suicidio.

El diagnóstico médico aplicando el aprendizaje automático permite generar mejores diagnósticos.

En la Universidad de Harvard han demostrado que los sistemas automáticos funcionan con datos objetivos (ecografías, análisis clínicos, tomografías).

Utilizando el aprendizaje automático, un grupo de investigadores en Estados Unidos han desarrollado un sistema que trata de relacionar los rasgos del rostro, con los sentimientos de las personas.

En Singapur están tratando de crear sillas de rueda autónomas que permitiría que los pacientes por el hospital puedan desplazarse sin ayuda de otras personas.

Se tiene investigación en el ámbito farmacológico: el descubrimiento de nuevos medicamentos podría ser mucho más eficiente y veloz gracias a la utilización de sistemas de inteligencia artificial y, en concreto, de aprendizaje máquina. (Pastor, J., 2016).

Aplicaciones para aprendizaje de idioma

En el MIT se ha creado una base de datos de palabras en inglés no nativo. El objetivo es perfeccionar el procesamiento de las máquinas del idioma. Se utiliza el aprendizaje automático al procesamiento del lenguaje (NLP).

Aplicaciones para control en redes sociales

En facebook se ha desarrollado un sistema, utilizando aprendizaje automático, que permite detectar contenidos prohibidos o censurados, en las retransmisiones en vivo y los bloque automáticamente.

El uso de los algoritmos de aprendizaje automático también ofrece soluciones para el spam y el envío de noticias falsas.

Aplicaciones en astronomía

En la Universidad de Toronto se hacen uso del aprendizaje automático, para tratar de acelerar la búsqueda de sistemas planetarios.

Lo que se busca es acelerar la búsqueda de planetas.

Reconocimiento de imágenes y objetos

El aprendizaje automático ha demostrado su utilidad en los sistemas de reconocimiento de imágenes. Se pueden utilizar para la organización de bibliotecas de fotos, reconocimiento de objetos y reconocimiento de personas en tiempo real.

Con el aprendizaje automático se ha conseguido que los algoritmos que hacen reconocimiento facial, identifiquen rostros y sirven para clasificar fotos en internet y hacer búsquedas muy precisas.

Aplicaciones para televisión, autos, celulares, etc.

Los algoritmos de aprendizaje automático son capaces de reconocer lo que está pasando en varios deportes y añadir comentarios al vídeo que se está emitiendo con apenas retraso.

Se han desarrollado sistemas basados en aprendizaje automático que están destinados a evitar errores al volante.

Para esto, con una cámara se analiza el lenguaje corporal y gestos para tratar de entender qué reacciones se tiene antes de realizar ciertas maniobras.

Huawei presentó su nuevo smartphone que cuenta con un software que aprende del usuario para mejorar la experiencia de uso.

El aprendizaje automático puede aportar ayuda en la comunicación entre los seres humanos y las máquinas.

Aplicaciones en la música

Se hace uso de sistemas de aprendizaje automático para descubrir el tipo de música que gusta al usuario, y mezclarlo con datos como ubicación, actividad o el clima para crear una lista de reproducción personalizada.

Se tienen aplicaciones que tras aprender la forma en la que estaban compuestas ciertas obras de música clásica, puede crear piezas que imitan estas composiciones y armonizar canciones existentes bajo ese mismo estilo.

Aplicaciones para optimización de procesos

El aprendizaje automático permitirá optimizar procesos y decisiones. Algunos, tan útiles como lograr que se ahorre dinero en la factura de la luz.

Como se puede apreciar, los avances en esta disciplina, son bastante amplios y en diferentes áreas. Todos ellos requieren de grandes volúmenes de información, para que se pueda realizar el entrenamiento de sus redes, para obtener los resultados esperados, los mismos que podrían cambiar nuestro mundo para siempre.

4.2. El uso de espectroscopia en el reconocimiento de frutas y sus propiedades

Se realizó una revisión del uso de espectroscopia en el reconocimiento de frutas y sus propiedades logrando identificar algunas características de cada uno de ellos, ventajas, inconvenientes y expectativas.

4.2.1. El grado de madurez de un fruto se puede determinar en base al contenido de antocianinas a partir de las longitudes de onda que refleja el fruto al ser sometido a radiaciones en el rango del infrarrojo cercano (NIR).

La idea es contar con un sensor de espectrofotometría de bajo costo que proporcione el contenido de antocianinas, y otros datos como el índice de madurez y firmeza de las frutas. Una sola herramienta, de bajo costo para poder monitorear continuamente la fruta.

Se comprueba la utilidad potencial de la espectroscopia NIR para predecir el nivel de antocianinas. La validación del modelo para dicha temporada arrojó un valor de 80% de efectividad

Se espera que no sea necesario tomar una muestra, sino solamente acercar un dispositivo al fruto y se obtendrá los datos requeridos. (RedAgrícola, 2018)

4.2.2. Se ha encontrado que, el color de la pulpa del melón, espárrago, tomate, pimiento verde, es uno de los índices utilizados para determinar el grado de madurez y calidad del mismo, siendo necesario disponer de medidas no destructivas que impidan el daño y la depreciación del producto. La tecnología NIRS ofrece enormes expectativas en este terreno, particularmente derivadas de su carácter de no destructiva, rápida y con posibilidad de ser incorporada a nivel de la línea de producción. Este trabajo, evaluó un instrumento NIRS de red de diodos para estimar el color del mesocarpio de

melones intactos. El desarrollo de los modelos de predicción de color se realizó utilizando un colectivo constituido por 158 muestras procedentes de Almería-España. Los valores de efectividad del orden de 88% en la fruta intacta indican la viabilidad inicial del empleo de la tecnología NIRS para la determinación de la madurez y calidad de frutos enteros (Flores et.al, 2009).

4.2.3 Ventajas e inconvenientes de la espectroscopia NIR

Ventajas:

La técnica no es destructiva ni invasiva, la muestra no requiere preparación. La obtención del espectro es rápida. Aumenta la capacidad analítica del laboratorio debido a la rapidez con la que se obtiene el espectro, permite el análisis de varios parámetros de forma simultánea.

Es una técnica idónea para procesos de control en planta debido a la ausencia de partes móviles en el aparato. Hay una tendencia hacia la miniaturización y compactación de los instrumentos.

Inconvenientes:

El precio de adquisición del espectrofotómetro NIR es elevado.

La complejidad de la señal NIR obliga a aplicar técnicas estadísticas complejas para la obtención de los modelos.

Es necesario un modelo de calibración distinto para cada parámetro que se necesite medir. El calibrado es difícil debido al gran número de muestras que es necesario medir para que el modelo de calibrado sea fiable. (Dolz, 2008).

Se ha encontrado que existen diversos proyectos y trabajos de investigación, que, en promedio desde hace diez años, se proponen aplicar el uso de la espectroscopia del rango del infrarrojo cercano NIR, para que, de forma no destructiva, se pueda determinar las características y propiedades que identifican a diferentes frutos, así como también su grado de madurez y otras propiedades físicas y químicas. Asimismo, se establecen las ventajas e inconvenientes de utilizar estos métodos de manera aislada. Debido a la complejidad de los datos obtenidos en los espectros NIR los obliga a aplicar técnicas matemáticas y estadísticas complejas para la obtención de los respectivos modelos.

4.3. Análisis el proceso selección y clasificación de arándanos en la empresa TalSA

Se realizó el análisis del proceso de selección y clasificación de arándanos en la empresa TalSA, encontrándose lo siguiente:

El proceso completo de arándano fresco, desde que llega a la empresa, empieza con el acopio de la fruta e inmediatamente se almacena en un ambiente sometido a enfriamiento a cero grados centígrados. De este almacenamiento, la materia prima se va ingresando a la línea de producción, se recepciona e ingresa al acondicionamiento en línea de proceso. Es en esta etapa, es donde se realiza la clasificación de los arándanos, por tamaños. Esto se consigue utilizando una faja transportadora que tiene orificios de diámetros determinados a través de los cuales van separándose los frutos de menor diámetro. A lo largo de la faja transportadora, los arándanos son observados por los trabajadores, quienes van separando aquellos que se encuentran en condiciones de calidad inadecuadas. Los arándanos que continúan en la línea de proceso son vertidos en contenedores del tamaño requerido y luego estos recipientes son sellados, etiquetados y paletizados, para luego ser sometidos nuevamente a un proceso de enfriamiento como producto terminado. Después son almacenados en una cámara de producto terminado a temperatura de cero grados centígrados. De este almacén son despachados a los clientes en el extranjero.

La clasificación se realiza en forma manual, en base a la observación y experiencia de los trabajadores.

En esta actividad se requiere de un trabajador con experiencia en estas labores. Asimismo, este trabajador puede realizar sus labores de manera eficiente un determinado tiempo, luego del cual su eficiencia disminuye.

En esta etapa de acondicionamiento de arándano en las líneas de proceso, en la selección y clasificación de arándanos para su envasado y posterior comercialización, en la empresa TalSA se observa lo siguiente:

- Empleo de muchos trabajadores en el proceso de selección y clasificación de los arándanos, lo que origina un sobre costo en mano de obra
- Se requiere que los trabajadores tengan experiencia para reconocer y diferenciar los frutos de calidad y los frutos defectuosos para poder realizar la clasificación.
- Un trabajador, aunque cuente con mucha experiencia, puede realizar sus labores de manera eficiente un determinado tiempo, luego del cual su eficiencia disminuye.

- Existen equipos que pueden realizar este trabajo de clasificación, pero son muy costosos, sus precios están en el orden de 1'250,000 Euros y darle mantenimiento tiene también un alto costo

4.4. Propuesta de una aplicación utilizando espectroscopia NIR y deep learning

Se propone el siguiente procedimiento:

4.4.1 Formación de la Base de Datos

En primer lugar, se procedió a realizar la captura de imágenes espectrales de los arándanos, utilizando un equipo de espectroscopia de infrarrojo cercano (NIR), este equipo, emite una radiación sobre el fruto, obteniéndose las longitudes de onda correspondientes al espectro infrarrojo cercano. El equipo de espectroscopia, está conectado a una computadora, la cual tiene instalado el software que viene con el equipo NIR, el mismo que permite que los espectros sean digitalizados automáticamente, formándose la base de datos de longitudes de onda digitalizadas, correspondiente a cada arándano.

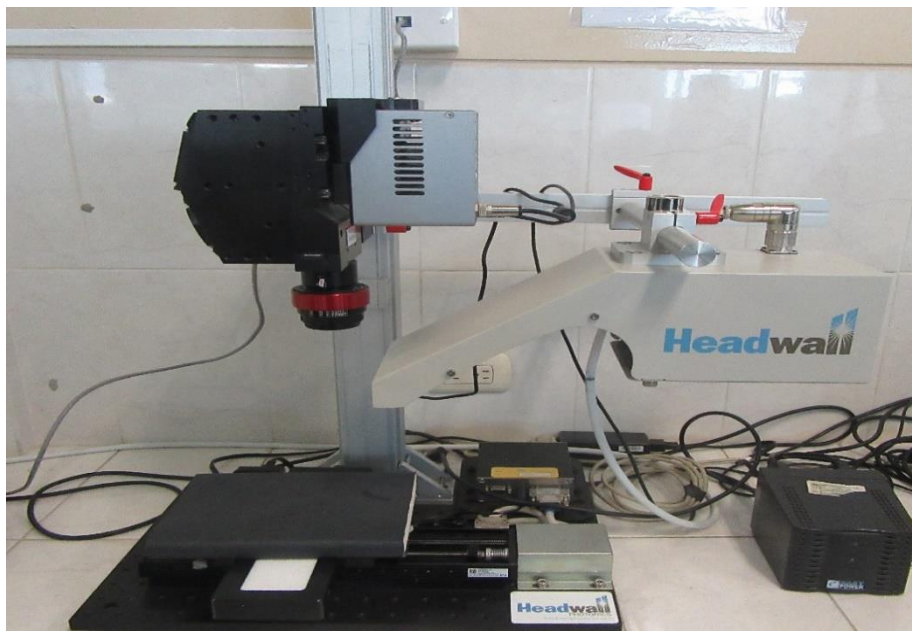


Fig. 16 :Equipo de espectroscopia NIR - HeadWall Photonics

Fuente: Elaboración propia



Fig. 17 : Equipo de espectroscopia NIR conectado a la computadora

Fuente: Elaboración propia



Fig. 18 : Equipo de espectroscopia NIR y la computadora en funcionamiento

Fuente: Elaboración propia

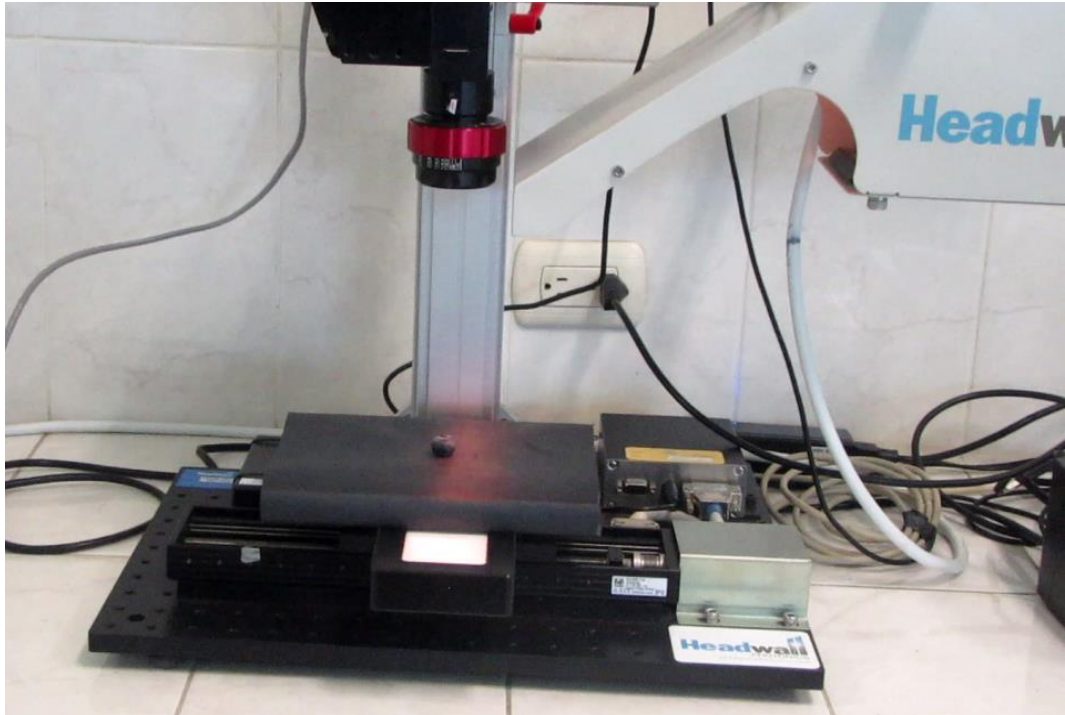


Fig. 19 : Proceso de emisión de las radiaciones y captura de espectros NIR

Fuente: Elaboración propia

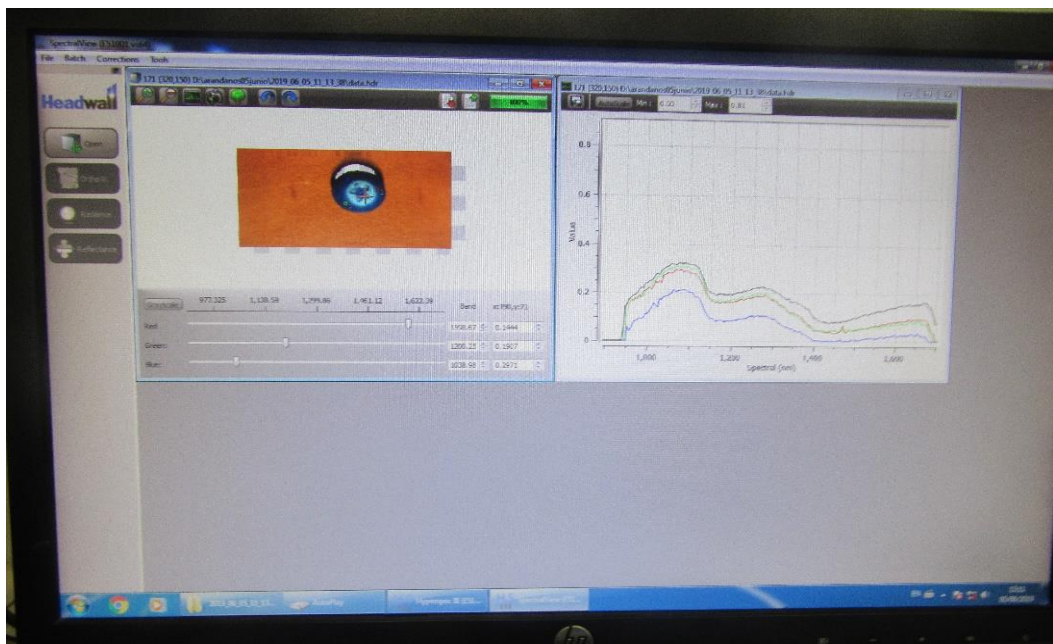


Fig. 20 : Proceso de captura de espectros NIR y digitalización

Fuente: Elaboración propia

Unnamed: 0	LONGITUD DE ONDA	LONGITUD DE ONDA.1	LONGITUD DE ONDA.2	LONGITUD DE ONDA.3	LONGITUD DE ONDA.4	LONGITUD DE ONDA.5	LONGITUD DE ONDA.6
ARÁNDANO 1	20679	26910	82303	137658	193173	248779	304408
ARÁNDANO 2	21096	27339	100644	156218	211787	267463	322899
ARÁNDANO 3	21029	27265	100779	156414	211885	267567	323135
ARÁNDANO 4	21068	27261	109358	164959	220459	276062	331624
ARÁNDANO 5	21141	27348	73524	129057	184627	240182	295823
ARÁNDANO 6	21146	27341	118071	173676	229231	284799	340428
ARÁNDANO 7	21023	27271	97894	153469	208984	264658	320181
ARÁNDANO 8	21139	27370	93544	149059	204611	260264	315808
ARÁNDANO 9	20958	27176	119725	175301	230831	286474	341987
ARÁNDANO 10	21113	27317	77146	132722	188296	243857	299445
ARÁNDANO 11	21162	27356	75732	131294	186869	242398	298001
ARÁNDANO 12	21162	27371	115804	171435	226934	282587	338101
ARÁNDANO 13	20803	27018	108754	164292	219832	275561	331034
ARÁNDANO 14	20541	26738	112509	168020	223621	279243	334760
ARÁNDANO 15	20824	27076	84650	140131	195664	251268	306845
ARÁNDANO 16	20683	26898	88916	144412	199960	255671	311086
ARÁNDANO 17	21042	27293	114792	170334	225913	281531	337031
ARÁNDANO 18	21122	27309	72798	128317	183950	239420	295167
ARÁNDANO 19	20764	27001	97112	152627	208180	263788	319396

Fig. 22 : Datos obtenidos de espectros NIR digitalizados

Fuente: Elaboración propia

En la imagen anterior, las columnas son el conjunto de valores que corresponden a las longitudes de onda y se selecciona aquellas que cambian significativamente de una medición a otra, y se representaron con la variable X. Adicionalmente se tiene el valor de la última columna, que se identifica con la variable Y, que almacena 1's o 0's. El valor 1 hace referencia a un arándano que se encuentra en óptimas condiciones de color, textura, forma y sabor, mientras que el valor 0 se refiere a un arándano que no cumple con las condiciones antes indicadas, ya sea por demasiada acidez o mal estado del arándano.

En la matriz se tienen las filas y columnas separadas por ','. Se utiliza ':' para hacer referencia a todas las filas, y en la lista después de la coma se hace referencia a todas las columnas del set de datos, en el caso de la variable dependiente Y, se seleccionan todas las filas y con el valor -1, se hace referencia solamente a la última columna.

En resumen, lo que hace este código es guardar el valor de todas las filas de las columnas específicas en las variables 'X' y 'Y'.

```
X = dataset.iloc[:, [2, 19, 37, 38, 55, 56, 73, 74, 91, 92, 109, 110, 111, 127, 128, 145, 146]].values
y = dataset.iloc[:, -1].values
```

Importando las librerías necesarias

En primer lugar, se debe importar las librerías necesarias. Entre estas se tienen las librerías de numpy, TensorFlow (éste será el framework sobre el que correrá Keras), Keras y unas librerías necesarias Scikit Learn, Pandas, etc.

4.4.3. Pasos para crear la red neuronal con python:

1.-Importar las librerías que serán necesarias para la obtención de los datos

```
1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
```

2.-Se importa el dataset, utilizando la librería pandas. Para esto se crea una variable y se almacena en ella los datos que se procesarán con la red neuronal

```
dataset = pd.read_csv('arandanosxls2.csv')
```

3.- El siguiente paso es dividir el conjunto de datos o dataset, en un conjunto de variables independientes (X) y la variable dependiente o variable a predecir (Y).

```
X = dataset.iloc[:, [3, 20, 38, 39, 56, 57, 74, 75, 92, 93, 110, 111, 112, 128, 129, 146, 147]].values
y = dataset.iloc[:, -1].values
```

La variable independiente, como se indicó anteriormente, toma los valores 2, 19, 37, etc., y hace referencia a las columnas o longitudes de onda que se seleccionaron del dataset, como las más importantes. Por Ejemplo, aquí se puede ver que la columna correspondiente a 'LONGITUD DE ONDA' y 'LONGITUD DE ONDA 1' que corresponden a las columnas 1 y 2, sus valores no varían o varían muy poco, esto nos indica que estas longitudes de onda no interactúan de manera apreciable con el arándano, así que lo que se hace en el código, es solo considerar aquellas columnas o longitudes de onda que tienen una interacción apreciable con el arándano como se ve en la columna de nombre 'LONGITUD DE ONDA 2' que corresponde al

índice 3 del dataset, así que solo se seleccionan las columnas o longitudes de onda que se puede ver que interactúan de manera significativa con el arándano, las cuales serían: 3, 20, 38, 39, 56, 57, 74, 75, 92, 93, 110, 111, 112, 128, 129, 146, 147 y se obtienen los valores para cada arándano.

La variable dependiente Y, es el valor que se desea predecir. Tomará los valores 0 o 1. Se trata de encontrar una relación entre la longitud de onda absorbida por los arándanos y sus características en óptimas condiciones.

Unnamed: 0	LONGITUD DE ONDA	LONGITUD DE ONDA.1	LONGITUD DE ONDA.2	LONGITUD DE ONDA.3	LONGITUD DE ONDA.4	LONGITUD DE ONDA.5	LONGITUD DE ONDA.6
ARÁNDANO 1	20679	26910	82303	137658	193173	248779	304408
ARÁNDANO 2	21096	27339	100644	156218	211787	267463	322899
ARÁNDANO 3	21029	27265	100779	156414	211885	267567	323135
ARÁNDANO 4	21068	27261	109358	164959	220459	276062	331624
ARÁNDANO 5	21141	27348	73524	129057	184627	240182	295823
ARÁNDANO 6	21146	27341	118071	173676	229231	284799	340428
ARÁNDANO 7	21023	27271	97894	153469	208984	264658	320181
ARÁNDANO 8	21139	27370	93544	149059	204611	260264	315808
ARÁNDANO 9	20958	27176	119725	175301	230831	286474	341987
ARÁNDANO 10	21113	27317	77146	132722	188296	243857	299445
ARÁNDANO 11	21162	27356	75732	131294	186869	242398	298001
ARÁNDANO 12	21162	27371	115804	171435	226934	282587	338101
ARÁNDANO 13	20803	27018	108754	164292	219832	275561	331034
ARÁNDANO 14	20541	26738	112509	168020	223621	279243	334760
ARÁNDANO 15	20824	27076	84650	140131	195664	251268	306845
ARÁNDANO 16	20683	26898	88916	144412	199960	255671	311086
ARÁNDANO 17	21042	27293	114792	170334	225913	281531	337031
ARÁNDANO 18	21122	27309	72798	128317	183950	239420	295167
ARÁNDANO 19	20764	27001	97112	152627	208180	263788	319396

Fig. 23 : Datos obtenidos de espectros NIR para entrenamiento

Fuente: Elaboración propia

- Se importan las librerías que servirán para dividir el set de datos en entrenamiento y testing. En este caso se elige un conjunto de test del 25%.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

Se utiliza la librería `train_test_split`, a la que se le pasa como parámetros el conjunto de datos X y Y. Asimismo se indica que el conjunto de datos para hacer el testing será el 25% del total de datos.

5.- Se importan las librerías para escalar las variables. Esto se hace para evitar que algunos datos dominen sobre otros.

```
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
```

6.- Se importan las librerías para crear la red neuronal. Para esto se utiliza keras, que corre sobre tensor Flow.

```
import keras
from keras.models import Sequential
from keras.layers import Dense
```

7.- Hay dos maneras de iniciar la red neuronal. Definiendo la secuencia de capas o definiendo el grafo de la relación de capas. En este caso, como va a ser una red neuronal que clasifique arándanos se le pone por nombre classifier. Se elige una secuencia de capas, donde cada capa se define como:

```
classifier = Sequential()
```

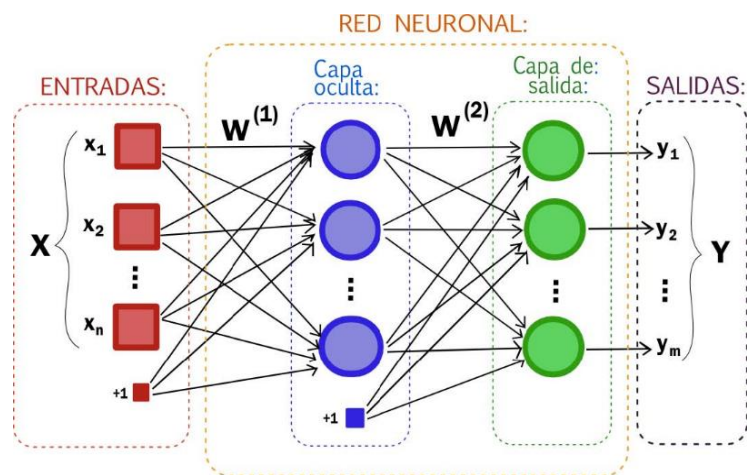


Fig. 24 : Esquema de una red neuronal

Fuente: <https://artfromcode.wordpress.com/2017/04/18/red-neuronal-en-python-con-numpy-parte-1/>

Una red neuronal se compone de tres capas: la capa de entrada, la capa oculta y la capa de salida.

La capa de entrada corresponde a las longitudes de onda obtenidas de cada arándano o variables independientes. La capa de salida corresponde a la variable de salida, la que se desea predecir. En la capa oculta se encuentran

las neuronas. Las neuronas de cada capa, tienen una función de activación. Depende de los datos que le lleguen o como se combinen estos datos para que una neurona se active y deje pasar información o determine que no es suficiente y no envíe nada para la salida.

8.- Luego se empieza a añadir las capas. La capa de entrada y la primera capa oculta. Con la función `add` se declara la primera capa. El parámetro `input_dim`, indica el número de nodos de la capa de entrada. El valor es 171 porque se tienen 171 valores de longitudes de onda por cada arándano, desde 0 hasta 170.

Primero se debe considerar las funciones de activación de cada capa. Las funciones de activación son funciones matemáticas que permitirán a la neurona la activación o la no activación de la misma. Si los datos que llegan a la neurona son suficientes para activar a la neurona, entonces esta despierta, procesa la información que le llega y la envía a la siguiente capa ya sea oculta o final. Las funciones de activación son las que deciden si la información que llega a la neurona es lo suficientemente importante como para despertar a la neurona. Estas son algunas funciones de activación:

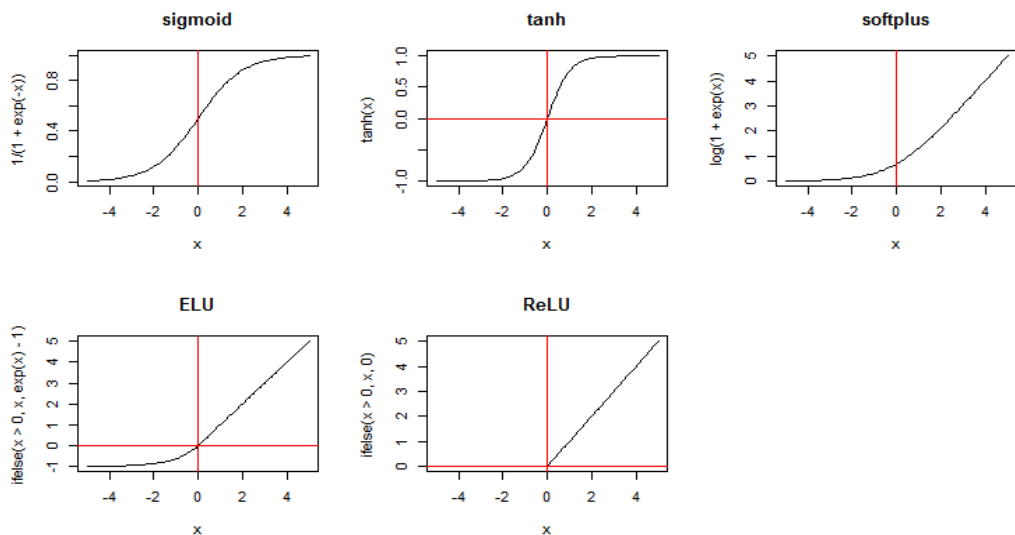


Fig. 25 : Algunas funciones de activación

Fuente: <https://www.mql5.com/es/articles/3473>

Aquí se define la capa de entrada y la primera capa oculta, donde los datos de entrada serán los tipos de datos que se tienen en el set de datos, ya sea nombre, edad, en este caso conjunto de rangos de longitud de onda. Se

```
classifier.add(Dense(units = 86, kernel_initializer = 'uniform', activation = 'tanh', input_dim = 171))
```

define el numero de neuronas, esto es a criterio de cada persona, se puede elegir el que permita obtener el mejor resultado; es experimentar en base a prueba y error.

La función de activación de la neurona es una tangente hiperbólica porque ayuda a centrar los datos y ayuda que el aprendizaje para la siguiente capa sea más fácil. El parámetro `kernel_initializer`, permite inicializar la red neuronal asignándoles pesos de manera uniforme y luego los va actualizando iterativamente a mejores valores.

El parámetro `units`, determina el número de neuronas de la primera capa oculta. En este caso es 86, ya que se ha calculado sumando los nodos de la capa de entrada más los nodos de la capa de salida y se divide entre 2. $(171+1)/2 = 172/2 = 86$.

No hay una regla estricta que indique cuantas neuronas se deben colocar.

Luego se define la capa de salida, que tendrá una sola neurona y se utilizará una función sigmoidea porque es útil para una clasificación binaria como 0 o 1, verdadero o falso, arándano en óptimas condiciones o no. Dado que el valor de una función sigmoidea está entre 0 y 1, se puede establecer que el resultado será 1 si es mayor o igual que 0.5, en caso contrario será 0. Por lo tanto, esta función sigmoidea nos dirá que tan probable es que el arándano se encuentre en buenas condiciones.

Se puede definir una segunda capa oculta con el mismo número de neuronas, se puede seleccionar más o menos neuronas, eso queda a elección, asignándole una función específica.

```
classifier.add(Dense(units = 72, kernel_initializer = 'uniform', activation = 'tanh'))
```

Por último, se define la capa de salida, la cual cuenta con una única neurona, ya que según el peso de los datos que le lleguen se activará o no. En este caso se utiliza una función de activación de tipo sigmoidea. Esta función transforma los valores que la activan a una escala de 0, 1, es parecida a la

función de tangente hiperbólica, se elige esta y no la función de tangente hiperbólica porque de acuerdo al juicio de expertos en el área de DATA SCIENTIST es la función que mejor rendimiento tiene en la última capa de una red neuronal de clasificación.

```
classifier.add(Dense(units = 1, kernel_initializer = 'uniform', activation = 'sigmoid'))
```

9.-Ahora se compila la red neuronal, con parámetros para optimizar los pesos, elegir el tipo de error a ajustar, y unas métricas para juzgar el rendimiento del modelo.

Se elige el tipo de optimizador ADAM porque mantiene una tasa de aprendizaje para cada peso en la red neuronal y se adapta por separado a medida que se desarrolla el aprendizaje, en otras palabras, cada peso de cada sinapsis se evalúa por separado y se va mejorando a medida que la red neuronal aprende. Otros motivos para trabajar con el optimizador ADAM es que es computacionalmente eficiente, es apropiado para objetivos que cambian con el tiempo, apropiado para problemas con gradientes muy ruidosos o dispersos y también que es muy adecuado para problemas con un gran conjunto de datos.

Se elige la pérdida de “Binary Cross Entropy”, ya que mide el rendimiento de un modelo clasificación binario (1 o 0, Si o No, True o False) cuyo resultado es un valor de probabilidad entre 0 y 1. La pérdida de entropía cruzada aumenta a medida que la probabilidad predicha difiere de la observación real.

Por último, se elige el tipo de métrica que se utiliza para monitorizar a la red neuronal la cual es “Accuracy”, con la cual se evalúa el porcentaje de aciertos, averiguando donde la red neuronal predice en mayor o menor número la correcta selección, es decir permite ver como está aprendiendo la red neuronal.

```
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

```
Epoch 600/600  
170/170 [=====] - 0s 108us/step  
- loss: 0.6155 - acc: 0.6412
```

10.- Ahora se entrena a la red neuronal con el conjunto de entrenamiento previamente creado, donde se corrigen los pesos y se seleccionan el número de iteraciones.

Lo que se indica en “batch_size” es cada cuantas iteraciones la red neuronal debe de ajustar su modelo en base a lo aprendido, cada vez que llegue al número de iteraciones requeridas, la red neuronal hará una propagación hacia atrás, esta es la parte donde la capa de salida envía el error hacia atrás a cada neurona que contribuyó a la estimación errónea; pero cada neurona no recibe todo el error, sino que cada neurona recibe una porción del error en base a su contribución o peso en la misma y así cada neurona ajustará los pesos los cuales serán usados en las siguientes iteraciones hasta que se vuelva a producir la propagación hacia atrás.

“Epochs” indica el numero total de iteraciones que hará la red neuronal con los mismos datos, no hay una regla para elegir estos parámetros, esto depende de cada persona, pero se debe tener en cuenta que se podría caer en un sobre ajustes del modelo (lo que indicaría más una memorización que aprendizaje de parte de la red neuronal) si se coloca demasiadas iteraciones o muy poco tamaño de lotes para reajustar el modelo, se debe de buscar el equilibrio.

```
classifier.fit(X_train, y_train, batch_size = 15, epochs = 600)
```

```
Epoch 584/600
170/170 [=====] - 0s 109us/step
- loss: 0.6226 - acc: 0.6471
Epoch 585/600
170/170 [=====] - 0s 106us/step
- loss: 0.6220 - acc: 0.6529
Epoch 586/600
170/170 [=====] - 0s 109us/step
- loss: 0.6234 - acc: 0.6529
Epoch 587/600
170/170 [=====] - 0s 110us/step
- loss: 0.6259 - acc: 0.6471
Epoch 588/600
170/170 [=====] - 0s 153us/step
- loss: 0.6224 - acc: 0.6529
Epoch 589/600
170/170 [=====] - 0s 109us/step
- loss: 0.6256 - acc: 0.6529
Epoch 590/600
170/170 [=====] - 0s 105us/step
- loss: 0.6230 - acc: 0.6529
Epoch 591/600
170/170 [=====] - 0s 106us/step
- loss: 0.6225 - acc: 0.6471
```

```
classifier.fit(X_train, y_train, batch_size = 20, epochs = 500)
loss: 0.1662 - acc: 0.9600
Epoch 338/500
75/75 [=====] - 0s 67us/step -
loss: 0.1647 - acc: 0.9733
Epoch 339/500
75/75 [=====] - 0s 85us/step -
loss: 0.1635 - acc: 0.9467
Epoch 340/500
75/75 [=====] - 0s 70us/step -
loss: 0.1606 - acc: 0.9600
Epoch 341/500
75/75 [=====] - 0s 68us/step -
loss: 0.1586 - acc: 0.9733
Epoch 342/500
75/75 [=====] - 0s 93us/step -
loss: 0.1581 - acc: 0.9600
Epoch 343/500
75/75 [=====] - 0s 68us/step -
loss: 0.1641 - acc: 0.9733
Epoch 344/500
75/75 [=====] - 0s 63us/step -
```

Se puede ver como va aumentando la exactitud a medida que la red neuronal va aprendiendo. Por ejemplo, se aprecia una exactitud de 96, 97, 94, 96, 97, 96, 97 %.

4.5. Uso de la aplicación propuesta en la detección y clasificación de arándanos de la empresa TalSA.

Esto se hace en la etapa de testing, utilizando la función predict. Se aplica la red neuronal al conjunto de testing para predecir el resultado y compararlo con los resultados reales y calcular el rendimiento de la red neuronal.

Se elige el umbral de 0.5 porque se considera que es suficientemente probable que el arándano tenga buenas condiciones por encima de 50% de probabilidad, lo cual clasificará al valor de la predicción como 1 si esta por encima del umbral y como 0 si esta por debajo.

```
y_pred = classifier.predict(X_test)
y_pred = y_pred > 0.5
```

```

loss: 0.0880 - acc: 1.0000
Epoch 496/500
75/75 [=====] - 0s 69us/step -
loss: 0.0888 - acc: 1.0000
Epoch 497/500
75/75 [=====] - 0s 82us/step -
loss: 0.0920 - acc: 1.0000
Epoch 498/500
75/75 [=====] - 0s 64us/step -
loss: 0.0899 - acc: 1.0000
Epoch 499/500
75/75 [=====] - 0s 101us/step -
loss: 0.0876 - acc: 1.0000
Epoch 500/500
75/75 [=====] - 0s 269us/step -

```

Se puede apreciar que la exactitud tiende a ser 1 o su equivalente a 100% en la mayoría de los casos.

4.6. Análisis estadístico e interpretación de los resultados

Se utiliza una matriz de confusión para ayudar a ver el número de aciertos y fallos y poder predecir el rendimiento de la red neuronal.

```

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)

```

Aquí se puede ver claramente los 0's que la red neuronal encontró como 0's son 140, y los 1's que la red neuronal encontró como 1's son 90, las demás cifras son los errores que cometió la red neuronal. Con estos valores se puede obtener el rendimiento de la red neuronal simplemente sumando los aciertos y dividiéndolo entre el total de números del conjunto de testing.

	0	1
0	140	10
1	10	90

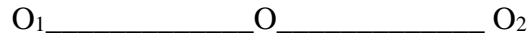
$$\text{Rendimiento} = (140 + 90) / 250$$

Rendimiento obtenido de 92 %

CAPÍTULO V

DISCUSIÓN

Post-Test (O₂): Corresponde a la nueva medición de la variable dependiente a ser utilizada.



O₁: Proceso de clasificación de arándanos sin utilizar la aplicación propuesta.

O: Aplicación la aplicación propuesta

O₂: Proceso de clasificación de arándanos utilizando la aplicación propuesta.

En base a los datos recolectados, durante la etapa de entrenamiento se obtuvo una exactitud promedio de hasta 97%.

```

loss: 0.1662 - acc: 0.9600
Epoch 338/500
75/75 [=====] - 0s 67us/step -
loss: 0.1647 - acc: 0.9733
Epoch 339/500
75/75 [=====] - 0s 85us/step -
loss: 0.1635 - acc: 0.9467
Epoch 340/500
75/75 [=====] - 0s 70us/step -
loss: 0.1606 - acc: 0.9600
Epoch 341/500
75/75 [=====] - 0s 68us/step -
loss: 0.1586 - acc: 0.9733
Epoch 342/500
75/75 [=====] - 0s 93us/step -
loss: 0.1581 - acc: 0.9600
Epoch 343/500
75/75 [=====] - 0s 68us/step -
loss: 0.1641 - acc: 0.9733
Epoch 344/500
75/75 [=====] - 0s 63us/step -

```

Fig. 26 : Exactitud de las mediciones en la etapa de entrenamiento

Fuente: Elaboración propia.

5.4 **Indicador 2:** Eficacia: porcentaje de arándanos clasificados correctamente

Este valor se obtiene al momento de utilizar la aplicación propuesta en modo test, obteniéndose un valor del 92% de eficacia.

	0	1
0	140	10
1	10	90

$$\text{Rendimiento} = (140 + 90) / 250$$

Rendimiento obtenido de 92 %

De acuerdo a la información proporcionada por los expertos de la empresa, se tiene los siguientes datos

La eficiencia en la selección de los arándanos, se maneja en base a un porcentaje máximo aceptable de defectos que se divide en dos tipos de defectos: calidad y condición. Éstos porcentaje varían según un rango de calificativo interno que va desde Excelente hasta Malo.

Para poder calcular los porcentajes se realiza un muestreo por cada paleta de producto terminado.

Producto Terminado: Se retiran entre 1 a 2 Kg (depende del formato que se procese) de distintos niveles de la paleta.

Para que un producto sea considerado bueno, el % de productos con defectos debe ser $\leq 10\%$.

Se utilizó la prueba χ^2 de Pearson, que se considera una prueba no paramétrica que mide la discrepancia entre una distribución observada y otra teórica, indicando en qué medida las diferencias existentes entre ambas, de haberlas, se deben al azar en el contraste de hipótesis. La fórmula que da el estadístico es la siguiente:

$$\chi^2 = \sum_i \frac{(\text{observada}_i - \text{teórica}_i)^2}{\text{teórica}_i}$$

Cuanto mayor sea el valor de X^2 , menos verosímil es que la hipótesis nula (que asume la igualdad entre ambas distribuciones) sea correcta. De la misma forma, cuanto más se aproxima a cero el valor de chi-cuadrado, más ajustadas están ambas distribuciones.

Se considera N-1, grados de libertad :

Criterio de decisión:

No se rechaza X_0 cuando . $X^2 < X_{\alpha}^2 ((r-1)(k-1))$, En caso contrario sí se rechaza.

Reemplazando datos: $X^2 = \frac{(92-90)^2}{90} = 0.044$

$$X^2 = 0.044 > X_{\alpha}^2 ((1-1)(1-1)) = 0$$

Como $X^2 = 0.044 > 0$ se rechaza la hipótesis nula y se acepta la hipótesis alterna.

H1: El uso de espectroscopia con infrarrojo y técnicas de Machine Learning y Deep Learning permitirán realizar la Detección y Clasificación de arándanos de forma automática en la empresa TalSA.

COMPARACIÓN DEL MÉTODO CON LOS ANTECEDENTES DE LA INVESTIGACIÓN

Antecedente 1

Trabajo de investigación, Fruit classification using computer vision and feedforward neural network.

Autores: Zhang, Y. Wang, S., Ji G., Phillips P. (2014).

Institución: U.S.A.

Análisis:

El trabajo tiene por objetivo el identificar y reconocer frutas con precisión. Utilizan un método de clasificación híbrido basado en el algoritmo de la colonia artificial de abejas caótica (FSCABC) y la red neuronal de alimentación directa (FNN). Su principal aporte al trabajo de investigación es que los resultados experimentales demostraron que el FSCABC – FNN fueron precisos y eficientes para diferenciar y reconocer las frutas.

En comparación al método propuesto, ellos utilizan visión por computadora, mientras que en el presente trabajo se utiliza los espectros infrarrojos cercanos (NIR) de los frutos, para en base a ellos caracterizar sus propiedades entrenando a las redes neuronales con la finalidad de poder identificar y diferenciar arándanos en óptimas condiciones, de aquellos que no lo están. Por lo que son trabajos que tienen diferentes enfoques.

Antecedente 2

Trabajo de investigación, DeepFood: Deep Learning-based Food Image Recognition for Computer-aided Dietary Assessment

Autor: Chang L., Yu C., Yan L., Guanling C., Vinod V., Yunsheng M

Institución: U.S.A.

Análisis:

Tiene como objetivo proponer un método basado en el aprendizaje profundo para enfrentar el problema del reconocimiento de imágenes de alimentos. Su principal aporte es que establecen algoritmos basados en redes neuronales convolucionales (CNN) con optimizaciones propuestas, como un modelo y una técnica de convolución optimizados.

Igual que en el caso anterior este trabajo se orienta al reconocimiento de imágenes utilizando algoritmos de redes neuronales, mientras que en el presente trabajo se utiliza redes neuronales con espectros electromagnéticos de infrarrojo cercano, que caracterizan al arándano en óptimas condiciones.

Antecedente 3

Trabajo de investigación: Deep Learning Based Food recognition

Autor: Qian Yu et.al

Institución: Stanford University. U.S.A.

Análisis:

El trabajo de investigación presenta un método de visión por computadora que tiene por objetivo reconocer y evaluar de manera eficiente la composición de los alimentos.

El principal aporte al trabajo de investigación es que permite el reconocimiento de alimentos con un método basado en redes neuronales convolucionales (CNN).

Se asemeja en algo al presente trabajo de investigación con la diferencia que utiliza un método de visión por computadora para reconocer y evaluar composición de alimentos, mientras que en el presente trabajo se utiliza el espectro electromagnético infrarrojo para identificar un alimento y determinar si se encuentra en condiciones óptimas o no.

Antecedente 4

Trabajo de investigación: Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. Computational Intelligence and Neuroscience

Autor: Sladojevic S., Arsenovic, M., Anderla A., Culibrk, D. Stefanović, D

Institución: U.S.A.

Análisis:

El trabajo de investigación presenta los resultados obtenidos utilizando una nueva generación de redes neuronales convolucionales (CNN) para el reconocimiento de imágenes. El principal aporte al trabajo de investigación es un modelo de reconocimiento de enfermedades de las plantas, que se basa en el tratamiento de imágenes de las hojas, mediante el uso de redes convolucionales profundas (CNN).

A diferencia del presente trabajo, propone un nuevo modelo de redes neuronales, mientras que en este trabajo se trata de reconocer y evaluar composición de alimentos, utiliza el espectro electromagnético infrarrojo para identificarlo y determinar su estado de madurez.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1. Se hizo una revisión del estado del arte de Machine Learning, Deep Learning, y Deep Neural Networks. Y se encontró que son áreas de mayor crecimiento en inteligencia artificial. Deep Learning utiliza algoritmos multinivel para crear diferentes niveles de abstracción de información. Algunas aplicaciones de esta son el reconocimiento de imágenes y voz; detección de fraudes, reconocimiento de caracteres manuscritos, descubrimiento de componentes farmacéuticos predicción de tendencias digitales y perfeccionamiento de procesos. Actualmente se aplica machine learning en muchas áreas tales como: el refinamiento y personalización de la interacción entre máquina y usuario, hacer predicciones de comportamiento y clasificación en marketing como la optimización la entrega de publicidad pagada y contenido a las audiencias seleccionadas. Se tienen también aplicaciones en medicina, psicología, lingüística, dispositivos móviles, procesamiento del lenguaje natural, robótica, controles en redes sociales, astronomía, reconocimiento facial, optimización de procesos, música, etc. Los avances en esta disciplina, requieren de grandes cantidades de datos para poder realizar los procesos de entrenamiento y lograr los resultados esperados que podrían cambiar nuestro mundo para siempre.
2. Se hizo una revisión del uso de espectroscopia en el reconocimiento de frutas y sus propiedades, encontrándose que existen diversos proyectos y trabajos de investigación, que, en promedio desde hace diez años, se proponen aplicar el uso de la espectroscopia del rango del infrarrojo cercano NIR, para que, de forma no destructiva, se pueda determinar las características y propiedades que identifican a diferentes frutos, tales como melón, espárrago, tomate, pimiento verde, arándanos, etc. así como también su grado de madurez y otras propiedades físicas y químicas. Asimismo, se establecen las ventajas e inconvenientes de utilizar estos métodos de manera aislada. Debido a la complejidad de los datos obtenidos en los espectros NIR los obliga a aplicar técnicas matemáticas y estadísticas complejas para la obtención de los respectivos modelos.
3. Se hizo el análisis del proceso de selección y clasificación de arándanos en la empresa TalSA. Se encontró que la selección y clasificación de los arándanos, para su

envasado y posterior comercialización, es parte de la etapa de acondicionamiento, en las líneas de proceso.

La clasificación se realiza en forma manual, en base a la observación y experiencia de los trabajadores.

Se requiere de un trabajador con experiencia para reconocer y diferenciar los frutos de calidad y los frutos defectuosos para poder realizar la clasificación de manera adecuada. Estas labores las puede realizar de manera eficiente un determinado tiempo, luego del cual su eficiencia disminuye.

Se emplea muchos trabajadores en el proceso de selección y clasificación de los arándanos, lo que origina un sobre costo en mano de obra

Existen equipos que pueden realizar este trabajo de clasificación, pero son muy costosos, sus precios están en el orden de 1'250,000 Euros y darle mantenimiento tiene también un alto costo

4. Se propuso una aplicación utilizando detección y procesamiento de imágenes por medio de espectroscopia con infrarrojo cercano y las Deep Neural Networks para la Detección y Clasificación de Arándanos en la empresa TalSA.

Se utilizó lenguaje de programación Python y las bibliotecas que brindan Keras y TensorFlow, que son plataformas de código abierto para el aprendizaje automático. Cuentan con un ambiente integral y flexible de herramientas, bibliotecas y recursos que permite impulsar el desarrollo de aplicaciones en Machine Learning y poder construir e implementar fácilmente aplicaciones con tecnología Machine Learning. Como parte del proceso se obtuvo los espectros NIR de las muestras de arándanos, se digitalizaron y se formó la base de datos que sirvió para entrenar las redes neuronales, alcanzándose niveles de exactitud de hasta 98%.

5. Se utilizó la aplicación propuesta en la detección y clasificación de arándanos de la empresa TalSA, consiguiéndose clasificar los arándanos con una exactitud de 100% en muchos de los casos que se utilizaron para probar la aplicación propuesta.
6. Se realizó el análisis estadístico de los datos obtenidos encontrándose que la aplicación propuesta permite detectar y clasificar los arándanos con una exactitud de 92%. Lo que nos indica que efectivamente si se puede utilizar la espectroscopia NIR y machine learning y deep learning para poder clasificar arándanos en la empresa TalSA.

RECOMENDACIONES

1. Se recomienda validar la aplicación propuesta con una mayor cantidad de arándanos a fin de maximizar la exactitud en la clasificación de arándanos.
2. Se recomienda utilizar la aplicación propuesta en otras empresas que también se dedican a la exportación de arándano fresco, con el fin de validar la propuesta con arándanos de otros tipos.
3. Se recomienda diseñar e implementar una interfaz que permita realizar la clasificación de los arándanos físicamente, de manera automática. en base a la aplicación propuesta.

REFERENCIAS BIBLIOGRÁFICAS

- ACM. (2014). Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Recuperado el 10 de abril del 2019 de <https://www.acm.org/binaries/content/assets/education/se2014.pdf>
- Aguilera J. (2017). Deep learning para clasificación en imágenes. Recuperado el 28 de octubre del 2018 de: <https://www.ackstorm.com/deep-learning-clasificacion-imagenes/>
- Alava Ingenieros. (2018). Sistemas de imagen para el Infrarrojo Cercano (NIR) para diversas aplicaciones. Recuperado el 8 de noviembre del 2018 de: <http://www.grupoalava.com/ingenieros/actualidad/sistemas-de-imagen-para-el-infrarrojo-cercano-nir-para-diversas-aplicaciones/>
- Andrade, T. (2013). Estudio de los principales tipos de redes neuronales y las herramientas para su aplicación- Andrade. Ecuador. Universidad Politécnica Salesiana. Recuperado el 10 de abril del 2019 de: <https://dspace.ups.edu.ec/bitstream/123456789/4098/1/UPS-CT002584.pdf>
- Bengio, Y., Courville, A., Vincent, P.(2013). Representation Learning: A Review and New Perspectives. IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, pp. 1798-1828.
- Bhargava A., Atul Bansal. (2018). A. Fruits and vegetables quality evaluation using computer vision: A review. Saud University. Recuperado el 9 de abril del 2019 de <https://reader.elsevier.com/reader/sd/pii/S131915781830209X?token=B05BFBE073196F6BD455E8D7579980BB356F7F4B0253EACFA35AA338CC72B3A0C0A62FE76417C6CEE2815E635CE96B21>
- Chang L., Yu C., Yan L., Guanling C., Vinod V., Yunsheng M. (2015). DeepFood: Deep Learning-based Food Image Recognition for Computer-aided Dietary Assessment. A White Paper. USA. Recuperado El 10 de setiembre del 2018 de: [file:///C:/Users/helpadministrator/Downloads/DeepFood_Deep_Learning-Based_Food_Image_Recognitio%20\(1\).pdf](file:///C:/Users/helpadministrator/Downloads/DeepFood_Deep_Learning-Based_Food_Image_Recognitio%20(1).pdf)
- Chollet, Fran (2015). Keras. La biblioteca de aprendizaje profundo de Python Recuperado el 9 de marzo del 2019 de <https://keras.io/>
- Dolz Z. (2008). Evaluación de la calidad de fruto en manzano: estudio de métodos no destructivos de análisis. Zaragoza. España. Recuperado el 9 de marzo del 2019 de <http://digital.csic.es/bitstream/10261/18601/1/Proyecto%20Pilar%20Dolz.pdf>
- Flores K. , Paz, P., Sánchez M., Pérez D., López M., Contreras M., Mejía J., Guerrero J., Garrido A. (2009). “Determinación no destructiva de parámetros de calidad de frutas y hortalizas mediante espectroscopía de reflectancia en el infrarrojo cercano. Universidad de Córdoba. España. Recuperado el 9 de marzo del 2019 de: <https://helvia.uco.es/xmlui/bitstream/handle/10396/2070/9788478019427.pdf>

- Gómez, G. (2018). Aprendizaje profundo. México. Inaoe. Recuperado el 10 de abril del 2019 de: <https://ccc.inaoep.mx/~pgomez/cursos/IC-I/acetatos/ApProfundo.pdf>
- Keras-Pandas (s.f.). Guía del usuario. Recuperado el 9 de marzo del 2019 de <https://keras-pandas.readthedocs.io/en/latest/intro.html>
- La fuente V. (2015). Aplicación de las técnicas de Espectroscopia Vis/NIR y de imágenes de retrodifusión de luz láser a la evaluación del estado de madurez de melocotón, manzana y cereza. (Tesis de Doctorado en Ciencias Químicas). Universidad de Zaragoza, España.
Recuperado el 11 de octubre del 2018 de:
http://digital.csic.es/bitstream/10261/126618/1/LafuenteV_TD-EEAD_2015.pdf
- Nogales J. (2016). Estudio del estado de madurez y la aptitud enológica en uva mediante análisis de imagen hiperespectral. (Tesis de Doctorado en Ingeniería Química). Universidad de Sevilla – España
Recuperado El 11 de noviembre del 2018 de:
https://idus.us.es/xmlui/bitstream/handle/11441/55531/Tesis_Julio_Nogales_Bu_eno_sin_publicaciones.pdf?sequence=1
- Pastor, J. (2016). 17 expectativas de cómo el machine learning va a cambiar el mundo. Recuperado el 10 de abril del 2019 de <https://www.xataka.com/robotica-e-ia/17-expectativas-de-como-el-machine-learning-va-a-cambiar-el-mundo>
- Precup, D. (2017) Introduction to Machine Learning. McGill University. Montreal
Recuperado el 10 de Julio 2018 de
<file:///H:/Tesis/DEEP%20Learning/Precup%20-%20dlss-intro-2017%20Introduction%20to%20machine%20Learning.pdf>
- Qian Yu et.al, (2016). Deep Learning Based Food recognition . Stanford University. USA.
Recuperado El 10 de setiembre del 2018 de:
<file:///H:/Tesis/Reconocimiento%20de%20alimentos/YuMaoWang-Deep%20Learning%20Based%20Food%20Recognition-report.pdf>
- RedAgrícola. (2018). Una exacta medición de madurez en arándano.
Recuperado el 9 de marzo del 2019 de:
<http://www.redagricola.com/pe/una-exacta-medicion-de-madurez-en-arandano/>
- Salomón, F. (2015). Tecnología NIR, sus Usos y Aplicaciones. V Congreso Argentino de Nutrición Animal - CAENA 2015
Recuperado el 20 de septiembre del 2018 de:
<https://www.engormix.com/balanceados/articulos/tecnologia-nir-sus-usos-t32534.htm>
- Samanamud Ríos, V. (2001). Estadística aplicada a la educación. Trujillo: Universidad Privada Antenor Orrego.
- Sladojevic S., Arsenovic, M., Anderla A., Culibrk, D. Stefanović, D. (2016). Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image

Classification. Computational Intelligence and Neuroscience. 2016. 1-11.
10.1155/2016/3289801. Serbia.
Recuperado el 26 de Mayo del 2018 de <http://dx.doi.org/10.1155/2016/3289801>

Shalev-Shwartz, E., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms (2014). Cambridge University Press, NY.
Recuperado el 20 de febrero e 2019 de
<https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf>

Smola A., Vishwanathan, S. (2008). Introduction to Machine Learning. United Kingdom: Cambridge University Press
Recuperado el 9 de abril del 2019 de <http://alex.smola.org/drafts/thebook.pdf>

Soriano, A., Soria, E., Martin, J.(2010).Redes neuronales artificiales. España. Universidad de Valencia. Recuperado el 10 de abril del 2019 de:
http://ocw.uv.es/ingenieria-y-arquitectura/1-2/libro_ocw_libro_de_redes.pdf

TensorFlow (s.f.). An end-to-end open source machine learning platform.
Recuperado el 9 de marzo del 2019 de <https://www.tensorflow.org/overview>

Zhang, Y. Wang, S., Ji G., Phillips P. (2014). Fruit classification using computer vision and feedforward neural network. A White Paper. EE: UU:
Recuperado el 9 de junio del 2018 de:
https://www.researchgate.net/publication/264546240_Fruit_classification_using_computer_vision_and_feedforward_neural_network

ANEXOS

ENCUESTA

Instrucciones

Responda de manera clara y concreta las siguientes preguntas

Nombre: _____

Cargo: _____

1.- ¿Qué porcentaje de eficiencia tienen o se estima en la calidad de los arándanos que producen o que porcentaje de error es considerado aceptable?

2.- Cuales son los valores o rangos establecidos para considerar que la producción es buena. ¿Cómo los obtienen o cómo los evalúan?

Fotos adicionales



Laboratorio de espectroscopia NIR



En el laboratorio de espectroscopia NIR