



UNIVERSIDAD
DE PIURA

FACULTAD DE INGENIERÍA

**Implementación de algoritmos de aprendizaje automático
para la medición de parámetros de calidad de uva**

Trabajo de Investigación para optar el Grado de
Bachiller en Ingeniería Mecánico - Eléctrica

**Rubén Alonso Calle Lapouble
Eduardo Emmanuel Correa Correa
Gustavo Edgardo Grosso Salazar
José Luis Huamantoma Pumallihua**

**Asesor:
Dr. Ing. William Ipanaqué Alama**

Piura, mayo de 2021



Resumen

Las uvas son una de las principales frutas de exportación para Perú. Poco a poco, ha ido adquiriendo un mayor aumento de sus exportaciones gracias a la mejora de su proceso productivo, la medición de la calidad y la mejora de las prácticas de cultivo. La medición y evaluación de la calidad de los frutos han sido uno de los aspectos más críticos de la logística. Los productos agrícolas para la agroexportación. Para el almacenamiento, transporte y venta de frutas, una medición de los parámetros y se requiere previamente una clasificación de la calidad de la fruta, siendo el color uno de los parámetros más característicos para representar el estado de calidad de los frutos, en este caso, siendo nuestro objeto de estudio la uva de mesa.

En este trabajo de investigación, la metodología propuesta para la estimación de los parámetros de calidad se compone de 3 parámetros principales: estimación del peso, estimación del calibre y apariencia estimación. Imágenes de racimos de uva obtenidas de la Sociedad Agrícola Saturno. S.A. Las imágenes de las uvas están segmentadas utilizando el Photoshop para mejorar la tarea de estimación en la que parámetro. Por último, la red neural multicapa de perceptrón es utilizada para estimar los pesos y el SVM se utiliza como final algoritmo para clasificar los racimos de uva como adecuados o no apto para la exportación por apariencia. Otro algoritmo como la CNN y el aprendizaje por transferencia se utilizan para comparar la precisión del modelo.

Se obtuvo un error de 7.7% para la estimación del calibre del racimo y para la estimación del peso en la que se trabajó con 3 variantes, 2 de ellas se trabajaron con redes neuronales en donde se obtuvo un error de 14.6% y 19.8%, como última variante se usó una regresión lineal multivariable donde se obtuvo un error de 13.3%. Para clasificar el estado del racimo, apta o no pata, se usó redes neuronales convolucionales junto con transfer learning en la que se obtuvo un 90.36% de precisión en la validación; y como segunda variante para clasificar el racimo se trabajó con Support Vector Machines en la que se obtuvo un 78.1% de precisión.

Según los resultados mostrados, se concluye que es viable el uso de algoritmos de aprendizaje automático y procesamiento de imágenes para estimar los parámetros de calidad de uva. También se concluye que es mejor usar algoritmos de machine learning cuando no se tiene suficiente data, como fue en el caso de la estimación del peso donde se obtuvo un menor error, en vez de algoritmos de Deep learning.



Tabla de contenido

Introducción	15
Capítulo 1	17
Generalidades	17
1.1. Planteamiento del problema y justificación.....	17
1.2. Objetivos.....	17
1.2.1. Objetivos generales.....	17
1.2.2. Objetivos específicos.....	18
1.3. Antecedentes.....	18
1.3.1. Estudio de un sistema de visión para control de pesos y tamaño de racimos de uva de mesa	18
1.3.2. Clasificación de calidad de fresas usando procesamiento de imágenes y máquinas de vectores de soporte	19
1.3.3. Determinación del estado de madurez de una cereza aplicando procesamiento de imágenes	19
1.3.4. Estimación del volumen en duraznos: Un estudio preliminar con procesamiento digital de imágenes.	19
1.3.5. Determinación del color del exocarpio como indicador de desarrollo fisiológico y madurez en la guayaba pera (psidium guajava cv. guayaba pera), utilizando técnicas de procesamiento digital de imágenes	20
1.3.6. Sistema de clasificación por visión artificial de mangos tipo Tommy	20
1.3.7. Quality Detection of Fruits by Using ANN Technique	21
1.3.8. Non-destructive and contactless quality evaluation of table grapes by a computer vision system.	21
Capítulo 2	23
Marco teórico.....	23

2.1.	La uva de mesa.....	23
2.1.1	Generalidades de la uva	23
2.1.1.1	Origen.....	23
2.1.1.2	Tipologías.....	23
2.1.1.3	Taxonomía.....	26
2.2.	Impacto de la uva.....	27
2.2.1.	Regional.....	28
2.2.2.	Nacional.....	29
2.2.3.	Internacional.....	30
2.3.	Características agroexportadoras.....	31
2.3.1.	Regionales.....	31
2.3.2.	Nacional.....	32
2.3.3.	Mundial.....	34
2.4.	Medidas de calidad.....	36
2.4.1.	Color.....	37
2.4.2.	Tamaño de bayas.....	37
2.4.3.	Textura.....	38
2.4.4.	Peso.....	38
2.5.	Espacio de color.....	39
2.5.1.	Espacio RGB.....	39
2.6.	Procesamiento de imágenes.....	40
2.6.1.	Preprocesamiento.....	40
2.6.2.	Procesamiento propiamente dicho.....	42
2.7.	Redes neuronales.....	46
2.7.1.	Red neuronal Artificial.....	46
2.7.2.	Aplicación de Redes Neuronales Convolucionales (CNN).....	52
2.8.	Support Vector Machine.....	53
2.8.1.	Missclassification cost.....	54
2.8.2.	Undersampling.....	55
2.8.3.	Normalización y Estandarización.....	55

2.8.3.1. Normalización.....	56
2.8.3.2. Estandarización.....	56
2.8.4. Cross-validation.....	56
2.8.5. Métricas	57
Capítulo 3	61
Metodología.....	61
3.1. Adquisición de imágenes.....	61
3.2. Pre - procesamiento y procesamiento	63
3.3. Procesamiento de imágenes para la medición de la baya de uva.....	65
3.3.1. Transformada de Watershed.....	66
3.3.2. Segmentación de watershed con marcadores.	67
3.3.3. Segmentación de watershed estocástico.....	68
3.3.4. Transformada de Hough.....	71
3.4. Extracción de parámetros para estimación del peso del racimo.	72
3.4.1. Bounding box	72
3.4.2. Cálculo de área.....	73
3.5. Estimación del estado de la uva con SVM.....	74
3.6. Estimación del estado de la uva con CNN.....	77
Capítulo 4	89
Análisis de Resultados	89
4.1. Resultados de la estimación del peso	89
4.2. Resultados de la estimación del calibre	90
4.3. Resultados de la clasificación de apariencia con SVM.....	90
4.4. Resultados de la clasificación de apariencia con CNN	93
Conclusiones	97
Recomendaciones	99
Referencias bibliográficas	101



Lista de tablas

Tabla 1. Principales variedades de uva en el Perú.....	24
Tabla 2. Cuadro de participación mundial del Perú.....	34
Tabla 3. Error promedio para las distintas variables.....	89
Tabla 4. Pruebas realizadas con SVM.....	91
Tabla 5. Matriz de Confusión - SVM color y textura.....	92
Tabla 6. Resultados de la validación para la arquitectura VGG-16.....	94





Lista de figuras

Figura 1. Estructura de la planta de vid.....	27
Figura 2. Exportaciones por productos en Piura.....	28
Figura 3. Empresas exportadoras en Piura.....	28
Figura 4. Exportación de uva de mesa por región en el 2017 (%).	29
Figura 5. Evolución mundial de exportaciones de uva.....	30
Figura 6. Ranking de exportaciones a nivel mundial.....	31
Figura 7. Principales nodos de producción en el Perú.	32
Figura 8. Cajas con uvas para exportación.	33
Figura 9. División de empresas agroexportadoras en el Perú.....	34
Figura 10. Exportación Uvas Mundialmente	35
Figura 11. Escala de color para racimos Red Globe. Desde bayas verdes en la parte izquierda a bayas rojas como extremo derecho.	37
Figura 12. Formas del racimo de la uva.....	38
Figura 13. Estructura básica del espacio de color RGB	39
Figura 14. Aumento de contraste.	41
Figura 15. Filtro Gaussiano.	41
Figura 16. Filtro mediana.....	42
Figura 17. Imagen Binarizada por Trheshold.....	43
Figura 18. Imagen original, histograma e imagen segmentada.	43
Figura 19. Imagen original, histograma e imagen segmentada.	44
Figura 20. Máscaras recomendadas para el filtro gaussiano.	44
Figura 21. a) imagen original; b) orientación; c) supresión no máxima; d) histéresis del umbral.	46
Figura 22. Algoritmo de Canny.....	46

Figura 23. Esquema de una neurona artificial.....	47
Figura 24. Capas de una red neuronal.....	48
Figura 25. Problemas de interés de las redes neuronales artificiales.....	49
Figura 26. Ejemplificación de la red perceptrón multi capa.....	50
Figura 27. Funciones de activación más utilizadas.	51
Figura 28. Gráfica del descenso de gradiente.....	51
Figura 29. Gráfica de una función de pérdida real.....	52
Figura 30. Máquina de soporte vectorial.	54
Figura 31. Matriz de confusión y las métricas que se pueden obtener.	57
Figura 32. Criterio de variación en las especificidad.	58
Figura 33. Fotografía tomada en el light box.....	61
Figura 34. Esquema de adquisición de datos.	62
Figura 35. Zona de recolección de data.....	62
Figura 36. Imagen de uva de mesa.....	64
Figura 37. Imagen segmentada con Image Segmenter.....	64
Figura 38. Imagen segmentada y binarizada con Image Segmenter.....	65
Figura 39. a) Imagen en escala de grises y b) representación 3-D de la imagen en escala de grises.	66
Figura 40. a) Imagen a segmentar, b) Imagen gradiente, c) fronteras watershed (presencia de sobre segmentación), d) Imagen sobre segmentada, e) fronteras watershed después de realizar un filtrado al gradiente y f) imagen segmentada.	67
Figura 41. a) Imagen a segmentar, b) gradiente, c) sobresegmentación, d) mínimos regionales del gradiente, e) marcadores internos, f) marcadores externos, g) marcadores internos y externos sobre la imagen original y h) resultado de la segmentación.....	68
Figura 42. a) Imagen original a segmentar, b) M realizaciones de N marcadores aleatorios, $mrki(x)$ y sus correspondientes contornos watershed, c) Función de densidad probabilidad de los contornos calculados para $N=10$ y $M=50$	69
Figura 43. a) Resultado inicial de la cuenca, b) Estructura del árbol y c) Resultado del proceso de limpieza.	70
Figura 44. Salida del algoritmo(derecha) y Detección de borde inicial (izquierda).....	71
Figura 45. Puntos de borde de un círculo y espacio acumulador.....	72
Figura 46. Bounding box.	72

Figura 47. Racimo de uva binarizada.....	73
Figura 48. Esquema de la red neuronal para la predicción del peso del racimo.....	74
Figura 49. Comand Window de las características de color en Matlab	75
Figura 50. Comand Window de las características de forma en Matlab	76
Figura 51. Comand Window de las características de textura en Matlab	76
Figura 52. Ventana del Classification Learner previos al entrenamiento del modelo.....	77
Figura 53. Capas y parámetros de la arquitectura VGG-16.....	78
Figura 54. Gráfica de accuracy en la CNN aplicando transfer learning para el modelo VGG-16.	79
Figura 55. Gráfica de loss en la CNN aplicando transfer learning para el modelo entrenado con VGG-16.....	79
Figura 56. Capas y parámetros de la arquitectura MobileNet.	81
Figura 57. Capas y parámetros de la arquitectura MobileNet aplicando Dropout.	81
Figura 58. Gráfica de accuracy en la CNN aplicando transfer learning para el modelo MobileNet.	82
Figura 59. Gráfica de loss en la CNN aplicando transfer learning para el modelo MobileNet.	83
Figura 60. Gráfica de Accuracy para la prueba 2 con la arquitectura ResNet50.....	84
Figura 61. Gráfica de Loss para la prueba 2 con la arquitectura ResNet50	84
Figura 62. Imagen de uva de mesa sin fondo	85
Figura 63. Imagen de uva de mesa sin fondo.	86
Figura 64. Proceso del entrenamiento en la herramienta Deep Network Designer	86
Figura 65. Detección de círculos para la imagen sin modificar(izquierda) y la modificada(derecha).....	90
Figura 66. Gráfica del mínimo error de clasificación para la SVM de textura y color con modificación de contraste.	91
Figura 67. Curva ROC para la SVM de textura y color con modificación de contraste.....	92
Figura 68. Gráfica accuracy en la CNN aplicando transfer learning para la red VGG-16.....	93
Figura 69. Gráfica loss en la CNN aplicando transfer learning para la red VGG-16.....	94
Figura 70. Inicio de la API Streamlit como servidor local.....	95
Figura 71. Clasificación de uva en la API Streamlit como servidor local.....	95

Figura 72. Clasificación de uva en la API Streamlit como servidor local.....96



Introducción

Perú es un país que tiene el privilegio de gozar de tierras muy fértiles gracias a las cuales se cultivan diferentes tipos de frutos. Siendo la uva, una de las principales frutas de exportación para el Perú. Poco a poco su exportación se ha incrementado gracias a la mejora continua de su proceso productivo, prácticas de cultivo y medición de calidad. Este es muy valorado en los amplios mercados de Asia, Europa y América siendo este último el mercado más fuerte.

Su producción ha sido mejorada con ayuda de programas del Estado. No obstante, aún hay algunas salvedades que pueden levantarse con ayuda de la tecnología actual. Tal es el caso de la aplicación de técnicas que implican el uso de AI (*Artificial Intelligence*) e IoT (*Internet of Things*) para ayudar a los agricultores a facilitar su labor de control y prevención de plagas y análisis de nutrientes del suelo de cultivo.

La medición y evaluación de la calidad en las frutas ha sido uno de los aspectos más importantes en la logística de los productos agrícolas destinado a la agroexportación. Para el almacenamiento, transporte y venta de frutas se requiere previamente una medición de parámetros y clasificación de la calidad de la fruta, siendo el color uno de los parámetros más característicos para representar el estado de calidad de las frutas, en este caso, siendo nuestro objeto de estudio, la uva de mesa.

A pesar de que el color, es uno de los rasgos característicos para la medición de calidad de uva, existen otros parámetros tales como la forma, el tamaño, los grados brix entre otros, los cuales son tan importantes como la coloración del fruto.

En este trabajo, se hará uso desde imágenes en el espacio RGB (Red, Green and Blue) con la finalidad de, por medio de distintos algoritmos de inteligencia artificial, lograr una estimación del peso y apariencia (color) de los racimos de uva., mientras que para el calibre se utilizó técnicas de procesamiento de imágenes.

La investigación realizada demuestra que, para el caso particular de uvas Red Globe de la Sociedad Agrícola Saturno S.A., en el caso del *Machine Learning*, se obtuvo un 78.1% en la estimación de apariencia y un 86.7% para la estimación del peso mientras que, para el caso del *Deep Learning*, se obtuvo un 90.36% y 82.79% en las mismas estimaciones. Para la estimación del calibre se obtuvo un 92.3% de precisión.



Capítulo 1

Generalidades

1.1. Planteamiento del problema y justificación

La región Piura, en los últimos años, viene destacando en el ámbito agroexportador, la cual representa el 4.3% del PBI del país con lo cual es fundamental su desarrollo para lograr un crecimiento a nivel nacional (MINCETUR, 2018b).

La uva de mesa peruana es altamente valorizada en los mercados internacionales, tales como Holanda, Estados Unidos, Reino Unido, China y otros, junto con ello, la demanda del producto se ha ido incrementando por lo que incentiva y lleva a los productores a adecuarse a los nuevos requerimientos de los consumidores, que se vuelven cada vez más exigentes (Huemura, 2018).

La calidad de la uva es un parámetro que suele ser controlado en sus variantes químicas por su mayoría pero, sin embargo no existe un parámetro específico para el análisis de la uva con respecto al color, el cual es un parámetro sumamente importante al momento en que el consumidor va a los supermercados a adquirir el producto, si bien, el nivel de acidez, dulzor, entre otros son parámetros que van de cierta forma relacionados con el aspecto visual de la uva, no está de más adicionar al proceso un control específico de color, ya que es el primer parámetro percibido por los consumidores al seleccionar la uva de mesa.

Por medio del presente trabajo se busca desarrollar un algoritmo de Machine Learning, el cual incrementa el nivel de competitividad de los productores peruanos de uva de mesa, a su vez reducir tiempos de producción y medición de calidad, y costos al interior de una planta agroexportadora. Es por lo que, uno de los objetivos principales es tanto descubrir las metodologías aplicadas en la industria para obtener los parámetros específicos que conlleva la medición de la calidad de la uva de mesa, como desarrollar soluciones aplicando los conocimientos obtenidos durante la carrera.

1.2. Objetivos

1.2.1. Objetivos generales

- Acercar al estudiante a una realidad regional en el sector agroindustrial con la cosecha y exportación de la uva.

- Descubrir las metodologías aplicadas en la industria para la medición de calidad en la uva.
- Implementar un proceso de medición de calidad en tiempo real dentro de la industria.
- Desarrollar las metodologías tanto teóricas como prácticas, teóricas en base a la investigación a realizar durante todo el semestre para optimizar los parámetros de medición de la calidad en la uva, mientras que, en el aspecto práctico se implementará un modelo que optimizará el proceso de medición de calidad.

1.2.2. Objetivos específicos

- Desarrollar un sistema con la ayuda de los clasificadores de machine learning para la medición de calidad de la uva y así poder seleccionar el clasificador más adecuado.
- Conseguir que el estudiante de ingeniería investigue, descubra y domine los conocimientos acerca de Deep learning, Machine learning y conceptos relacionados a la Industria 4.0.
- Implementar nuevos métodos artificiales que puedan reemplazar los métodos tradicionales para la medición de la calidad de la uva y así poder optimizar procesos en la selección de la uva de mesa en base a su calidad.

1.3. Antecedentes

La aplicación de las nuevas tecnologías como machine *learning* y *deep learning* siguen un proceso de mejora continua, cada vez perfeccionando y ampliando su campo de estudio en donde pueden ser aplicadas estas disciplinas científicas.

En este apartado se comentará brevemente los diferentes trabajos relacionados con el estudio del procesamiento de imágenes tanto en uva como en otros frutos con una similitud en su crecimiento y cosecha.

1.3.1. Estudio de un sistema de visión para control de pesos y tamaño de racimos de uva de mesa (Basilio Vela, 2016)

En su tesis de maestría estudia cuán viable es el uso de redes neuronales y un sistema de visión para determinar el peso de los racimos de uvas de distintos tipos, para extraer el tamaño y área propone la siguiente metodología:

Después de leer una imagen captura se procede a realizar un filtrado espacial sobre la imagen, seguido de la binarización de la imagen para poder separar el racimo de la faja transportadora para finalmente extraer los valores de tamaño y área. Para calcular el tamaño de la uva o el calibre, ellos toman una foto patrón con la finalidad de tener una constante que relacione la longitud de una regla con los pixeles que esta ocupa, cabe resaltar que aplica esta metodología partiendo de la premisa que la cámara estará a una distancia definida por el usuario y no se moverá verticalmente.

1.3.2. Clasificación de calidad de fresas usando procesamiento de imágenes y máquinas de vectores de soporte (Marin & Madrigal, 2016)

En este artículo el autor desarrolla una metodología para adquirir características de forma, tamaño y color en las fresas. Inicia con la toma de imágenes en la que usó un sistema de iluminación tipo domo con luz blanca difusa.

Para la clasificación uso una SVM, en esta etapa se usó 120 imágenes para el entrenamiento y 80 imágenes para hacer las pruebas, obteniéndose una precisión de 100% para clasificar formas. Para la medición del tamaño se obtuvo usaron una constante que relaciona pixeles/mm, esto lo lograron tomando una imagen a una foto con medidas conocidas, obtuvieron un error del 6.5% al medir el diámetro de las fresas.

1.3.3. Determinación del estado de madurez de una cereza aplicando procesamiento de imágenes (Ramírez et al., 2017)

En este trabajo se combina métodos de morfología matemática y detección de bordes y círculos para evaluar el estado de madurez de la cereza africana en 4 estados fisiológicos, cabe mencionar que usaron imágenes RGB. La metodología que usaron fue: Primero usaron la detección de bordes por el algoritmo de Canny, la detección de círculos mediante la transformada de Hough con la finalidad de separar la fruta del exterior. De un total de 407 imágenes obtuvieron 396 imágenes segmentadas correctamente, un 97.3% de precisión, después de segmentar la imagen procedieron a encontrar el promedio para cada canal de color obteniendo así el color promedio que se tiene dentro para cada día en un intervalo total de tiempo de 15 días.

Se puede rescatar de este trabajo que para la detección de bordes la efectividad del algoritmo de Canny es muy alta puesto que las uvas tienen forma geométrica similar a las cerezas, aunque se debería corroborar esto pues al tratarse de racimos puede existir pequeñas modificaciones.

1.3.4. Estimación del volumen en duraznos: Un estudio preliminar con procesamiento digital de imágenes (Ituarte, 2011).

En este trabajo se usa un conjunto de métodos para determinar el peso y el volumen del durazno, uno de ellos fue experimental en la que se usó el desplazamiento del agua que provocaba el durazno, pero en el segundo método usaron 2 fotos de un durazno tomadas en planos distintos, esto lo logró haciendo girar el durazno 90° grados con un dispositivo mecánico.

El procesamiento digital comienza con la transformación de la imagen en RGB a HSI, tono(H), saturación(S) e intensidad(I), seguido de un filtrado de mediana para eliminar el ruido, para su posterior binarización mediante la detección de un umbral global.

Para calcular el volumen del durazno uso un frostrum cónico, similar a un cono truncado, en la que se puede calcular el volumen de este ya que se conoce el diámetro mayor y menor, relacionado con la cantidad de pixeles que ocupa, para esto previamente se halló una constante que relaciona la distancia con los pixeles. Así el volumen del durazno será la sumatoria de enésimos frostrums cónicos.

Para el caso de las uvas se debe corroborar si es viable este método, porque si se hace un corte trasversal a un racimo no se tiene exactamente un solo círculo por lo que se debería adecuar este método, un ejemplo podría ser a considerar enésimas esferas en la que cada una representa una baya del racimo.

1.3.5. Determinación del color del exocarpio como indicador de desarrollo fisiológico y madurez en la guayaba pera (*psidium guajava* cv. *guayaba pera*), utilizando técnicas de procesamiento digital de imágenes (Castro et al., 2013)

En este artículo se realiza un análisis de imágenes digitales de frutas de guayaba pera de diferentes estaciones, por medio de las coordenadas colorimétricas, trabajando con los espacios de color RGB y CIE-L*a*b con la finalidad de apreciar la evolución de los frutos en postcosecha, llegando a la conclusión de que, la tasa de respiración es un factor clave ya que influye directamente en la coloración del fruto en el periodo de postcosecha.

Este artículo nos ayudará para poder describir la evolución del fruto y su relación que presenta con la maduración de este. En caso se tomen muestras en diferentes periodos, se puede replicar las condiciones de manipulación y almacenamiento de tal forma que se pueda determinar la vigencia del producto.

La toma de imágenes se realizó con una cámara digital Kodak con sensor CDD, de modelo C180 a una distancia de 25cm del fruto y con iluminación fluorescente a 1m del mismo. La guayaba crece en arbustos y su cosecha se da por unidades.

1.3.6. Sistema de clasificación por visión artificial de mangos tipo Tommy (Romero et al., 2015)

Este paper utiliza una potente herramienta disponible en MATLAB con la finalidad de clasificar mango de acuerdo a dos características, color y tamaño. Se resalta que es un sistema ya implementado en una planta, debido a que los mangos salen por una faja transportadora y una cámara web tomas las fotos en grupos de 3.

El algoritmo clasificador utilizado es una red decisión programada también en el mismo MATLAB mediante 3 coloraciones de mango y 2 de tamaño, se determina si el mango es adecuado y óptimo para la exportación.

Se puede replicar el trabajo y quizá proponer un módulo adecuado a la uva para su implementación en la planta con la que estamos trabajando ya que, reduciría

considerablemente los tiempos y a su vez costos por el personal que ya no ejecutarían estas funciones.

Respecto a los materiales utilizados, se utiliza una cámara web del tipo seguridad, los toolbox de MATLAB y el computador que procesará los datos, a diferencia de la uva, el mango se recolecta por unidad, pero será cuestión de adecuarlo a nuestro caso e implementarlo.

1.3.7. *Quality Detection of Fruits by Using ANN Technique (Kaur & Sharma, 2015)*

En esta investigación usan un sistema que utiliza técnicas de procesamiento de imágenes para clasificar y determinar la calidad de las frutas. Comúnmente las imágenes de frutas son bidimensionales y se clasifican según métodos de análisis basados en la forma y el color. Sin embargo, diferentes imágenes de frutas pueden tener valores de color y forma similares o idénticos dificultando el resultado, es por eso que de esta forma aún no es lo suficientemente efectivo para, llegando a la conclusión de que, la tasa de respiración es un factor clave ya que influye directamente en la coloración del fruto en el periodo de postcosecha.

Este artículo nos ayudará para poder describir la evolución del fruto y su relación que presenta con la maduración de este. En caso se tomen muestras en diferentes periodos, se puede replicar las condiciones de manipulación y almacenamiento de tal forma que se pueda determinar la vigencia del producto.

La toma de imágenes se realizó con una cámara digital Kodak con sensor CDD, de modelo C180 a una distancia de 25cm del fruto y con iluminación fluorescente a 1m del mismo. La guayaba crece en arbustos y su cosecha se da por unidades.

1.3.8. *Non-destructive and contactless quality evaluation of table grapes by a computer vision system (Cavallo et al., 2019)*

En el siguiente artículo, se propone un sistema de visión por computadora para poder medir la calidad de la uva mediante técnicas analíticas objetivas e implementando las técnicas de procesamiento de imágenes. El sistema fue verificado en 2 cultivos de uva de mesa con diferentes clasificaciones dependiendo los parámetros como de calidad, donde se eligieron 5 niveles, de los cuales el 5 y 4 son los más comercializados (aptos para la exportación), el nivel 3 que es un límite, y, por último, el nivel 2 y 1 son los menos aptos para su exportación.

La implementación de las técnicas del procesamiento de imágenes utilizado en la siguiente investigación es muy importante ya que, como se especifica, ha sido correctamente calibrada para cada cultivo con una cámara digital, siempre y cuando no haya contacto con los racimos.

Esta investigación es de gran importancia para nuestro estudio, ya que especifica los parámetros medidos y los cuidados necesarios para la obtención de imágenes, con el fin de optimizar el procesamiento de las imágenes.



Capítulo 2

Marco teórico

2.1. La uva de mesa

La uva de mesa en la actualidad ha llegado a posicionarse como uno de los productos de mayor valor en los mercados tanto a nivel nacional como internacional, seguido del banano, mango y cacao. Además, la agroexportación está creciendo cada vez más, siendo hoy en día uno de los dos motores que impulsan la economía peruana (Ramos, 2016). La uva peruana en el año 2018 representó el 12% (US\$ 817,9 millones) del total de agroexportaciones en el Perú (MINAGRI, 2019).

2.1.1 Generalidades de la uva

2.1.1.1 Origen. Según las investigaciones, las plantaciones de uva fueron uno de los primeros cultivos del hombre por lo que, juega un papel fundamental en la historia de la economía, desde la antigüedad hasta la actualidad.

Luego de la propagación de los mitos acerca del vino por parte del cristianismo, el cultivo de uva experimentó un gran crecimiento que se mantiene hasta el día de hoy. Fue con la llegada de los españoles que se introduce la vid en el Perú. Según Garcilaso de la Vega, fue Francisco de Caravantes quien las trajo, provenientes de las Islas Canarias (Pisco, 2019).

Un dato importante que señalan los cronistas es que fue en la hacienda Marcahuasi (Cuzco), donde se elaboró la primera vinificación en Sudamérica. Sin embargo no fue en Cuzco sino en Ica donde se expandió el cultivo de este producto debido a las favorables condiciones edafoclimáticas y es por ello que Ica se ha desarrollado como un gran exponente en la industria de vinos (Pisco es Perú, 2019).

2.1.1.2 Tipologías. Según (Bhandari, 2020), el crecimiento exponencial de la exportación de las uvas peruanas se dio a causa de la aparición de la uva de tipo "*Red Globe*", la cual apareció por los años 90 y, actualmente es la de mayor demanda en los mercados internacionales debido a sus buenas condiciones a bajas temperaturas.

Entre las variedades de uva que el Perú exporta se encuentran:

Tabla 1. Principales variedades de uva en el Perú

Variedad		Características				
		Color	Forma	Presencia de semillas	Grados brix	Observaciones
Red Globe		Rojo oscuro ligeramente brillante	Redonda. Calibre promedio: 24 - 28 mm	Si	15 - 18°	Cáscara firme, pulpa crujiente y carnosa
Superior		Verde claro	Elíptico ovoidal	No	15°	Pulpa crujiente
Crimson		Rojo	Cónica. Calibre promedio: 17 - 22 mm	No	18°	Sabor neutro muy dulce, de pulpa crujiente y cáscara gruesa
Flame		Roja	Esférica. Calibre promedio: 18 mm	No	16°	Pulpa crujiente, consistente e incolora
Thompson		Verde claro	Cónica. Calibre promedio: 18 - 20 mm	No	18°	Sabor neutro muy dulce, de pulpa crujiente y cáscara gruesa
Sweet Celebration		Rojo brillante	Redonda a ovalada. Calibre promedio: 25 mm	No	18°	Muy crujiente

Variedad		Características				
		Color	Forma	Presencia de semillas	Grados brix	Observaciones
Magenta		Rojo	Redonda. Calibre promedio: 23 - 25 mm	No	17°	-
Timpson		Verde pálido	Ovoide. Calibre promedio: 21 - 23 mm	No	19°	Sabor afrutado, moscatel
Sweet Globe		Verde claro	Ovoide. Calibre promedio: 22 - 24 mm	No	17°	-
Jack's Salute		Roja	Redonda. Calibre promedio: 20 mm	No	18°	-
Autumn Royal		Negra púrpura a negra	Cónica. Calibre promedio: 20 - 24 mm	No	18°	Sabor neutral
Early Sweet		Verde claro	Ovalada. Calibre promedio: 22 mm	No	16°	-

Variedad		Características				
		Color	Forma	Presencia de semillas	Grados brix	Observaciones
Timco		Rojo intenso	Bayas grandes y levemente alargadas	No	20°	Excelente sabor, buena vida post-cosecha, sensible a ácido giberélico
Midnight Beauty		Negro	Ovoide. Calibre promedio: 22 - 24 mm	No	16°	-
Sweet Enchantment		Negro	Ovalada. Calibre promedio: 18 - 20 mm	No	17°	Buen sabor y textura
Summer Royal		Negro	Ovoide	No	17°	Bayas firmes y crujiente, sabor dulce

Fuente: Castillo & Correa (2019)

2.1.1.3 Taxonomía. Se debe recalcar que la uva a analizar en el presente proyecto es la uva de mesa ya que, la uva utilizada para la elaboración de vino es diferente, los métodos utilizados van de acuerdo con el producto último. Para el caso del vino, se basa en los rendimientos adecuados con el fin de obtener un alto grado de calidad mientras que, la uva de mesa se busca el mayor rendimiento de la fruta (Arias Cuya, 2017).

Comúnmente, la vid está compuesta de arbustos trepadores, similares a lianas, de un tallo, en su mayoría, sarmentoso, caso contrario, herbáceo y, además, tiene zarcillos en dirección contraria a las hojas.

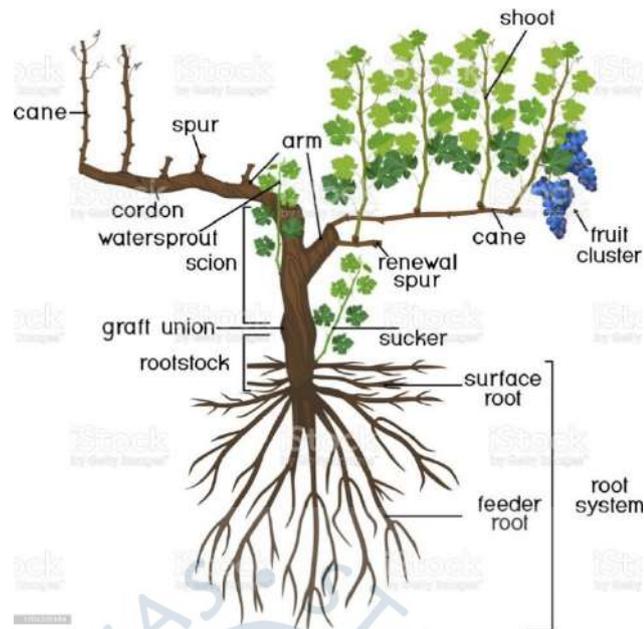


Figura 1. Estructura de la planta de vid.

Fuente: Flaya (2020).

La vid pertenece a la familia de las *vitáceas* y presenta la siguiente taxonomía:

- División: *Espermafitas o Fanerógamas*
- Subdivisión: *Angiospermas*
- Clase: *Dicotiledóneas*
- Subclase: *Dialipétalas*
- Orden: *Ramnales*
- Familia: *Vitáceas*
- Género: *Vitis*
- Especie: *Vitis Vinifera*

2.2. Impacto de la uva

La producción mundial de uva en el periodo de 2000 – 2017 ha mostrado un considerable incremento de 16.8%, lo que significa un crecimiento anual aproximado de 0.9%, lo que significa que el consumo ha crecido de la mano de la producción.

Los principales productores de uva en el mundo son China (18% de producción mundial), situándose como principal productor y consumidor, seguido de Italia con 9.6% de producción mundial, Estados Unidos en tercer lugar con un 8.3% de producción mundial, Chile con un 2.7% en la novena posición. El Perú se ubica en el puesto 21, representando el 0.9% de la producción mundial en el año 2017. Sin embargo, en este periodo (2000 – 2017), la producción creció en un 503%, es decir, un tasa anual de 11.1% aproximadamente (MINAGRI, 2019).

2.2.1. Regional

La uva piurana ha llegado a 37 mercados en el año 2019. Sus principales destinos de exportación son: Holanda, EEUU y el Reino Unido con 13,122, 8,649 y 3,277 toneladas métricas respectivamente (Vera, 2019).

Piura: Exportaciones por Productos y Destinos (US\$ Millones)									
N°	Descripción	Principales Destinos	Anual		Var %	Enero - Junio		Part. %	
			2016	2017	17/16	2017	2018	18/17	2018
1	Pota congelada	España, Corea del Sur, China	124,1	135,7	9%	89,7	180,2	101%	13,2%
2	Pota en conserva	China, Corea del Sur, Japón	122,3	140,5	15%	95,2	161,9	70%	11,9%
3	Mangos frescos o secos	Países Bajos, EEUU, Reino Unido	161,5	142,9	-12%	97,7	151,1	55%	11,1%
4	Fosfatos de calcio naturales	EEUU, Brasil, Indonesia	299,8	206,8	-31%	80,8	112,0	39%	8,2%
5	Residual 6	Panamá, México, EEUU	114,2	175,6	54%	77,2	105,0	36%	7,7%
6	Bananas frescas	Países Bajos, EEUU, Alemania	146,2	138,8	-5%	73,8	84,6	15%	6,2%
7	Gasolina sin tetraetilo de plomo	EEUU, Argentina, Panamá	74,3	90,9	22%	34,8	59,3	70%	4,3%
8	Demás diesel 2	Bolivia, Nicaragua	31,9	120,4	277%	63,1	42,9	-32%	3,1%
9	Demás aceites de pescado	Canadá, EEUU, Australia	79,7	106,2	33%	57,4	37,3	-35%	2,7%
10	Harina de pescado	China, Taipéi Chino, Corea del Sur	35,3	69,7	97%	41,7	32,0	-23%	2,3%
11	Uvas frescas	EEUU, Países Bajos, Rusia	261,3	217,3	-17%	31,8	29,7	-7%	2,2%
12	Oro (incl. oro platinado)	Suiza	5,2	18,2	248%	4,4	24,1	444%	1,8%
13	Filetes congelados de demás pescados	EEUU, Ecuador, Corea del Sur	28,0	23,2	-17%	6,3	23,5	274%	1,7%
14	Colas de Langostinos con caparazón	EEUU, Canadá, España	25,0	48,0	92%	31,3	23,2	-26%	1,7%
15	Carburorreactores tipo queroseno	EEUU, Brasil, Colombia	24,5	31,6	29%	14,6	18,1	24%	1,3%
	Otros		492,3	458,0	-7%	217,3	280,1	28,9%	20,5%
	Total		2 025,6	2 123,7	4,8%	1 017,1	1 365,1	34,2%	100%

Figura 3. Exportaciones por productos en Piura.

Fuente: MINCETUR (2018)

Piura: Exportaciones por Empresas y Productos (US\$ Millones)									
N°	Part. % 2018	Empresa	Principales Productos 2018	Anual					Var % 18/17
				2014	2015	2016	2017	2018	
1	18,2%	Petroperú	Residual 6; Los demás diesel 2	624,0	299,3	248,1	400,2	495,9	24%
2	8,5%	Miski Mayo	Fosfatos de calcio	313,4	346,0	299,4	205,5	230,6	12%
3	4,0%	Seafrost	Conservas de atunes; Pota congelada	69,9	54,2	53,2	75,6	109,3	45%
4	3,5%	Sociedad Agrícola Rapel	Uvas; Demás cajas	44,2	59,4	69,3	74,3	95,4	28%
5	2,8%	Dsm Marine Lipids Perú	Demás aceites de pescado, Demás acidos	95,3	68,7	80,2	106,7	77,5	-27%
6	2,5%	Ecosac Agrícola	Uvas; Demás hortalizas	38,1	49,6	65,1	62,7	68,3	9%
7	2,1%	Copeinca	Harina de pescado; Grasas, aceites de pescado	14,9	32,3	4,2	42,8	56,4	32%
8	2,0%	Productora Andina de Congelados	Pota congelada; Pota en conserva	70,8	50,2	47,7	40,2	55,2	37%
9	1,8%	Inversiones Prisco	Colas de langostinos	23,8	20,6	32,0	47,3	48,3	2%
10	1,6%	Minera Las Lomas Doradas	Oro (incl. oro platinado)	-	6,0	6,4	18,3	44,8	145%
11	1,6%	Costa Mira	Pota en conserva; Pota congelada	20,1	13,3	17,1	25,1	44,3	76%
12	1,6%	Altamar Foods Perú	Pota congelada; Pota en conserva	24,7	18,7	18,4	33,3	43,2	30%
13	1,6%	El Pedregal	Uvas; wilkings y cítricos simil.	31,0	38,9	37,2	24,3	42,3	74%
14	1,3%	Camposol	Mango; Uvas	38,7	29,8	23,8	18,7	35,3	89%
15	1,2%	Sucroalcolera del Chira	Alcohol etílico; Alcohol carburante	7,1	23,0	49,4	29,7	33,0	11%
	45,7%	Otros		1 552,4	1 069,4	974,1	918,0	1 243,3	35,4%
	100%	Total		2 968	2 179	2 026	2 123	2 723	28,3%
		N°Empresas exportadoras		557	580	576	601	632	5%

Figura 2. Empresas exportadoras en Piura.

Fuente: MINCETUR (2018).

Piura posee aproximadamente el 31% del total de hectáreas certificadas en el país. Además, nuestra región ocupa el 3° puesto al interior del país, representando el 4.3% del PBI, después de Arequipa y la Libertad. La economía diversificada de la región justifica su posición ya que es el primer productor de mango, limón y es el segundo productor de uva, después de Ica.

2.2.2. Nacional.

A nivel nacional, las exportaciones de uva han ido creciendo de tal manera que, el Perú ocupa el quinto lugar a nivel mundial en exportaciones de uva fresca y, próximamente, podría convertirse en el abastecedor de China (Huemura, 2018). A su vez, ocupa el tercer lugar en lo que respecta a la productividad, superando a China, Italia, España y Estados Unidos, según lo informado por el por el Ministerio de Agricultura (GESTIÓN, 2017).

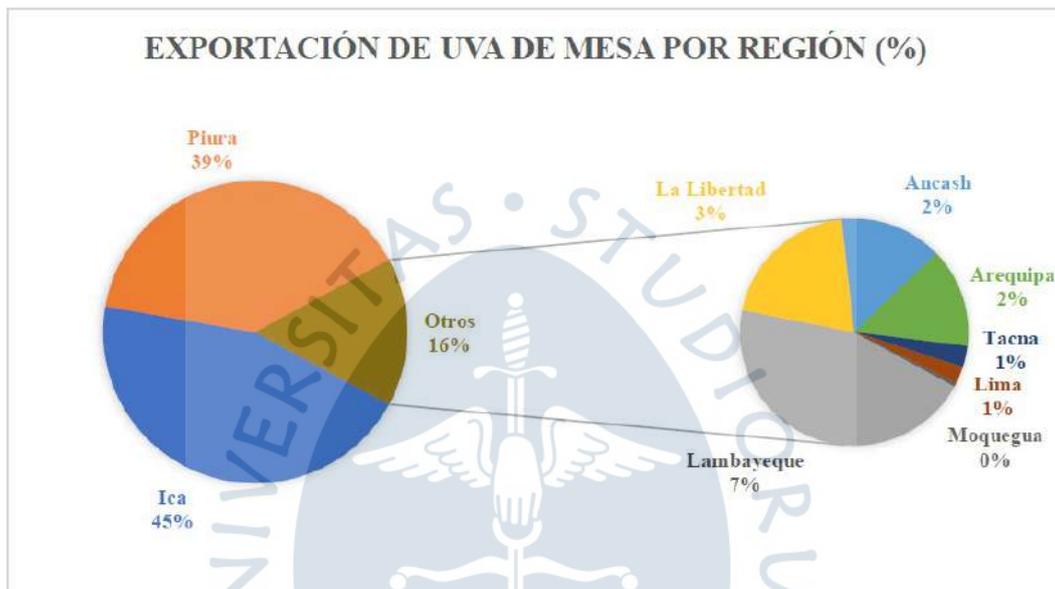


Figura 4. Exportación de uva de mesa por región en el 2017 (%).

Fuente: Huemura (2018)

Al observar la Figura 4. y, hacer una comparativa con la campaña anterior (2015 -2016), se aprecia que regiones como Piura, Tacna, Lima y Ancash han logrado incrementar sus niveles de exportación, cabe resaltar que Tacna y Ancash han sido los que mejor acogida han tenido con la producción de uva de mesa y esto se refleja en el buen rendimiento que han obtenido. Por otra parte, Ica, principal exportadora, ha disminuido en un 5% junto con Lambayeque en un 13% en lo que respecta a sus exportaciones. Esta disminución permite a la región Piura acercarse al podio en la exportación de uva de mesa ya que se disminuyó la brecha que los separaba, pasando de 6.3 millones a 2.1 millones (Huemura, 2018).

Según el Ministerio de Agricultura, las extraordinarias condiciones edafoclimáticas, tanto en el norte como sur del país es una de las razones del buen rendimiento del producto. La variedad de mayor demanda es la uva "Red Globe" que, en el 2016, llegó a cerca de 40 países alrededor del mundo, entre ellos, Holanda, Japón China y Estados Unidos.

2.2.3. Internacional

Debido al gran aporte de nutrientes, vitaminas y minerales de la uva, su demanda ha ido creciendo al ser considerado dentro de los alimentos saludables. Es por ello que en países como México, Estados Unidos, China y la India se incrementa su demanda.

Entre los países que lideran la exportación de uva de mesa a nivel mundial se encuentra: Chile en primer lugar alcanzando un 15% del total exportado en el 2017, luego se encuentra Italia aportando con el 10% de exportaciones mundiales, sin embargo, su participación en el mercado mundial ha ido disminuyendo en una tasa promedio de 1.2% por año.



Figura 5. Evolución mundial de exportaciones de uva.

Fuente: MINAGRI (2019)

En tercer lugar, se encuentra Estados Unidos con el 8% de exportaciones y, similar a Italia, presenta una disminución en su participación de 1.1% y, en cuarto lugar, se encuentra Sudáfrica con un aporte del 7%.

Perú ha logrado posicionarse dentro del mercado mundial desde el 2001 al 2017 con un sorprendente incremento de aproximadamente 4000%, pasando de exportar 7,000tm a 268,000tm lo que significa una tasa de crecimiento anual de 26.1%. A nivel de exportaciones representa el 6% del total de exportaciones a nivel mundial y es considerado como uno de los más importantes comercializadores y proveedores de uva fresca.

EXPORTACIONES MUNDIALES DE UVAS FRESCAS
(Miles de toneladas)

Orden PAÍSES	2001	2005	2010	2011	2012	2013	2014	2015	2016	2017	2018
Total Mundo	2 560	3 620	3 752	4 492	4 038	4 217	4 177	4 203	4 445	4 714	
1 Chile	443	810	779	854	813	856	732	751	708	704	822
2 Italia	594	504	485	502	491	510	453	469	481	490	
3 Estados Unidos	458	539	408	415	422	474	445	393	386	385	
4 Sudafrica	180	230	260	253	268	283	298	324	305	337	324
5 China	1	21	89	106	122	105	126	208	254	280	-
6 Turquía	84	156	238	240	209	203	258	175	173	278	180
7 Países Bajos	94	172	227	256	248	279	276	223	241	269	
8 Perú	7	19	78	122	149	177	268	314	294	268	
9 Hong Kong, China	52	61	81	110	117	147	153	167	205	202	
10 México	98	190	171	138	168	150	153	164	156	196	
11 India	13	52	64	75	114	149	137	76	147	185	
12 España	104	114	127	138	128	140	145	149	157	144	
13 Uzbekistán	32	109	67	112	119	21	59	107	96	136	
14 Egipto	5	25	84	624	116	88	113	116	113	113	
15 Australia	34	54	31	32	42	78	83	87	114	111	114
Otros países	363	564	563	514	512	557	478	480	614	616	

Figura 6. Ranking de exportaciones a nivel mundial.

Fuente: MINAGRI (2019)

2.3. Características agroexportadoras

2.3.1. Regionales

El principal foco productivo de uva en el Perú, gracias a sus riquezas climatológicas y un suelo adecuado, se encuentra ubicado en la región de Ica, la cual tiene concentrada más del 40% de la producción total nacional. Siguiendo a nivel de provincias, las zonas de Ica, Chincha y Pisco, debido a su importante producción en cuanto a volumen.

En la parte norte, se impone la región Piura, la cual se ha desarrollado y comparte junto a Ica la agroindustria de producción y exportación.

Debido al alto rendimiento que se han tratado los cultivos y a su potencial exportadora, la cantidad total de uva cosechada ha crecido sostenidamente y ha alcanzado un rendimiento considerable promedio nacional de 22 toneladas métricas por hectárea cuadrada. “A nivel regional, Ica y Piura son las regiones con mayor rendimiento agrícola, alrededor de 20 – 30 toneladas métricas por hectárea, respectivamente.”(Briceño, 2016).

Actualmente Piura es la tercera región más importante del interior del país después de las regiones de La Libertad y Arequipa y alberga a casi 3 millones de habitantes de los 33 millones en total, por lo que, se puede decir que contribuye con la exportación del país, con lo que sus agroexportaciones representan el 12% de las exportaciones agroindustriales del país. Los principales productos que la región exportó fueron: uva, banano y mango. Las exportaciones de estos productos alcanzaron récords, siendo en cuestión de agroexportación la más importante de la región la Agrícola Rapel (MINCETUR, 2019).

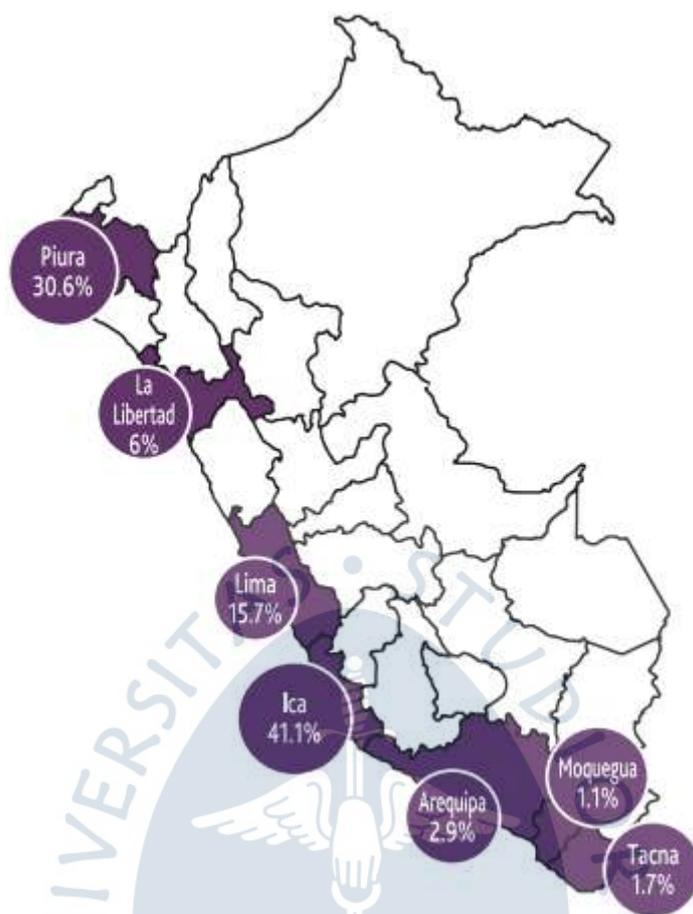


Figura 7. Principales nodos de producción en el Perú.

Fuente: Briceño (2016).

En las bases productivas de uva, su gran mayoría de la producción es realizada por las grandes empresas agropecuarias que producen diferentes productos. En Piura, los complejos agroindustriales más destacados son El Pedregal, Sociedad Agrícola Rapel -perteneciente al consorcio chileno Verfrut- y la empresa agrícola San Juan. Lo mismo ocurre en los clústeres de La Libertad e Ica, donde destacan grandes empresas como Camposol, Agrokasa, Beta y Don Ricardo. Todas estas empresas funcionan como productores, acopiadores y exportadores; es decir, manejan la cadena productiva de manera integrada. Si bien existen pequeños productores de uva, estos, por lo general venden sus cosechas a las empresas grandes o para su consumo local (Briceño, 2016).

2.3.2. Nacional

Como bien se ha mencionado, la mayoría de agroexportadoras en el Perú están concentradas en provincias como Piura, Ica, siendo en estas localidades donde mayor concentración de empresas, tales como: El Pedregal S.A, Sociedad AGRÍCOLA RAPEL S.A.C., ECOSAC AGRÍCOLA S.A.C, COMPLEJO AGROINDUSTRIAL BETA S.A, EXPORTADORA SAFCO PERU S.A, AGRICOLA ANDREA S.A.C., SOCIEDAD AGRÍCOLA SATURNO S.A, AGRICOLA DON RICARDO S.A.C., CORPORACION AGROLATINA S.A.C.(Briceño, 2016), las cuales representan más del 50%

de las exportaciones totales, se encargan tanto de la producción como de la exportación y acopio del cultivo; es decir, se encargan de toda la cadena productiva del valor de la uva.

Estos tipos de agroexportadores son los más comunes en este mercado, aunque no son los únicos; algunas empresas grandes (como Agroexportadora Cachiche y Lamas Import/Export) realizan solo el acopio de la uva para su exportación, ya que compran el producto a pequeños agricultores (Moreyra, 2019).



Figura 8. Cajas con uvas para exportación.

Fuente: Briceño (2016).

En el Perú, las principales empresas exportadoras de uvas durante el año 2019 fueron Sociedad Agrícola Rapel SAC, El Pedregal SA y Ecosac Agrícola SAC. Según Promperu. El Pedregal S.A. envió 32,708 toneladas de uva por US\$ 94 millones. En donde, la empresa creció en valor y aumentó sus exportaciones con un incremento alrededor del 25% con respecto al año anterior. El buen desempeño de la empresa le permitió ser pionero en los últimos 3 años como el principal agroexportador de uvas del Perú, con un 9% de participación en las exportaciones totales (Moreyra, 2019).

Asimismo, la Sociedad Agrícola Rapel S.A.C., cuya empresa representa importantes exportaciones disminuyó sus envíos durante la campaña del 2019 un 35% en valor y en volumen aproximadamente (Moreyra, 2019).



Figura 9. División de empresas agroexportadoras en el Perú.

Fuente: Elaboración propia.

2.3.3. Mundial

Según las estadísticas de FAOSTAT la producción mundial de uva se ha incrementado un 17% en el periodo de 2000-2017, es decir un crecimiento promedio anual de 0,9%, el cual se relaciona con la demanda y consumo mundial de la uva y para el año 2017 la producción mundial fue de 87,4 millones, cifra que se ha ido incrementando con la llegada de nuevos mercados a la agroindustria de la uva (Moreyra, 2019).

Las exportaciones mundiales de uvas fresca representan alrededor del 6% de la producción mundial, y entre los tipos de uva que destacan para mayor exportación se encuentran la Thompson Seedless, Crimson Seedless, Red Globe, y la Flame Seedless.

Tabla 2. Cuadro de participación mundial del Perú

Países exportadores	Participación mundial al 2017	Millones de toneladas
China	18 %	13.2
Italia	9.6%	-
Estados Unidos	8.3%	-
Chile	2.7%	-
Perú	0.9%	-

Fuente: Elaboración propia

EXPORTACIONES MUNDIALES DE UVAS FRESCAS
(Miles de toneladas)

Orden PAÍSES	2001	2005	2010	2011	2012	2013	2014	2015	2016	2017	2018
Total Mundo	2 560	3 620	3 752	4 492	4 038	4 217	4 177	4 203	4 445	4 714	
1 Chile	443	810	779	854	813	856	732	751	708	704	822
2 Italia	594	504	485	502	491	510	453	469	481	490	
3 Estados Unidos	458	539	408	415	422	474	445	393	386	385	
4 Sudafrica	180	230	260	253	268	283	298	324	305	337	324
5 China	1	21	89	106	122	105	126	208	254	280	-
6 Turquía	84	156	238	240	209	203	258	175	173	278	180
7 Países Bajos	94	172	227	256	248	279	276	223	241	269	
8 Perú	7	19	78	122	149	177	268	314	294	268	
9 Hong Kong, China	52	61	81	110	117	147	153	167	205	202	
10 México	98	190	171	138	168	150	153	164	156	196	
11 India	13	52	64	75	114	149	137	76	147	185	
12 España	104	114	127	138	128	140	145	149	157	144	
13 Uzbekistán	32	109	67	112	119	21	59	107	96	136	
14 Egipto	5	25	84	624	116	88	113	116	113	113	
15 Australia	34	54	31	32	42	78	83	87	114	111	114
Otros países	363	564	563	514	512	557	478	480	614	616	

Figura 10. Exportación Uvas Mundialmente.

Fuente: Trademap.

En un inesperado 8vo lugar, el Perú se posiciona en el mercado mundial en pocos años, ha pasado de exportar pocas decenas de miles de toneladas a comienzos del 2000 a exportar hasta 268 mil toneladas en el año 2017, con una importante participación de 6% del total de exportaciones mundiales (Moreyra, 2019).

Las exportaciones peruanas no se cruzan con las chilenas debido a que no se superponen las épocas de cosecha, la cosecha chilena es posterior a las fechas de cosecha peruanas y agregando que la calidad y producción de uvas chilena ha ido disminuyendo en los últimos años debido a su dinamismo y problemas climáticos, lo cual dificulta la disponibilidad de agua y tierra, situación la cual el mercado peruano ha sacado ventaja y se ha impuesto en el mercado mundial exportador. Sin embargo, con Sudáfrica el mercado peruano compite directamente porque las cosechas y exportaciones salen en el mismo período (Moreyra, 2019).

Entre los principales países con mayor producción y exportación mundial de uva se encuentran: China, Francia, Italia, Estados Unidos, y España. Sin embargo, actualmente entre los países que encabezan las exportaciones mundiales de uvas se encuentra Chile con un 15% del total exportado en el 2017 (704 mil toneladas). (Moreyra, 2019)

Como otra potencia exportadora se encuentra Italia, con una participación del 10% del total de exportaciones mundiales, sin embargo, cada año se va reduciendo su participación como productora y exportadora, en los últimos 15 años se ha reducido su participación en 18%, perdiendo peso en el mercado mundial.

Estados Unidos es otro de los grandes países con relevante participación de 8% en las agroexportaciones mundiales. En el orden de importancia significativa después de Estados Unidos en las exportaciones mundiales le siguen Sudáfrica (7%), China (6%), Turquía (6%) y Países Bajos (6%), poniendo a Holanda como potencia comercializadora gracias a su puerto de Ámsterdam y su ubicación estratégica, así como su logística disponible, convirtiéndose en un importante centro de distribución mundial, el cual ayudaría a reexportar frutos provenientes de América Latina hacia mercados muy lejanos de Europa del Este, Asia y Oceanía (Moreyra, 2019).

Entre los 10 últimos primeros lugares, el Perú se ha posicionado en el impotente mercado mundial, en pocos años ha pasado de exportar 7000 toneladas a exportar 270 mil toneladas a fines de la cosecha del 2017, con una importante participación del 6% del total de exportaciones mundiales, llegando un país prometedor y dinámico del sector exportador, y que viene posicionándose en el mercado como uno de los más importantes países comercializadores y productores del mundo.” (Moreyra, 2019).

2.4. Medidas de calidad

Para llegar a los mercados tanto locales como internacionales, es importante que los productos lleguen con la mejor calidad posible, más aún si el destino es internacional, ya que debe competir como un producto óptimo de calidad global, esta calidad es definida por muchos parámetros como apariencia, textura y sabor, y una serie de características que se deben cumplir momento previamente a la cosecha. Esto, con el fin de garantizar un buen potencial de almacenamiento y transporte, así como la aceptabilidad por parte del consumidor final.

La uva de mesa presenta diversas características morfológicas considerables en la etapa de pre y postcosecha, por lo que para alcanzar el nivel de calidad exigido por los mercados de destino es necesario poder conocer estas características. Entre ellas destacan una alta relación entre la superficie y volumen, sabor, color y la existencia de una estructura vegetativa como el raquis sin protección cerosa.

Como cualquier otro producto, la fruta como el raquis cambia constantemente, tanto en estructura como en composición, lo que ocasiona alteraciones en la apariencia y en el sabor de la fruta por deshidratación, pérdida de agua desde la baya y raquis y degradación de ácidos orgánicos, acompañado de pardeamientos en el raquis y la baya.

Estos cambios se intensifican a medida que aumenta el período entre la cosecha y después, en la venta del producto, la cual puede superar los 90 días estando en el mercado dependiendo de la variedad y de su destino (Torres et al., 2017).

En el proceso de producción, durante la etapa de previa a la cosecha y cosecha, se definen tanto las características organolépticas de la variedad, como el potencial de almacenamiento. En estas variedades llegan a existir parámetros relacionados con los

estándares de calidad con el fin de cumplir con los requisitos que emite el mercado de destino. A nivel de manejo del producto en el campo, es importante disponer de índices de madurez para observar el grado de avance en el desarrollo de la fruta.

2.4.1. Color

El color óptimo de la baya de uva es un atributo que depende de las variedades de uva que existe. Siendo para el caso de las uvas rojas, una tarea más difícil de cumplir por las condiciones agroclimáticas, en donde el color se ve afectado por la luminosidad, temperatura, manipulación o uso de reguladores de crecimiento en las plantas que afectan al vigor de esta.

En la mayoría de los casos se aplican productos químicos regulados como el etileno y ácido abscísico para mejorar la homogeneidad. En variedades rojas es muy importante el desarrollo de color, ya que complementa los índices de cosecha como su relación con SST (sólidos solubles totales), sino se ha completado el desarrollo del propio color de la variedad, no estará en condiciones de ser cosechado y posteriormente llevado al mercado (Torres et al., 2017).

A nivel agroindustrial, la forma más común para evaluar el desarrollo del color rojo es a través del uso de escalas hedónicas, las cuales permiten identificar y evaluar visualmente la cobertura a nivel de baya, intensidad del color y a nivel de racimo.



Figura 11. Escala de color para racimos Red Globe. Desde bayas verdes en la parte izquierda a bayas rojas como extremo derecho.

Fuente: Torres et al (2017).

En la variedad de uvas verdes, la calidad se define mucho por el color de esta, en términos comerciales, existen requisitos muy exigentes por mercado de destino mundialmente y, por lo tanto, se pueden distinguir categorías de color de la baya. Similar a las variedades coloreadas, el color verde es mayormente afectado por condiciones de manejo, estado de madurez en la cosecha y condiciones climáticas al momento de ser transportado a su destino final.

2.4.2. Tamaño de bayas

Al momento previo a la cosecha, los agricultores comúnmente se fijan en un estándar de calidad definido por la vista humana, el tamaño del racimo de la baya. El cual luego viene definido por mediciones del tamaño y así después poder organizar, en donde se puede

categorizar a la uva final respecto al destino que requiera tal característica. Estos atributos apuntan a tener una forma de racimo, que puede ser globosa, cilíndrica, o cónica (Torres et al., 2017).



Figura 12. Formas del racimo de la uva

Fuente: Torres et al (2017)

La forma de la baya tiene que ser la característica de la variedad, tanto en su diámetro polar como ecuatorial y estará determinada por las manipulaciones y labores de manejo utilizada en interacción con las condiciones agroclimáticas. A nivel de diámetro ecuatorial, esta mostrará el calibre de la baya, el cual, está muy ligado a la categoría de calidad al momento de entregar el producto (Torres et al., 2017).

2.4.3. Textura

La textura es por último un parámetro relevante y no tan notable en la calidad de la uva de mesa, debido a que el elemento más conocido es la firmeza, caracterizado con la consistencia, dureza, entre otros, teniendo un efecto directo en la percepción organoléptica por parte del consumidor.

Si bien es afectado por el manejo del ambiente y la manipulación de esta, existe una alta influencia de la variedad o genotipo de la uva. A nivel agro comercial, la firmeza es evaluada con equipos tecnológicos tales como Firmtech o Durofel, cualitativamente de forma manual o utilizando métodos cuantitativos como un manual actualizado y completo, lo cual esto último igual dificulta mucho la rápida labor de medición de calidad en la agroindustria, al no estar automatizado (Torres et al., 2017).

2.4.4. Peso

La estimación del peso es un dato importante porque dependiendo cómo se use nos permite determinar otro tipo de información, como un monitoreo del total de producción durante la producción; además de esto el peso se puede usar para encontrar un aproximado de los grados Brix (Cruz, 2016), en esta tesis de grado se aproxima el peso del racimo a través

de variables independientes como el número de baya, peso de baya, calibre, longitud del racimo y grados Brix. Pero al trabajar con imágenes RGB es muy difícil obtener los grados brix por lo que se plantea a futuro encontrar este parámetro de calidad de manera indirecta, primero encontrando el peso del racimo para después encontrar la estimación de los grados Brix.

2.5. Espacio de color

El espacio de color para el trabajo de una imagen es un factor fundamental dentro del campo del *Machine Learning* debido a que la imagen es un rasgo característico el cual brinda información relevante acerca de un objeto de estudio determinado. Dentro de este apartado se tomará en cuenta los espacios de color básicos como lo es RGB, HSV, CIE L*a*b entre otros, con el fin de estudiar su estructura y su aplicación como el fin de analizar los diferentes espacios de color asignados.

2.5.1. Espacio RGB

El espacio de color RGB por sus siglas en inglés R(Red), G(Green) and B(Blue), es el espacio de color aditivo definido por los colores primarios rojo, verde y azul, donde se puede obtener cualquier color de acuerdo con las proporciones de peso en cada color. Cuantos más colores se agreguen, se obtendrá un color más claro debido a que emiten luz.

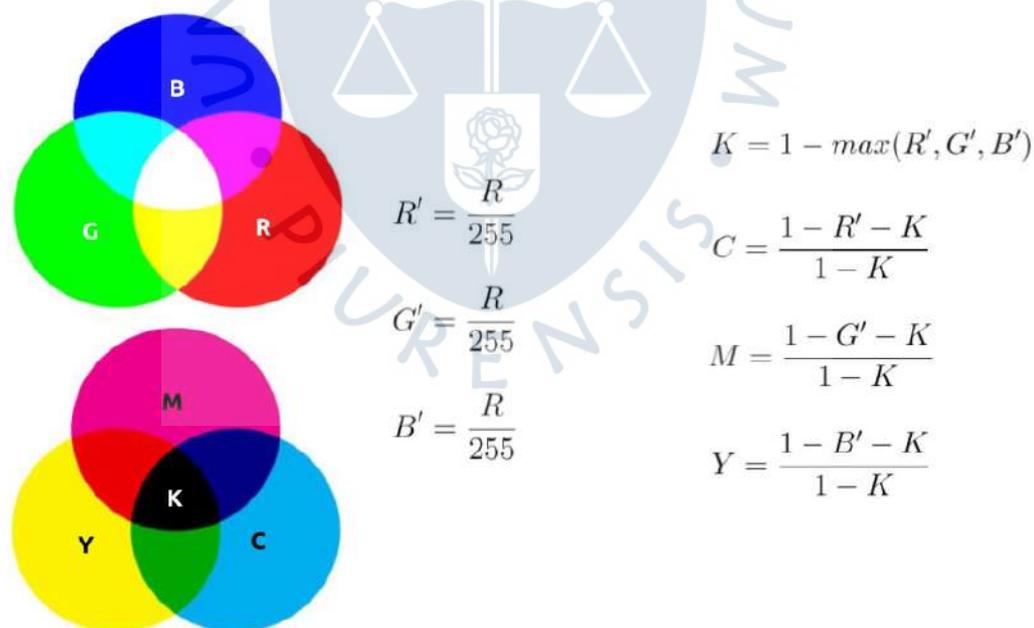


Figura 13. Estructura básica del espacio de color RGB.

Fuente: Thevenot (2019).

Al definir colores en estas dimensiones, es necesario conocer la secuencia de colores en el espectro de colores, por ejemplo, que una mezcla 100% de rojo y verde produce

amarillo. El modelo de color RGB a menudo se representa como un cubo al mapear las dimensiones rojo, verde y azul en los ejes x, y e z en el espacio 3D. Esto se ilustra en el ejemplo interactivo a continuación, donde todas las posibles mezclas de colores se representan dentro de los límites del cubo.

El modelo de color RGB no es un modelo especialmente intuitivo para crear colores en código. Si bien es posible que pueda adivinar la combinación de valores a usar para algunos colores como el amarillo (cantidades iguales de rojo y verde) o el color rojo que se usa en las botellas de Coca-Cola (mucho rojo con un poco de azul), menos los colores puros son mucho más difíciles de adivinar en este modelo de color

2.6. Procesamiento de imágenes

Se puede definir como un conjunto de métodos o algoritmos con la finalidad de extraer las características más importantes, además de eliminar aquellos elementos que no representen ninguna importancia después del preprocesamiento o del procesamiento propiamente dicho.

2.6.1. Preprocesamiento

También conocido como procesamiento de bajo nivel, tiene como finalidad eliminar el ruido como consecuencias de falta de iluminación o problemas propias de cámara, también puede mejorar algunos aspectos como el brillo de la imagen o eliminar sombras de esta, se puede decir que se puede mejorar la imagen excluyendo o mejorando ciertas partes de la imagen original (Basilio Vela, 2016).

2.6.1.1. Aumento y reducción de contraste. El contraste hace referencia a la diferencia entre las partes más oscuras y claras en una imagen. Se dice que una imagen tiene buen contraste si la imagen posee diversos valores desde el negro al blanco. Durante una variación del contraste el valor de todos los píxeles es modificado, esto con la finalidad de hacer más uniformes las intensidades de tonos claros y oscuros, a mayor contraste la imagen se acerca a una imagen a blanco y negro, caso contrario cuando el contraste disminuye esta se acerca a una imagen en escala de grises.

2.6.1.2. Filtro lineal. El filtrado es un método para modificar la imagen de entrada, permite eliminar ciertas partes que tengan ruido y/o resaltar ciertos atributos. La operación matemática con la que se representa un filtrado es la convolución esta se puede definir mediante la siguiente ecuación:

$$I'(x, y) = \sum_{(i, j) \in R} I(x + i, y + j) \cdot H(-i, -j)$$

También se puede escribir como:

$$I' = I * H$$

Donde x e y son las coordenadas de los píxeles en la imagen de entrada.



Figura 14. Aumento de contraste.

Fuente: Martínez Libreros & Potes Blandon, (2019)

Dentro de los filtros lineales más comunes para eliminar el ruido se tiene el filtro Gaussiano y el filtro mediano, ambos pueden ser usados con una matriz de 3×3 o 5×5 , el primero es ideal para quitar el ruido Gaussiano y, pero a medida que se aumenta la matriz del kernel se desenfoca la imagen original. El filtro mediano además de quitar el ruido permite obtener un mejor contraste entre el objeto y la imagen de fondo.

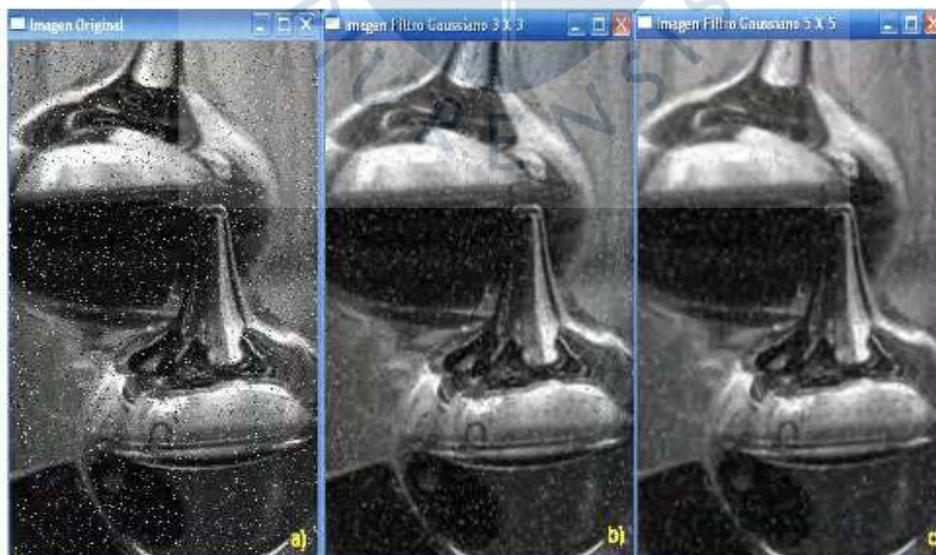


Figura 15. Filtro Gaussiano.

Fuente: Martínez Libreros & Potes Blandon (2019).

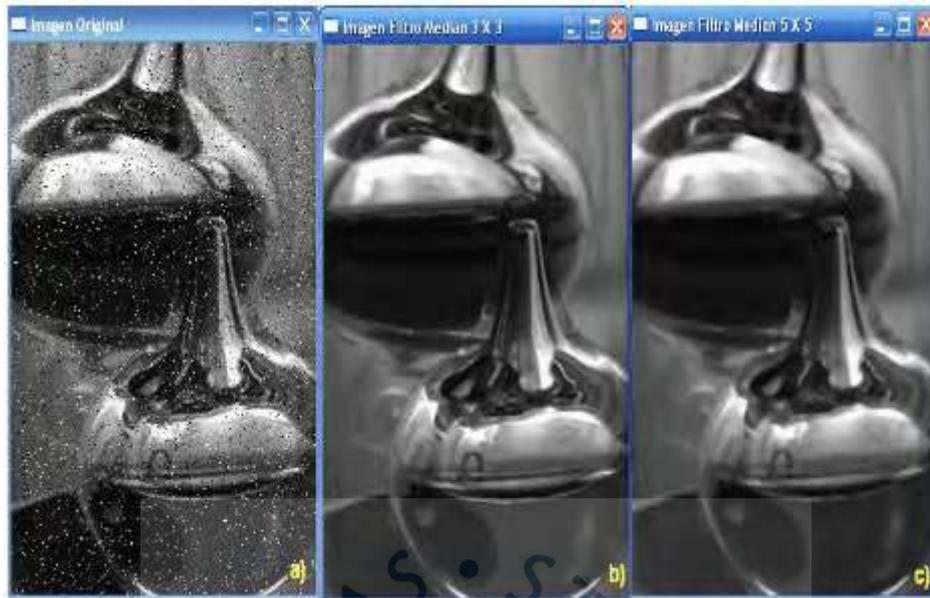


Figura 16. Filtro mediana.

Fuente: Martínez Libreros & Potes Blandon (2019).

2.6.2. Procesamiento propiamente dicho

2.6.2.1. Segmentación. La segmentación es un proceso por el cual se separa una imagen en regiones con la finalidad de hacer más fácil un posterior análisis automático (Bustamante, 2014).. Encontrar los límites de una palabra, localizar la cara o en concreto como se estudiará en este trabajo, encontrar los límites o contornos de la uva dentro de una imagen son ejemplos de problemas de segmentación.

2.6.2.2. Threshold. Es un método que sirve para binarizar una imagen, convierte imágenes en escala de grises en imágenes binarias, esto lo logra asignándoles a todos los píxeles un valor de 0 o 1 dependiendo del valor umbral (*threshold*) que es fijado, pero determinar este valor de umbral a veces puede ser difícil, los umbrales podrán ser locales como globales y en ciertas ocasiones puede haber más de 2 valores de umbrales.

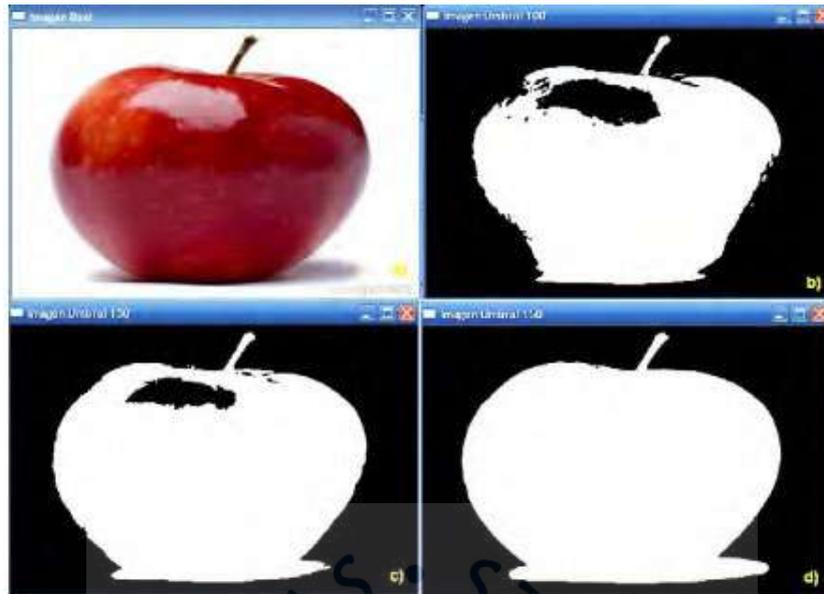


Figura 17. Imagen Binarizada por Trheshold.

Fuente: Martínez Libreros & Potes Blandon (2019).

2.6.2.3. Método de Otsu Binivel. También conocido como segmentación por umbralización, para obtener el umbral máximo se maximiza la varianza, medida de la dispersión de valores, entre clases. El objetivo del método Otsu es encontrar el umbral que haga que cada clase tenga la menor dispersión, y a su vez. Para encontrar el umbral máximo se puede apoyar del histograma de la imagen de entrada, que nos proporcionan los valores de pixeles en la escala de grises.

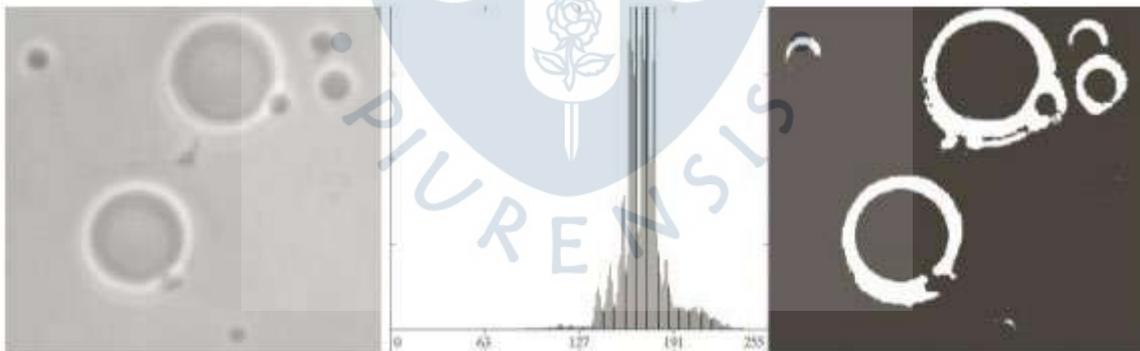


Figura 18. Imagen original, histograma e imagen segmentada.

Fuente: Jiménez Hernández (2016)

2.6.2.4. Método de Otsu Multinivel. El método de Otsu puede ser aplicado a un número mayor de clases, dependiendo de la salida de imagen que se quiera obtener, una imagen puede tener un histograma con 3 picos, o modas diferentes, en este caso se buscaran esos valores de umbrales que determinarán en cuantas partes se divide la imagen (Jiménez Hernández, 2016).

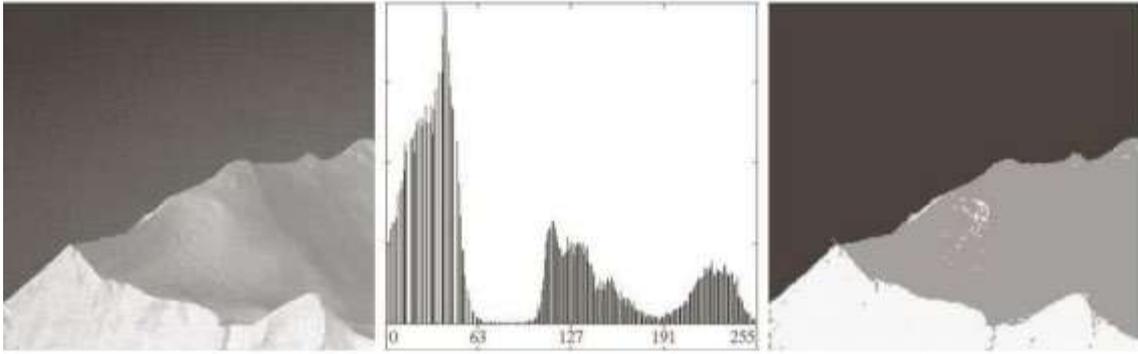


Figura 19. Imagen original, histograma e imagen segmentada.

Fuente: Jiménez Hernández (2016)

2.6.2.5. Algoritmo de Canny. Es uno de los principales métodos para la detección de bordes, esto lo logra mediante una serie de convoluciones y se basa en 3 criterios: Evitar la eliminación de bordes importantes y no dar falsos bordes, la distancia entre la posición real y la localizada del borde se debe minimizar y como último criterio es no identificar múltiples pixeles de contornos donde solo exista un único contorno (Martínez Libreros & Potes Blandon, 2019).

Como primera pasa del algoritmo de Canny se tiene la obtención del gradiente en cual se calcula la orientación y magnitud del vector gradiente de cada pixel. Dentro de este lo primero que se hace es pasar un filtro gaussiano para suavizar la imagen y eliminar posibles ruidos. A continuación, se muestran dos máscaras para realizar el filtro gaussiano.

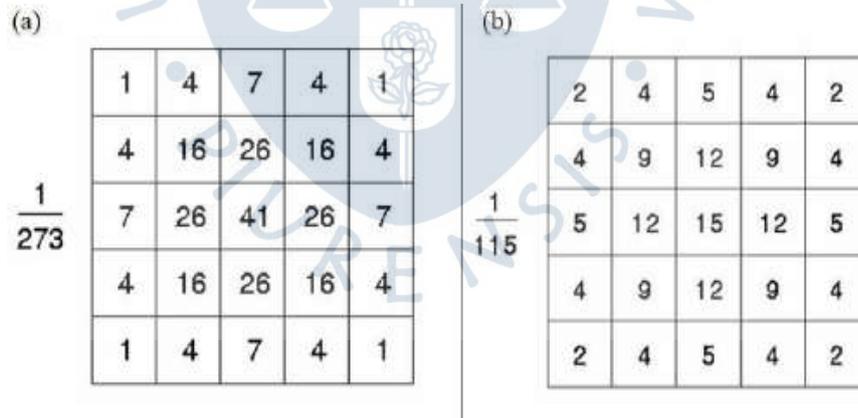


Figura 20. Máscaras recomendadas para el filtro gaussiano.

Fuente: Valverde (2007).

Como segundo paso se tiene la supresión no máxima del gradiente, en la cual se tiene 4 direcciones en 0° , 45° , 90° y 135° respecto al eje horizontal, en la que para cada pixel se encuentra la dirección que mejor se acerca a la dirección del Angulo de la gradiente.

Para explicar mejor se tiene:

El gradiente de una imagen $f(x, y)$ en un punto (x, y) se define como un vector en dos dimensiones el cual está definida por:

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} f(x, y) \\ \frac{\partial}{\partial y} f(x, y) \end{bmatrix}$$

El vector es perpendicular al borde, donde la dirección del vector G apunta en la dirección máxima de f en el punto (x, y) , entonces se cumple que :

$$|G| = \sqrt{G_x^2 + G_y^2} = |G_x| + |G_y|$$

$$\theta(x, y) = \tan^{-1} \frac{G_x}{G_y}$$

Si el valor de la magnitud del gradiente es menor a uno de sus dos vecinos en la dirección obtenida se asigna 0 a dicho pixel, caso contrario se asigna el valor que tenga la magnitud del gradiente. A la salida se tendrá una imagen I_n .

Como tercer paso se tiene la histeresis de umbral de supresión no máxima,

En resumen se tiene:

E_m : imagen de entrada, magnitud del gradiente

E_θ : imagen de entrada, orientación del gradiente

I_n : imagen de salida

Para cada uno de los pixeles (i, j) :

- Encontrar la dirección d_k que mejor se aproxima a la dirección $E_\theta(i, j)$, el cual es la pendiente del borde.
- Si $E_m(i, j)$ es más pequeño que uno de sus dos vecinos en la dirección d_k , al pixel (i, j) de I_n se le asigna el valor de 0, $I_n(i, j) = 0$ (supresión), caso contrario $I_n(i, j) = E_m(i, j)$

- Devolver I_n

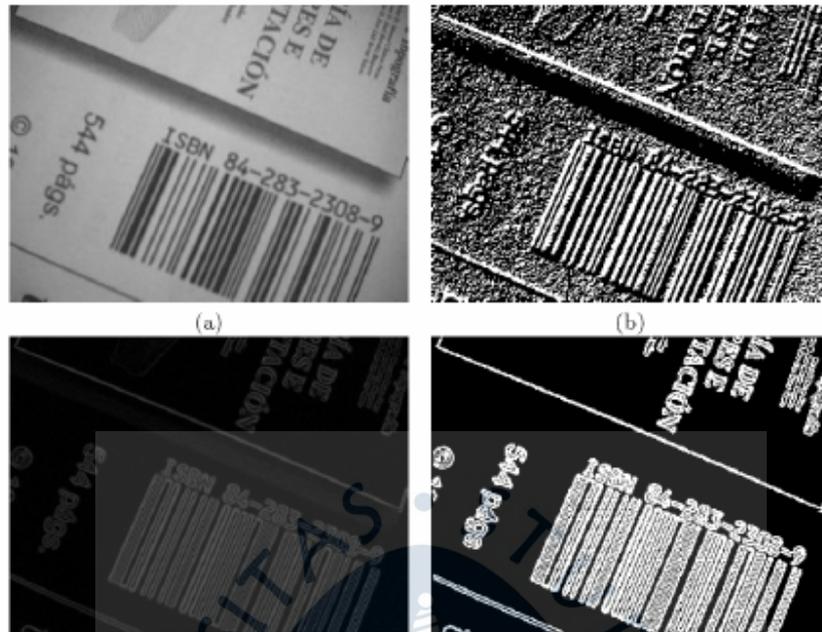


Figura 22. a) imagen original; b) orientación; c) supresión no máxima; d) histéresis del umbral.

Fuente: (Valverde, 2007)

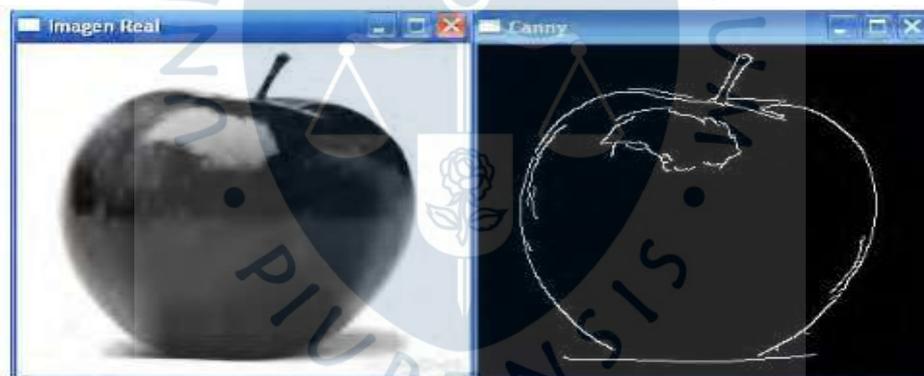


Figura 21. Algoritmo de Canny.

Fuente: Martínez Libreros & Potes Blandon (2019).

2.7. Redes neuronales

2.7.1. Red neuronal Artificial

Una neurona artificial se basa en el funcionamiento de una neurona biológica que responde a impulsos nerviosos, a diferencia de una neurona biológica una neurona artificial responderá a entradas numéricas, las n neuronas x_i están enviando señales de entradas, que son los valores numéricos de “algo”. Los valores w_{ij} representan los pesos sinápticos, el primer y segundo índice representan de donde y hacia qué neurona va la información.

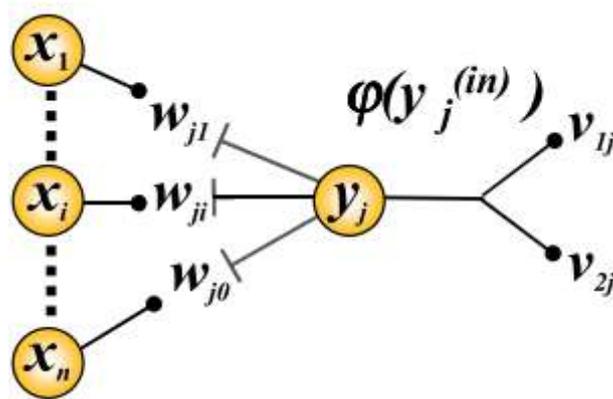


Figura 23. Esquema de una neurona artificial.

Fuente: Izaurieta & Saavedra (1999).

Matemáticamente se puede expresar mediante la siguiente expresión.

$$Y_j^{in} = \sum_{i=1}^n W_{ji} X_i$$

Los pesos sinápticos representan el valor o la importancia que tienen los valores de entrada, adicionalmente se suele agregar un umbral(θ_i) que resta el potencial al resultado post-sináptico(Larrañaga, n.d.), por lo que matemáticamente quedaría la siguiente expresión:

$$Y_j^{in} = \sum_{i=1}^n W_{ji} X_i - \theta_i$$

Una red neuronal artificial es el conjunto de capas donde cada una de estas posee un conjunto finito de neuronas, las capas se pueden dividir en las siguientes:

- Capa de Entrada (INPUT): Reciben información del exterior.
- Capas ocultas: Las cuales están encargadas de realizar el trabajo de la red, en el siguiente gráfico vendrían a ser las capas 1 y 2(layer 1 y layer 2).
- Capas de salida (OUTPUT): Poseen la información de todo el trabajo realizado antes

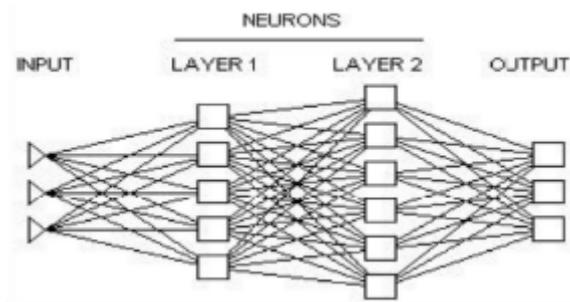


Figura 24. Capas de una red neuronal.

Fuente: Requena et al. (2017).

Las redes neuronales hoy en día se pueden utilizar para desarrollar distintos problemas, tales como: clasificación de patrones, agrupamiento, modelación, predicción y optimización (Mohammadhassani et al., 2013).

Para el problema de clasificación de patrones se trabaja con la identificación de patrones desconocidos en las variables de entrada y, con ayuda del aprendizaje automático supervisado (Supervised machine learning), se determinan las características de cada clase presente para poder clasificar las variables de entrada.

En el caso del agrupamiento, se trabaja un concepto similar con respecto a la clasificación de patrones, la diferencia radica en el aprendizaje del algoritmo ya que, aquí se trabaja con el aprendizaje no supervisado (Unsupervised machine learning), el secreto de este aprendizaje es que el mismo algoritmo se va dando cuenta de las semejanzas y diferencias entre las variables de entrada y de esta forma los va agrupando.

La modelación y predicción se resuelve fácilmente con los modelos de ANN, ya que al realizar un entrenamiento y una validación se pueden crear modelos aptos para realizar una predicción de un suceso determinado.

Finalmente, para la optimización se busca maximizar o minimizar una determinada función con sus respectivas restricciones para obtener una salida lo más cercana al valor esperado

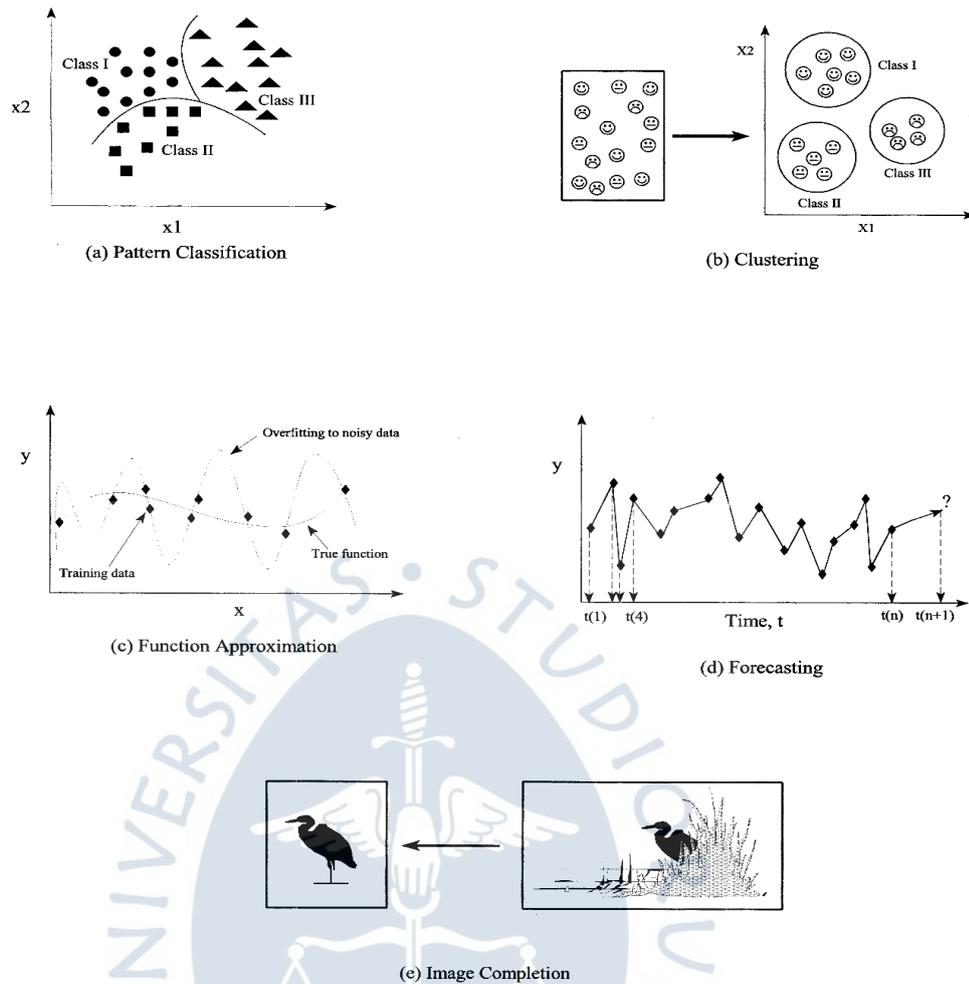


Figura 25. Problemas de interés de las redes neuronales artificiales.

Fuente: Mohammadhassani et al., (2013).

El modelo perceptrón fue el primer modelo propuesto en el año 1958 por Rosenblatt, el cual traía consigo una ley de aprendizaje de pesos los cuales se iban ajustando. Sin embargo, la red perceptrón no podía desarrollar los problemas de linealidad XOR, por lo que llega la propuesta de las MLP (Multi layer perceptrón) en las que se logra resolver este impedimento por medio de más capas de forma secuencial.

Esto no altera el concepto inicial de la red perceptrón de una sola capa ya que, como se sabe, se basa en una regresión lineal, lo que si sumamos varias regresiones si sigue obteniendo como resultado la regresión lineal.

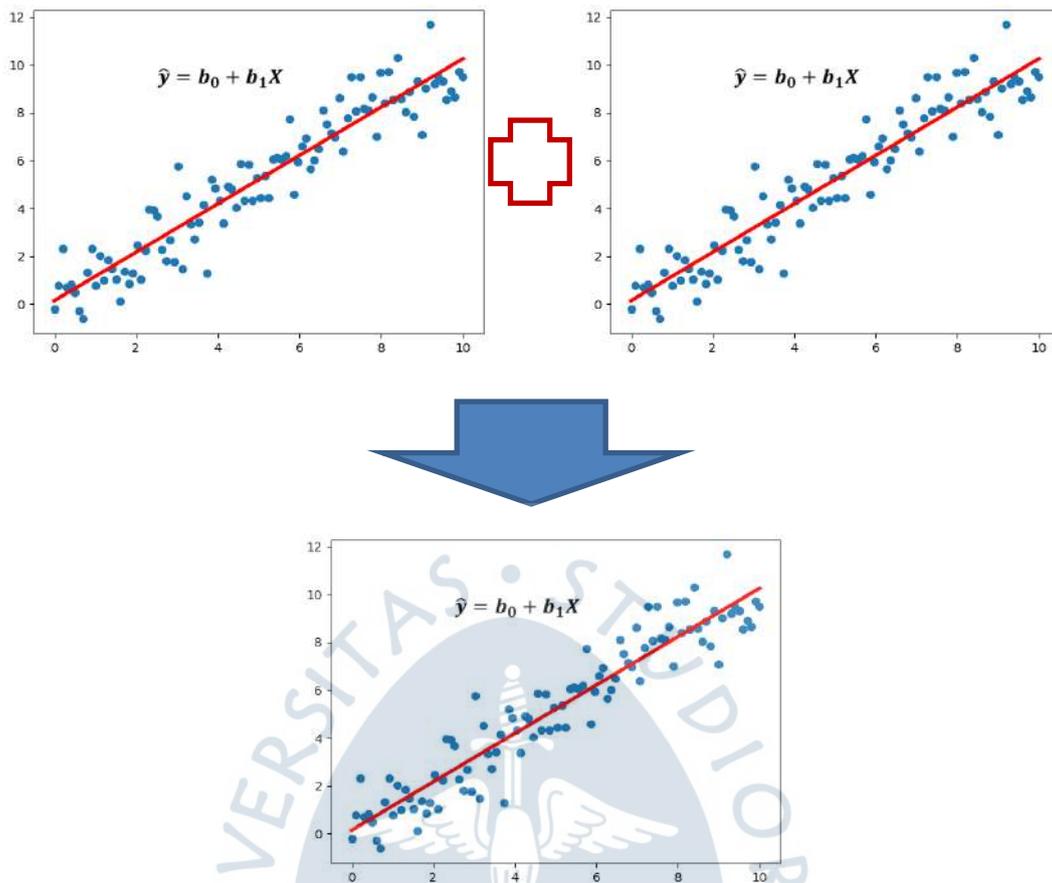


Figura 26. Ejemplificación de la red perceptrón multi capa.

Fuente: Elaboración propia.

Luego de ello, se vienen a implementar las funciones de activación las cuales determinan si el valor de entrada de la neurona es relevante para el aprendizaje, de ser así, se adquiere este nuevo conocimiento de lo contrario se toma como no decisivo en el aprendizaje.

Para minimizar el error de aprendizaje se aplica el gradiente descendiente como optimizador ya que, es uno de los que mejores resultados a dado junto con el optimizador Adam.

Este optimizador busca minimizar el error una vez iniciado el entrenamiento. Sin embargo, existe la posibilidad de que no se caiga en el mínimo global ya que, en los casos reales, los planos no son así, es por ello que se debe elegir adecuadamente una tasa de aprendizaje de tal forma que se evite el overfitting y a la vez se obtenga un modelo que generalice correctamente.

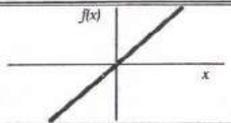
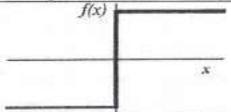
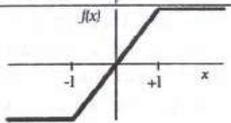
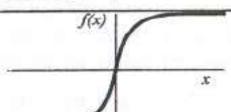
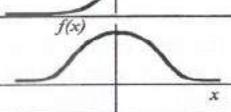
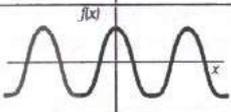
	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(ax + \varphi)$	$[-1, +1]$	

Figura 27. Funciones de activación más utilizadas.

Fuente: ibiblio.org (2003)

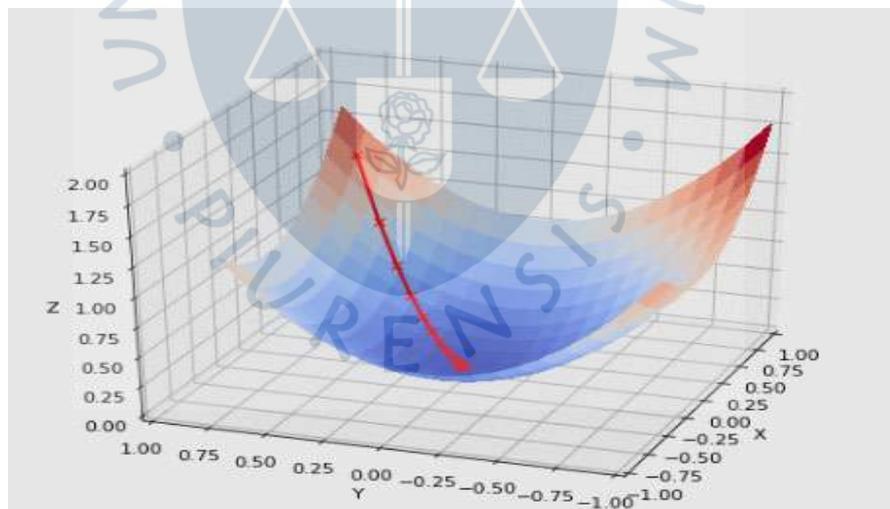


Figura 28. Gráfica del descenso de gradiente.

Fuente: Kathuria (2018)

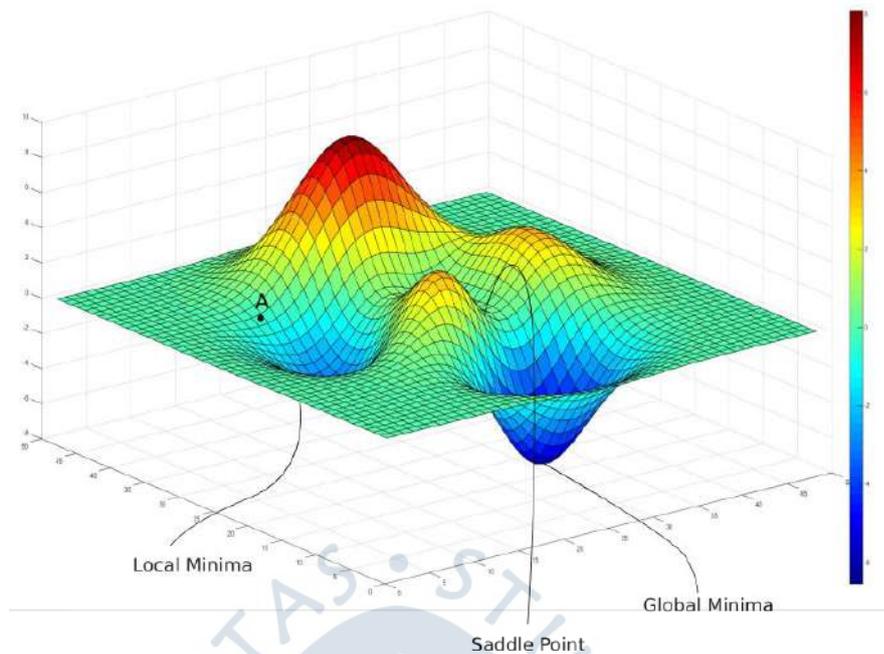


Figura 29. Gráfica de una función de pérdida real.

Fuente: Kathuria (2018)

2.7.2. Aplicación de Redes Neuronales Convolucionales (CNN)

Según lo explicado anteriormente sobre el proceso matemático que trabaja una red neuronal convolucional, se puede rescatar su proceso simplificado y útil para el entrenamiento de imágenes. El reconocimiento y clasificación de imágenes basadas en redes neuronales ha mejorado y evolucionado en el campo de la inteligencia artificial, siendo el sector alimenticio beneficiado considerablemente debido a las mejoras en tecnologías que se usan para la producción, transporte y entrega de los productos, permitiendo que se automaticen los recursos que se emplean para tener el mejor producto posible y con la mejorar calidad del mercado (Aguilar & Campoverde, 2020)

Un buen identificador automatizado de frutas debe considerar 3 parámetros importantes para definir su calidad y valor en el mercado, siendo estos el color, la forma y tamaño de la fruta, es por ello por lo que, para lograr esta correcta identificación es necesaria la rapidez y efectividad con la que se haga.

Esta clasificación lleva consigo varios procesos, que parte desde la recolección y toma de imágenes de las frutas, en este caso la uva de mesa, para después iniciar el pre - procesamiento de imágenes, en donde se prioriza filtrar las imágenes, eliminando ruidos y partes de la imagen que puedan interrumpir o dificultar la clasificación y entrenamiento. El procesamiento viene seguido de una segmentación de las imágenes, para así descartar información no relevante y el etiquetado a todas las imágenes, haciendo que haya una mejor

comparación entre el elemento que se quiera entrenar (uva de mesa) y su background (Aguilar & Campoverde, 2020).

El entrenamiento de la red neuronal convolucional es previsto de 2 clasificaciones según el aspecto de la uva de mesa: Uva en buena calidad y uva de mala calidad, estas dos clasificaciones se harán a través de Python con la librería TensorFlow para el aprendizaje de la red, donde se verá si el producto (uva de mesa) es apto por los requisitos de calidad de la empresa y para posteriormente un consumidor.

Posteriormente, la validación de los resultados, la cual mide los tiempos de respuesta de la red y la eficiencia de nuestra red neuronal convolucional entrenada. Al tener estos pasos completos y bien desarrollados nos asegura un alto porcentaje de *accuracy* en la validación y un tiempo adecuado de respuesta.

2.8. Support Vector Machine

Una máquina de soporte vectorial es un algoritmo computacional que tiene como objetivo el etiquetar objetos. Por ejemplo, una SVM (*Support Vector Machine*), puede aprender a reconocer dígitos escritos a mano al compararlos con una base de datos. (Noble, 2006).

Una máquina de soporte vectorial es una entidad matemática, que busca maximizar una función matemática con respecto a un conjunto determinado de datos. Para ello se deben definir 4 conceptos previos: (a) Hiperplano separador, (b) Hiperplano de margen máximo, (c) Margen blando y (d) la función de Kernel.

a) Hiperplano separador:

Una máquina de vectores de soporte (SVM) realiza la clasificación mediante la búsqueda del hiperplano que maximiza el margen entre las dos clases, también denominado hiperplano separador.

b) Hiperplano de margen máximo

El hiperplano de margen máximo es el que separa con la mayor distancia los datos. El objetivo es maximizar el margen bajo las restricciones que todos los puntos de datos deben estar en el lado correcto del hiperplano.

c) Margen Blando

Por supuesto, no queremos que la SVM permita demasiadas clasificaciones erróneas. el margen blando requiere la introducción de los parámetros que controla especificados por el usuario. Aproximadamente, cuántos ejemplos pueden violar el hiperplano de separación y qué tan lejos a través de la línea se les permite ir. El definir este parámetro resulta complicado por el hecho de que todavía queremos un margen mayor con respecto a las muestras. El margen blando muestra una compensación entre las violaciones del hiperplano y el tamaño del margen (Noble, 2006).

d) La función de Kernel

Resulta que la manera más simple de separar las clases es mediante una línea o plano recto. Sin embargo, en la realidad las dimensiones son mayores a lo ideal por lo que los algoritmos de SVM deben tratar con más de 2 variables a predecir o el hecho de que los datos no están completamente separados o el multiclase.

Como las computadoras no pueden procesar este tipo de datos en N dimensiones las funciones de kernel facilitan su trabajo proyectando en un espacio con dimensiones mayores, aumentando así la capacidad de las SVM.

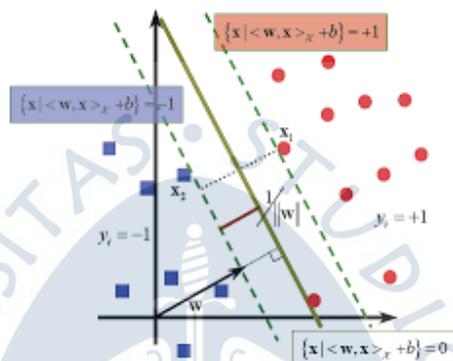


Figura 30. Máquina de soporte vectorial.

Fuente: Noble (2006).

2.8.1. *Missclassification cost*

Como se sabe el aprendizaje en los algoritmos de machine *learning* suele tener una alta relevancia al momento de clasificar una clase. El error de predicción puede aumentar si se trata de data desbalanceada ya que, le dará mayor peso a la clasificación de mayor tamaño de muestras e incluso puede llegar a descartar las de la clase restante como ruido (en casos extremos).

El aprendizaje sensible a los costos es un subcampo de aprendizaje automático que tiene en cuenta los costos de los errores de predicción (y potencialmente otros costos) al entrenar un modelo de aprendizaje automático. Es un campo de estudio que está estrechamente relacionado con el campo del aprendizaje desequilibrado que se refiere a la clasificación en conjuntos de datos con una distribución de clases sesgada.

En primer lugar, debemos comprender que no todos los errores de clasificación se deben tratar de forma similar. Relacionado con el curso, la clasificación es un problema de modelado predictivo el cual implica predecir una etiqueta de clase para una observación.

La mayoría de los algoritmos de aprendizaje automático diseñados para la clasificación suponen que hay un número igual de ejemplos para cada clase observada, pero en la realidad no es así. En nuestro caso al tratarse de una empresa de exportación es uva es claro que no se

van a tener el mismo número de muestras de uvas en mal estado con las muestras de uvas en buen estado.

Con estas definiciones podemos entender el *missclassification cost* como la penalización asociada con una predicción incorrecta y, por tanto, de esta forma se optimiza la precisión del algoritmo minimizando así el error de clasificación.

Por una convención, usualmente se le asigna la etiqueta 0 a la clase mayoritaria y 1 a la clase minoritaria. En otros casos se les asigna una nomenclatura negativa frente a una positiva.

El objetivo del aprendizaje sensible a los costos es minimizar el costo de un modelo en un conjunto de datos de entrenamiento. Para amortiguar los errores de clasificación se suele comprobar la precisión de un algoritmo por medio de la matriz de confusión

Una matriz de confusión es un resumen de las predicciones realizadas por un modelo sobre las muestras destinadas para la clasificación. Es una tabla que resume el número de predicciones realizadas para cada clase, separadas por la clase real a la que pertenece cada ejemplo. Conceptualmente, el costo de etiquetar un ejemplo incorrectamente siempre debe ser mayor que el costo de etiquetarlo correctamente.

Como se mencionó anteriormente este error de clasificación aumenta cuando se trabaja con data desbalanceada y es por ello que existen diversos métodos para contrarrestar este desbalance. Para el presente trabajo se tomó el conocido como *undersampling* (Brownlee, 2020b).

2.8.2. Undersampling

Los métodos de remuestreo están diseñados para cambiar la composición de un conjunto de datos de entrenamiento para una tarea de clasificación desequilibrado. Sin embargo, se ha desarrollado un conjunto de técnicas para disminuir la clase mayoritaria.

El *undersampling* o submuestreo hace referencia a un grupo de técnicas diseñadas para equilibrar la distribución de clases de un dataset de clasificación que tiene una distribución de clases sesgada. Por tanto, se puede decir que el submuestreo consiste en reducir los datos mediante la eliminación de ejemplos pertenecientes a la clase mayoritaria con el objetivo de igualar el número de ejemplos de cada clase (Brownlee, 2020a).

2.8.3. Normalización y Estandarización

La normalización y estandarización son métodos que se utilizan como preprocesamiento de data cuando esta no sigue una determinada distribución. Algunos algoritmos de aprendizaje automático son sensibles al escalado de características, mientras que otros son prácticamente invariables a ella

2.8.3.1. Normalización. La normalización es una técnica en la cual se reescalan los valores muestreados y se desplazan de tal forma que terminan en un rango de 0 y 1. También es conocido como escalado Min-Max.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Donde X_{max} y X_{min} son los valores máximos y mínimos del conjunto de datos respectivamente.

- Cuando el valor de X es el valor mínimo en la columna, el numerador será 0, y por lo tanto X' es 0
- Por otro lado, cuando el valor de X es el valor máximo de la columna, el numerador es igual al denominador y, por lo tanto, el valor de X' es 1
- Si el valor de X está entre el valor mínimo y el valor máximo, el valor de X' está entre 0 y 1

2.8.3.2. Estandarización. Similar a la normalización, esta técnica se centra en el escalado alrededor de la media con una desviación estándar igual a 1.

$$X' = \frac{X - \mu}{\sigma}$$

Donde μ es la media de los valores de la muestra y σ es la desviación estándar de los valores de la muestra

- La normalización es buena cuando se sabe que la distribución de los datos no sigue una distribución gaussiana. Esto puede ser útil en algoritmos que no asumen ninguna distribución de los datos como *K-Nearest Neighbors* y *Neural Networks*.
- La estandarización, por otro lado, puede ser útil en los casos en que los datos siguen una distribución gaussiana. Sin embargo, esto no tiene que ser necesariamente cierto. Además, a diferencia de la normalización, la estandarización no tiene un rango delimitador. Por lo tanto, incluso si tiene valores atípicos en sus datos, no se verán afectados por la estandarización (Bhandari, 2019).

2.8.4. Cross-validation

Cuando se obtienen los resultados de clasificación de un modelo se necesita estar seguros de que los resultados son confiables de que el modelo tiene la mayoría de los patrones de los datos correctos, y no está captando ruido, o en otras palabras un bajo sesgo y varianza.

Cuando se trabaja con una cantidad pequeña de datos es complicado aplicar técnicas de incremento o aumento de data porque se puede caer en el *overfitting* o *underfitting* ya que, al reducir los datos de entrenamiento, se corre el riesgo de perder patrones importantes en el conjunto de datos, lo que a su vez aumenta el error inducido por el sesgo. Por lo tanto, lo que se necesita es un método que proporcione amplios datos para entrenar el modelo y

también deje amplios datos para la validación. Esto es lo que hace la validación cruzada por medio de *k-folds*.

En la validación cruzada por *k-folds* los datos se dividen en subconjuntos *k*. Ahora el método de retención se repite *k* veces, de modo que cada vez, uno de los subconjuntos *k* se utiliza como conjunto de pruebas/conjunto de validación y los otros subconjuntos *k-1* se juntan para formar un conjunto de entrenamiento. La estimación del error se promedia en todos los ensayos *k* para obtener la eficacia total del modelo. Como se puede ver, cada punto de datos llega a estar en un conjunto de validación exactamente una vez, y llega a estar en un conjunto de entrenamiento *k-1* veces. Esto reduce significativamente el sesgo, ya que estamos utilizando la mayoría de los datos para el ajuste, y también reduce significativamente la varianza, ya que la mayoría de los datos también se utilizan en el conjunto de validación. El intercambio de los conjuntos de entrenamiento y pruebas también aumenta la eficacia de este método. Como regla general y evidencia empírica, por lo general se prefiere *K 5* o *10*. Al final se puede tomar cualquier valor que mejore los resultados del modelo sin caer en *overfitting* (Gupta, 2017).

2.8.5. Métricas

Cuando se habla de modelos predictivos, se está hablando de un modelo de regresión (salida continua) o de un modelo de clasificación (salida nominal o binaria). Las métricas de evaluación utilizadas en cada uno de estos modelos son diferentes.

En los problemas de clasificación, utilizamos dos tipos de algoritmos, las salidas de clase y las de probabilidad. El primer tipo es de utilidad cuando se trabaja con algoritmos como SVM y KNN mientras que los algoritmos de probabilidad son cuestión de determinar la probabilidad dado un umbral (Srivastava, 2019).

a. Matriz de Confusión

Una matriz de confusión es una matriz $N \times N$, donde *N* es el número de clases que se predicen.

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	Positive Predictive Value	$a/(a+b)$
	Negative	c	d	Negative Predictive Value	$d/(c+d)$
		Sensitivity	Specificity	Accuracy = $(a+d)/(a+b+c+d)$	
		$a/(a+c)$	$d/(b+d)$		

Figura 31. Matriz de confusión y las métricas que se pueden obtener.

Fuente: Srivastava (2019)

- **Accuracy:** la proporción del número total de predicciones que eran correctas.
- **Valor predictivo positivo o precision:** la proporción de casos positivos que se identificaron correctamente.
- **Valor predictivo negativo:** la proporción de casos negativos que se identificaron correctamente.
- **Sensibilidad o recall:** la proporción de casos positivos reales que se identifican correctamente.
- **Especificidad:** la proporción de casos negativos reales que se identifican correctamente.

b. F-1 score

F1-Score es la media armónica de los valores de *Precision* y *Recall* para un problema de clasificación. La fórmula para F1-Score es la siguiente:

$$F_1 = \left(\frac{\text{recall}^{-1} + \text{precision}^{-1}}{2} \right)^{-1} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

c. Curva AUC – ROC

Esta es de nuevo una de las métricas populares utilizadas en la industria. Si nos fijamos en la matriz de confusión mostrada anteriormente, observamos que, para un modelo probabilístico, obtenemos un valor diferente para cada métrica (Srivastava, 2019).

Por lo tanto, para cada sensibilidad, obtenemos una especificidad diferente. Los dos varían de la siguiente manera:

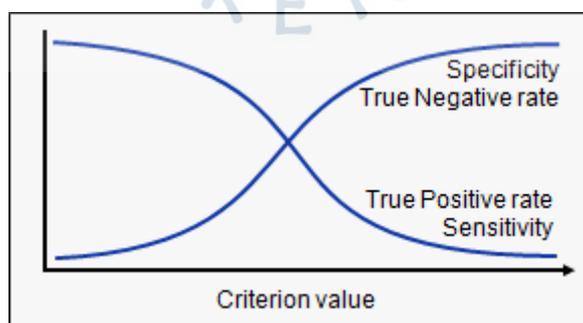


Figura 32. Criterio de variación en la especificidad.

Fuente: Srivastava (2019)

A continuación, se presentan algunas reglas generales:

- .90-1 - Excelente
- .80-.90 – Muy Bueno
- .70-.80 - Bueno
- .60-.70 - Malo
- .50-.60 – Muy Malo





Capítulo 3

Metodología

3.1. Adquisición de imágenes

El proyecto inicia con la toma de data ya que, al trabajar con una empresa local la data será propia obtenida de los racimos de la Sociedad Agrícola Saturno S.A, para ello se ha optado por la utilización de un light box casera ya que esto nos facilitará labores posteriores de segmentación y otros. Esta herramienta nos permitirá realizar las tomas fotográficas en un fondo totalmente blanco.



Figura 33. Fotografía tomada en el light box.

Fuente: Elaboración propia.

Este light box tiene unas medidas de 36cm de alto x 28 cm de base y 20cm de ancho, se ha acondicionado lo mayor posible a una real, donde en vez de la tela utilizada se ha forrado de papel blanco. Los laterales de la caja se iluminan con 2 lámparas, las cuales utilizan focos espirales de 32W E27.

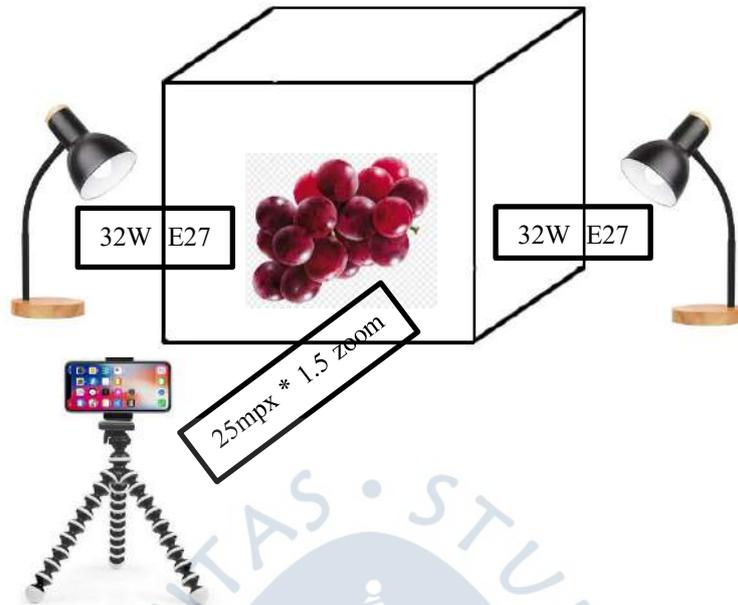


Figura 34. Esquema de adquisición de datos.

Fuente: Elaboración propia

Las imágenes fueron tomadas con la cámara de un celular Samsung A50, de 25mpx colocada a 1.5 de zoom colocada a una distancia de 20 cm del borde exterior del light box.

Para el primer grupo de racimos alcanzados por la empresa se han tomado cerca 137 imágenes de uva Red Globe las cuales fueron tomadas con el dispositivo móvil dentro de la *lightbox* iluminada por las lámparas.



Figura 35. Zona de recolección de data.

Fuente: Elaboración propia.

3.2. Pre - procesamiento y procesamiento

Si bien la toma de datos de uvas de mesa para el entrenamiento de la red neuronal tiene que cumplir ciertos parámetros de calidad, el pre – procesamiento de imágenes viene a ser la reducción de posibles defectos que tengan las imágenes, como exceso o déficit en el contraste, y el uso de filtros que puedan reducir el ruido.

Este pre - procesamiento es un proceso no muy especializado, en donde principalmente podemos corregir ruidos provocados por una posible poca iluminación al momento de tomar las muestras, estos ruidos nos pueden arrojar resultados en píxeles que no reflejen la verdadera intensidad de la imagen, es por lo que, se puede codificar en Matlab para poder aplicar estos filtros y eliminar posibles ruidos.

Para el tema de aumento y reducción de contraste, usaremos una herramienta del *toolbox*, *Color Tresholder* la cual podremos ir variando el contraste, diferenciando así las partes más claras y oscuras en una imagen, con el fin que la red pueda reconocer de forma más precisa entre el *background* y la forma de la uva para la clasificación de la uva.

Con el pre – procesamiento podemos proceder a lo que viene a ser el procesamiento en sí de las imágenes, es decir una vez obtenidas las imágenes por parte del pre-procesamiento con un tamaño cualquiera, se procede a preparar la imagen a través del *toolbox* de Matlab, como *Image Segmenter* y *Color Tresholder*, la cual es una herramienta que permite segmentar imágenes, convirtiéndolas en imágenes binarias, asignándole a todos los píxeles valores de 0 a 1, basado en su color en múltiples espacios de colores, con el fin de poder resaltar la parte más importante de la imagen, la cual nos ayudaría de acuerdo a las características que queremos medir, como son: el color y la forma de la uva de mesa.

Otra herramienta útil para el procesamiento es la *Image batch Processor*, ya que permitiría que esto se haga a todas las imágenes que se ingresen y no una por una. Así mismo, para la identificación de bordes en la clasificación de la uva de mesa de acuerdo con su forma, se puede usar el algoritmo de Canny, el cual a través de código en Matlab trabajaremos con el vector gradiente de cada píxel para así poder detectar de una forma más precisa los bordes en las imágenes y así poder tener un mejor *accuracy* al momento de validarla.



Figura 36. Imagen de uva de mesa.

Fuente: Elaboración propia.



Figura 37. Imagen segmentada con Image Segmenter.

Fuente: Elaboración propia.

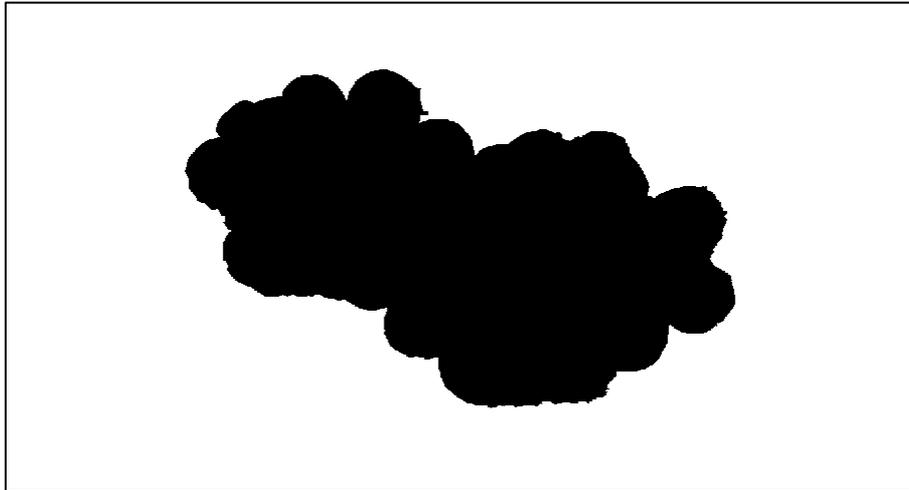


Figura 38. Imagen segmentada y binarizada con Image Segmenter.

Fuente: Elaboración propia

Una vez procesadas las imágenes, se pueden cargar a Matlab o que el código llame a las imágenes para empezar a entrenarlas, estas imágenes para el entrenamiento serán en un 70% de las imágenes tomadas en total, lo cual corresponde a 324 imágenes de uvas de mesa.

3.3. Procesamiento de imágenes para la medición de la baya de uva.

Se debe mencionar que los parámetros a estimar en el presente trabajo serán: estado de la fruta (apto o no apto), de acuerdo con la pudrición presente en el racimo o no y el calibre del racimo. Adicional a ello, se presenta una propuesta para la estimación del peso. Sin embargo, no es muy viable la estimación de peso de uva ya que, al tratarse de exportaciones con mercados internacionales vale la pena el invertir un poco más de tiempo pesando cada racimo y dar un peso exacto que realizar una estimación no tan precisa con el fin de ahorrarse un poco de tiempo.

Para el caso de la estimación del calibre y estimación del peso se hará uso de la Transformada de *Watershed* para poder diferenciar entre las bayas presentes en el racimo junto a ello se utiliza la Transformada de Hough. Seguido de estos pasos ya se calcula un *bounding box* apropiado para calcular el área del racimo, bayas y así poder estimar el calibre y el peso.

Para el caso del estado de la fruta, es decir, apto o no apto, se trabaja con algoritmos de *Machine Learning* y *Deep Learning*, con máquinas de soporte vectorial y con redes neuronales convolucionales respectivamente con la finalidad de comparar resultados y ver que algoritmo se adecua mejor a nuestro caso de estudio.

3.3.1. Transformada de Watershed

Como su nombre lo dice trata de segmentar imágenes encontrando cuencas, para entender mejor esto se piensa en un plano $f(x, y)$ donde se representa la altura de una imagen previamente transformada a escala de grises, en presencia de lluvias por ejemplo el agua se almacenaría en una de las etiquetas denominadas cuencas hidrográficas. La figura de la imagen izquierda viene a ser una imagen en escala de grises y la figura de la derecha vendría a ser su representación 3D. En resumen, la transformada de *Watershed* es una alternativa para segmentar imágenes encontrando cuencas hidrográficas en una imagen en escala de grises. (Cabanilles Mengual & Martínez, 2014).

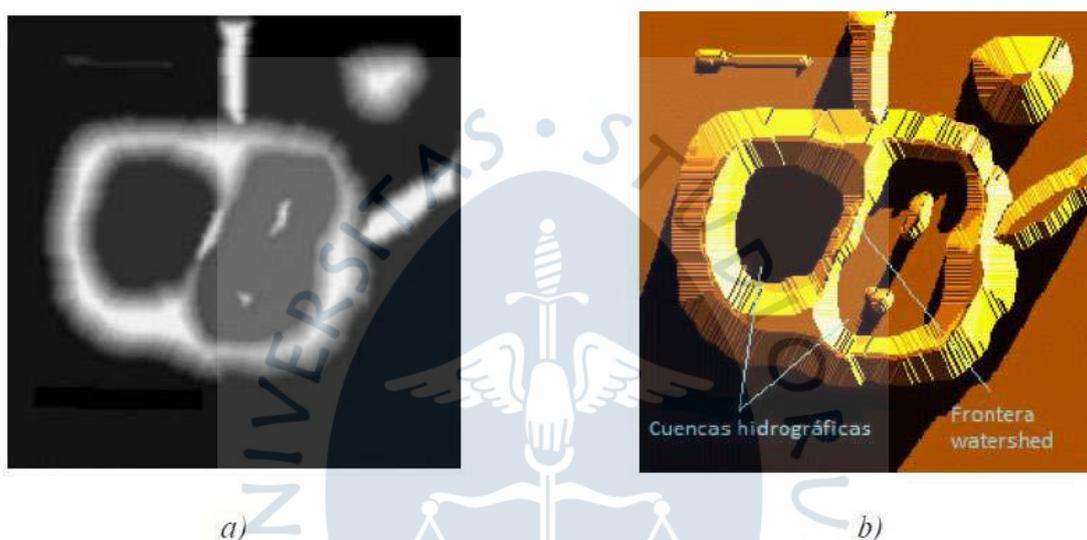


Figura 39. a) Imagen en escala de grises y b) representación 3-D de la imagen en escala de grises.

Fuente: Cabanilles Mengual & Martínez (2014).

(Acea Mena, 2012). Menciona que la implementación de la transformada de *Watershed* se basan en dos algoritmos: Los de cascada (*waterfall*) y los basados en la inundación topográfica. El primero de ellos consiste en suponer que el agua cae desde cada posición del píxel de la imagen, el agua recorrerá la distancia mínima hasta llegar a un mínimo regional de la topografía, así se queda definido los mínimos regionales.

Al trabajar con la transformada de *Watershed* se busca que las cuencas sean los objetos que se deseen segmentar, como en nuestro caso las bayas de las uvas, pero generalmente se presentan ruidos e irregularidades del gradiente lo que hace que existan demasiadas cuentas lo que conlleva a una sobre segmentación.

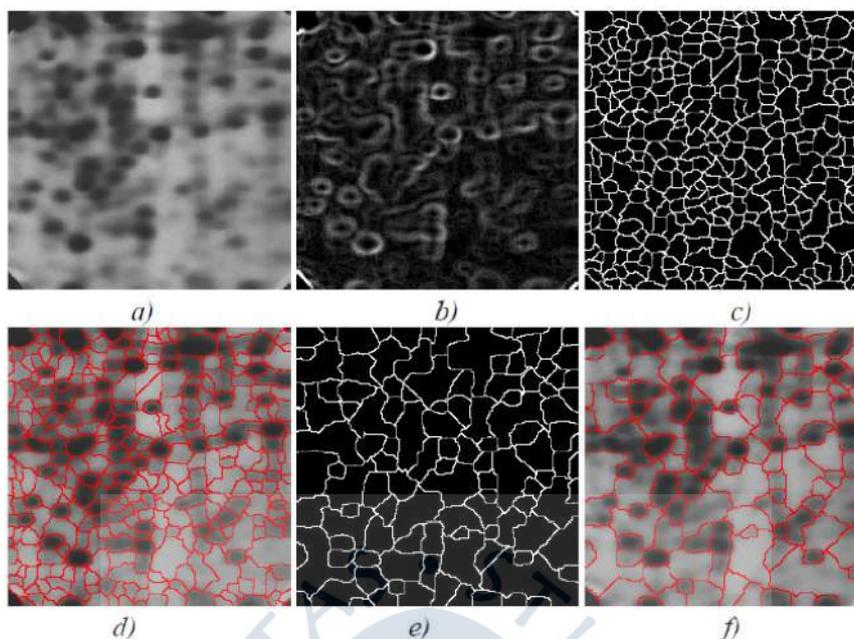


Figura 40. a) Imagen a segmentar, b) Imagen gradiente, c) fronteras watershed (presencia de sobre segmentación), d) Imagen sobre segmentada, e) fronteras watershed después de realizar un filtrado al gradiente y f) imagen segmentada.

Fuente: Cabanilles Mengual & Martínez (2014).

3.3.2. Segmentación de watershed con marcadores.

Este método tiene por objetivo solucionar el problema de la sobre segmentación, este método trata de reducir la cantidad de cuencas o regiones al incorporar una etapa de preprocesamiento con la finalidad de añadir información para una buena segmentación.

Este método introduce unos denominados “marcadores”, uno dentro de cada región con la finalidad de inundar las zonas *watershed* desde los marcadores, también encontrar marcadores que me limiten la inundación asegurando que estén fuera del objeto. (Cabanilles Mengual & Martínez, 2014).

Sea $f(x): A \subset \mathbb{R} \rightarrow \mathbb{R}$, sea $x_0 \in A$ y sea $P = (x_0, f(x_0))$ un punto cualquiera perteneciente a la función. Se dice que P es un mínimo local de f si existe un entorno reducido de centro x_0 , en símbolos $E'((x_0))$ donde para todo elemento x de $E'((x_0))$ se llega a cumplir que $f(x) \geq f(x_0)$.

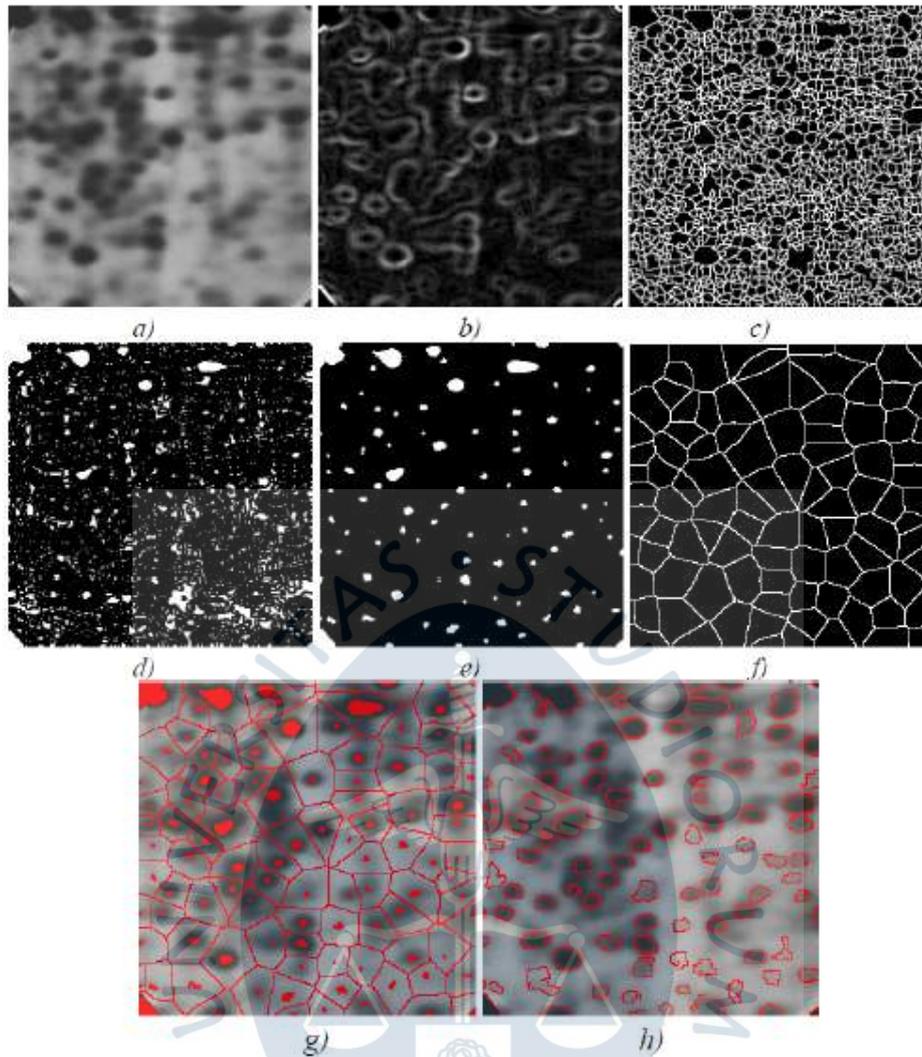


Figura 41. a) Imagen a segmentar, b) gradiente, c) sobre segmentación, d) mínimos regionales del gradiente, e) marcadores internos, f) marcadores externos, g) marcadores internos y externos sobre la imagen original y h) resultado de la segmentación.

Fuente: Cabanilles Mengual & Martínez (2014)

3.3.3. Segmentación de watershed estocástico.

La finalidad de este método es introducir nuevos marcadores aleatorios para crear un nuevo modelo que pueda lidiar contra variaciones en las condiciones de la segmentación, esto con la finalidad de construir una función de densidad de probabilidad de todos los contornos para que se puedan establecer las regiones más predominantes o significativas. (Cabanilles Mengual & Martínez, 2014).

La función densidad $pdf(x)$, *probability density function*, se origina a partir de M realizaciones, el cual se puede estimar mediante las ventanas de Parzen:

Sea $x_1, x_2, \dots, x_M \in \mathbb{R}^n$

$$pdf(x) = \frac{1}{N_h} \sum_{i=1}^N K\left(\frac{x - x_i}{h}\right)$$

$K(x)$ se denomina núcleo, h un parámetro de suavizado, ancho de banda, y N es la cantidad de marcadores aleatorios. Usualmente, $K(x)$ es tomada como una función Gaussiana con media cero y varianza σ^2 .

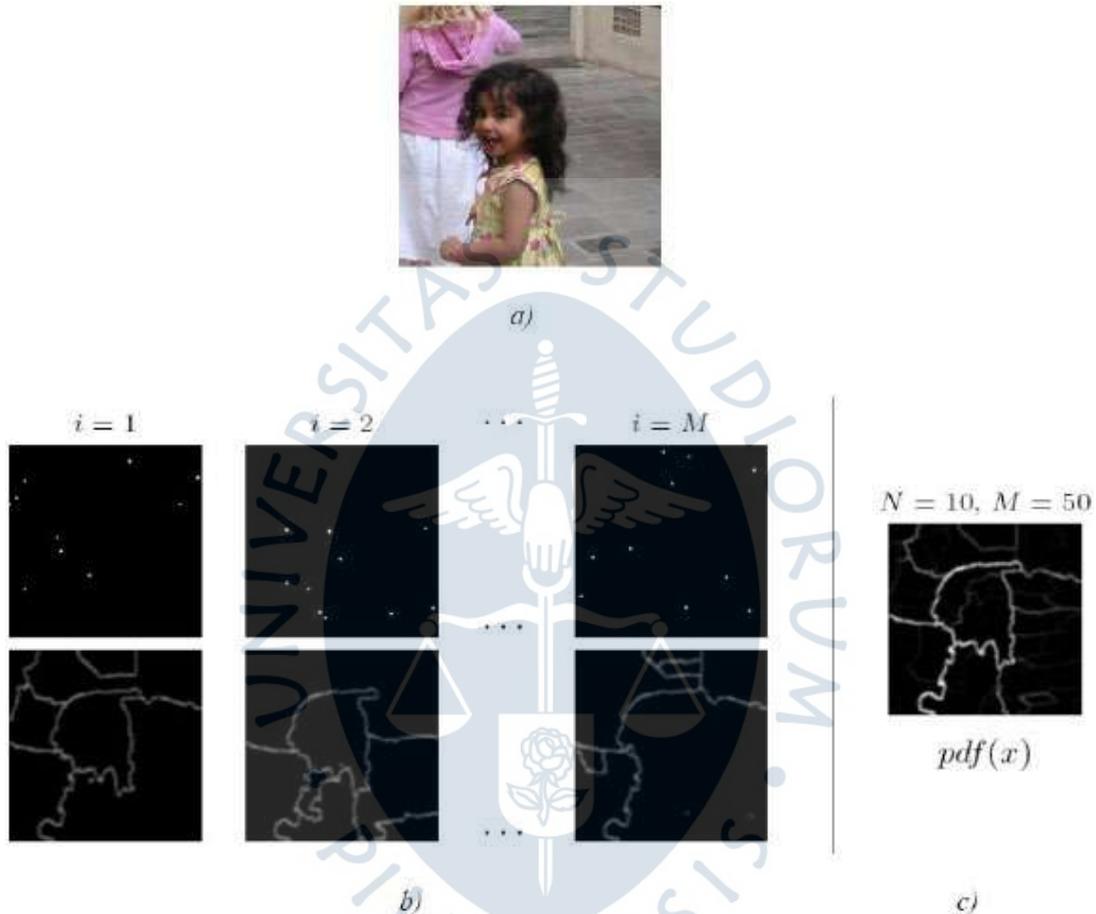


Figura 42. a) Imagen original a segmentar, b) M realizaciones de N marcadores aleatorios, $mrk_i(x)$ y sus correspondientes contornos watershed, c) Función de densidad probabilidad de los contornos calculados para $N=10$ y $M=50$.

Fuente: Cabanilles Mengual & Martínez (2014).

En este artículo (Rabatel & Guizard, 2007), proponen una metodología de procesamiento de imágenes para estimar el tamaño de la baya de las uvas mediante la adquisición de imágenes en el campo.

Para la adquisición de imágenes usan la herramienta REV que ha sido diseñada para adquirir imágenes de uvas aisladas para que el procesamiento posterior sea el adecuado, con un fondo en forma de “v” de color naranja

Primero inician con la detección de bordes del racimo de uva mediante la segmentación por *watershed*, en la que consideran un desafío recuperar el perfil exacto de tantas bayas como sea posible, en las que se incluye las que están parcialmente ocluidas.

Posterior a esto se aplican varios pasos a esta imagen para obtener una lista de arcos:

- Se analizan todos los bordes de la cuenca, buscando intersecciones de bordes y se almacenan como una estructura de árbol de segmentos de contornos individuales con interconexiones recursivas (Imagen 3.b)
- Se eliminan los segmentos terminales en la estructura de árbol con una longitud por debajo de un umbral dado (para su caso se usó 10 píxeles). Se vuelve al paso anterior para conectar los segmentos restantes en segmentos más grandes. Da como resultado un contorno más limpio (Imagen 3.c)
- Finalmente, cada segmento de contorno en la estructura de árbol se descompone en arcos elementales de curvatura constante, que se almacenan en una lista de arcos.

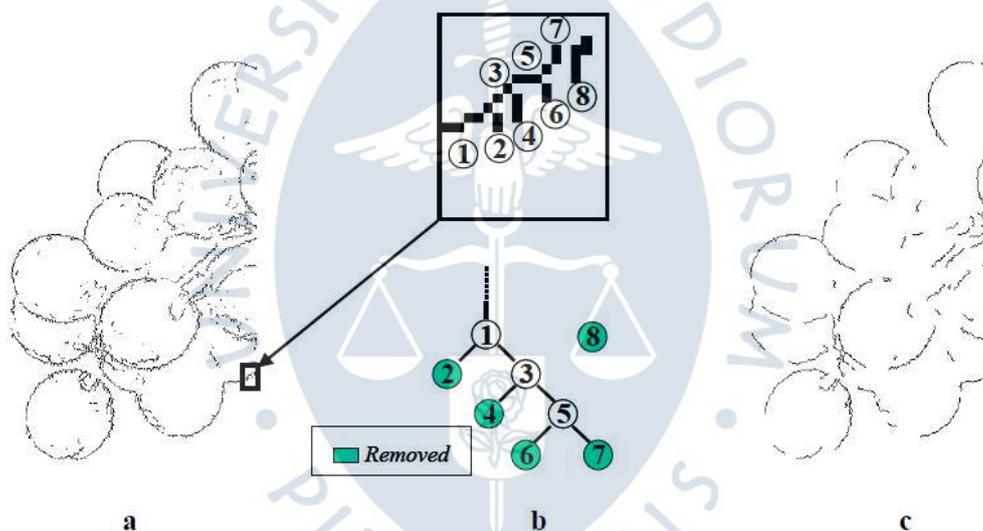


Figura 43. a) Resultado inicial de la cuenca, b) Estructura del árbol y c) Resultado del proceso de limpieza.

Fuente: Rabatel & Guizard (2007).

Después de este procesamiento de la imagen se procede a ajustar la lista de arcos a elipses, pero para ajustar las curvas se presentan algunos problemas, por ejemplo, aunque se conociera el centro de una baya es seleccionar adecuadamente los bordes.

Implementan el siguiente procedimiento, aplicar directamente el modelo de elipse a un conjunto de arcos de contorno compatibles, utilizado el algoritmo de mínimos cuadrados. Pero se tiene el problema de definir el conjunto relevante de arcos de contorno en cada caso. Por lo que definen el procedimiento combinatorio como sigue:

- Para un centro de elipse dado, se seleccionan todos los segmentos de contorno en la lista de arcos que son compatibles con él (en términos de distancia, curvatura, orientación).
- Cada posible subconjunto de segmentos de contorno entre los contornos compatibles encontrados en el paso anterior se construye sucesivamente. Por ejemplo, con 3 segmentos S1, S2, S3, se construirá 7 subconjuntos: {S1}, {S2}, {S3}, {S1 S2}, {S1 S3}, {S2 S3}, {S1 S2 S3}.
- El algoritmo de ajuste de FitzGibbon se aplica a cada subconjunto, y solo se mantiene el mejor ajuste, de acuerdo con un criterio de calidad.

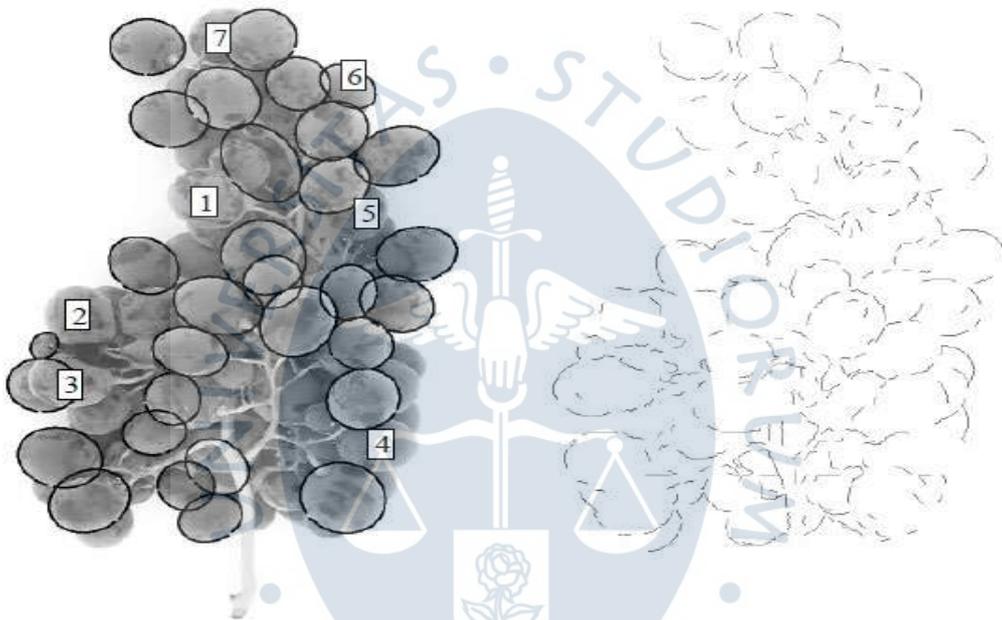


Figura 44. Salida del algoritmo (derecha) y Detección de borde inicial (izquierda).

Fuente: Rabatel & Guizard (2007).

3.3.4. Transformada de Hough

Es una técnica que puede localizar formas como líneas, círculos o también elipses en una imagen, se basa en la búsqueda de patrones. Este algoritmo es usado muy a menudo para el procesamiento de imágenes con la finalidad de buscar formas.

La transformada de Hough para la detección de círculos se usa la ecuación de la forma:

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

Para entender mejor este concepto tenemos la siguiente imagen, en la primera imagen cada punto ubicado en el borde define un conjunto de círculos en el espacio acumulador, pero cada uno de estos puntos tienen un conjunto infinito de posibles valores de radio, el espacio acumulador se agranda solo para puntos que se encuentren dentro del rango deseado, ya que otros puntos no se podrán aproximar a esa circunferencia. En la segunda figura se puede

apreciar que todas las circunferencias se intersecan en un punto y este será el centro del radio elegido.

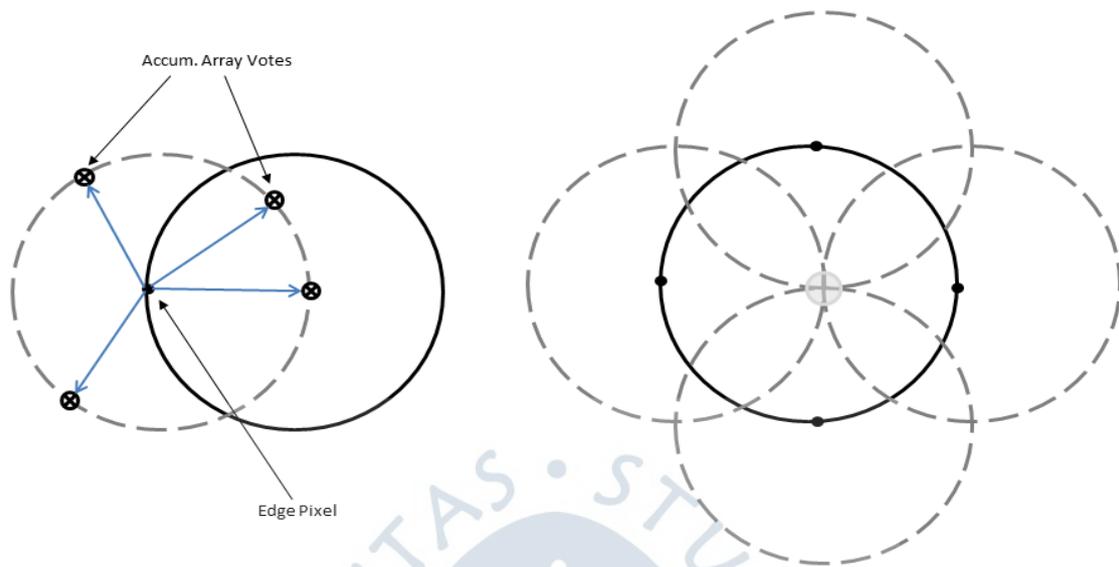


Figura 45. Puntos de borde de un círculo y espacio acumulador.

Fuente: <https://es.mathworks.com/help/images/ref/imfindcircles.html>

3.4. Extracción de parámetros para estimación del peso del racimo.

3.4.1. Bounding box

El *bounding box* se puede definir como el cuadrado mínimo con el que se puede encerrar un objeto. Este rectángulo se puede definir mediante 2 puntos:

$$BB(O) = (x_{min}, x_{max}, y_{min}, y_{max})$$



Figura 46. Bounding box.

Fuente: Elaboración propia.

3.4.2. Cálculo de área.

El área del racimo corresponderá al número de píxeles que corresponde al racimo después de ser segmentada y binarizada previamente.

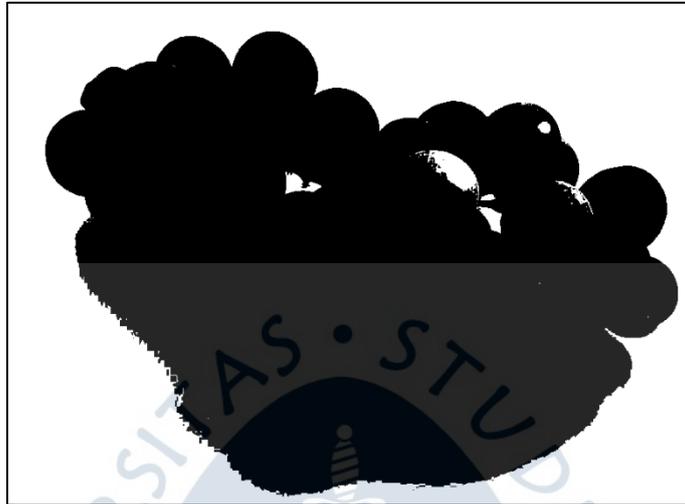


Figura 47. Racimo de uva binarizada.

Fuente: Elaboración propia.

Se puede observar que la segmentación por umbralización no permite segmentar adecuadamente el racimo de uva, por lo que posteriormente se probará con nuevos métodos de segmentación.

Los datos de entrada para estimar el peso del racimo serán el alto, ancho y el área que ocupa el racimo dentro del *bounding box*, entonces se puede expresar de la siguiente manera:

$x_1 = \text{Largo del bounding box}$

$x_2 = \text{Ancho del bounding box}$

$x_3 = \text{Área del racimo binarizada}$

$y = \text{Estimación del peso}$

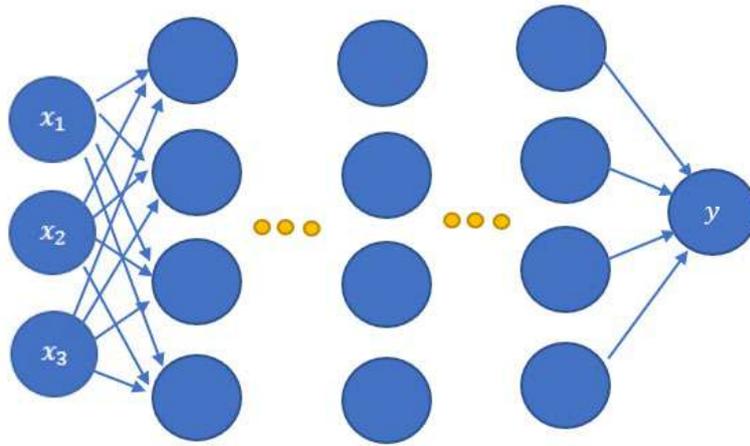


Figura 48. Esquema de la red neuronal para la predicción del peso del racimo.

Fuente: Elaboración propia.

3.5. Estimación del estado de la uva con SVM.

Para la aplicación de una máquina de soporte vectorial a nuestro conjunto de datos primero se debe tener estos datos en forma numérica, al tratarse de imágenes de racimos de uva, debemos hacer una extracción de características en cada imagen para hacer uso de una SVM.

La extracción se realizará en Matlab, y una vez almacenados en formato .csv, se hará uso del algoritmo en cuestión.

Las características por extraer serán tres: color, forma y textura cada una de ellas será explicada a más detalle en la siguiente sección

```

%%Color
R = img(:, :, 1);
G = img(:, :, 2);
B = img(:, :, 3);

Mr = mean2(R); %Mean Red
Mg = mean2(G); %M1
Mb = mean2(B);

Rnormalized = R./(R+G+B);
Gnormalized = G./(R+G+B);
Bnormalized = B./(R+G+B);

NMr = mean2(Rnormalized); %Mean of Normalised Red
NMg = mean2(Gnormalized);
NMb = mean2(Bnormalized);

MCHr = mean2(imhist(R));
MCHg = mean2(imhist(G));
MCHb = mean2(imhist(B));

VCHr = var(imhist(R));
VCHg = var(imhist(G));
VCHb = var(imhist(B));

Stdr = std2(R);
Stdg = std2(G); %M2
Stdb = std2(B);

%Indice que me indica que el verde es mas dominante que el rojo NDI_rg > 0
NDI_rg = abs(R-G)./(R+G);
NDIrg = mean2(NDI_rg);
%Indice que me indica que el azul es mas dominante que el rojo NDI_rb > 0
NDI_rb = abs(R-B)./(R+B);
NDIrb = mean2(NDI_rb);
%Indice que me indica que el azul es mas dominante que el verde NDI_gb > 0
NDI_gb = abs(G-B)./(G+B);
NDIgb = mean2(NDI_gb);

Kurtosis = kurtosis(double(img(:))); %M3
Skewness = skewness(double(img(:))); %M4

```

Figura 49. Comand Window de las características de color en Matlab.

Fuente: Elaboración propia.

```

%% Forma
R = img(:, :, 1);
G = img(:, :, 2);
B = img(:, :, 3);

Rnormalized = R./(R+G+B);
Gnormalized = G./(R+G+B);
Bnormalized = B./(R+G+B);

R_bw = imcomplement(im2bw(R, graythresh(img)));
G_bw = imcomplement(im2bw(G, graythresh(img)));
B_bw = imcomplement(im2bw(B, graythresh(img)));

NR_bw = imcomplement(im2bw(Rnormalized, graythresh(img)));
NG_bw = imcomplement(im2bw(Gnormalized, graythresh(img)));
NB_bw = imcomplement(im2bw(Bnormalized, graythresh(img)));

Ar = bwarea(R_bw);
Ag = bwarea(G_bw);
Ab = bwarea(B_bw);

NAr = bwarea(NR_bw);
NAg = bwarea(NG_bw);
NAb = bwarea(NB_bw);

```

Figura 50. Comand Window de las características de forma en Matlab.

Fuente: Elaboración propia.

```

%% Textura
img_gray = rgb2gray(img);
glcm = graycomatrix(img_gray);
stats = graycoprops(glcm, 'Contrast Homogeneity Correlation Energy');
Contrast = stats.Contrast;
Homogeneity = stats.Homogeneity;
Correlation = stats.Correlation;
Energy = stats.Energy;

gaborArray = gaborFilterBank(5, 8, 39, 39);
GaborFeatures = gaborFeatures(img_gray, gaborArray, 4, 4);
Mean_GF = mean2(GaborFeatures);
Variance_GF = var(GaborFeatures);

```

Figura 51. Comand Window de las características de textura en Matlab

Fuente: Elaboración propia.

Una vez extraídas las características y con ayuda de la app de Matlab “*Classification Learner*”, se procede a la creación del modelo y la búsqueda de los mejores parámetros para obtener los mejores resultados.

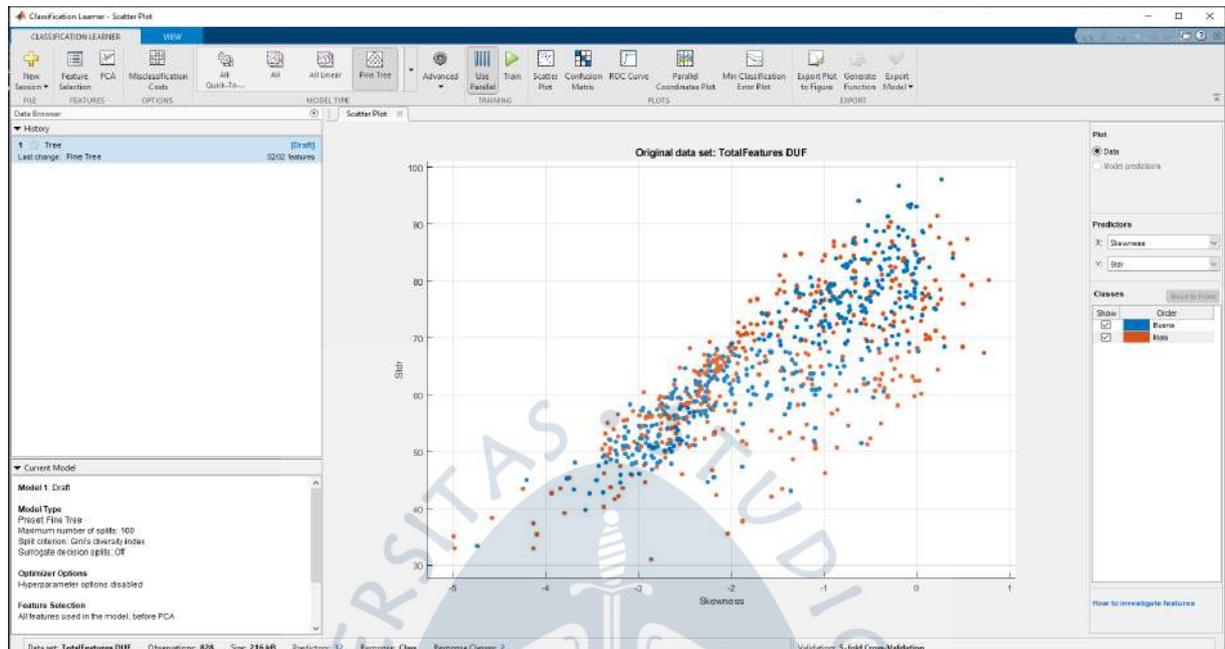


Figura 52. Ventana del Classification Learner previos al entrenamiento del modelo.

Fuente: Elaboración propia.

Una vez dentro del programa, lo único que se debe hacer es ir variando parámetros, los cuales fueron explicados en el capítulo 2, y lograr obtener el modelo que mejor generalice nuestra clasificación de uvas buenas o malas.

3.6. Estimación del estado de la uva con CNN.

Para la estimación de la uva, aplicando redes neuronales convolucionales, se utilizará la técnica de aprendizaje por transferencia o *transfer learning*. Esta técnica consiste en entrenar la red neuronal con un aprendizaje previo, obtenido por el dataset de ImageNet. Lo que se hará es la variación de los hiperparámetros para obtener los mejores resultados en la calidad de la uva.

Las imágenes ingresadas a la red son las procesadas anteriormente. Estas imágenes serán clave para poder obtener los mejores resultados, ya sea para el entrenamiento y validación de la red neuronal.

Los hiperparámetros a variar son los siguientes:

- Arquitectura por utilizar.
- Número de capas congeladas.
- Tamaño de imagen.
- *Batch size*.

- Optimizador.
- Función de *loss* o pérdida.
- Número de épocas.
- Últimas capas agregadas (*Dropout, flatten, dense*).

3.6.1. VGG-16 Transfer Learning

Los hiperparámetros utilizados ha sido en base a la investigación de (Krinitskiy et al., 2018), donde utiliza la arquitectura VGG-16 con una función de pérdida entropía cruzada, un optimizador Adams y la métrica *accuracy* para observar el rendimiento de la red.

Los valores iniciales han sido extraídos de (Keras, n.d.) para observar el número de clases para cada arquitectura, En el caso de VGG-16, se trabaja con una entrada de 224*224. Para el número de épocas, se ha utilizado un valor de 50, siendo el valor de *batch size* de 32.

Lo siguiente será cargar la arquitectura de red neuronal para poder trabajar con todas las capas que caracterizan al modelo. En el caso de VGG-16, se cargará el modelo con los parámetros entrenados del conjunto de datos de 'imagenet'.

Layer (type)	Output Shape	Param #
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
output (Dense)	(None, 1)	4097

Total params: 134,264,641
 Trainable params: 4,097
 Non-trainable params: 134,260,544

Figura 53. Capas y parámetros de la arquitectura VGG-16.

Fuente: Elaboración propia.

Al aplicar *transfer learning*, se congelarán todas las capas, excepto la última capa donde se trabajará con una capa densa en la cual se trabajarán 4097 parámetros.

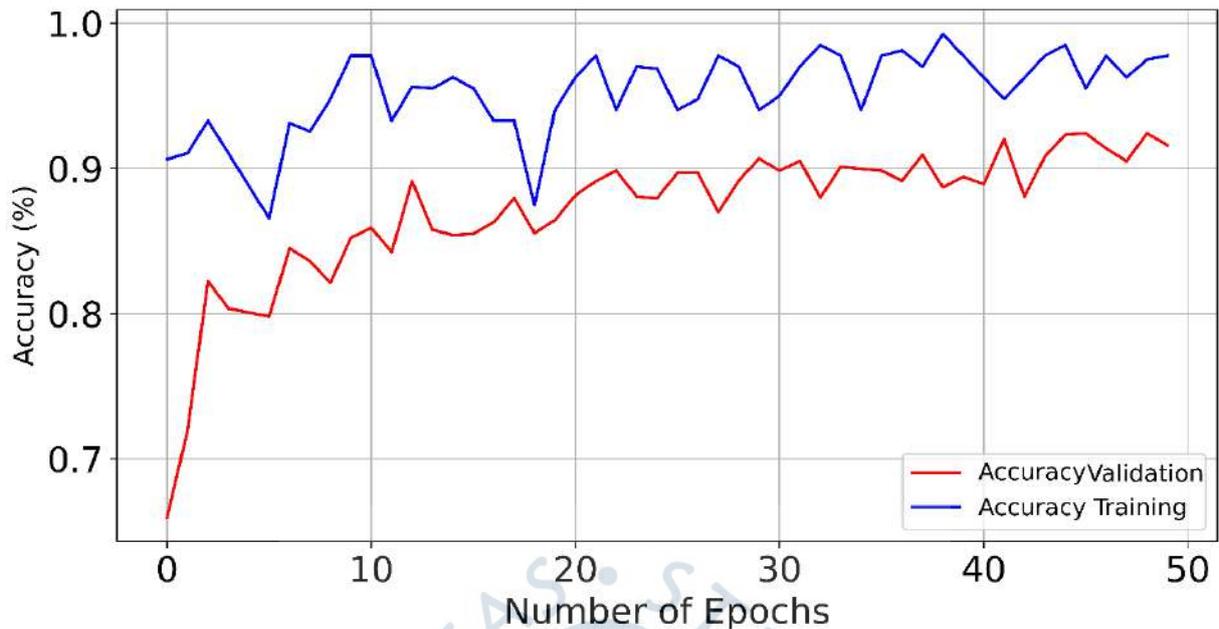


Figura 54. Gráfica de accuracy en la CNN aplicando transfer learning para el modelo VGG-16.

Fuente: Elaboración propia.

El análisis de la Figura 54 está determinado mayormente por la curva de validación, ya que los parámetros del modelo ya han aprendido y corregido en el entrenamiento, en este caso, se observa desde el primer punto un valor aproximado de 0.9 en *accuracy*. El valor promedio para la validación es de 0.91, obteniendo un resultado mayor. Para observar si el modelo no se ha sobre ajustado a los datos, se observará el comportamiento de la gráfica *loss*.

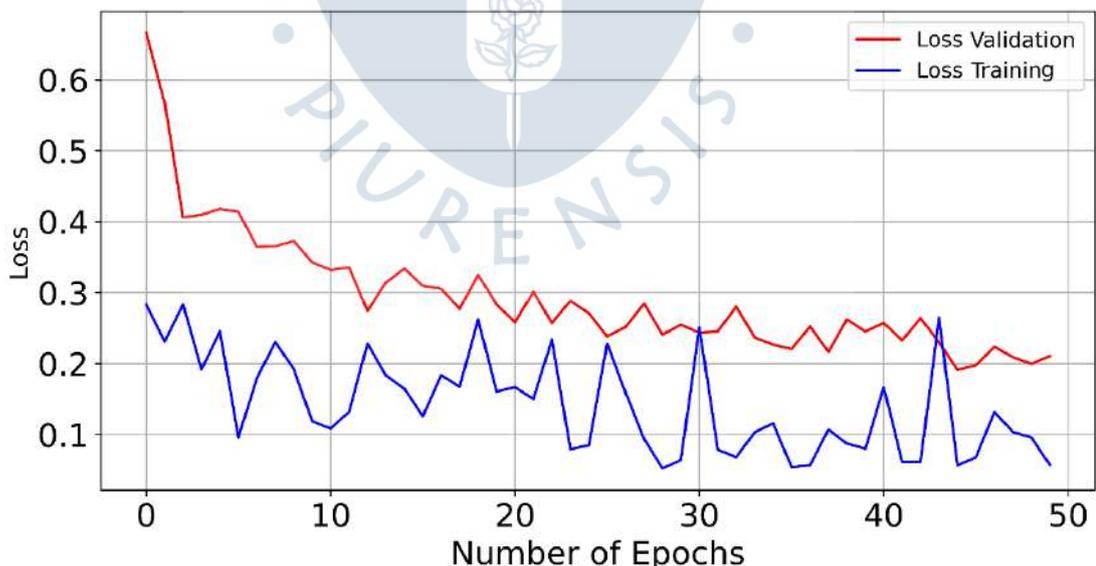


Figura 55. Gráfica de loss en la CNN aplicando transfer learning para el modelo VGG16.

Fuente: Elaboración propia.

El análisis de la Figura 55 tiene valores de pérdida bastante aceptables, debido a que se encuentran entre 0 y 0.3. Generalmente se encuentran pérdidas mayores en el entrenamiento que en la validación ya que se utiliza *Dropout*.

Mientras menor sea la pérdida, mejor será el modelo, teniendo en cuenta que el modelo no se haya sobre ajustado a los datos de entrenamiento. A comparación de la curva *accuracy*, la cual tiene valores en porcentaje, la curva de *loss* es una suma de los errores cometidos para cada ejemplo, tanto en el conjunto de entrenamiento como en el de validación. Para este caso, la función de activación ha sido la entropía cruzada.

3.6.2. MobileNet Transfer Learning

La función de pérdida será *crossentropy* para la clasificación binaria, teniendo como optimizador adadelta y la métrica será el *accuracy*.

El optimizador ADADELTA se eligió en base a la investigación de (Aral et al., 2019), donde utiliza tanto el optimizador ADADELTA como ADAMS aplicado en la arquitectura MobileNet.

La función de pérdida será *crossentropy* en este caso *binary crossentropy* debido a que se trata de una clasificación binaria. Esta función de pérdida ha sido implementada en la investigación de (Gavai et al., 2018), donde aplica la función de pérdida junto con la arquitectura MobileNet utilizando Tensorflow.

Al aplicar *transfer learning*, se congelarán todas las capas de la red, excepto la última capa, donde se trabajará con un valor de *Dropout* de 0.3. La capa densa de salida tendrá como función de activación "Sigmoide" por ser una de las más utilizadas para el caso de clasificación binaria, en este caso, "high quality" y "low quality".

Como se observa, el valor total de los parámetros es de 3 288 864, el cual aumentará al aplicarle el *Dropout*. Cabe resaltar que la red ha sido cargada teniendo como última capa una convolucional con función de activación ReLu, teniendo un valor total de 0 parámetros. Lo que se realizará es la congelación de todas las capas, con todo el entrenamiento obtenido con el conjunto de datos de 'ImageNet'.

conv_pw_11 (Conv2D)	(None, 14, 14, 512)	262144
conv_pw_11_bn (BatchNormaliz	(None, 14, 14, 512)	2048
conv_pw_11_relu (ReLU)	(None, 14, 14, 512)	0
conv_pad_12 (ZeroPadding2D)	(None, 15, 15, 512)	0
conv_dw_12 (DepthwiseConv2D)	(None, 7, 7, 512)	4608
conv_dw_12_bn (BatchNormaliz	(None, 7, 7, 512)	2048
conv_dw_12_relu (ReLU)	(None, 7, 7, 512)	0
conv_pw_12 (Conv2D)	(None, 7, 7, 1024)	524288
conv_pw_12_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_pw_12_relu (ReLU)	(None, 7, 7, 1024)	0
conv_dw_13 (DepthwiseConv2D)	(None, 7, 7, 1024)	9216
conv_dw_13_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_dw_13_relu (ReLU)	(None, 7, 7, 1024)	0
conv_pw_13 (Conv2D)	(None, 7, 7, 1024)	1048576
conv_pw_13_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_pw_13_relu (ReLU)	(None, 7, 7, 1024)	0
=====		
Total params:	3,228,864	
Trainable params:	3,206,976	
Non-trainable params:	21,888	

Figura 56. Capas y parámetros de la arquitectura MobileNet.

Fuente: Elaboración propia.

conv_dw_13_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_dw_13_relu (ReLU)	(None, 7, 7, 1024)	0
conv_pw_13 (Conv2D)	(None, 7, 7, 1024)	1048576
conv_pw_13_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_pw_13_relu (ReLU)	(None, 7, 7, 1024)	0
flatten (Flatten)	(None, 50176)	0
fc1 (Dense)	(None, 128)	6422656
dropout_3 (Dropout)	(None, 128)	0
fc2 (Dense)	(None, 128)	16512
dropout_4 (Dropout)	(None, 128)	0
output (Dense)	(None, 1)	129
=====		
Total params:	9,668,161	
Trainable params:	9,646,273	
Non-trainable params:	21,888	

Figura 57. Capas y parámetros de la arquitectura MobileNet aplicando Dropout.

Fuente: Elaboración propia.

Al aplicar el *Dropout*, el valor de los parámetros totales se ha triplicado, teniendo en cuenta que se trabajó con un valor de 0.3. Como se explicó en el apartado anterior, se ha agregado una capa de salida, aplicando *Dropout* y teniendo una capa densa de salida, con un total de 129 parámetros. Al obtener mayores parámetros en la red, su eficiencia de entrenamiento mejorará, aunque tendrá mayor tiempo para terminar el entrenamiento de la red.

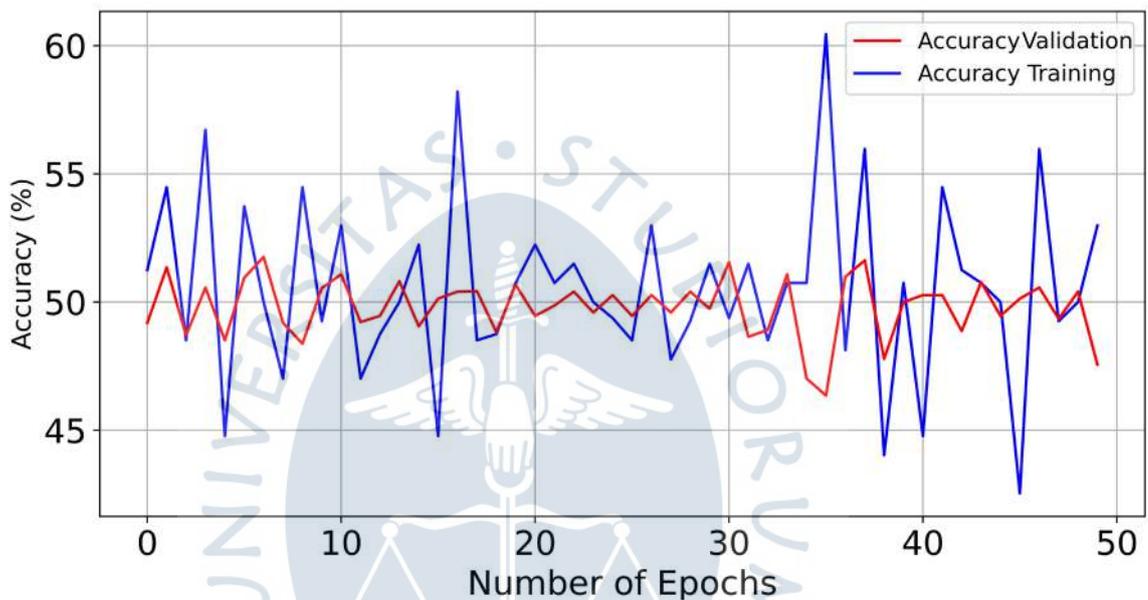


Figura 58. Gráfica de accuracy en la CNN aplicando transfer learning para el modelo MobileNet.

Fuente: Elaboración propia.

El análisis de la Figura 58. está determinado mayormente por la curva de validación, ya que los parámetros del modelo ya han aprendido y corregido en el entrenamiento, en este caso, se observa desde el primer punto un valor aproximado de 0.5 en *accuracy*. El valor promedio para la validación es de 0.53, obteniendo un resultado mayor. Estos valores nos indican que la red no ha sido entrenada con los mejores hiper parámetros o son insuficientes las capas ocultas para poder extraer las características de cada clase y poder clasificar de manera óptima.

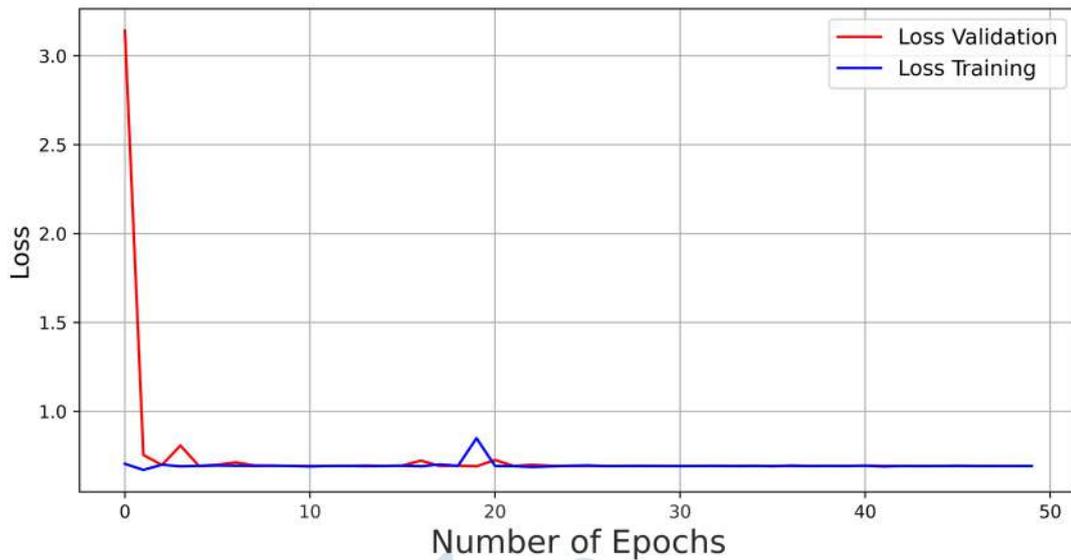


Figura 59. Gráfica de loss en la CNN aplicando transfer learning para el modelo MobileNet.

Fuente: Elaboración propia.

En la Figura 59 se observa el valor de *loss*, donde se verifica un valor constante en el entrenamiento y la validación. Esto quiere decir que la red tiene el problema del *underfitting*, comparando las dos gráficas de *accuracy* y *loss*.

3.6.3. ResNet50 Transfer Learning

Para esta tercera prueba, se congelaron todos los pesos de la red pre entrenada con el conjunto de imágenes de “imagenet”. En la última capa se agregaron 6 capas las cuales varían entre capas de *Dropout*, *Flatten* y capas densas. Los parámetros iniciales para ingresar a la red es una imagen de 224*224, se utilizó un *batch size* de 16 y la función de pérdida es *binary crossentropy* (debido a la clasificación binaria), el optimizador utilizado es ADAM con un *learning rate* de 0.0005. Para esta tercera prueba se utilizaron 300 épocas.

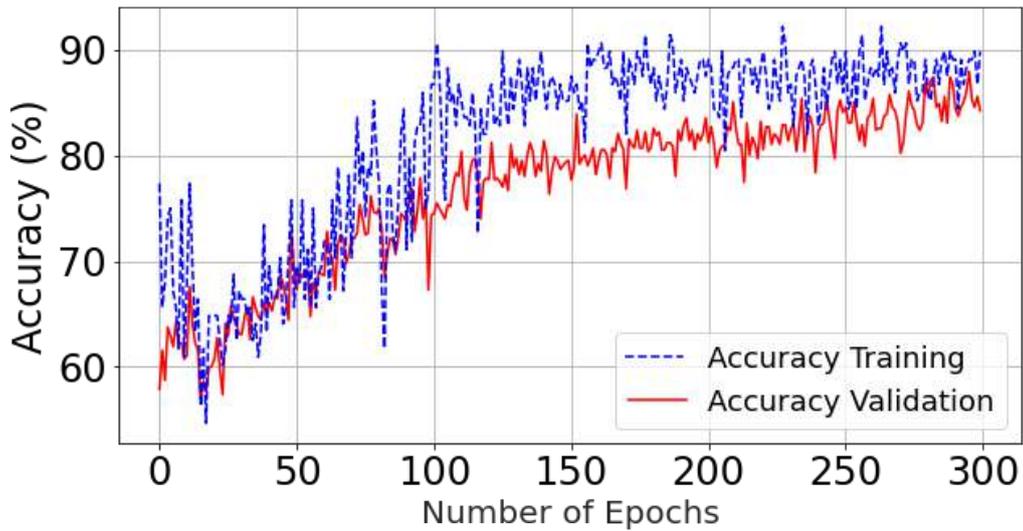


Figura 60. Gráfica de Accuracy para la prueba 2 con la arquitectura ResNet50.

Fuente: Elaboración propia.

El análisis de la Figura 60 está determinado mayormente por la curva de validación, ya que los parámetros del modelo ya han aprendido y corregido en el entrenamiento, en este caso, se observa desde el primer punto un valor aproximado de 0.9 en *accuracy*. El valor promedio para la validación es de 0.91, obteniendo un resultado mayor. Para observar si el modelo no se ha sobre ajustado a los datos, se observará el comportamiento de la gráfica *loss*.

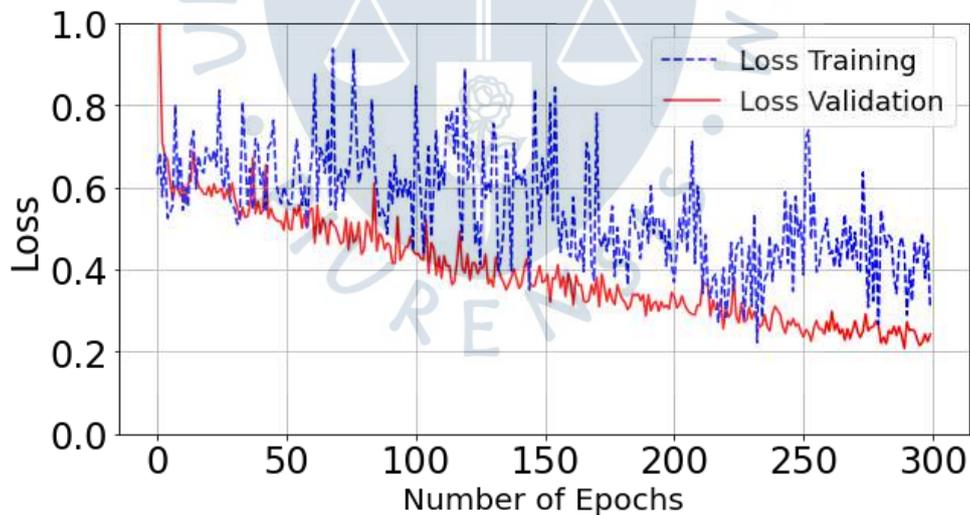


Figura 61. Gráfica de Loss para la prueba 2 con la arquitectura ResNet50.

Fuente: Elaboración propia.

En la Figura 61 se observa el valor de *loss*. Comparando las dos gráficas de *accuracy* y *loss*, para esta tercera prueba, se puede apreciar que el valor de la gráfica de *accuracy*, tanto entrenamiento como validación tienden hacia un valor de 90%. Observando la gráfica de error, tiende a un valor de 0.2 a medida que aumentan las épocas.

3.6.4. Redes Neuronales usando herramientas de Matlab

Se buscó desarrollar un algoritmo de aprendizaje automático que pudiera clasificar la apariencia de la uva, por lo que se eligió una red neuronal artificial, se propuso una metodología alternativa con el apoyo de herramientas del programa Matlab, la cual a través de su *toolbox Deep Network Designer* se podía implementar, trabajar y editar arquitecturas de redes neuronales convolucionales. Con la data obtenida se pudo cargar a la herramienta y dividirla para el entrenamiento y validación de esta. En este caso se aplicó CNN (*Convolutional Neural Networks*) con la arquitectura VGG19.

Debido al estado en bruto en el que se capturaron las imágenes y al fondo que podría dificultar el entrenamiento en la red, se intentó utilizar herramientas para segmentar y eliminar el fondo. Sin embargo, no fue posible eliminar todo el fondo de forma automática, por lo que fue necesario recurrir al programa Photoshop y así eliminar el fondo manualmente

Además, se utilizó la caja de herramientas para cambiar los espacios de color a las imágenes y poder estimar qué espacio es el más apropiado. Al hacerlo, tenemos 2 valores finales: RGB, HSV.



Figura 62. Imagen de uva de mesa sin fondo.

Fuente: Elaboración propia.

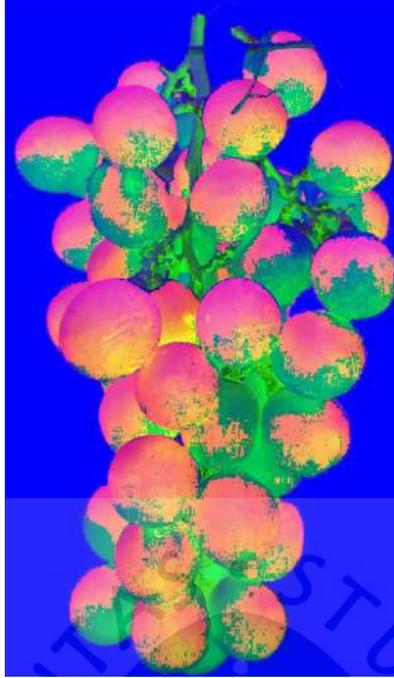


Figura 63. Imagen de uva de mesa sin fondo.

Fuente: Elaboración propia.

Usando el solucionador sgdm, un tamaño de lote mínimo de 20 y aplicando 20 épocas, se logró una precisión del 60%.

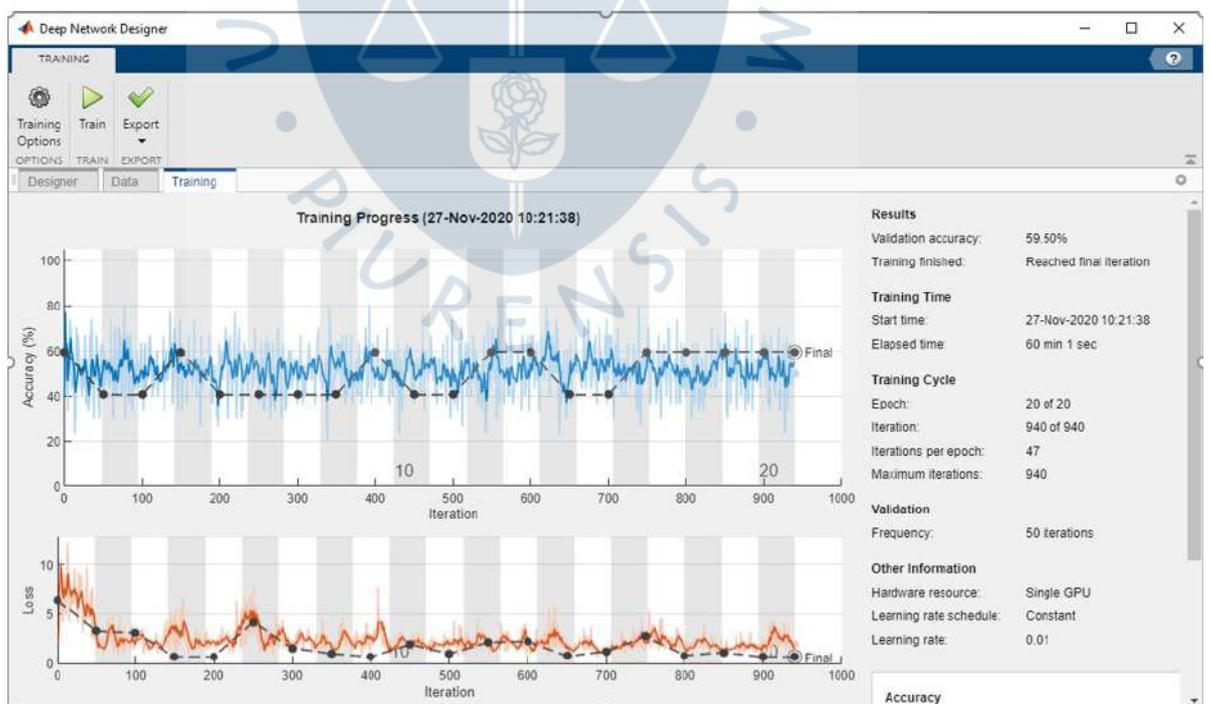


Figura 64. Proceso del entrenamiento en la herramienta Deep Network Designer

Fuente: Elaboración propia.

Es destacable observar el bajo porcentaje en la validación de Precisión obtenida a través de la red, una de las razones son los datos obtenidos, debido a que la uva es un fruto con muchas características como el color, el raquis y los tonos pueden variar debido a su heterogeneidad. Asimismo, fue difícil encontrarlos a todos en el mismo estado de madurez y procesar todas las imágenes en las mismas circunstancias, ya que se tomaron datos en 2 diferentes ocasiones por motivos externos.





Capítulo 4

Análisis de Resultados

4.1. Resultados de la estimación del peso

En nuestra primera variante se tomó la data de una manera no uniforme para el número de racimos grandes y pequeños, en el primer lote de imágenes la cantidad de imágenes de racimos pequeños y medianos eran distintos, siendo un número mayor las imágenes de racimos pequeños. En la segunda variante se escogió la misma cantidad de racimos pequeños y medianos, en este caso se trabajó con 68 imágenes en total. Para ambas variantes se estimó el peso con una red neuronal (3 neuronas de entrada, 5 y 6 neuronas en la capa oculta).

En la tercera variante se estimó el peso del racimo mediante una regresión lineal multivariable, para este caso se usó el primer lote de imágenes.

Tabla 3. Error promedio para las distintas variables.

Estimación del peso	Error (%)
Primera variante (Red neuronal)	14.63
Segunda variante (Red neuronal)	19.8
Tercera variante (Regresión lineal)	13.3

Fuente: Elaboración propia.

El error promedio que se obtuvo en la prueba de testeo es de 19.8%, esto se puede explicar porque la cantidad de imágenes con la que se trabajo es pequeña (solo 68 imágenes en total), ya que las redes neuronales trabajan mejor mientras más datos cuentan para el entrenamiento y validación.

Para la tercera variante se procedió a hacer una regresión lineal del tipo multivariable donde se obtuvo un error promedio de 13.3 % para un testeo de 10 imágenes, tal como se esperaba los algoritmos de *Machine Learning* tienen un mejor comportamiento para una data pequeña.

4.2. Resultados de la estimación del calibre

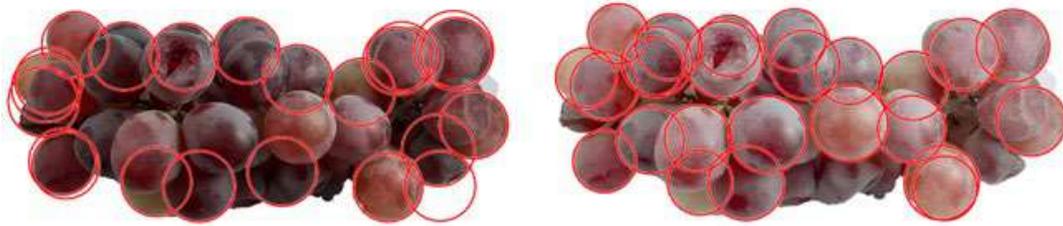


Figura 65. Detección de círculos para la imagen sin modificar (izquierda) y la modificada (derecha).

Fuente: Elaboración propia.

Para la correcta estimación del calibre se procedió con los siguientes pasos: Mejora del contraste, detección de Bordes, transformada de Hough y como último paso el promedio de los círculos aproximados.

Finalmente, al promedio se multiplica por una constante que me relaciona pixeles con milímetros, en nuestro caso se halló esta constante tomando una distancia ya conocida en la imagen.

$$K = \left(\frac{210}{2907}\right)\text{mm/pixeles}$$

$$K = 0.0722 \text{ mm/pixeles}$$

$$\text{Calibre}_{\text{estimado}} = 2 * r * K$$

$$\text{Calibre}_{\text{estimado}} = 2 * 185.68 * 0.0722 = 26.83\text{mm}$$

El calibre real fue de 24 mm por lo que el error es de 7.7%, esto se puede explicar porque se necesita un rango de diámetros para la aproximación, se puede obtener mejores resultados ajustando este valor. Otra causa puede ser que se detecta 2 o 3 círculos por cada baya, para un trabajo futuro se debe corregir este problema.

4.3. Resultados de la clasificación de apariencia con SVM

Para el caso de las máquinas de soporte vectorial, se utilizaron con la finalidad de clasificar la apariencia de los racimos de uva.

En total se realizaron 16 experimentos con diferentes modelos en busca del que arroje mejores resultados. A continuación, se muestran los más relevantes:

Tabla 4. Pruebas realizadas con SVM

Parámetros	Accuracy
5 CARACTERISTICAS	69.8%
TODAS	76.9%
COLOR	77.2%
COLOR + TEXTURA	78.1%
TEXTURA	74.0%

Fuente: Elaboración propia.

Como se aprecia, la máquina de soporte vectorial que mejor resultados arrojó fue la que contenía los parámetros de color y textura, llegando a un 78.1% de precisión. A continuación, se detallarán los resultados de esta SVM de color y textura con modificación de contraste.

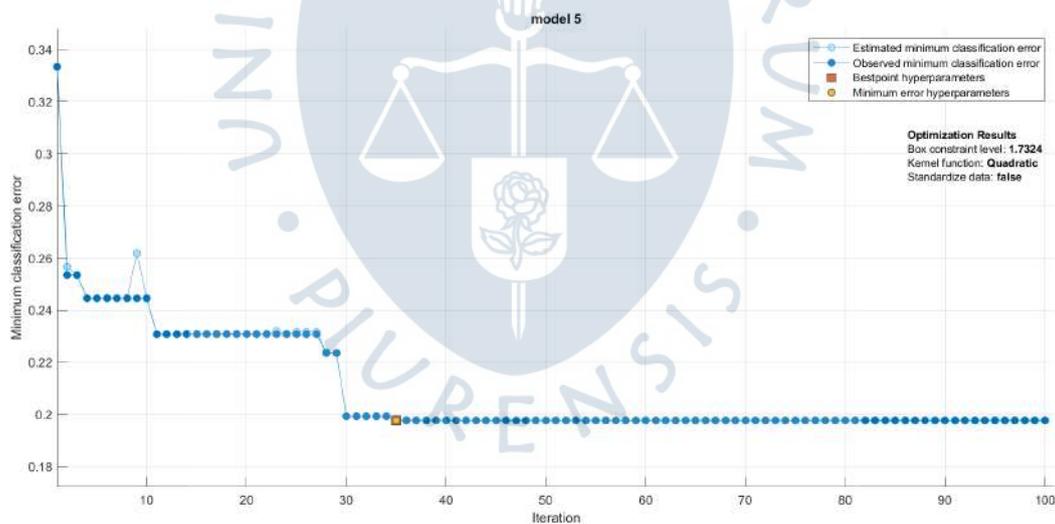


Figura 66. Gráfica del mínimo error de clasificación para la SVM de textura y color con modificación de contraste.

Fuente: Elaboración propia.

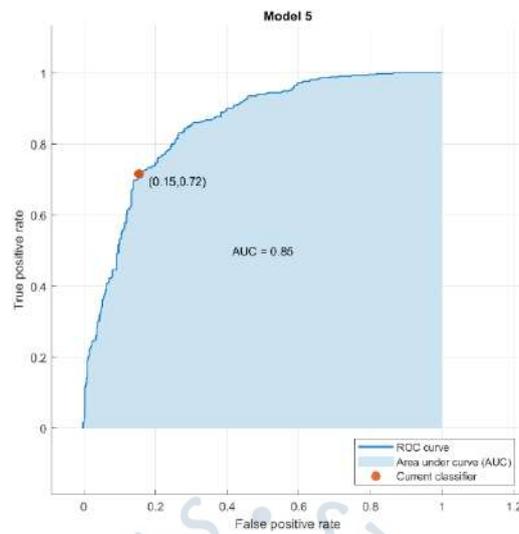


Figura 67. Curva ROC para la SVM de textura y color con modificación de contraste.

Fuente: Elaboración propia.

Tabla 5. Matriz de Confusión - SVM color y textura.

RESULTADOS							
		BUENA S	MALA S	Precision (%)	Recall (%)	F-1 Score (%)	Accuracy (%)
COLOR + TEXTUR A	BUENAS	296	117	71.7%	82.2%	76.6%	78.1%
	MALAS	64	349	84.5%	74.9%		

Fuente: Elaboración propia

Para este caso se han cruzado las características de textura y color debido a todos los entrenamientos realizados anteriormente, son las características con las que mejor características se obtienen. Se aprecia que es el mejor modelo obtenido con un AUC de 0.85 que, según la teoría detallada previamente, califica al modelo como “Muy Bueno”.

El modelo realiza una predicción con una precisión de clasificación de Uvas buenas de 72% y de clasificar uvas malas de 85%.

4.4. Resultados de la clasificación de apariencia con CNN

De acuerdo con los resultados obtenidos con redes neuronales convolucionales, utilizando la técnica de *transfer learning*, los mejores resultados se obtuvieron con la arquitectura VGG 16, teniendo como hiperparámetros una función de pérdida a *crossentropy*, un optimizador ADAM y la métrica *accuracy* para observar el rendimiento de la red neuronal.

Los valores iniciales han sido extraídos de (Keras, n.d.) para observar el número de clases para cada arquitectura, En el caso de VGG-16, se trabaja con una entrada de 224×224 . Para el número de épocas, se ha utilizado un valor de 50, siendo el valor de *batch size* de 32.

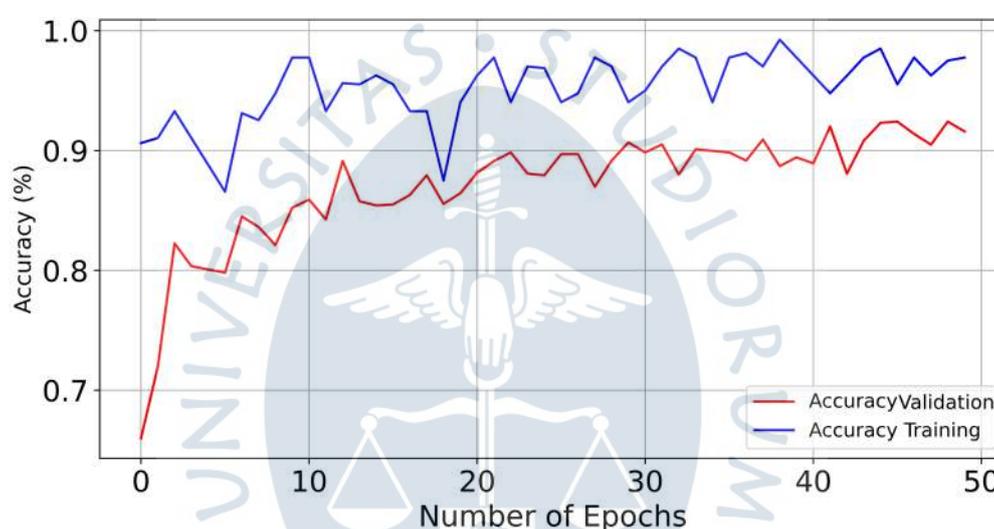


Figura 68. Gráfica accuracy en la CNN aplicando transfer learning para la red VGG-16

Fuente: Elaboración propia.

El análisis de la Figura 68 está determinado mayormente por la curva de validación, ya que los parámetros del modelo ya han aprendido y corregido en el entrenamiento, en este caso, se observa desde el primer punto un valor aproximado de 0.9 en *accuracy*. El valor promedio para la validación es de 0.91, obteniendo un resultado mayor. Para observar si el modelo no se ha sobre ajustado a los datos, se observará el comportamiento de la gráfica *loss*.

El análisis de la Figura 69 tiene valores de pérdida bastante aceptables, debido a que se encuentran entre 0 y 0.3. Generalmente se encuentran pérdidas mayores en el entrenamiento que en la validación ya que se utiliza *dropout*.

Mientras menor sea la pérdida, mejor será el modelo, teniendo en cuenta que el modelo no se haya sobre ajustado a los datos de entrenamiento. A comparación de la curva *accuracy*, la cual tiene valores en porcentaje, la curva de *loss* es una suma de los errores

cometidos para cada ejemplo, tanto en el conjunto de entrenamiento como en el de validación. Para este caso, la función de activación ha sido la entropía cruzada.

Para la validación, se aplicó un total de 83 datos para cada clase, esto se puede observar en la Tabla 1, donde:

- TP = *True positive* o verdaderos positivos
- FP = *False positive* o falsos positivos

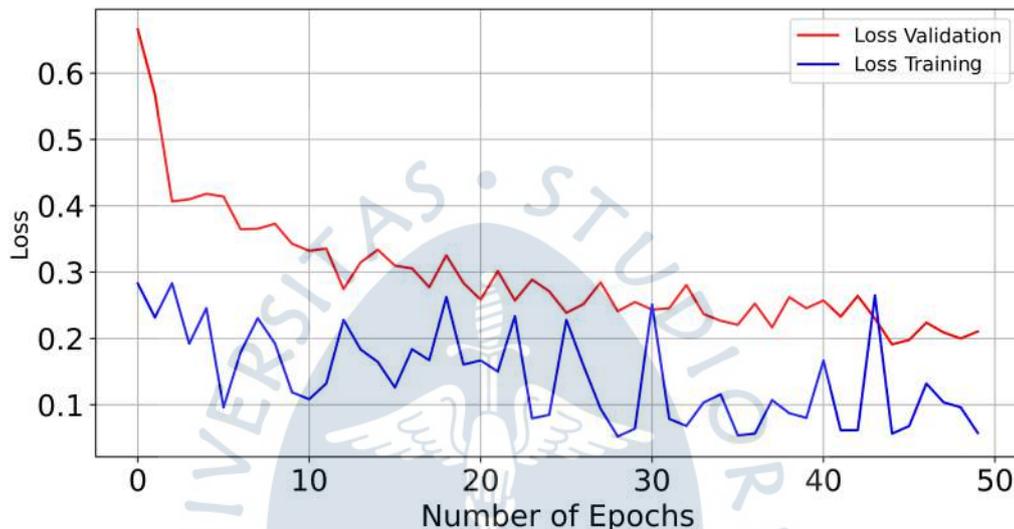


Figura 69. Gráfica loss en la CNN aplicando transfer learning para la red VGG-16
Fuente: Elaboración propia.

Tabla 6. Resultados de la validación para la arquitectura VGG-16.

	Total de muestras	TP	FP	Accuracy	Global accuracy
High quality	83	77	6	92.77%	90.36%
Low Quality	83	73	10	87.95%	

Fuente: Elaboración propia.

El valor global de *accuracy* entre las dos clases será calculado de la siguiente fórmula:

$$Accuracy = \frac{\text{Correct number of predictions}}{\text{Total number of predictions}} = \frac{77 + 73}{83 + 83} = 90.36\%$$

Este valor de 90.36% es aceptable en el campo del *deep learning*, por lo que esta red ha sido la que ha tenido mejores resultados tanto en entrenamiento como validación.

Como resultado final, se creó una interfaz que tendrá como finalidad ser el clasificador de calidad de uva red globe, la cual se puede trabajar como servidor local. Dentro de la interfaz se puede cargar imágenes hasta 200 MB de capacidad, siendo los formatos aceptables “PNG”, “JPG” y “JPEG”. Esta interfaz se puede observar en la Figura 70.



Figura 70. Inicio de la API Streamlit como servidor local.

Fuente: Elaboración propia.

Para observar la correcta predicción de la API, lo siguiente será predecir una imagen de la uva red globe. Para el primer ejemplo, se utiliza una uva de buena calidad. Se observa cómo se imprime la respuesta del estado de la uva, siendo la predicción correcta.

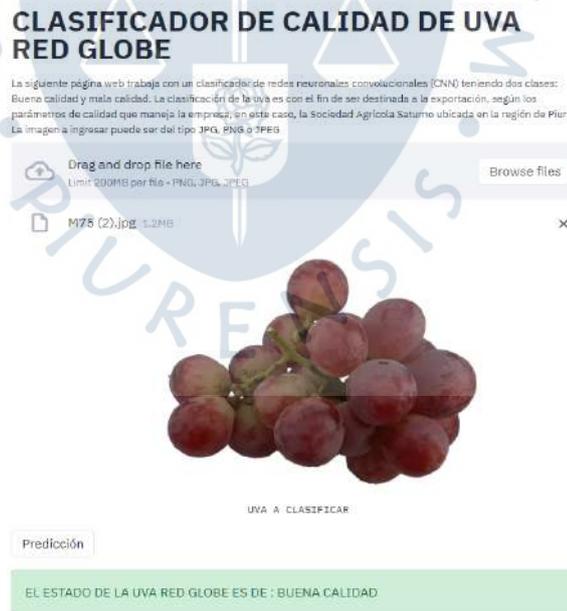


Figura 71. Clasificación de uva en la API Streamlit como servidor local.

Fuente: Elaboración propia.

Para el segundo ejemplo, se utiliza una imagen de mala calidad como se observa en la Figura 71. Para este ejemplo, la predicción también fue correcta, por lo que se concluye que la API funciona de manera correcta con respecto a la arquitectura utilizada.

CLASIFICADOR DE CALIDAD DE UVA RED GLOBE

La siguiente página web trabaja con un clasificador de redes neuronales convolucionales (CNN) teniendo dos clases: Buena calidad y mala calidad. La clasificación de la uva es con el fin de ser destinada a la exportación, según los parámetros de calidad que maneja la empresa, en este caso, la Sociedad Agrícola Saturno ubicada en la región de Piura. La imagen a ingresar puede ser del tipo JPG, PNG o JPEG.

Drag and drop file here
Límite: 200MB por file • PNG, JPG, JPEG

Browse files

20201031_0920251.jpg 677KB



UVA A CLASIFICAR

Predicción

EL ESTADO DE LA UVA RED GLOBE ES DE : MALA CALIDAD

Figura 74. Clasificación de uva en la API Streamlit como servidor local.

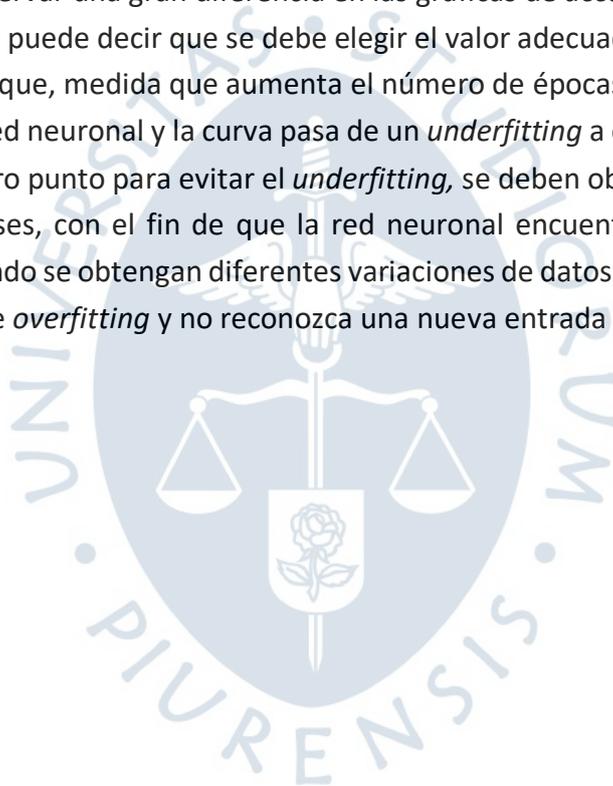
Fuente: Elaboración propia.

Conclusiones

1. Los algoritmos desarrollados por parte de la apariencia nos dan un *accuracy* del 78.1% para la clasificación de uvas con la categoría “Buenas o Aptas para exportar” aplicando SVM mientras que, las CNN nos brindan alta precisión en la validación (90.36%) mientras que en el testeo final del modelo no se lograban grandes resultados.
2. Con respecto a la estimación de calibre, se obtuvo un *accuracy* de 86.7% utilizando *Machine Learning*, mientras que con redes neuronales se obtuvo un 82.79%. Nuevamente los algoritmos de aprendizaje automático superaron a los de aprendizaje profundo, dando a notar que para este tipo específico de clasificación (racimos de uva), los algoritmos de aprendizaje automático otorgan mejores resultados.
3. En el caso de la estimación del peso por medio de técnicas de procesamiento de imágenes el resultado único obtenido fue de 92.3% de precisión. Es cierto que es un nivel aceptable, pero, la relevancia del parámetro dentro del sector agroindustrial exige un mayor nivel de precisión ya que ese 7.7% de error puede representarse en pérdidas para la empresa exportadora o inconvenientes con los mercados internacionales.
4. La investigación demuestra que, los algoritmos de *Machine Learning* brindan mejores resultados cuando se trabaja con racimos de uvas, debido a la complejidad del fruto. La detección de bayas como conjunto (racimo), hace que la precisión de los algoritmos no sea tan elevada. Sin embargo, no hay investigación a la fecha en la que se profundice la medición de los tres parámetros de calidad en racimos de uva presentados en esta investigación por medio de imágenes RGB, la mayoría investigaciones se limitan a la detección de racimos en viñedos.
5. Según las experiencias y pruebas repasadas en muchos estudios, la mejor red neuronal para la clasificación de imágenes es la convolucional, ya que, en comparación con las otras redes, ya sea profunda o recurrente, la convolucional se encargan de ir reduciendo la imagen a través de las convoluciones y el *Max Pooling* conforme va avanzando, reduciéndose así en cada capa y dejando lo más resaltante de la imagen hasta llegar a la capa de salida. Sin embargo, esto depende de muchos factores como la cantidad y calidad de las imágenes con las que se trabajen.
6. Es muy importante el procesamiento que se le haga a las imágenes, como la aplicación de segmentación, filtros, ya que un buen tratamiento de estas implica que la red pueda

reconocer y entrenar de una manera correcta y así tener valores altos en la validación con el menor tiempo posible.

7. El uso de un light box ha sido de mucha utilidad en el preprocesamiento de imágenes ya que, al brindarnos las fotografías con fondo blanco facilita las tareas de segmentación e incluso se llegó a investigar que existe una paleta de colores en las cual hay un color óptimo para resaltar de mejor manera el color de las uvas Red Globe.
8. Analizando los resultados obtenidos para cada red neuronal convolucional donde se ha aplicado transfer *learning*, se concluye que, mientras mayores sean los parámetros entrenados en la red, los valores de *accuracy* serán mayores, obteniendo así un mejor rendimiento del modelo. En el caso de MobileNet, se utilizaron 9 646 273 parámetros totales, mientras que con VGG-16 se utilizaron 134 260 544 parámetros totales, llegando a observar una gran diferencia en las gráficas de *accuracy* y *loss*.
9. Finalmente, se puede decir que se debe elegir el valor adecuado tanto para las épocas a entrenar, ya que, medida que aumenta el número de épocas, se modifica más veces el peso en la red neuronal y la curva pasa de un *underfitting* a óptimo y de óptimo a un *overfitting*. Otro punto para evitar el *underfitting*, se deben obtener un *dataset* mayor de las dos clases, con el fin de que la red neuronal encuentre los valores óptimos, siempre y cuando se obtengan diferentes variaciones de datos para que la red no tenga el problema de *overfitting* y no reconozca una nueva entrada a clasificar.



Recomendaciones

1. En vista de que la adquisición de imágenes es limitada, se trabajó con 2 algoritmos que implican el uso de inteligencia artificial, aplicando conceptos de *Machine Learning* y *Deep Learning: Support Vector Machine* y *Convolutional Neural Networks*, el SVM se elige ya que es óptimo para trabajar con escasa cantidad de datos, a comparación de las CNN las cuales trabajan con una cantidad amplia de datos. La CNN ha sido elegida con vistas a futuro, cuando se pueda aumentar el *dataset* y así, implementarlo a nivel industrial ya que, los resultados con escasa cantidad de datos no son fiables ya que, los algoritmos de *Deep Learning* necesitan de muchos datos para poder generalizar correctamente el problema en cuestión.
2. La adquisición de imágenes ha sido tomada en un estudio fotográfico casero con el fin de que el procesamiento sea más preciso a la hora de segmentar y filtrar los ruidos de cada imagen, evitando el preprocesamiento manual desarrollado en softwares de diseño de imágenes como Photoshop, Corel Draw, entre otros. Como trabajo a futuro, se plantea que en espacio donde se tome la imagen del racimo en una industria, pueda contar con un espacio almacenado, con una luminosidad determinada para que el clasificador pueda aprender de manera óptima.



Referencias bibliográficas

- Acea Mena, L. (2012). *Efectos de la interpolación sobre la calidad de la segmentación empleando la transformada watershed* (tesis de pregrado). Universidad Central de Las Villas, Santa Clara, Cuba.
- Aguilar, J. y Campoverde, M. (2019). *Clasificación de frutas basadas en redes neuronales convolucionales*. *Polo del Conocimiento*, 41(5), 3-22. <https://doi.org/10.23857/pc.v5i01.1210>
- Arias Cuya, F. A. (2017). Situación y experiencia en el cultivo de uva vinífera (*Vitis vinifera* L.) en el valle de Ica (tesis de pregrado). Universidad Nacional Agraria La Molina, Lima, Perú. <http://repositorio.lamolina.edu.pe/handle/UNALM/2952>
- Basilio Vela, O. A. (2016). Estudio de un sistema de visión para control de pesos y tamaño de racimos de uva de mesa (tesis de maestría). Pontificia Universidad Católica Del Perú, Lima, Perú. <http://tesis.pucp.edu.pe/repositorio/handle/123456789/6985>
- Bhandari, A. (2019). *Feature Scaling for Machine Learning: Uderstanding the difference Between Normalization vs Standardization*. India: *Analytics Vidhya*. Recuperado de: <https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Brownlee, J. (2020a). *Cost-Sensitive Learning for Imbalanced Classification*. Australia: *Machine Learning Mastery*. Recuperado de: <https://machinelearningmastery.com/cost-sensitive-learning-for-imbalanced-classification/>
- Brownlee, J. (2020b). *Undersampling Algorithms for Imbalanced Classification*. Australia: *Machine Learning Mastery*. Recuperado de: <https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/>
- Bustamante, S. (2014). *Algoritmos de procesamiento de imagen aplicados a la detección de figuras geométricas y sus propiedades espaciales*. Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile. http://opac.pucv.cl/pucv_txt/Txt-4500/UCE4968_01.pdf
- Cabanilles, P. (2014). *Segmentación Automática del cerebro mediante técnicas de tratamiento*

- digital de imagen* (tesis de pregrado). Universidad Politécnica de Valencia, Valencia, España. https://riunet.upv.es/bitstream/handle/10251/37996/cabanillesmengual_paula.pdf?sequence=1&isAllowed=y
- Castillo, F., & Correa, S. (2019). *Propuesta de mejora de procesos de una planta de empaque de uva de mesa y determinación de indicadores* (tesis de pregrado). Universidad de Piura, Piura, Perú.
- Castro, J., Cerquena, N. y Gutiérrez, N. (2013). Determinación del color del exocarpio como indicador de desarrollo fisiológico y madurez en la guayaba pera (*Psidium guajava* cv . Guayaba pera), utilizando técnicas de procesamiento digital de imágenes. *Revista EIA*, 10(19), 79–89.
- Cavallo, D. Pietro, Cefola, M., Pace, B., Logrieco, A. F., & Attolico, G. (2019). Non-destructive and contactless quality evaluation of table grapes by a computer vision system. *Computers and Electronics in Agriculture*, 156(June 2018), 558–564. <https://doi.org/10.1016/j.compag.2018.12.019>
- Cruz Córdova, S. (2016). *Aplicación de la regresión múltiple para predecir las variables que influyen en el peso del racimo en el cultivo de uva Red Globe* (tesis de grado). Universidad Nacional de Piura, Piura, Perú. <http://repositorio.unp.edu.pe/handle/UNP/543>
- Flaya, M. (2020, February 8). *Ilustración de Partes De La Planta Morfología De La Planta De Vid Con Sistema Radicular Aislado Sobre Fondo Blanco Estructura De La Planta De Vid y más Vectores Libres de Derechos de Agricultura*. España: iStock. Recuperado de: <https://www.istockphoto.com/es/vector/partes-de-la-planta-morfología-de-la-planta-de-vid-con-sistema-radicular-aislado-gm1204220189-346413213>
- Gestión (17 de diciembre de 2017). Perú ocupa el tercer lugar en productividad de uva en el mundo. Recuperado de: <https://gestion.pe/economia/peru-ocupa-tercer-lugar-productividad-uva-mundo-222930-noticia/>
- Gupta, P. (2017, June 5). *Cross-Validation in Machine Learning. Towards Data Science*. Recuperado de: <https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f>
- Huemura, N. R. (2018). *Evaluación de una propuesta de mejora para reducir el tiempo de traslado en la cosecha de uva de mesa* (tesis de pregrado). Universidad de Piura, Piura, Perú.
- Ituarte, L. E. (2011). Estimación del Volumen en Duraznos: Un Estudio preliminar con Procesamiento Digital de Imágenes. *CAI 2011 – Congreso de Agroinformática 2011*. 1–12.
- Izaurieta, F., & Saavedra, C. (1999). Redes Neuronales Artificiales. *Charlas de Física*, 1–15. [https://doi.org/10.1016/S0210-5691\(05\)74198-X](https://doi.org/10.1016/S0210-5691(05)74198-X)
- Jiménez Hernández, S. (2016). *Aplicación de técnicas de procesado de imagen para la segmentación de núcleos en muestras histológicas humanas*. (Tesis de pregrado).

- Universidad Politécnica de Valencia, Valencia, España.
https://riunet.upv.es/bitstream/handle/10251/70432/47446589G_TFG_14733577572447932831980581352993.pdf?sequence=2&isAllowed=y
- Kaur, M., & Sharma, R. (2015). Quality Detection of Fruits by Using ANN Technique. *IOSR Journal of Electronics and Communication Engineering Ver. II, 10(4)*, 2278–2834.
<https://doi.org/10.9790/2834-10423541>
- Keras. (n.d.). *VGG16 y VGG19*. Recuperado de: <https://keras.io/api/applications/vgg/>
- Larrañaga, P. , Inza, I. y Moujahid, A. *Tema 8. Redes Neuronales*. 1–19. Universidad del País Vasco/Euskal Herriko Unibertsitatea, Biscay, España.
<http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t8neuronales.pdf>
- Marin, N. y Madrigal, C. (2016). *Clasificación de calidad de fresas usando procesamiento de imágenes y máquinas de vectores de soporte* (tesis de pregrado). 157–163. Universidad Inca Garcilazo de la Vega, Lima, Perú. <http://repositorio.uigv.edu.pe/handle/20.500.11818/664>
- Martinez, M., y Potes, J. (2019). *Diseño e implementación de un sistema de visión artificial para detectar una especie de arácnidos y nuevos brotes de flora en un hábitat alterado por un incendio* (tesis de pregrado). 53(9), 1689–1699. Universidad Autónoma de Occidente, Cali, Colombia. <http://hdl.handle.net/10614/5274>
- MINAGRI. (2019). *LA UVA PERUANA: Una oportunidad en el Mercado Mundial*. 51. <file:///C:/Users/USER/Downloads/Informe-Uva-peruana.pdf>
- MINCETUR. (2018). Reporte de Comercio Regional Piura-2018. https://www.mincetur.gob.pe/wp-content/uploads/documentos/comercio_exterior/estadisticas_y_publicaciones/estadisticas/reporte_regional/RRC_Apurimac_2018_Anual.pdf
- MINCETUR. (2018). *Reporte regional de comercio Piura - 2018*. 3. https://cdn.www.gob.pe/uploads/document/file/410278/Reporte_de_Comercio_-_Reporte_Comercio_Regional_-_RCR_-_Piura_2018_-_Anual20191030-24204-jbgu7n.pdf
- Ministerio de Comercio Exterior y Turismo. (2016). *Análisis Integral de la Logística en el Perú - Producto UVA. Primera Ed*, 34. https://cdn.www.gob.pe/uploads/document/file/31327/P_Uva.pdf
- Ministerio de Comercio Exterior y Turismo. (2019). *Reporte de Comercio Regional Arequipa-2018*. https://cdn.www.gob.pe/uploads/document/file/410220/Reporte_de_Comercio_-_Reporte_Comercio_Regional_-_RCR_-_Arequipa_2018_-_Anual20191030-24204-1q30770.pdf
- Moreyra, J. (2019). *LA UVA PERUANA : Una oportunidad en el Mercado Mundial*. 40–43. http://agroaldia.minagri.gob.pe/biblioteca/download/pdf/tematicas/f-taxonomia_plantas/f01-cultivo/2019/Informe-uva-peruana.pdf

- Noble, W. S. (2006). What is a support vector machine? *Nature Biotechnology*, 24(12), 1565–1567. <https://doi.org/10.1038/nbt1206-1565>
- Potentia. (2019). Llegada de la uva al Perú. Perú: *Pisco es Perú!*. Recuperado de: <https://pisco.net.pe/llegada-de-la-uva-al-peru/>
- Pisco es Perú. (2019). El Origen (Parte I). Perú: Pisco es Perú. Recuperado de: https://www.piscoesperu.com/el_origen_parte_i.php
- Rabatel, G., y Guizard, C. (2007). Grape berry calibration by computer vision using elliptical model fitting. *Precision Agriculture 2007 - Papers Presented at the 6th European Conference on Precision Agriculture, ECPA 2007*, 581–587.
- Ramírez, W. A. M., Durán, M. T., Morales, R. A. L., Yépez, E. C., y Ledesma, L. M. C. (2017). Determinación Del Estado De Madurez De Una Cereza Aplicando Procesamiento De Imágenes. *Jóvenes En La Ciencia*, 3(2), 2685–2689.
- Ramos, E. (2016). ¿Cuáles son los 10 productos agropecuarios peruanos más exportados este año?. Perú: Agraria.pe. Recuperado de: <https://agraria.pe/noticias/cuales-son-los-10-productos-agropecuarios-12372>
- Requena, K., Rozo, D, y Arévalo, J. (2017). Principios y características de las redes neuronales artificiales. *Desarrollo e Innovación En Ingeniería*, 2(40), 173–183.
- Romero, A., Marín, A., & Jiménez, J. (2015). Sistema de clasificación por visión artificial de mangos tipo Tommy Classification system for artificial vision type Tommy mango. *Revista UIS Ingenierías*, 14(1), 10.
- Salas, D. (2020, June 18). *Cultivo de Uvas*. Recuperado de: <https://proyectosperuanos.com/uvas/>
- Salford Systems. (2014). Random Forests for Beginners. *Salford Systems*, 71. <https://es.scribd.com/document/352251625/RANDOM-FORESTS-FOR-BEGINNERS-pdf>
- Srivastava, T. (2019). Evaluation Metrics Machine Learning. India: *Analytics Vidhya*. Recuperado de: <https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/>
- Thevenot, A. (2019). Understand and Visualize Color Spaces to Improve Your Machine Learning and Deep Learning Models. *Towards Data Science*. Recuperado de: <https://towardsdatascience.com/understand-and-visualize-color-spaces-to-improve-your-machine-learning-and-deep-learning-models-4ece80108526>
- Torres, E., Rivera, S., Mueña, V., Corradini, F., Sepúlveda, P., Abarca, P., Defilippi, B., Muñoz, I., Selles Van Sch., G., Rodriguez, F., Campos, A., & Riquelme, J. (2017). Principales Variedades de uva de mesa cultivadas en Chile. *Manual Del Cultivo de Uva de Mesa. Convenio INIA-INDAP*, 18, 152.
- Valverde, J. (2007). Detección de bordes mediante el algoritmo de Canny. *ResearchGate*.

https://www.researchgate.net/publication/267240432_Deteccion_de_bordes_mediante_el_algoritmo_de_Canny

Vera, A. (2019). Piura: exportación de uva supera las 54 mil toneladas en el inicio de la campaña. Perú: *El regional Piura*. Recuperado de: <https://www.elregionalpiura.com.pe/index.php/regionales/150-piura/38304-piura-exportacion-de-uva-supera-las-54-mil-toneladas-en-el-inicio-de-la-campana>

