UNIVERSIDAD ANDRÉS BELLO

FACULTAD DE INGENIERÍA

INGENIERÍA CIVIL INFORMÁTICA

CLASIFICACIÓN AUTOMÁTICA DE LA CALIDAD DE LOS ARÁNDANOS USANDO IMÁGENES EN ESPECTRO SWIR CON DEEP LEARNING

Tesis de pregrado para optar al título de Ingeniero Civil Informático

Alumno:

Ariel Sebastián Ignacio Chandía Delgado

Profesor guía: Ph.D. Juan Eduardo Tapia Farias

Santiago de Chile, 2019

AGRADECIMIENTOS

Totalmente agradecido de aquellas personas que mostraron preocupación durante todo este desafiante proceso. Principalmente a mi familia, por toda la paciencia y el esfuerzo que han hecho, además de la confianza que han tenido sobre mí durante todo este tiempo. Hugo Chandía Alarcón, Elizabeth Delgado Gonzalez y mi querida hermana Vania Chandía Delgado, mis principales fuentes emocionales que realmente facilitaron el camino para lograr mis metas.

Por otra parte, gustaría demostrar mi gratitud hacia mi profesor guía Ph.D. Juan Tapia Farias por la oportunidad de desarrollar esta investigación, además agradecer por dar acceso y conocimiento a tecnologías de difícil acceso para un estudiante.

La amistad es algo invaluable y agradezco enormemente a aquellos que la han compartido durante todos estos años, sin duda alguna ha sido un apoyo real.

Finalmente mis compañeros con los quienes he recorrido este trayecto, fue una experiencia increíble y les deseo un éxito tremendo, además de mis gracias correspondientes por las experiencias en el aula.

Dedicado a todos aquellos que piensan en hacer un cambio real a través de la ciencia y la investigación.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

El análisis empírico de la calidad de los arándanos en la etapa de post-cosecha presenta problemas reflejados en la calidad del producto empaquetado. Aprovechando los beneficios de utilizar el espectro SWIR se capturan imágenes de arándanos de buena y mala calidad, para luego realizar un diseño de base de datos de arándanos con 36.469 imágenes de arándanos de buena calidad y 60.615 arándanos de mala calidad, sumando un total de 97.044 imágenes. Con estas imágenes se realizan cuatro rondas de experimentos principales, dónde las primeras dos tienen un enfoque en el análisis de la intensidad de los píxeles y las otras dos analizan la textura de las imágenes, utilizando dos redes neuronales convolucionales, LeNet y Smaller VGG. El análisis de las texturas (filtro BSIF 3x3) presenta un mejor comportamiento en Smaller VGG, dónde el mejor resultado registrado se alcanza con un 96,34 % de train y un 94,05 de test, manteniendo un análisis en el fruto reflejado en un mapa de calor.

Palabras Claves: Arándanos; Aprendizaje Profundo; Espectro SWIR; ; Clasificación

CAPÍTULO 1. INTRODUCCIÓN

1.1 CONTEXTO

El arándano es un fruto que nace en racimos de la planta Vaccinium Corymbosum (Arándano), fruto que es exportado a gran parte del mundo, principalmente Estados Unidos. Este fruto se caracteriza por su sabor dulce, el cual puede llegar a ser usado para distintos fines culinarios.

”Con apenas 80 calorías por taza y prácticamente sin grasas, los arándanos ofrecen importantes beneficios para nuestra nutrición y salud. Los arándanos son una excelente fuente de vitamina C. Una porción contiene del orden de 14mg o casi el 25 % de las necesidades diarias de vitamina C, la cual es necesaria para la formación de colágeno y para mantener encías saludables. También ayuda en la absorción de hierro y contribuye a un sistema inmunológico saludable.”[1]

Durante los últimos años 9 se ha observado un crecimiento en materias de exportación de arándanos en Chile, superando las 100 mil toneladas en la última temporada [2]. Se observa figura 1.1 la cantidad de toneladas de arándanos exportados desde las temporadas 2008-2009 al 2017-2018.

FIGURA 1.1: Toneladas de arándanos exportados en las últimas diez temporadas [2].

Los puntos destacables evidenciados en la figura 1.1 son:

- El crecimiento de toneladas de arándanos exportados desde el primer periodo al ultimo es de un 166,23 %. [2]
- El crecimiento promedio de toneladas de arándanos exportados anualmente es de un 10,29 %. [2]
- Sólo dos temporadas excepcionales tuvieron una baja cantidad de exportación en comparación al periodo anterior, siendo 2013-2014 la temporada más considerable. Estas bajas no son atribuidas a ningún fenómeno o al menos no son mencionados en el compendio. [2] Aunque se menciona en el Crop & Exports Report 2017-2018, que las temporadas heladas afectan directamente al crecimiento del fruto, por lo tanto, es un factor importante en temas de producción [3].

Si tomamos en cuenta la cantidad de toneladas exportadas en contraste a la cantidad de cajas de arándanos exportados evidenciados en la figura 1.2, el peso promedio de cada caja es de 2,8035 Kg aproximadamente. El valor promedio por caja puede variar bastante según el sector de exportación, siendo Norteamérica (Canadá y Estados Unidos) el mayor consumidor tienen el promedio más alto, luego lo sigue Europa y finalmente Asia, todos mantienen la misma relación de consumo y peso promedio por caja [2].

FIGURA 1.2: Número de cajas de arándanos exportados en las últimas nueve temporadas [2].

1.2 MOTIVACIÓN

Chile actualmente es considerado como el segundo mayor exportador de arándanos frescos a nivel mundial [1]. Los arándanos mayoritariamente son seleccionados de manera manual, de acuerdo a la capacidad y experiencia que posee cada operador, siendo por tanto una medida de calidad muy subjetiva.

Actualmente, en el mercado no existe una unidad de medida establecida de manera objetiva que permita ser interpretada por diferentes grupos de agricultores a nivel nacional como en otros países. Por tanto, la calidad se ve influenciada por la jornada del temporero, condiciones ambientales, duración de turnos, entre otros. La cosecha se concentra en tiempos cortos por jornadas extensas.

La fatiga visual afectará el temporero, por lo cual los primeros frutos seleccionados serán de mejor calidad. como se puede apreciar en la figura 1.3. Debido a estas condiciones, existe la necesidad de realizar este trabajo de manera eficiente y asegurar una producción de calidad. Entonces, se necesita minimizar la cantidad de fruto rechazado o devuelto cuando es recibido en la ciudad de destino.

Esta tesis busca estudiar la posibilidad de seleccionar los frutos en base a su calidad (Buena y/o Mala) usando la experiencia de los temporeros al capturar las imágenes seleccionadas como Buenas y “Malas” durante el proceso de packing y usarlas para entrenar una red convolucional mediante técnicas de Deep learning.

FIGURA 1.3: Trabajadora que separa los arándanos según su calidad (separar buenos y malos) [1].

La realización de esta tesis busca apoyar al sector agricultor, específicamente a las pequeñas y medianas empresas (Pymes) que no tienen acceso a máquinas industriales de clasificación por su precio y esta falta causa una desventaja frente a los competidores de alta gama. Dando una oportunidad de seguir y crecer en el mercado.

Es necesario tomar en cuenta todos los aspectos a los que está relacionado la calidad del fruto, para saber bajo que estándares se va a trabajar. Entonces, identificar el fruto, sus cualidades y si estas denotan ser perjudiciales o positivas en el fruto para finalmente saber la calidad de este.

El arándano es un fruto delicado y su calidad varía en el tiempo según factores a los que se expone, ya sea por temperatura, humedad, manipulación, deterioro por hongos o nivel de madurez. A nivel visible del arándano se considera el color del fruto, el cual guarda relación directa con nivel de madurez del fruto, considera también el nivel de rugosidad relacionado con la madurez y la cantidad de agua y las infecciones que presenta, las cuales suelen ser hongos de color marrón (antracnosis) o blancos (botrytis) como se puede ver en la figura 1.4 en el apartado de hongos. En otros casos la tierra puede tener altos niveles de potasio y el fruto puede agrietarse [4] como se puede ver en la figura 1.4, en el apartado de cicatrices.

La calidad del fruto se puede ver desde tres perspectivas: calidad visible, calidad organoléptica y calidad nutritiva. De todas estas, la visible es fácilmente identificable, pues considera el color (completamente azul idealmente), tamaño, forma y ausencia de infecciones, grietas o rugosidad [4].

A continuación en la figura 1.4 se presentan distintos arándanos diferenciándose por sus cualidades.

FIGURA 1.4: Afiche de defecto de arándanos. Fuente: `http://www.comitedearandanos.cl/lo-que-usted-debe-saber-tecnicas-de-manejo-de-cosecha-y-poscosecha-en-arandanos/`

El afiche del comité de arándanos mostrado en la figura 1.4 sirve a modo de guía para tener en consideración qué tipos de arándanos están aptos para su comercialización y así tener ciertos estándares de calidad con los cuales trabajar. Aquellos arándanos afectados muy levemente por el frio o por exceso de potasio pueden ser considerados de buena calidad. En casos como insectos, hongos o materias extrañas, pueden impactar en los demás arándanos donde son almacenados. Existe un documento llamado Manual del manejo agronómico del arándano [4],

el cual dicta una manipulación adecuada para el fruto, tanto en etapa de pre-cosecha cómo en post-cosecha.

Para saber el estado de los arándanos es posible recurrir al uso de imágenes y/o grabaciones, dado que estas tendrán información fruto mismo. La luz se muestra a distintos umbrales de longitud de onda electromagnética y según su rango de trabajo podría mostrar datos útiles.

El espectro electromagnético de la luz tiene diversas clasificaciones según su longitud de onda y frecuencia. La luz visible considera lo que es capaz de percibir el ojo humano y su representación se puede visualizar en la figura 1.5.

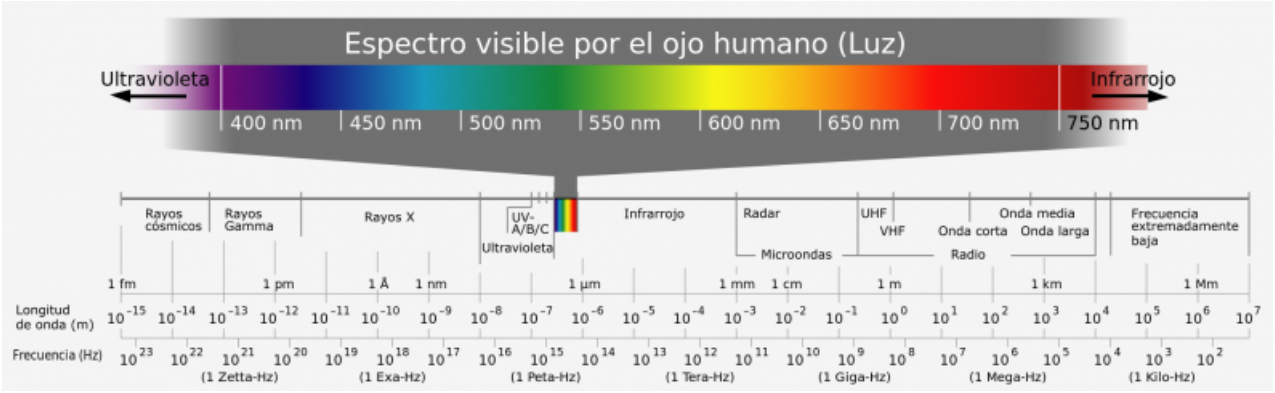


FIGURA 1.5: Espectro electromagnético de la luz. Fuente: `https://www.vix.com/es/btg/curiosidades/2011/10/02/el-espectro-visible-de-luz`

Dentro de la clasificación infrarroja, existe el espectro short-wavelength infrared (SWIR) el cual comprende las longitudes desde 0.7μm hasta 3.0μm aproximadamente como se puede observar en la figura 1.6. Por parte de las imágenes capturadas en este espectro, son capaces de revelar información que no son visibles para el ojo humano, el cual es capaz captar ondas de longitud entre 380nm hasta 780nm. Por lo tanto, estas imágenes pueden evidenciar características no observables por el ojo humano y ser analizadas para extraer información de interés de la imagen. En la siguiente figura se puede observar la clasificación del espectro infrarrojo.

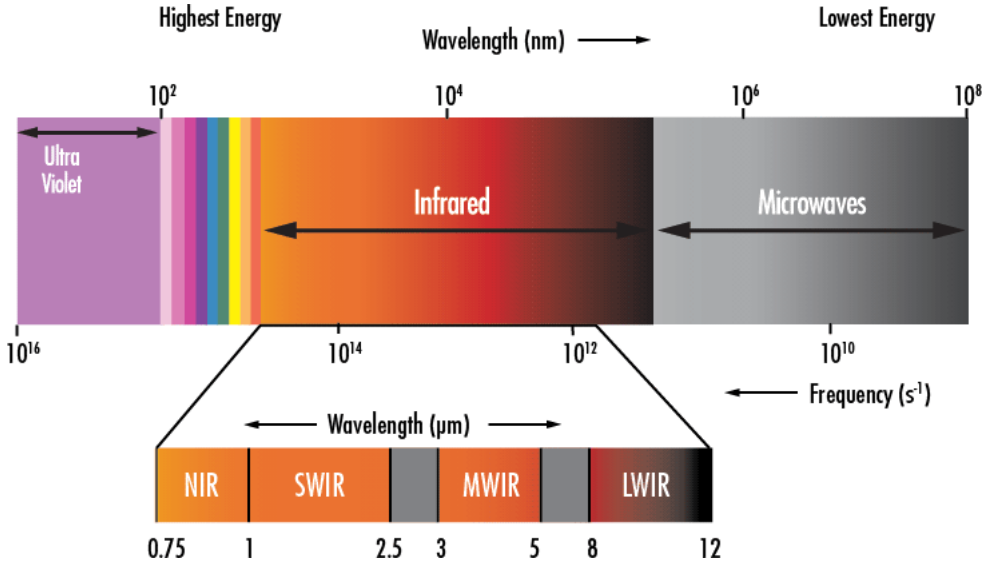


FIGURA 1.6: Espectro Infrarrojo y sus clasificaciones. Fuente: `https://www.edmundoptics.com/resources/application-notes/imaging/what-is-swir/`

Las imágenes tienen gran cantidad de datos potenciales a ser procesadas con distintos algoritmos o conjuntos de estos, siendo uno de ellos Deep Learning (aprendizaje profundo) que dentro del campo de visión artificial es dónde se destaca por sus buenos resultados, el cual se explicará en detalle en el capítulo 5.

CAPÍTULO 2. ESTADO DEL ARTE

2.1 INTRODUCCIÓN

En el año 2018 M.-H. Hu, Y. Zhao y G.-T. Zhai, en Shanghai Jiao Tong University, utilizaron técnicas de Machine Learning, aprendizaje supervisado específicamente aprendizaje activo, utilizando imágenes en espectro hiperespectral. El foco principal de este estudio es verificar que tan eficiente puede ser un clasificador al ocupar algoritmos de aprendizaje activo en combinación con el espectro mencionado anteriormente, ocupando también muy poca cantidad de datos etiquetados. El algoritmo usado fue el Estimated error reduction, en comparación a algoritmos de aprendizaje supervisado como support vector machine (SVM) [6] y self-organizing maps(SOM) [7]. Los resultados obtenidos pueden ser visualizados a continuación en la tabla 2.1:

TABLA 2.1: Medidas de rendimiento de los algoritmos estudiados por [5].

Machine learning technique	Accuracy	Precision	Recall Rate	No. of training sample
Estimated error reduction	0.87	0.94	0.78	9
SOM	0.46	0.51	0.34	557
SVM	0.88	0.84	0.97	228

En el año 2013 G. A. Leiva-Valenzuela y J. M. Aguilera, en la Pontificia Universidad Católica de Chile, realizaron un estudio sobre la detección automática de enfermedades que podrían tener arándanos. Estudio orientado a la visión computacional que su fin busca mejorar la calidad del almacenamiento después de la cosecha. Para esto en primera parte se separó, una por la parte del tallo y por la contraparte de esta en el fruto como partes a analizar, además de definir cuatro clases definidas como ”control”, ”shriveled”, ”fungally decayedz ”mechanically damaged”.

Luego se utilizó una herramienta de Matlab denominada ”Balu”[9] para la extracción de características, las cuales resultaron con un número amplio de 951 características. Luego se

combinaron 5 algoritmos para obtener 13 características asociadas a las cuatro clases predefinidas anteriormente. Con cross validation se verificó cuales algoritmos se trabajaría para la extracción de características, los cuales fueron ”SFS-LDA”, ”SFS-FISHER”, ”SFS-5-KNN”, ”RANK-ROCz ”RANK-t TEST”[8], [10].

Finalmente para obtener los resultados finales y verificar qué clasificador tiene el mejor comportamiento, se decidió separar los arándanos en buen estado de los en mal estado, los resultados están expresados en rangos de valores, siendo el promedio el valor tomado en cuenta como comportamiento y se pueden visualizar en la siguiente tabla en base al clasificador con mejor comportamiento:

TABLA 2.2: Resultados obtenidos de la clasificación [8].

Experiment	Performance (C.I.)*		
	Stem end	Calyx end	Both orientations
Two classes detection:			
Control vs. Shriveled	96.67 (91.42-100)[a]	95.00 (88.64-100)[a]	93.3 (88.18-98.48)[d]
Control vs. Fungally decayed	100.0 (100-100)[a]	98.89 (95.35-100)[b]	97.00 (83.23-100)[e]
Control vs. Mechanically damaged	88.33 (78.96 - 97.71)[a]	86.67 (76.74-96.59)[a]	90.00 (83.80-96.20)[d]
Control vs. Shriveled + Fungally decayed + Mechanically damaged	96.67 (92.69 - 100.0)[d]	93.67 (88.29-99.04)[d]	89.34 (84.69-94.00)[c]
Stem end vs. Calyx end	-	-	96.82 (94.17-99.47)[c]
Four Classes detection:			
Control vs. Shriveled vs. Fungally decayed vs. Mechanically damaged	77.50 (68.88-86.12)[d]	71.33 (61.35-81.32)[d]	67.86 (60.82-74.90)[c]

*Performances and confidence interval were calculated using 10 fold cross validation with 95 % of confidence.
[a]60 images, classification with 4 best presicion selected features.
[b]45 images, classification with 4 best selected features.
[c]225 images, classification with 13 best selected features.
[d]120 images, classification with 12 best selected features.
[e]105 images, classification with 10 best selected features.

Otra investigación realizada en el año 2018 por J. Kuzy, Y. Jiang y C. Li, tomó lugar en University of Georgia, consiste en la detección de golpes en el arándano. Para esto se tomaron en cuenta dos tipos de arándano: Meadowlark y Farthing. Cabe destacar que fue usado el espectro térmico, con la cámara termal FLIR T440 (`http://arquitecsolar.com/termografia/145-flir-t440-camara-termografica.html`), en condiciones de luz y calor favorables (cuatro focos de 325-W). Además se contaba con aire acondicionado en la sala de registros para mantener la temperatura de la sala.

Las imágenes fueron extraídas de los frames de videos que se grabaron con la cámara. Primero se dejaba 30 minutos para que el sistema evitar cambios en la temperatura en plena medición.

Para la extracción de características se uso un algoritmo nativo de MATLAB llamado Fast Fourier Transform (FFT) [12]. Luego para la selección de características predominantes fue usado el algoritmo RELIEFF [13] implementado en WEKA (The University of Waikato, Hamilton,

New Zealand).

Finalmente para la clasificación se utilizaron cinco algoritmos, linear discriminant analysis (LDA) [14], support vector machine (SVM) [6], random forest [15], logistic regression [16], y K-nearest-neighbors (KNN) [17]. Los resultados finales de la clasificación se encuentran en la siguiente tabla:

TABLA 2.3: Resultados obtenidos de la clasificación [11].

Cultivar	Feature Set	SVM Normalized		SVM Non-normalized		Random Forest	LDA	Logistic Regression	K Nearest Neighbors
		RBF	Linear	RBF	Linear				
Farthing	Waveform	81.84	87.67	68.06	69.90	78.92	84.28	89.50	82.52
Farthing	Amplitude	67.69	82.57	51.09	81.99	77.60	82.57	79.90	73.73
Farthing	Phase	61.77	65.38	54.30	59.40	72.17	61.22	58.56	52.49
Farthing	7	67.99	66.49	46.47	68.18	72.13	78.18	71.58	66.21
Farthing	13	70.51	79.72	48.37	67.84	80.94	70.76	84.39	73.48
Farthing	20	74.99	83.34	54.59	84.06	81.01	82.58	78.53	67.39
Meadowlark	Waveform	78.63	76.39	77.71	78.63	74.94	76.26	77.89	73.72
Meadowlark	Amplitude	68.63	75.76	55.00	73.97	75.53	74.68	70.78	67.10
Meadowlark	Phase	53.53	74.11	58.39	65.11	78.06	73.99	70.11	62.14
Meadowlark	7	76.94	76.01	61.57	75.56	73.89	77.26	76.93	69.28
Meadowlark	13	78.61	79.06	55.00	74.46	78.99	77.74	76.76	72.56
Meadowlark	20	79.74	80.72	55.00	75.36	78.85	77.67	75.43	73.43

El pasado año 2018 se llevo realizó un estudio por K. P. Ferentinos, que consiste en la detección y diagnóstico de enfermedades en plantas. Utilizando técnicas de Deep Learning y una base de datos con 87.848 imágenes en espectro visual, la cual solía estar disponible, sin embargo no se ha logrado acceder a ella. Se consideraron plantas de manzana, banana, arándano, repollo, tapioca, melón, apio, cereza, maíz, pepino, berenjena, uva, cebolla, naranja, calabaza, entre otros, algunos más de un tipo entre ellos, dando un total de 58 clases diferentes. Utilizando imágenes en distintos ambientes, se lograron obtener resultados bastante buenos, siendo el más alto de un 99.48 % utilizando la red VGG [19]. Los resultados pueden ser visualizados en la siguiente tabla 2.4:

TABLA 2.4: Resultados obtenidos de la clasificación con Deep Learning con imágenes en ambiente de laboratorio y en terreno [18].

	Original images				Pre-Processed images			
Model	Success rate	Average error	Epoch	Time(s/epoch)	Success rate	Average error	Epoch	Time(s/epoch)
AlexNet	99.06	0.0354	47	7034	98.64	0.0658	50	1022
AlexNetOWTBn	99.44	**0.0192**	46	7520	99.07	0.0332	45	1125
GoogLeNet	97.27	0.0957	45	7845	97.06	0.0984	40	2670
Overfeat	98.96	0.0412	45	6204	98.26	0.0848	49	1570
VGG	**99.48**	0.0223	48	7294	98.87	0.0542	49	4208

Es necesario destacar que luego, se utilizaron imágenes en ambientes separados, es

decir, las fotografías en ambientes de laboratorio fueron utilizadas como train y las en ambiente de terreno como test, y viceversa, dando resultados bastante bajos. Los resultados se pueden visualizar en la tabla 2.5:

TABLA 2.5: Resultados obtenidos de la clasificación, utilizando imágenes en ambientes diferentes en test y en train [18].

	Training: Laboratory - Testing: Field				Training: Field - Testing: Laboratory			
Model	Success rate	Average error	Epoch	Time(s/epoch)	Success rate	Average error	Epoch	Time(s/epoch)
AlexNetOWTBn	32.23	**3.5484**	53	4375	62.57	1.9369	104	-
VGG	**33.27**	7.8541	54	4901	65.69	2.6786	134	-

Finalmente, se hizo una prueba con las imágenes originales (sin preprocesar) como set de entrenamiento, y ver como se comporta el modelo con el set de test. Esto con las dos clasificadores que tuvieron un mejor comportamiento. Los resultados se pueden visualizar en la siguiente tabla:

TABLA 2.6: Resultado final de los modelos con mejor comportamiento [18].

Model	Success rate	Average error	Epoch	Time(s/epoch)
AlexNetOWTBn	99.49	**0.0174**	121	6647
VGG	**99.53**	0.0223	67	7034

Para visualizar de mejor manera los resultados, tecnologías y parámetros de interés sobre cada investigación estudiada, se presenta una tabla que resume los aspectos más importantes. Se destaca el nivel alto de desempeño en Deep Learning en la investigación de K. P. Ferentinos.

TABLA 2.7: Tabla resumen que muestra los principales trabajos en el estado del arte.

Autores	Año	Algoritmo	Tasa de clasificación	Espectro	N° de imágenes	Orientación	Disponibilidad BDD
Meng-Han, H., Zhao, Y. & Zhai, G.	2017	Estimated error reduction	0.87	Hiperespectral	557 (solo 9 de train)	Daño en arándano	No
Leiva-Valenzuela, G. & Aguilera, J.	2013	SVM	0.89	Espectro visible y escala de grises	Desconocido	Detección de enfermedades en el arándano	No
Kuzy J., Jiang, Y. & Li, C.	2018	Logistic Regression	0.89	Termal	378 vídeos (17 segundos c/u)	Detección de moretones (golpes) en arándanos	No
Ferentinos, K.	2018	CNN (VGG)	**0.9984**	Espectro visible	87.848	Detección de enfermedades en plantas	No

CAPÍTULO 3. HIPÓTESIS Y OBJETIVOS

3.1 HIPÓTESIS

- Se estima que es posible capturar imágenes en espectros infrarrojos para determinar la calidad del arándano.
- Se espera que las imágenes capturadas en el espectro SWIR aporten información complementaria a la calidad de los arándanos.
- Se estima que las técnicas de Deep Learning permitirán extraer características relevantes de la calidad del arándano.

3.2 OBJETIVOS

3.2.1 Objetivo general

Diseñar un sistema de clasificación automática de arándanos basado en Deep Learning utilizando imágenes en espectro SWIR.

3.2.2 Objetivos específicos

- **OE01:** Capturar, procesar y etiquetar imágenes positivas y negativas de arándanos utilizando una cámara SWIR, y determinar las características más relevantes.

- **OE02:** Diseñar, implementar y evaluar una característica que permita determinar la calidad de los arándanos.

- **OE03:** Implementar la Red Neuronal Convolucional (CNN) de mejor desempeño para la extracción de características del arándano.

CAPÍTULO 4. METODOLOGÍA

4.1 METODOLOGÍAS

Para realizar los objetivos propuestos y lograr éxito en ello, es necesario hacer un texto detallado sobre los planes de cómo se realizará cada proceso, esto es conocido como metodología. A continuación las metodologías de cada objetivo.

4.1.1 Metodología - 01

Para la captura de imágenes en espectro SWIR se utilizó la cámara Xenics Bobcat-320, algunas de las especificaciones se encuentran en la siguiente tabla:

TABLA 4.1: Especificaciones de matriz de la cámara Xenics Bobcat-320. Fuente: `http://www.xenics.com/es/camera/bobcat-320`

Especificaciones de matriz y cámara	Bobcat-320
Tipo de matriz	InGaAs
Banda espectral	0.9 μm hasta 1.7 μm
Resolución	320 x 256
Tamaño de pixel	20 μm
Full well capacity	125 k e-
Consumo de energía	2.8 W
Fuente de alimentación	12 V
Temperatura ambiente de operabilidad	-40ºC a 70ºC
Dimensiones (W x H x L mm^3)	55 x 55 x 72

Para tener una apreciación gráfica sobre la cámara, a continuación se presentan imágenes de Xenics Bobcat-320.

FIGURA 4.1: Vista frontal de la cámara Xenics Bobcat-320. Fuente: `http://www.xenics.com/es/camera/bobcat-320`

FIGURA 4.2: Vista trasera de la cámara Xenics Bobcat-320. Fuente: `http://www.xenics.com/es/camera/bobcat-320`

FIGURA 4.3: Vista lateral de la cámara Xenics Bobcat-320. Fuente: `http://www.xenics.com/es/camera/bobcat-320`

Los arándanos de mala calidad (imágenes negativas) fueron proporcionados por la empresa Berries Bio-Bio, la mala calidad de los arándanos capturados fue en base al criterio de los trabajadores con experiencia del lugar.

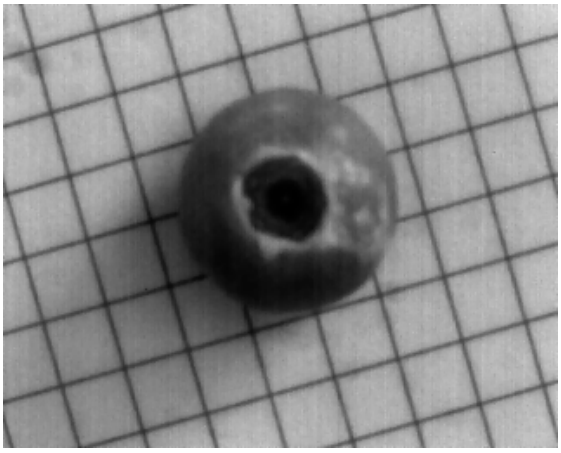


FIGURA 4.4: Imagen de arándano considerado de mala calidad. Imagen capturada con Xenics Bobcat-320. Fuente: Elaboración propia.

Por otra parte, los arándanos de buena calidad fueron comprados en diferentes supermercados de distintas marcas, la mayoría con fecha de elaboración el día anterior a la captura (excepto 2.5 cajas de 300gr que tenían 3 días de diferencia entre la captura y la elaboración). Cabe destacar que habían arándanos de mala calidad también en estas cajas, los cuales simplemente fueron desechados.

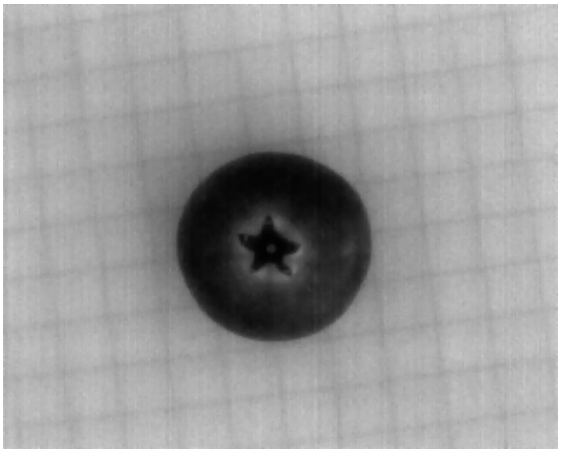


FIGURA 4.5: Imagen de arándano considerado de buena calidad. Imagen capturada con Xenics Bobcat-320. Fuente: Elaboración propia.

La captura consta de grabar, con la cámara mencionada anteriormente, los arándanos por ambos lados (lado del tallo y su contraparte), por aproximadamente dos segundos cada lado del arándano. Operando a un promedio de 30 ”frames per second”(FPS), se logran obtener 120 imágenes por arándano de forma aproximada (60 por cada lado del arándano, en un aproximado de dos segundos por lado). La cámara se sitúa a una altura de 35 cm aproximadamente (altura

de un tarro de pintura) como se ve en la figura 4.6.

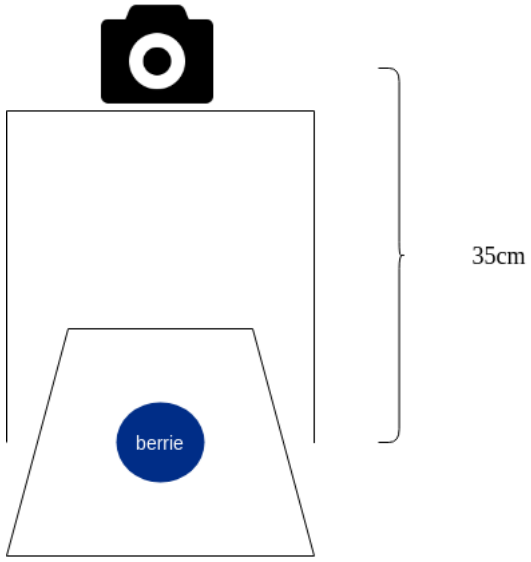


FIGURA 4.6: Esquema del montaje de la captura de imágenes. Fuente: Elaboración propia.

. En la captura de arándanos de buena calidad fue necesario usar un foco para evitar los cambios de sombra (dado que la captura fue hecha en la mañana y la sombra producida por la posición del sol cambiaba abruptamente). Mientras que la captura de los arándanos de mala calidad fue durante la tarde a luz ambiente.

Por lo tanto, la base de datos se construye en base a estas imágenes capturadas, luego filtradas (separando las que no aportan información, cómo por ejemplo la que se muestra en la figura 4.7). Luego es necesario marcar cada imagen en las coordenadas para ser recortadas, el código de recorte utilizado almacena las coordenadas como $x1, y1, w, h$, siendo w el ancho y h el alto del rectángulo marcado, es decir, que $y2 = y1 + h$ y $x2 = x1 + w$. Sin embargo, en la base de datos se guardarán las coordenadas bajo el formato x1, y1, x2, y2.

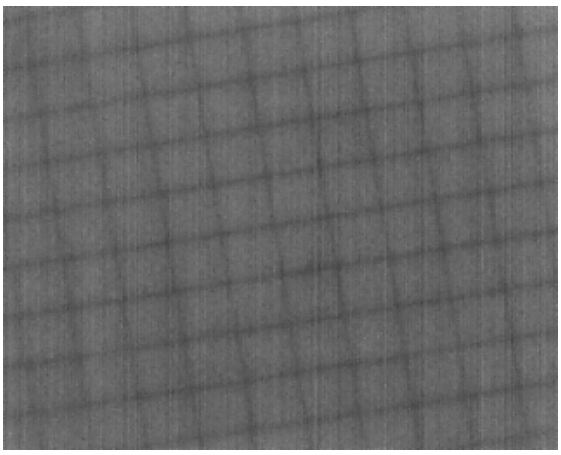


FIGURA 4.7: Imagen sin arándano, que puede considerarse para una posible tercera clase ”No arándano”. Imagen capturada con Xenics Bobcat-320. Fuente: Elaboración propia.

La base de datos de imágenes estará en constante cambio a través de los meses, agregando cada vez más datos. El formato se puede visualizar en la figura 4.8.

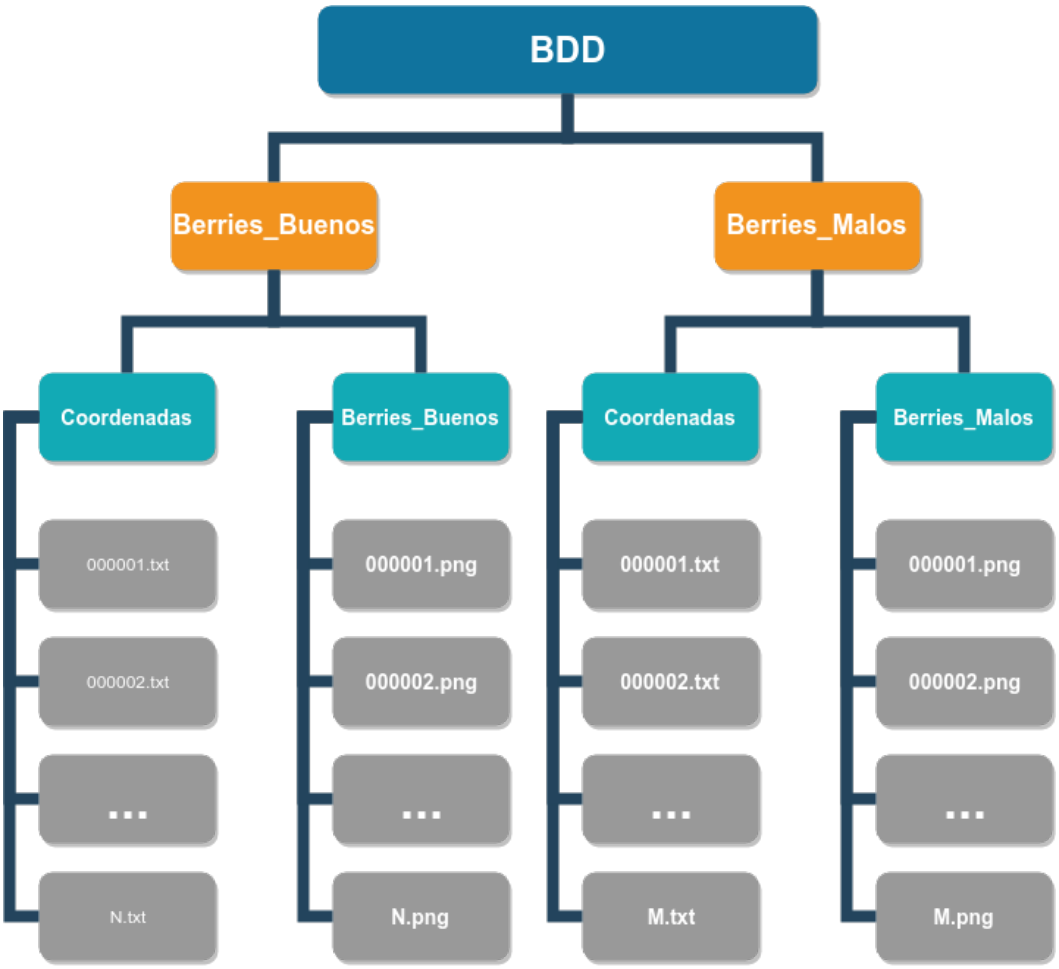


FIGURA 4.8: Organización de archivos de la BDD de imágenes. Fuente: Elaboración propia.

4.1.2 Metodología - 02

Se buscará la forma de aprovechar la información almacenada en cada una de las imágenes, es decir aprovechar la intensidad de los píxeles para realizar el proyecto. En caso de que esto no sea suficiente, es posible que sea necesario resaltar la información ya almacenada, procesando la imagen. Para esto es necesario utilizar una herramienta que permita destacar texturas, o extraer histogramas, histogramas con segmentación de imágenes, o extracción de espectrogramas de energía de la imagen.
Dado la naturaleza del problema, y además la utilidad vista en la literatura, se realizará un rescatado de texturas de las imágenes, lo cual está explicado en los capítulos 5 y 6. Con este trabajo realizado, las imágenes estarán preparas para servir como entrada al proceso que se realizará durante la metodología 03 descrita a continuación.

4.1.3 Metodología - 03

Se evaluarán distintos métodos para entrenar los modelos, ya sea From Scratch o por Transfer Learning. Siendo From Scratch desde cero, es decir, se entrenan los modelos sin extraer características antes, mientras que Transfer Learning se utilizan modelos ya entrenados para extraer características y así pasar esto a nuestro modelo para obtener mejores resultados, idealmente. Ambos conceptos explicados de mejor forma en el capítulo 5. En detección de enfermedades asociadas a plantas, se han obtenido buenos resultados con CNN: VGG, arquitectura con la cual se entrenará el modelo.
A modo preliminar, se utilizarán experimentos con la arquitectura CNN: LeNet, con el objetivo de familiarizar los conceptos de Deep Learning con redes más pequeñas y menos complejas (Shallow Net), conceptos explicados en el capítulo 5.La evaluación de los distintos métodos considerarán una grilla dónde se tomará nota sobre cada experimento asociado a cada método, con sus respectivos resultados y parámetros con los cuales fueron ejecutados. En caso de no llegar a resultados concluyentes, será necesario el uso de otra red pequeña como Smaller VGG. Para entrenar los diferentes modelos es necesario realizar una serie de pasos, dónde se debe

procesar el conjunto de imágenes, de tal forma que su organización no afecte al desempeño del modelo. También se aplicará un algoritmo de Data augmentation, explicado en el capítulo 5. Estas etapas pueden se visualizadas en la figura 4.9.

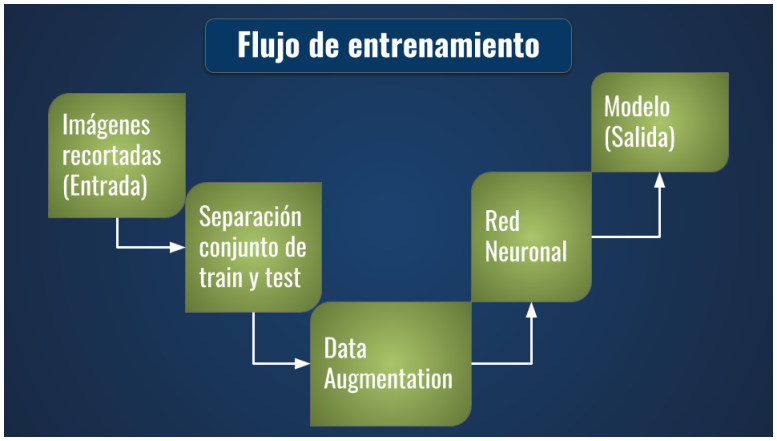


FIGURA 4.9: Flujo de la realización de experimentos. Fuente: Elaboración propia.

Se automatizará la realización de experimentos y así poder realizar varios de estos en una sola ejecución, realizando una eficiencia en el tiempo de entrenamiento y también aprovechar los momentos de noche o madrugada donde los computadores no están siendo utilizados.

CAPÍTULO 5. MARCO TEÓRICO

5.1 MACHINE LEARNING

Machine learning es un área de estudio en ciencias de la computación orientado a la inteligencia artificial. Estudia la creación de sistemas que sean capaces de aprender automáticamente y así poder reconocer patrones, predecir futuros comportamientos y/o escenarios, con el fin de tomar decisiones [20], [21].

Existen algoritmos o conjuntos de algoritmos que ayudan a resolver distintos problemas, los cuales tienen como base diferentes comportamientos matemáticos. Algunos funcionan a modo de agrupamiento, mientras que otros en base a modelos probabilísticos o incluso en base a dimensiones. Por lo tanto, dependiendo del modelo es como se va a discernir por sobre los elementos nuevos que se le presenten al sistema entrenado.

5.2 DEEP LEARNING

Deep Learning o aprendizaje profundo es un conjunto de algoritmos que toman como base las técnicas de machine learning. Utilizado en el área de visión computacional, su característica principal es que es procesado en la tarjeta gráfica (GPU), por lo tanto es necesario tener acceso a las GPUs modernas.

Utiliza cascadas de múltiples capas de procesamiento no lineal, para la extracción de características, reconocimiento de patrones o clasificación. Esto se puede lograr a través de aprendizaje supervisado o no supervisado, conceptos explicados en 5.3. Cada capa utiliza como entrada la salida de la capa anterior [22]. Existen varias arquitecturas que trabajan como Deep Learning,

siendo las Convolutional Neural Networks (CNN) las más comunes, las cuales tratan de simular el comportamiento neural de un cerebro biológico a través de nodos interconectados de forma similar.

5.2.1 Convolutional Neural Networks

Las arquitecturas CNN, son utilizadas principalmente en el campo de visión artificial para el análisis y procesamiento de imágenes. Modelando en base a pequeñas cantidades de información, las cuales se combinan en etapas (capas) posteriores. Esta tipo de redes por lo general está compuesta por tres tipos de capas, una de convolución, otra de reducción (pooling) y una capa de clasificación, que nos dará el resultado final de la red al estar completamente conectada. Por lo tanto, al tener esta capa de convolución es lo que la hace diferente al resto de redes neuronales, al tener un comportamiento tridimensional.

En la primera capa, la convolucional es dónde se aplican filtros de menor tamaño que la imagen original para resaltar características en la región donde se aplica el filtro. La convolución es capaz de trabajar con entradas de distinto tamaño, lo cual puede ser usado a favor. Luego de la capa de convolución por lo general se aplica la capa de pooling, la cual reduce el las dimensiones de la entrada para la próxima convolución. Finalmente, al final se utilizan capas conectadas de forma unidimensional [23], [24], como una ultima capa que contiene el resultado de la red, un ejemplo se puede visualizar en la figura 5.1.

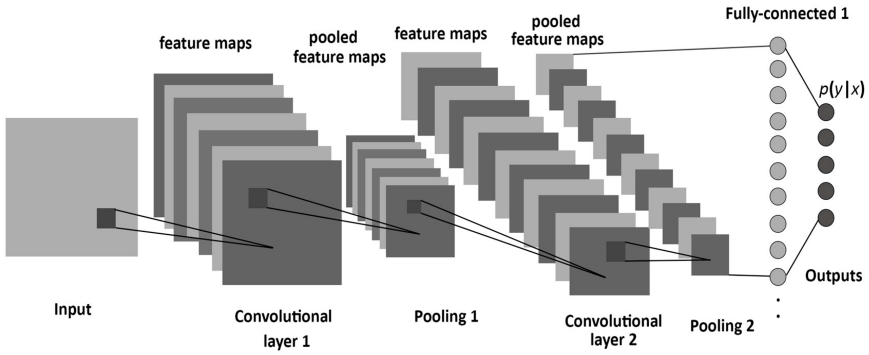


FIGURA 5.1: Imagen ejemplo de una red CNN [24].

5.3 SUPERVISED LEARNING Y UNSUPERVISED LEARNING

Supervised Learning o Aprendizaje Supervisado, y Unsupervised Learning o Aprendizaje no Supervisado, son técnicas ocupadas tanto en Machine Learning como en Deep Learning, dónde Supervised Learning se encarga de entrenar el modelo de tal forma, que se le presentan los datos con una etiqueta, dónde al algoritmo se le evidencia cual es la clase positiva y cual es la negativa, en este tipo de entrenamiento es donde predominan los modelos en base a dimensiones o geométricos, modelos que separan linealmente (o en base a una función matemática que lo permita) las diferentes clases. Mientras que a diferencia en Unsupervised learning los datos se le entregan con una etiqueta que explicite a qué pertenece cada elemento, aquí es dónde predominan los modelos orientados a la agrupación (clustering).

5.4 REDES NEURONALES UTILIZADAS

Durante el desarrollo de este proyecto se utilizaron dos redes neuronales principalmente, y su uso está explicado en detalle en el capitulo 6.

5.4.1 LeNet

Esta red neuronal convolucional (CNN) que se considera como un buen punto de partida para relacionarse ellas. Fue utilizada en 1998 para el reconocimiento de números escritor a mano [31]. La arquitectura utilizada [32] consiste principalmente de 3 sets, cada set consiste en una etapa de convolución, otra de activación y finalmente otra de pooling. Dónde la etapa de convolución consiste en aplicar cierta cantidad de filtros con un tamaño especifico para resaltar ciertas características de la imagen, los filtros dependen de la operación matemática (convolución) que exista por detrás de esta fase. La siguiente etapa es una función de activación, la cual se utiliza en las redes neuronales para transformar el discriminador linear en una neurona o unidad, dónde la función utilizada en esta investigación corresponde a ReLu (rectified linear unit) [33] que

en palabras simples transforma los valores negativos a cero, dónde su fórmula matemática se estipula de la siguiente manera f(x) = Max(0, x) , además esta función se utiliza dada su eficiencia en el cálculo y sin mayores impactos en la generalización del modelo. Y finalmente una función de pooling, la cual consiste principalmente en reducir las dimensiones espaciales del volumen de entrada para la siguiente capa, esto tiene un impacto beneficioso (aunque no lo parezca a simple vista) dado que la sobrecarga para las siguientes capas va disminuyendo y a su vez trabaja de tal forma que disminuye la probabilidad de overfitting o sobreajuste (sobreentrenamiento). La función utilizada como pooling en esta investigación es Maxpooling, la cual, divide la imagen de entrada en distintas regiones rectangulares y en base al valor de estos, guarda los de mayor valor cómo la nueva imagen.

Se varió su arquitectura, aumentando la cantidad de capas de convoluciones, activaciones y pooling, en una cada una, para lograr mejores resultados.

5.4.2 Smaller VGG

Esta arquitectura es una versión acotada de la red VGG-16 [34], que en este caso consta de 5 capas convolucionales que a diferencia de la LeNet anterior que tenía 4, utiliza 6 capas de RelU y una de softmax característica de una red VGG, en contraste de las 5 ReLu utilizadas en LeNet. También, se utilizan 3 capas pooling, pero en LeNet se utilizan 4, esto es porque en LeNet se utiliza una por cada convolución. Además se agregan 2 funciones al modelo, una función de dropout, que en simples palabras elimina cierto porcentaje de las neuronas totales, esto se hace con el fin de combatir el overfitting o sobreajuste, el cual, en este caso se utiliza 4 veces, donde 3 de ellas son de 0.25 y la última de 0.5. La otra función es BatchNormalization, el cual consiste en normalizar los valores obtenidos de la función de activación, con el fin de que los números resultantes, no sean ni muy grandes ni muy pequeños. Esta última se utiliza 6 veces, una por cada ReLU.

5.5 LIBRERÍAS Y FRAMEWORKS

Para el desarrollo de todo lo que engloba esta investigación fue necesario el uso de distintas librerías y Frameworks que facilitan el trabajo a fin. Evidentemente hay unos mas cruciales que otros que es pertinente explicar su uso en esta investigación.

5.5.1 OpenCV

Utilizado para el procesamiento general de imágenes, es una librería [27] con un amplio catalogo de herramientas de trabajo. Utilizado durante toda la investigación, para el recorte de imágenes, para pre-procesamiento antes del entrenamiento (excepto Data Augmentation) y guardado de imágenes con conservación de ciertas características de estas. Es necesario destacar que las imágenes procesadas por OpenCV son de tipo BGR, a menos que se utilicen los comandos necesarios para procesarlas en RGB.

5.5.2 Matplotlib

Esta librería [28] fue utilizada en su gran parte para generar los gráficos de las curvas de aprendizaje y guardado de imágenes en RGB (Heat Maps específicamente, explicados en el capítulo 5.6.1). A diferencia de OpenCV esta librería maneja las imágenes en RGB, además de tener compatibilidad con Python y una semejanza con MATLAB, siendo su fuerte el trazado 2D, por lo cual facilita cualquier tipo de trabajo con histogramas, gráficos, espectros, entre otras utilidades.

5.5.3 Keras & Tensorflow

Keras [29] es una API (application programming interface) de redes neuronales de alto nivel, escrita en python capaz de ejecutarse con Tensorflow, CNTK y Theano. Se utiliza frecuentemente para el desarrollo de investigaciones dada su modularidad, que permite una configuración a distintos niveles y con pocas restricciones, y extensabilidad, que permite añadir nuevos módulos sin problemas de compatibilidad. Además, soporta redes convolucionales y recurrentes, además de tener la capacidad de ejecutarse en CPU y en GPU.

Por otra parte Tensorflow [30] es una librería open source para el calculo numérico utilizando gráficos de flujos de datos. Desarrollada por investigadores e ingenieros que trabajan en el equipo Google Brain en la organización de investigación de Machine Learning. Tiene una arquitectura flexible capaz de ejecutarse en uno o más unidades de procesamiento (CPUs tanto como GPUs).

Ambas herramientas son totalmente compatibles con Python, por lo cual, son compatibles con las tecnologías descritas anteriormente.

5.6 ALGORITMOS ÚTILES

5.6.1 Grad-CAM

Grad-CAM [35] es un algoritmo que es capaz de reflejar el comportamiento de una red a la hora de clasificar utilizando un Heatmap (mapa de calor). Una vez la red entrenada, basta con tener el modelo y la imagen que se desea clasificar, este algoritmo procesa la imagen por cada una de las capas del modelo destacando con tonalidades rojizas la zona de interés para el clasificador. La escala oscila entre un azul hasta un rojo, ambos intensos, como puede ser visualizada en la figura 5.2, siendo el rojo mas intenso el lugar de mayor interés y así va decreciendo a medida que el clasificador y tornándose completamente azul en los píxeles que considera menos importantes para clasificar. Este algoritmo tiene una limitación y es que se pueden generar las imágenes de calor sólo en base al escalado de imagen utilizado en el entrenamiento, por lo que es necesario

redimensionar las imágenes y con imágenes tan pequeñas no se puede visualizar de muy buena forma el comportamiento del modelo, como se podrá ver en el capitulo 7.

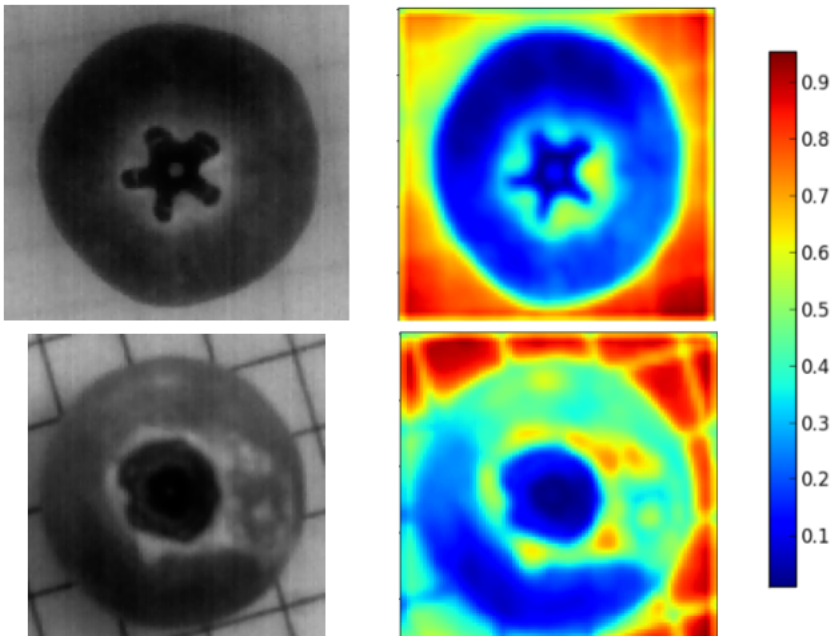


FIGURA 5.2: Imagen comparativa de una imagen original con su heatmap a partir de un modelo en los primeros experimentos con LeNet, junto a la escala utilizada.

5.6.2 Algoritmo BSIF

BSIF (binarized statistical image features) es un algoritmo tiene la característica de resaltar texturas de una imagen en base a filtros de tamaños determinados. Fue utilizado un estudio [36] en el cual se logra predecir el color de los ojos con imágenes en espectro Near Infrared (NIR), utilizando filtros de BSIF para resaltar características importantes del ojo (basados en su textura). Cada filtro tiene cierta dimensión y bits asociados, los cuales frente a la interacción con la imagen que se le quiera aplicar el filtro, va a generar una nueva imagen, destacando texturas que no se destacan en la intensidad de los píxeles. Dada la similitud en el espectro de imágenes se tomó la decisión de ver los resultados en imágenes de arándanos en SWIR. Cabe destacar que este algoritmo está implementado en Matlab, para lo cual se utilizó GNU Octave, una opción open source para el procesamiento de algoritmos hechos en Matlab.

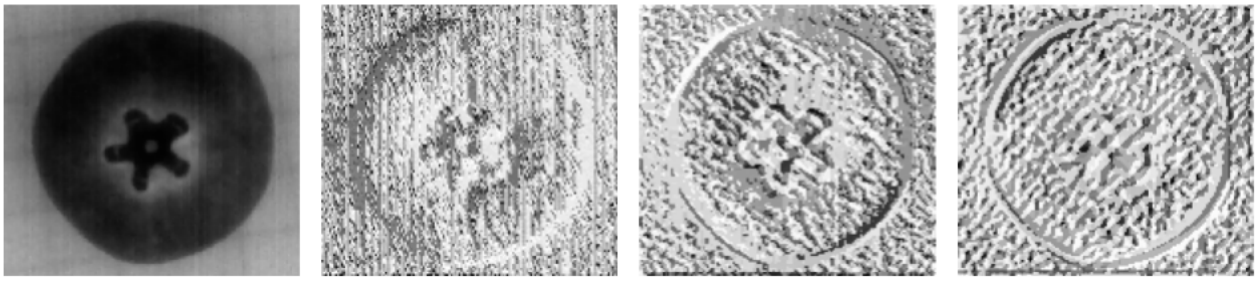


FIGURA 5.3: Imagen comparativa entre los distintos filtros de BSIF utilizados. Imagen original, filtro BSIF de 3x3, 5x5 y 7x7, respectivamente.

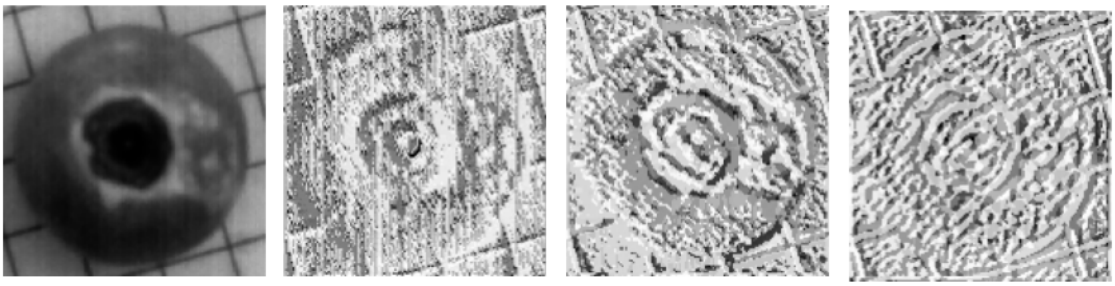


FIGURA 5.4: Imagen comparativa entre los distintos filtros de BSIF utilizados. Imagen original, filtro BSIF de 3x3, 5x5 y 7x7, respectivamente.

5.7 MÉTRICAS DE DESEMPEÑO

Utilizado en Supervised Learning, consiste en evaluar el desempeño de cualquier modelo, existen cálculos asociados que tienen como base una matriz de confusión. Una matriz de confusión se construye en base a los resultados que arroja un modelo con el fin de demostrar su desempeño de clasificación. Por una parte se tiene el ”ground truthrepresentado en las filas y por otra parte está la predicción representado en las columnas, en un ejemplo de dos clases (una positiva y negativa), como se muestra en la tabla 5.1.

TABLA 5.1: Matriz de confusión.

		Predicción	
		positivo	**negativo**
Valor Real	**positivo**	True Positive	False Negative
	negativo	False Positive	True Negative

Donde:

- True Positive: Es el número de elementos de la clase positiva, clasificados exitosamente como positivos.
- False Positive: Es el número de elementos de la clase negativa clasificados erróneamente como clase positiva.
- False Negative: Es el número de elementos de la clase positiva clasificados erróneamente como clase negativa.
- True Negative: Es el número de elementos de la clase negativa, clasificados exitosamente como negativos.

5.7.1 Accuracy

Accuracy muestra los elementos clasificados de manera correcta frente al total de elementos clasificados. En caso de tener un 100 % de accuracy, es porque el modelo clasificó exitosamente todos los elementos de manera correcta.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.1}$$

5.7.2 Precision

La precision es un cálculo basado en la relación de elementos de la clase positiva clasificados como tal, frente a la cantidad total de clasificaciones positivas.

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

5.7.3 Sensitivity

Sensitivity, recall o TPR (true positive rate) es la relación de elementos de la clase positiva clasificados como tal, frente a la cantidad total de elementos de la clase positiva. Es decir, muestra el desempeño al momento de clasificar la clase positiva.

$$Sensitivity = \frac{TP}{TP + FN} \tag{5.3}$$

5.7.4 Specificity

Specificity, o TNR (true negative rate) es la relación de elementos de la clase negativa que se clasifican como tal, frente a la cantidad total de elementos de la clase negativa. Es decir, muestra el desempeño al momento de clasificar la clase negativa.

$$Specificity = \frac{TN}{TN + FP} \tag{5.4}$$

5.7.5 Fallout

Fallout o FPR (false positive rate) es la relación de elementos de la clase negativa que fueron clasificados de forma errónea como clase positiva, frente a la cantidad total de elementos de la clase negativa.

$$Fallout = \frac{FP}{FP + TN} \tag{5.5}$$

5.7.6 F-score

F-score es la media armónica de los valores Precision y Sensitivity. Utilizado para evaluar la exactitud de una prueba de un modelo, tomando en cuenta los valores anteriormente mencionados. Un valor para β = 1 es dónde tanto Precision como Senitivity están en perfecta ponderación, de otra forma la ponderación se tiende a desbalancear a favor de la Precision o Sensitivity.

$$F_\beta = \frac{(1 + \beta^2) \times Presicion \times Sensitivity}{\beta^2 \times Presicion \times Sensitivity} \tag{5.6}$$

La evaluación de F_1 Score alcanza su mejor valor en 1 y su peor es alcanzado en 0.

$$F_1 = \frac{2 \times Presicion \times Sensitivity}{Presicion \times Sensitivity} \tag{5.7}$$

5.8 DATA AUGMENTATION

Data Augmentation es un algoritmo usado a menudo en los modelos de aprendizaje, que cumple la función de aumentar la cantidad de datos de algún dataset, con el fin de mejorar el modelo a entrenar [25].

Usado principalmente en el campo de visión artificial. Los mejores modelos, por lo general, tienen un dataset bastante grande (decenas de miles de datos o incluso millones) y a menudo es difícil tener acceso a una base de datos tan grande. Este algoritmo es capaz de utilizar diferentes métodos para lograr su objetivo y hacer el modelo más robusto. Es necesario ser cuidadoso al momento de hacer las variaciones con este algoritmo, pues algunas funciones, según sus parámetros, pueden dejar fuera datos importantes o regiones de interés, por lo que estos parámetros dependen totalmente de qué es lo que se espera que reconozca el modelo a entrenar. Algunas de las técnicas más básicas son:

- Flip: Esta técnica consiste principalmente en crear una segunda imagen (o más) como efecto espejo. También se pueden lograr flip verticales, que se logran rotando la imagen en 180° y un flip horizontal.

- Rotation: Cómo se dijo en el punto anterior, la rotación consiste en girar la imagen bajo ciertos grados de inclinación. Dependiendo de las imágenes originales, es que tan efectivo puede ser rotar en ciertos grados.

- Scale: Escalar la imagen puede dar como resultado la imagen original de forma alargada o más pequeña. Prácticamente es hacer un efecto de ”zoom”, dónde este varía según el parámetro scale.

- Crop: Consiste en recortar aleatoriamente la imagen original y así sacar varias muestras de una misma imagen. Luego estas imágenes se redimensionan al tamaño de la imagen original.

- Traslation: La traslación consiste en variar los ejes x e y para trasladar el área de interés u objeto que se quiere estudiar a distintas posiciones. Esta técnica puede ayudar a una CNN a buscar los patrones a lo largo de toda la imagen.

- Gaussian Noise: Funciona como distorsionador de frecuencias, añadiendo ”ruido.ª la imagen original, dónde los valores del ruido que puede tomar está definido por la distribución Gaussiana [26]. Su función matemática es la siguiente:

$$P(g) = \sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{(g-\mu)^2}{2\sigma^2}} \tag{5.8}$$

  Dónde g = valor gris, σ = desviación estándar y μ = media [26].

A continuación se muestran dos ejemplos con una función de Data Augmentation aplicada a dos imágenes, una de un arándano de buena calidad y otra con un arándano de mala calidad. Generándose así imágenes con ciertas rotaciones y escalados, cuidando de que parte del arándano no quede fuera de la imagen.

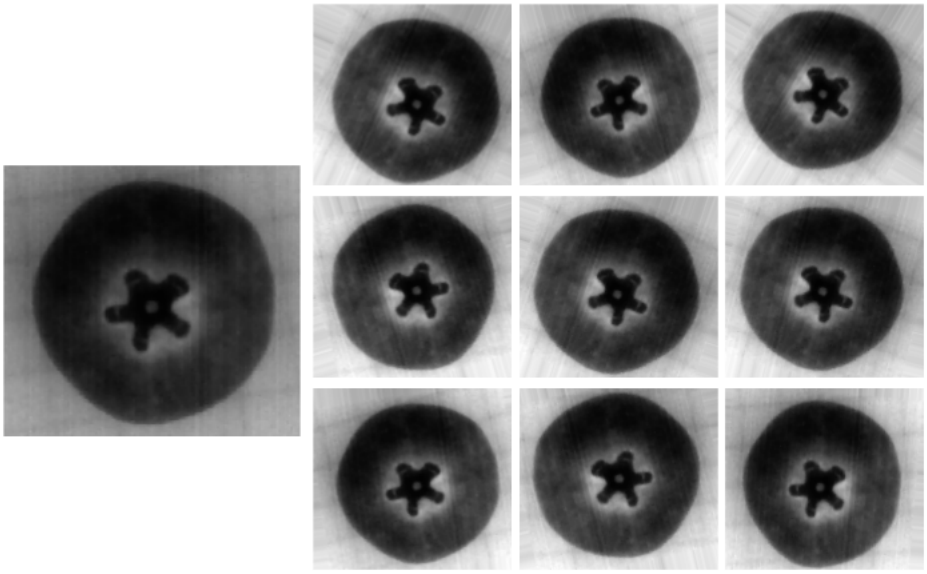


FIGURA 5.5: Imagen ejemplo de Data Augmentation con arándanos de buena calidad.

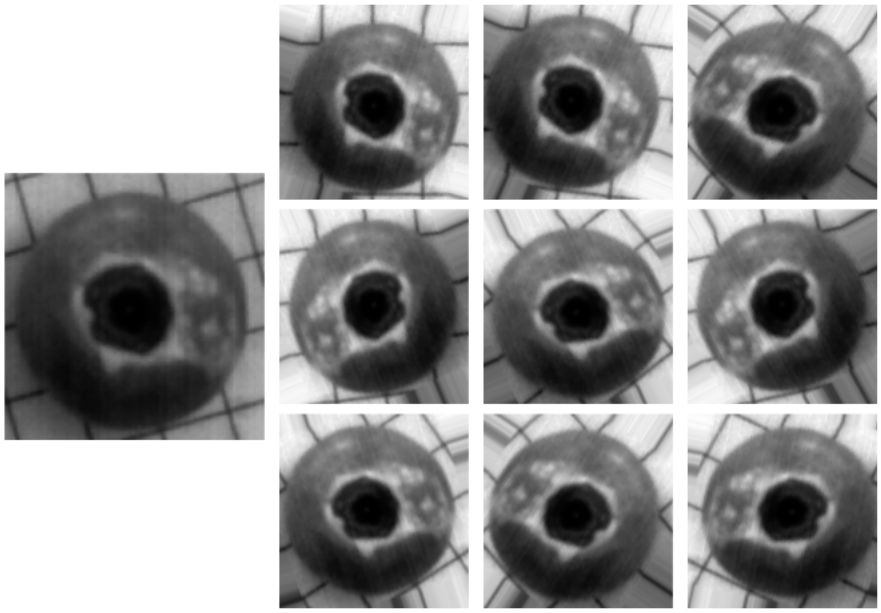


FIGURA 5.6: Imagen ejemplo de Data Augmentation con arándanos de mala calidad.

CAPÍTULO 6. EXPERIMENTOS

Cómo se mencionó en el capítulo 4 se realizaron experimentos para conocer cómo funciona las redes neuronales. Estos experimentos preliminares se caracterizan porque el pre-procesamiento del dataset de entrada estaba siendo realizado de tal forma, que las imágenes estaban en espacio de colores, algunas pruebas con diferencias entre los bits del espacio de colores. Estas pruebas fueron útiles para dar cuenta de la necesidad de que los datos utilizados para el entrenamiento deben estar en condiciones iguales. Otra característica importante es que en la primera fase de estos experimentos, el dataset era sujeto a una función random, dónde se mezclaban las imágenes, y por lo tanto repetían imágenes de train en test.

Dados los resultados que presentan un sesgo en test (como se verá en el capítulo 7), también se optó por utilizar una CNN diferente pero que también fuese pequeña para comparar los comportamientos y resultados de dichas redes y modelos resultantes. Esta red fue Smaller VGG la cual fue explicada en el capítulo 5.

Con los parámetros ya definidos, entonces, se comienzan a trabajar los siguientes cuatro experimentos, donde los primeros dos están enfocados en la intensidad de los píxeles, mientras que los últimos dos experimentos están hechos en base a la textura de estos.

6.1 EXPERIMENTO 1 (LENET)

El primer experimento llevado a cabo, con la red LeNet se decide optar por una variabilidad en el tamaño de las imágenes y Dense. Las imágenes sólo se trabajan en base a la intensidad de píxeles, utilizando escala de grises y 8 bits. Se utilizan filtros pequeños en las capas de convolución (5x5), dado el tamaño de las imágenes. Estos experimentos son cruciales, dado que reflejan que el sesgo del modelo persiste a pesar de que los problemas de entrada

en el dataset fueron arreglados. Por lo tanto, se comenzó a pensar en alguna herramienta que fuese capaz de visualizar el comportamiento del modelo. Esto se ratificó con los resultados de la sección 6.2 Experimento 2 (Smaller VGG).

6.2 EXPERIMENTO 2 (SMALLER VGG)

El experimento 2 se caracteriza por la necesidad de saber si el comportamiento de los resultados vistos en 6.1 son independientes de la Red neuronal. Esta red neuronal convolucional es más compleja y profunda que LeNet como fue explicada en el capítulo 5. Fue en este momento dónde se comenzó a aplicar el algoritmo Grad-CAM para visualizar el comportamiento de los modelos resultantes, esto también fue aplicado para los modelos de la sección 6.1 Experimento 1 (LeNet). Desde este punto en adelante se decide utilizar una vía alternativa para el entrenamiento, buscando una forma de realizar una variabilidad en la entrada de imágenes. Es por esto que se utilizó el algoritmo BSIF para tener un enfoque en texturas en vez de las intensidades de los píxeles, algoritmo explicado en el capítulo 5. El tamaño de los filtros BSIF utilizados son de 3x3, 5x5 y 7x7, todos de 8 bits. El hecho de que el problema del sesgo de la red sea recurrente, generó un retraso en el avance por los experimentos, tratando de buscas las soluciones propuestas hasta el momento, este sesgo además es provocado dadas las diferencias en las imágenes de arándanos de buenas calidad versus las de mala calidad, dado que el fondo de estas imágenes eran diferente, presentaban un fondo prácticamente en blanco y el otro con las líneas del cuaderno bien definidas, respectivamente.

6.3 EXPERIMENTO 3 (BSIF - LENET)

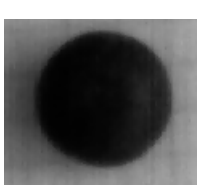
Este experimento consiste en realizar las pruebas realizadas en 6.1 experimento 1 (LeNet), pero esta vez con diferentes filtros en dataset de entrada. Por lo tanto hay un set de pruebas para el filtro de 3x3, otro para el filtro de 5x5 y un último para el filtro de 7x7. Por lo tanto, fue necesario realizar tres copias a la base de datos y aplicar en cada una de ellas un filtro distinto

de BSIF para recién comenzar a realizar los experimentos.

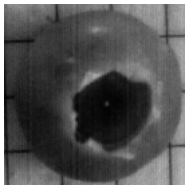
6.4 EXPERIMENTO 4 (BSIF - SMALLER VGG)

Este experimento consiste en realizar exactamente lo mismo que en 6.3 Experimento 3 (BSIF - LeNet), pero orientado a Smaller VGG. Esta vez se tomó la decisión de calcular los mapas de calor sólo para los resultados llamativos.

Todas las imágenes utilizadas para las pruebas con Grad-CAM son la registrada como ”020001.png”de la base de datos de imágenes recortadas o BSIF según corresponda. A continuación en la figuras 6.1, 6.2,6.3 y 6.4 se presentan la lista de imágenes utilizadas:

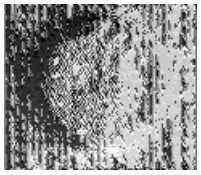


(a) Arándano de buena calidad.

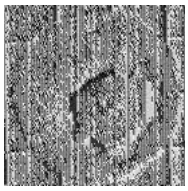


(b) Arándano de mala calidad.

FIGURA 6.1: Arándanos utilizados para Grad-CAM sin filtro BSIF.



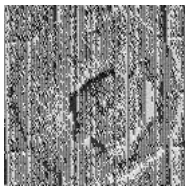
(a) Arándano de buena calidad.



(b) Arándano de mala calidad.

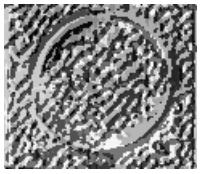
FIGURA 6.2: Arándanos utilizados para Grad-CAM con filtro BSIF 3x3.

(a) Arándano de buena calidad.



(b) Arándano de mala calidad.

FIGURA 6.3: Arándanos utilizados para Grad-CAM con filtro BSIF 5x5.



(a) Arándano de buena calidad.

(b) Arándano de mala calidad.

FIGURA 6.4: Arándanos utilizados para Grad-CAM con filtro BSIF 7x7.

CAPÍTULO 7. RESULTADOS

7.1 RESULTADOS

Los primeros experimentos se comenzaron a realizar cómo se mencionó en el capítulo 4, para entender el funcionamiento de las redes y los distintos modelos generados, en base a los distintos parámetros en conjunto a la base de datos. Es por esto que sólo se plasmarán resultados.

TABLA 7.1: Tabla de resultados de pruebas con LeNet con imágenes sin recortar (cargadas como RGB).

Experimento	Epochs	N Imagenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	Dense	Accuracy Train	Accuracy test
Experimento 1	100	20.000	$1e^{-5}$	100	64x64	40 %	20(5x5)	40(5x5)	60(5x5)	80(5x5)	512	0.8426	0.9829
Experimento 2	100	20.000	$1e^{-5}$	100	96x96	40 %	20(5x5)	40(5x5)	60(5x5)	80(5x5)	512	0.9090	0.9950
Experimento 3	100	20.000	$1e^{-5}$	100	128x128	40 %	20(5x5)	40(5x5)	60(5x5)	80(5x5)	512	0.8988	0.9930
Experimento 4	100	20.000	$1e^{-5}$	100	256x256	40 %	20(20x20)	40(15x15)	60(10x10)	80(5x5)	10240	0.9826	0.9995

Durante los experimentos vistos en 7.1, se destacan resultados con un alto Accuracy tanto de test como de train. Siendo el de test muy, muy cercano a uno. Luego se comenzó a pensar en el por qué de estos resultados tan altos, y sobretodo el patrón de tener un Accuracy de test tan alto. Por lo que se decidió repetir parte de los experimentos realizados pero con imágenes recortadas, con redimensiones menores, dado que el resultado obtenido en el experimento 1 de la tabla 7.1, es el con más posibilidades de crecer, además que las imágenes recortadas tienen un tamaño mínimo cercano al 100x100.

TABLA 7.2: Tabla de resultados de pruebas con LeNet (16bit - 8bit).

Experimento	Epochs	N Imagenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	Dense	Accuracy Train	Accuracy test
Experimento 1	100	20.000	$1e^{-5}$	100	96x96	$\approx$ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	1440	0.9998	1.0
Experimento 2	100	20.000	$1e^{-5}$	100	64x64	$\approx$ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	640	1.0	1.0
Experimento 3	100	20.000	$1e^{-5}$	100	32x32	$\approx$ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	180	0.9998	1.0
Experimento 4	100	20.000	$1e^{-5}$	100	96x96	$\approx$ 6000	20(20x20)	40(15x15)	60(10x10)	80(5x5)	1440	1.0	1.0
Experimento 5	100	20.000	$1e^{-5}$	100	64x64	$\approx$ 6000	20(20x20)	40(15x15)	60(10x10)	80(5x5)	640	1.0	1.0
Experimento 6	100	20.000	$1e^{-5}$	100	32x32	$\approx$ 6000	20(20x20)	40(15x15)	60(10x10)	80(5x5)	180	0.9997	1.0

Esta vez, en la tabla 7.2 se decidió variar cada experimento con distintos tamaños de

filtros (desde el experimento 4 al 6). Se mantuvo un Dense equivalente al 50 % en relación al flatten de salida de la red, esto se conservará para todas las pruebas posteriores, dado que antes se manejaban números constantes para esta variable y quedando gran parte de la última capa de la red fuera, con valores demasiado bajos para abarcar una parte suficiente del comportamiento real de la red neuronal. En este set de experimentos de la tabla 7.2, los valores de Accuracy son demasiado altos, pero esto tiene una explicación y que dado el formato de la imágenes utilizadas (16 bits en imágenes positivas, mientras que las negativas son de 8 bits), la información contenida en cada píxel teniendo tantas diferencias en su profundidad de tonalidades, es posible que la red haya tomado esas diferencias y haya hecho su clasificación en base a esto.

Además se arregló un problema importante del set de datos de entrenamiento, que hacía que algunas imágenes de train quedaran en test, dado que se utilizaba una función random en todo el set de datos y cómo existen varias imágenes de un mismo arándano, esto podía causar distorsión en los resultados finales. Por lo cual, se separó manualmente hasta qué imagen se consideraría para train y dejando el resto para test. Así para las dos clases, verificando que no se repliquen imágenes en los set de datos.

TABLA 7.3: Tabla de resultados de pruebas con Smaller VGG (16 bits - 8 bits).

Experimento	Epochs	N Imagenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	Dense	Accuracy Train	Accuracy test
Experimento 1	100	20.000	$1e^{-5}$	100	96x96	≈ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	1440	0.9997	1.0
Experimento 2	100	20.000	$1e^{-5}$	100	64x64	≈ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	640	0.9991	1.0
Experimento 3	100	20.000	$1e^{-5}$	100	32x32	≈ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	180	0.9980	1.0

Cómo los resultados no estaban de acuerdo a un comportamiento normal (con Accuracy de test por sobre la de train) se pensó en utilizar otra red pequeña con la cual comparar la red LeNet, obteniendo así los resultados registrados en la tabla 7.3. La cual mostró un comportamiento similar a la red LeNet en cuanto a este comportamiento anormal y resultados cercanos al 100 %. Sin embargo, ocurre el mismo efecto que en los entrenamientos de la tabla 7.2 dados los bits de cada imagen.

7.1.1 Experimento 1 (LeNet)

Cómo se mencionó en el capítulo 6 todo el dataset de entrada tiene el mismo formato, por lo tanto, ya se pueden tomar conclusiones en base a los modelos en sí y no en base a la incongruencia de la data. Sin embargo en algunos casos se presenta el mismo problema que existía antes, el sesgo del modelo (Accuracy de test superior al de train). Para verificar el comportamiento de los modelos entrenados, se utilizó cómo se mencionó anteriormente Grad-CAM.

TABLA 7.4: Tabla de resultados de pruebas con LeNet 8bit-grayscale.

Experimento	Epochs	N Imagenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	Dense	Accuracy Train	Accuracy test
Experimento 1	100	20.000	$1e^{-5}$	100	96x96	≈ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	1440	0.9838	0.9831
Experimento 2	100	20.000	$1e^{-5}$	100	64x64	≈ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	640	0.9698	0.9976
Experimento 3	100	20.000	$1e^{-5}$	100	32x32	≈ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	180	0.9367	0.9992

En la tabla 7.4 se muestran experimentos, siempre con filtros pequeños en cada capa, aunque aumentando su cantidad en cada una de ellas. Todos los parámetros utilizados en estos experimentos fueron replicados en los experimentos venideros, para saber la diferencia entre ellos.

A continuación se muestran las curvas de aprendizaje de cada una de los experimentos. Caracterizándose todas por un alto valor en el Accuracy de test en etapas muy tempranas del entrenamiento (antes de la décima época), mientras que en train, la curva se va dando siempre abajo y no sube hasta el tope sino hasta etapas muy posteriores. Esto evidencia cierta anomalía dado que las imágenes de train son diferente a las de test, donde el modelo no debería reconocer de mejor forma imágenes que jamás ha visto.

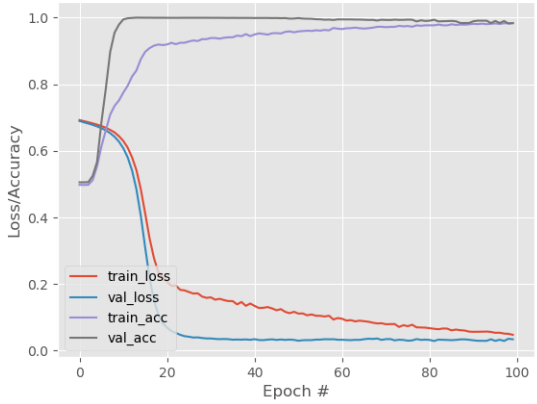
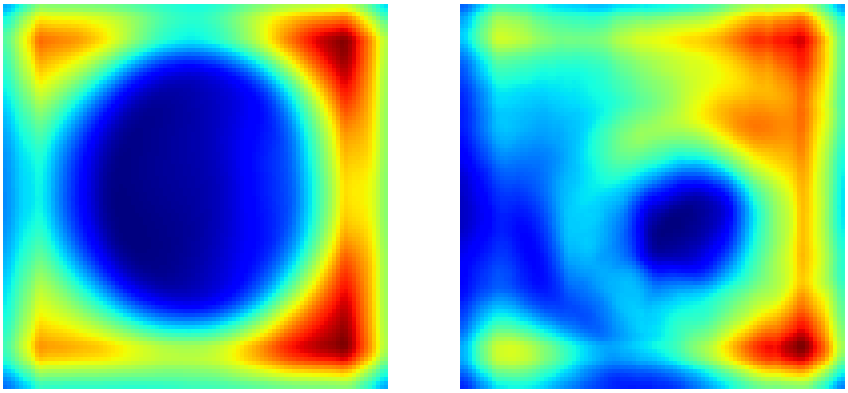


FIGURA 7.1: Curva de aprendizaje del primer experimento con la red LeNet utilizando imágenes recortadas.



(a) Heatmap de un arándano de buena calidad.

(b) Heatmap de un arándano de mala calidad.

FIGURA 7.2: Heatmap obtenido a partir del modelo correspondiente a la figura 7.1

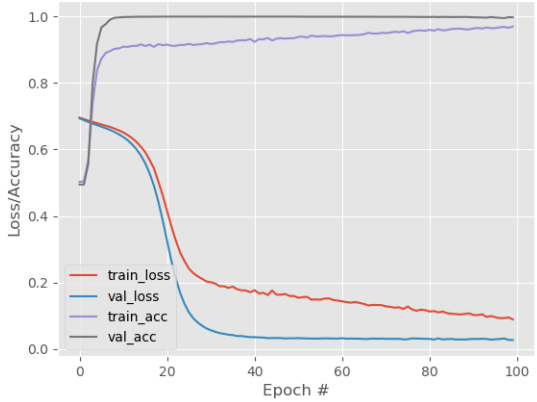
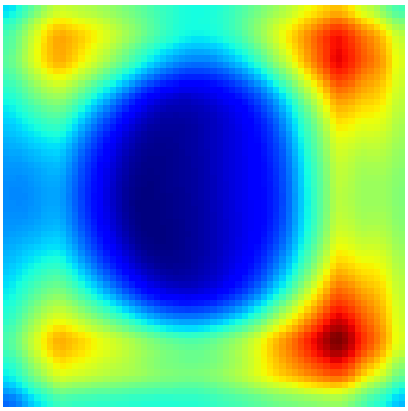
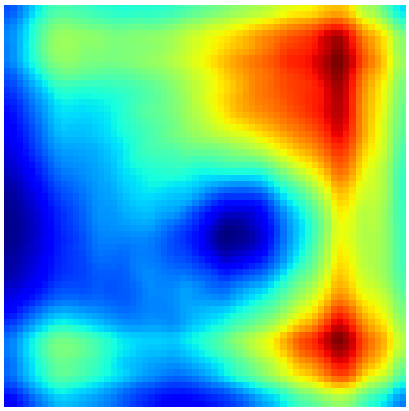


FIGURA 7.3: Curva de aprendizaje del segundo experimento con la red LeNet utilizando imágenes recortadas.



(a) Heatmap de un arándano de buena calidad.



(b) Heatmap de un arándano de mala calidad.

FIGURA 7.4: Heatmap obtenido a partir del modelo correspondiente a la figura 7.3

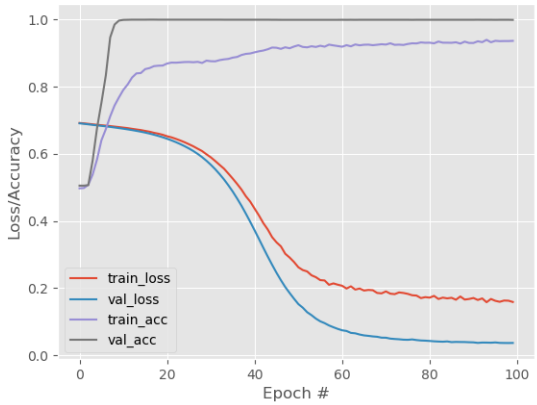
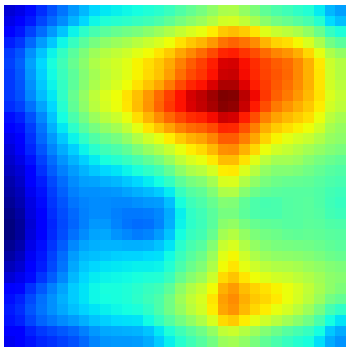
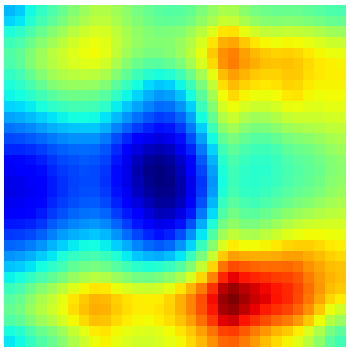


FIGURA 7.5: Curva de aprendizaje del tercer experimento con la red LeNet utilizando imágenes recortadas.



(a) Heatmap de un arándano de buena calidad.



(b) Heatmap de un arándano de mala calidad.

FIGURA 7.6: Heatmap obtenido a partir del modelo correspondiente a la figura 7.5

En los heatmaps generados 7.2 y 7.4, se evidencia que se está considerando partes

fuera del arándano para la clasificación, mientras que en el heatmap 7.6 no se logra identificar que es lo que está realmente viendo el modelo. Es por esto que el sesgo del modelo se hace evidente, dado que para el modelo es más fácil clasificar en base al patrón de fondo de la imagen que por la información contenida en la intensidad de los píxeles del arándano.

7.1.2 Experimento 2 (Smaller VGG)

En cuanto a Smaller VGG la situación es diferente sólo en los mapas de calor, de hecho los resultados son aún más altos como lo muestra la tabla 7.5. Estas pruebas tienen exactamente los mismos parámetros que las pruebas en LeNet y llegan incluso a 1.0 de Accuracy (es decir 100 %) en test.

TABLA 7.5: Tabla de resultados de pruebas con Smaller VGG 8bit-grayscale.

Experimento	Epochs	N Imagenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	c5	Dense	Accuracy Train	Accuracy test
Experimento 1	100	20.000	$1e^{-5}$	100	96x96	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	1440	0.9994	1.0
Experimento 2	100	20.000	$1e^{-5}$	100	64x64	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	640	0.9993	1.0
Experimento 3	100	20.000	$1e^{-5}$	100	32x32	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	180	0.9986	1.0

Si se observa cada una de las curvas de aprendizaje de las pruebas con Smaller VGG, es decir, las curvas 7.7, 7.9 y 7.11, todas logran sus resultados cerca de la época 4 o 5 en test, mientras que en train estos resultados son alcanzados en etapas posteriores, sin embargo aún en muy tempranas épocas.

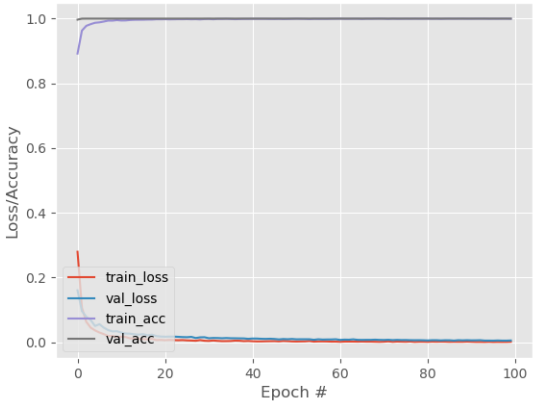
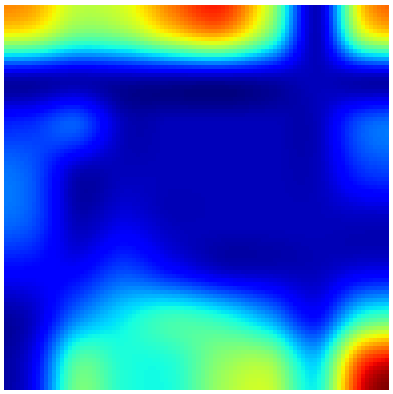
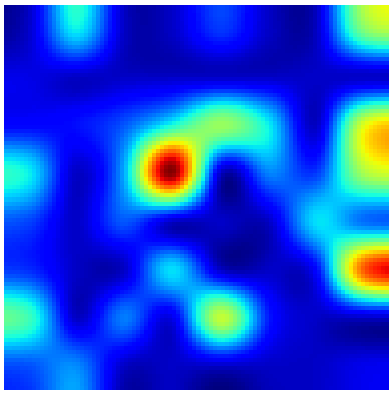


FIGURA 7.7: Curva de aprendizaje del primer experimento con la red Smaller VGG utilizando imágenes recortadas.



(a) Heatmap de un arándano de buena calidad.



(b) Heatmap de un arándano de mala calidad.

FIGURA 7.8: Heatmap obtenido a partir del modelo correspondiente a la figura 7.7.

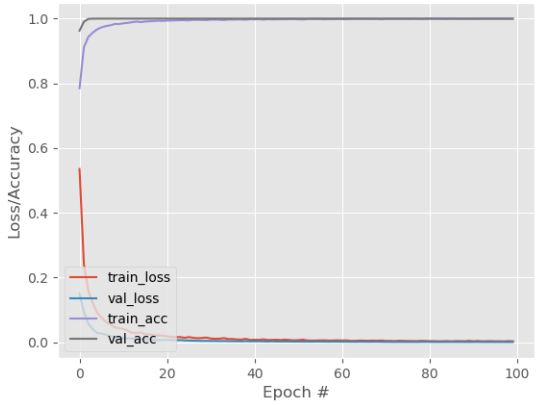
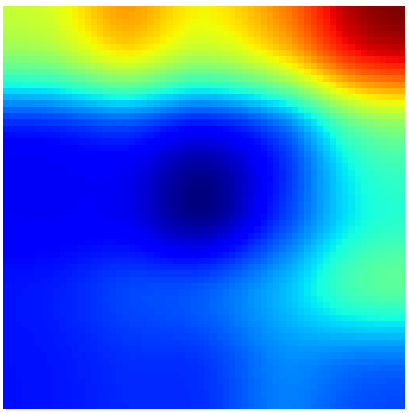
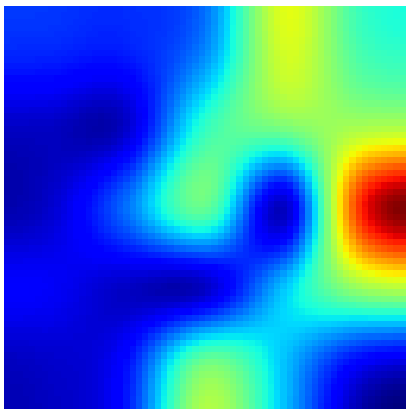


FIGURA 7.9: Curva de aprendizaje del segundo experimento con la red Smaller VGG utilizando imágenes recortadas.



(a) Heatmap de un arándano de buena calidad.



(b) Heatmap de un arándano de mala calidad.

FIGURA 7.10: Heatmap obtenido a partir del modelo correspondiente a la figura 7.9.

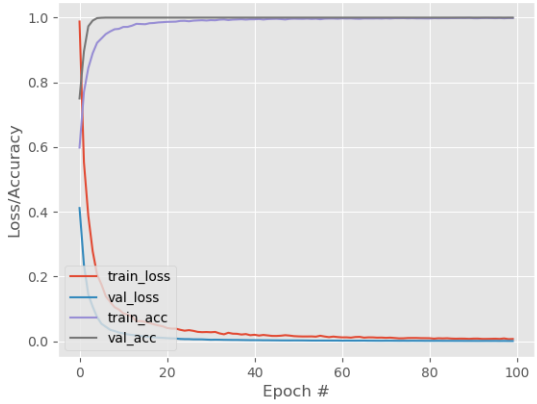
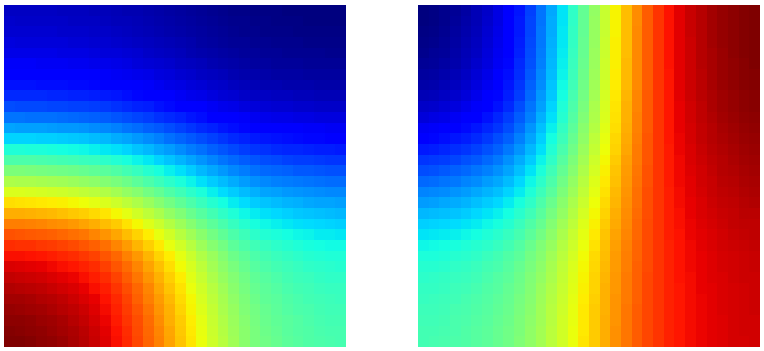


FIGURA 7.11: Curva de aprendizaje del tercer experimento con la red Smaller VGG utilizando imágenes recortadas.



(a) Heatmap de un arándano de buena calidad.

(b) Heatmap de un arándano de mala calidad.

FIGURA 7.12: Heatmap obtenido a partir del modelo correspondiente a la figura 7.11.

Se puede apreciar en los mapas de calor algo totalmente diferente a lo visto con LeNet, esta vez pareciera fijarse en puntos fijos, lejos del arándano. Aunque en el primer entrenamiento en la figura 7.8, para ambos pareciera fijarse en los cuadrados formados por la separación de lineas echas por el cuaderno. En el heatmap del segundo experimento en la figura 7.10, se ve un efecto similar al del experimento anterior. Mientras que en el último experimento en la figura 7.12 se ve un comportamiento que en realidad no tiene mucho sentido en cuanto a algún patrón en especifico, sin embargo algo es seguro y es que prácticamente ninguno de los 3 experimentos se guía por el arándano.

Cómo se mencionó en el capítulo anterior, es en este punto dónde se comienzan las pruebas con BSIF, debido a la característica de los resultados y su sesgo en base a los patrones marcados por Grad-CAM encontrados en la parte exterior del arándano por parte de LeNet, mientras que Smaller VGG no parece seguir un patrón en particular.

7.1.3 Experimento 3 (BSIF - LeNet)

TABLA 7.6: Tabla de resultados de pruebas con LeNet con filtro BSIF 3x3.

Experimento	Epochs	N Imagenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	Dense	Accuracy Train	Accuracy test
Experimento 1	100	20.000	$1e^{-5}$	100	96x96	$\approx$ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	1440	0.9976	1.0
Experimento 2	100	20.000	$1e^{-5}$	100	64x64	$\approx$ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	640	0.9973	0.9570
Experimento 3	100	20.000	$1e^{-5}$	100	32x32	$\approx$ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	180	0.9847	0.8185

El primer experimento de la tabla 7.6 refleja una situación similar a lo visto anteriormente, sin embargo a medida que bajan las dimensiones de las imágenes de entrada, el accuracy de test va bajando, situación que no había sucedido anteriormente en LeNet.

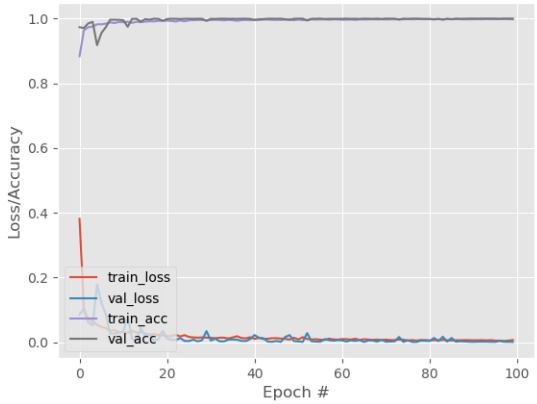
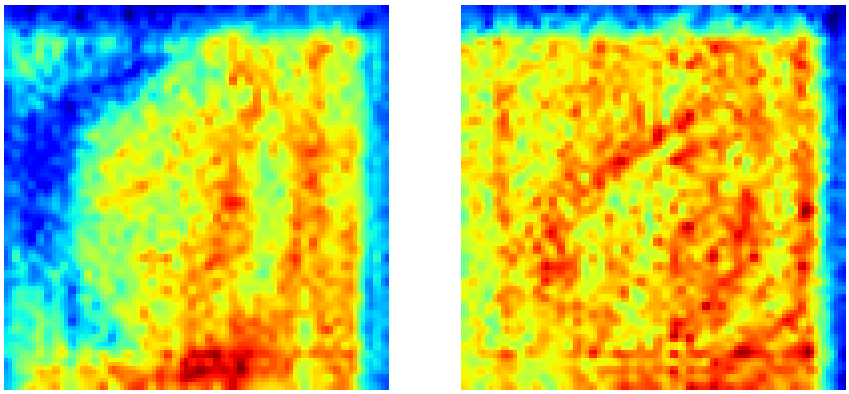


FIGURA 7.13: Curva de aprendizaje del primer experimento con la red LeNet utilizando imágenes recortadas y con un filtro BSIF de 3x3 de 8 bits.

En el primer experimento de la tabla 7.6, su curva tiende a alcanzar los niveles máximos en etapas tempranas y si vemos el heatmap en la figura 7.14 se nota una leve tendencia a dejar de lado lo que es el exterior del arándano, por lo tanto, esta vez el patrón del fondo de la imagen ya no es importante en esta prueba.



(a) Heatmap de un arándano de buena calidad.

(b) Heatmap de un arándano de mala calidad.

FIGURA 7.14: Heatmap obtenido a partir del modelo correspondiente a la figura 7.13.

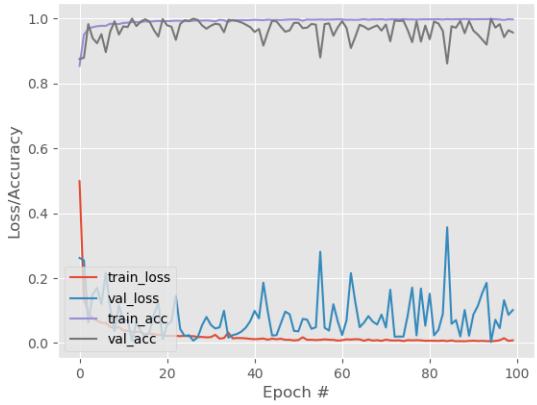
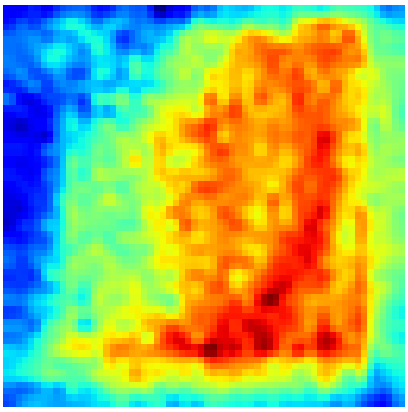
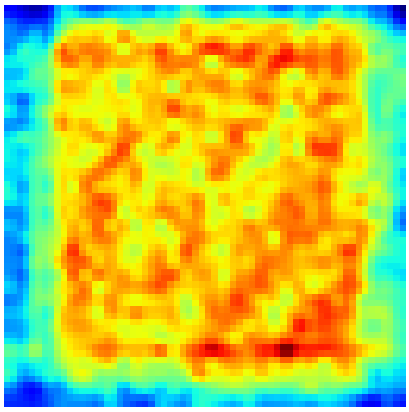


FIGURA 7.15: Curva de aprendizaje del segundo experimento con la red LeNet utilizando imágenes recortadas y con un filtro BSIF de 3x3 de 8 bits.

En este segundo experimento correspondiente al filtro BSIF de 3x3 con LeNet, si se ve la curva de la figura 7.15, se nota que la curva de test tiene un comportamiento por debajo que la de train. También se ve en la curva de pérdida oscilaciones que reflejan un sobreajuste o overfitting. El Heatmap de la figura 7.16 cumple con el patrón del experimento anterior.



(a) Heatmap de un arándano de buena calidad.



(b) Heatmap de un arándano de mala calidad.

FIGURA 7.16: Heatmap obtenido a partir del modelo correspondiente a la figura 7.15.

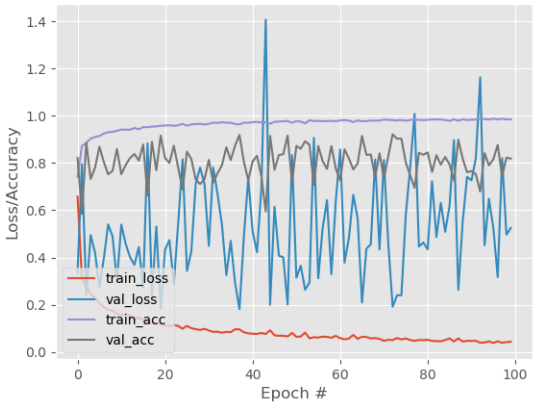
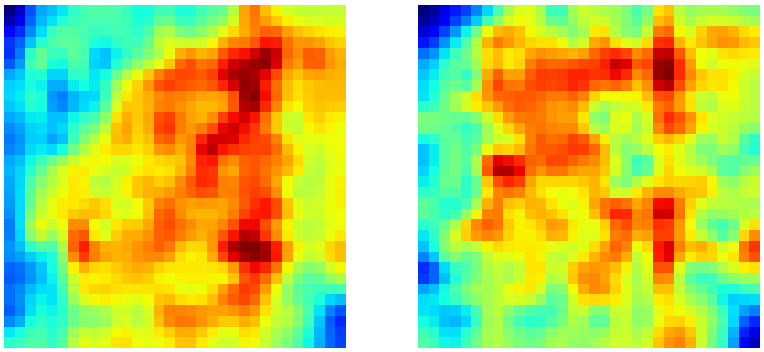


FIGURA 7.17: Curva de aprendizaje del tercer experimento con la red LeNet utilizando imágenes recortadas y con un filtro BSIF de 3x3 de 8 bits.

En cuanto al tercer experimento, reflejado en la curva 7.17, si bien la curva de entrenamiento tiene un comportamiento parecido a los experimentos anteriores, se debe destacar que la curva de test es variable dónde en realidad refleja que la red no aprendió en realidad, el error es muy grande.



(a) Heatmap de un arándano de buena calidad.

(b) Heatmap de un arándano de mala calidad.

FIGURA 7.18: Heatmap obtenido a partir del modelo correspondiente a la figura 7.17.

Cómo segundo filtro a aplicado, es el BSIF de 5x5 de 8 bits. Nuevamente los resultados muestran valores para el Accuracy totalmente elevados en los tres experimentos. Aunque esta vez la tendencia en la baja del desempeño en test no es tan drástica como en los experimentos con filtros de 3x3.

TABLA 7.7: Tabla de resultados de pruebas con LeNet con filtro BSIF 5x5.

Experimento	Epochs	N Imagenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	Dense	Accuracy Train	Accuracy test
Experimento 1	100	20.000	$1e^{-5}$	100	96x96	≈ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	1440	0.9955	0.9998
Experimento 2	100	20.000	$1e^{-5}$	100	64x64	≈ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	640	0.9958	0.9998
Experimento 3	100	20.000	$1e^{-5}$	100	32x32	≈ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	180	0.9800	0.9443

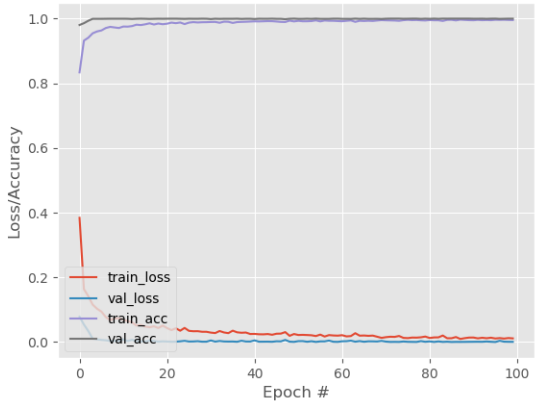
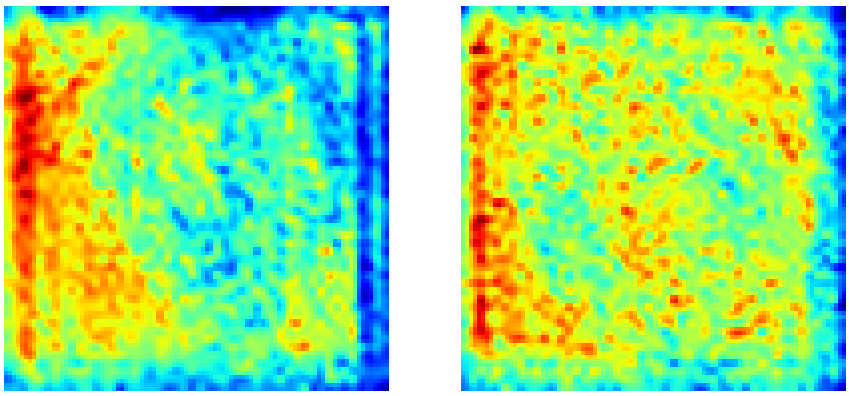


FIGURA 7.19: Curva de aprendizaje del primer experimento con la red LeNet utilizando imágenes recortadas y con un filtro BSIF de 5x5 de 8 bits.

En la figura 7.19 se nota una tendencia al sesgo en épocas muy tempranas. Dónde la curva de train está por debajo de la de test. Correspondiente a este experimento, las imágenes heatmap en la figura 7.20, muestran una importancia en la parte lateral de la imagen, siendo esto

prominente en la imagen del arándano de buena calidad, mientras que en la imagen del arándano de buena calidad además de esta fijación en la parte lateral y en gran parte de la imagen, sin embargo el arándano en sí no tiene mayor importancia en el modelo.



(a) Heatmap de un arándano de buena calidad.

(b) Heatmap de un arándano de mala calidad.

FIGURA 7.20: Heatmap obtenido a partir del modelo correspondiente a la figura 7.19.

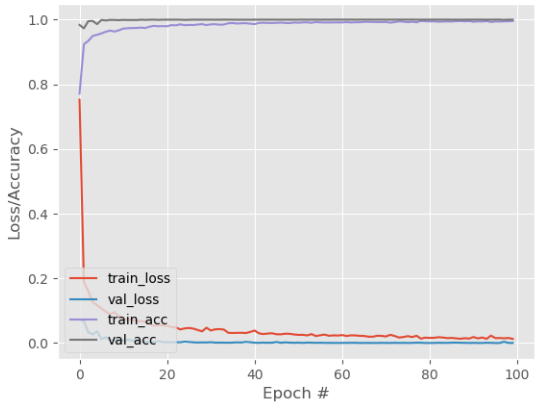
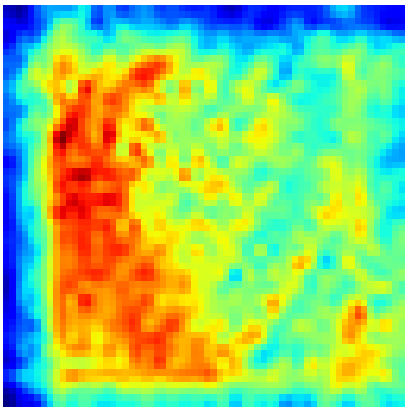
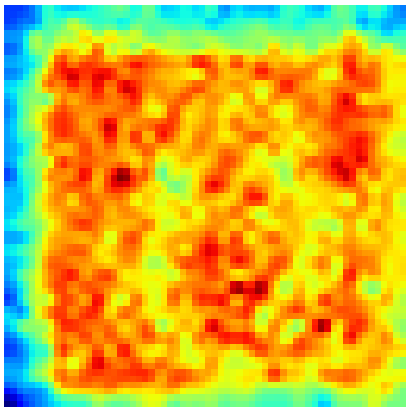


FIGURA 7.21: Curva de aprendizaje del segundo experimento con la red LeNet utilizando imágenes recortadas y con un filtro BSIF de 5x5 de 8 bits.

En el entrenamiento reflejado en la curva de la figura 7.21, al menos en términos de curva se nota un comportamiento muy similar al del experimento anterior. Sin embargo, si se presta atención a las imágenes de la figura 7.22 en la imagen del arándano de mala calidad, es posible visualizar que comprende todo el arándano, aunque también alcanza parte del rango exterior de este. Este es otro caso de sesgo dada la forma de las curvas de aprendizaje que se pueden visualizar en la figura 7.19.



(a) Heatmap de un arándano de buena calidad.



(b) Heatmap de un arándano de mala calidad.

FIGURA 7.22: Heatmap obtenido a partir del modelo correspondiente a la figura 7.21.

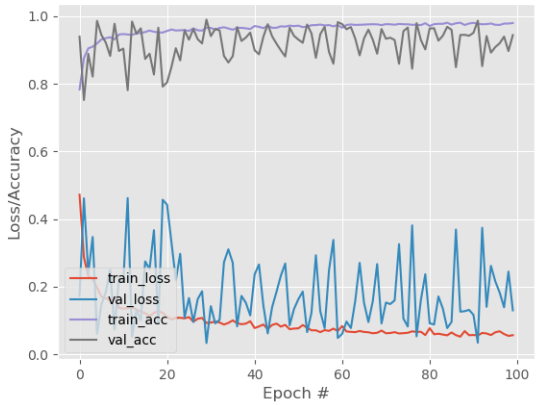
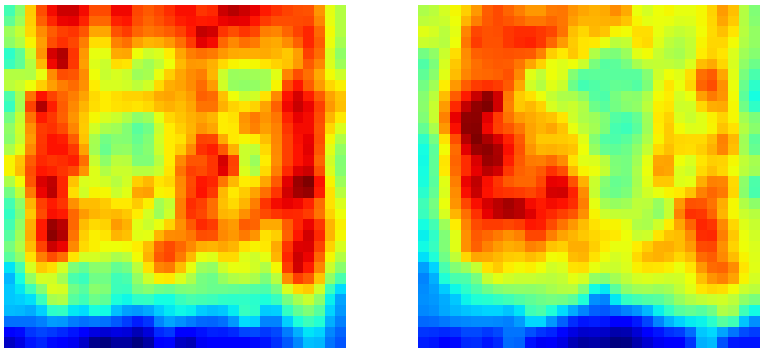


FIGURA 7.23: Curva de aprendizaje del tercer experimento con la red LeNet utilizando imágenes recortadas y con un filtro BSIF de 5x5 de 8 bits.

Los resultados de este entrenamiento pueden ser interesantes, sin embargo si vemos la curva de la figura 7.23, nos damos cuenta que la variabilidad tan drástica de la pérdida en test, es bastante amplia, sin embargo guarda cierta tendencia a bajar. Por otro lado la curva de test se mantiene por debajo, en gran parte de las épocas, de la curva de train. Cómo trabajo futuro es necesario variar los parámetros como las épocas y la tasa de aprendizaje, y ver nuevamente los resultados. Con respecto a los heatmaps, que pueden ser visualizados en la figura 7.24. Estos muestran un comportamiento algo más cercano al arándano en sus dos imágenes, aunque podría mejorar eventualmente.



(a) Heatmap de un arándano de buena calidad.

(b) Heatmap de un arándano de mala calidad.

FIGURA 7.24: Heatmap obtenido a partir del modelo correspondiente a la figura 7.23.

Los últimos filtros aplicados, para entrenamiento con LeNet. Los resultados vuelven a mostrar un comportamiento con una tasa de clasificación demasiado alta, como se aprecia en la tabla 7.8.

TABLA 7.8: Tabla de resultados de pruebas con LeNet con filtro BSIF 7x7.

Experimento	Epochs	N Imágenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	Dense	Accuracy Train	Accuracy test
Experimento 1	100	20.000	$1e^{-5}$	100	96x96	≈ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	1440	0.9937	0.9998
Experimento 2	100	20.000	$1e^{-5}$	100	64x64	≈ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	640	0.9913	0.9977
Experimento 3	100	20.000	$1e^{-5}$	100	32x32	≈ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	180	0.9729	0.9928

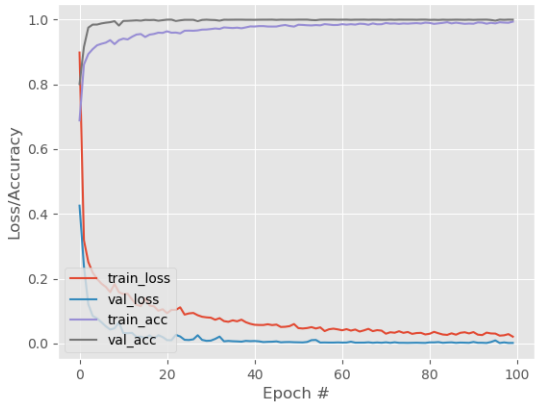
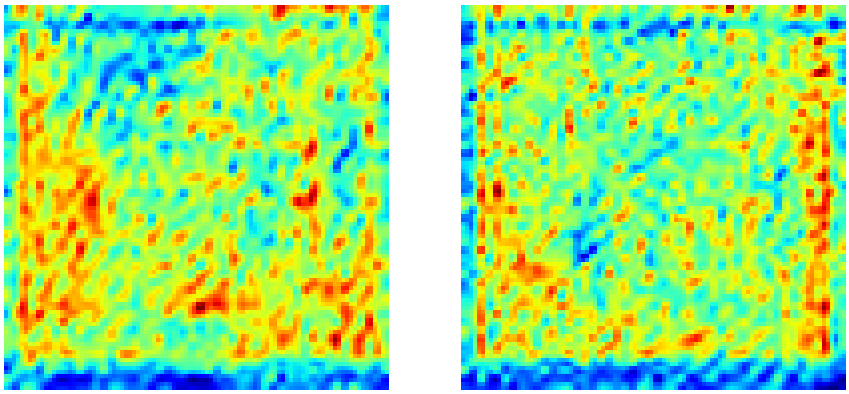


FIGURA 7.25: Curva de aprendizaje del primer experimento con la red LeNet utilizando imágenes recortadas y con un filtro BSIF de 7x7 de 8 bits.

Esta prueba cuya curva se muestra en la figura 7.25 se parece bastante en el comportamiento y su heatmap, al primer experimento evidenciado en la tabla 7.7. Con un sesgo desde las primeras etapas de entrenamiento, el resultado de su heatmap reflejado en la figura 7.26 no muestran un patrón general, para la discriminación de arándanos.



(a) Heatmap de un arándano de buena calidad.

(b) Heatmap de un arándano de mala calidad.

FIGURA 7.26: Heatmap obtenido a partir del modelo correspondiente a la figura 7.25.

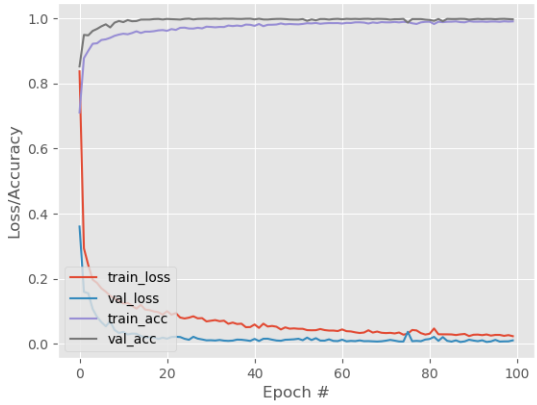
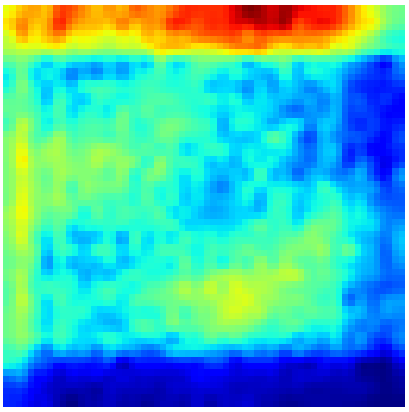
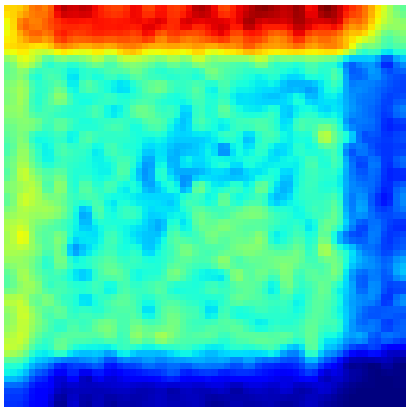


FIGURA 7.27: Curva de aprendizaje del segundo experimento con la red LeNet utilizando imágenes recortadas y con un filtro BSIF de 7x7 de 8 bits.

En la segunda prueba, dónde su curva se muestra en la figura 7.27, se muestra nuevamente con sesgo en épocas tempranas. Dónde las imágenes de heatmap de la figura 7.28, muestra un comportamiento totalmente errado, sólo haciendo una principal y fuerte actuar a las partes superiores de cada imagen, esto debido al sesgo mencionado anteriormente.



(a) Heatmap de un arándano de buena calidad.



(b) Heatmap de un arándano de mala calidad.

FIGURA 7.28: Heatmap obtenido a partir del modelo correspondiente a la figura 7.27.

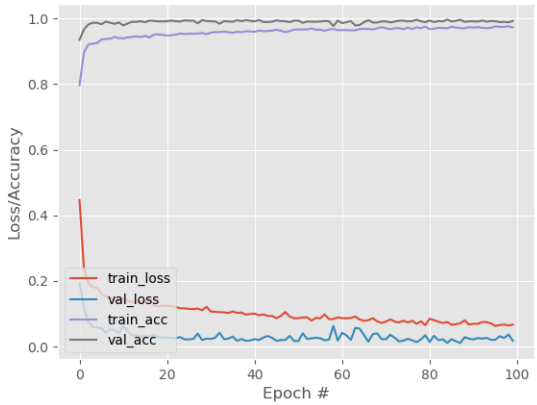
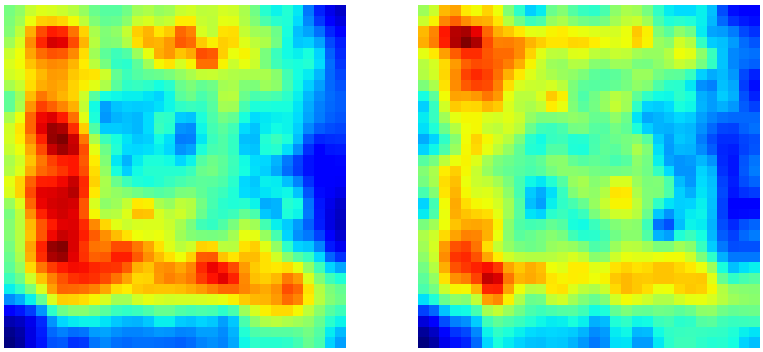


FIGURA 7.29: Curva de aprendizaje del tercer experimento con la red LeNet utilizando imágenes recortadas y con un filtro BSIF de 7x7 de 8 bits.

El tercer experimento también arroja altos valores y con un sesgo evidenciado en la curva de aprendizaje en las primeras etapas de entrenamiento. Al momento de generar los heatmap de la figura 7.30, se muestra una cobertura en cierta parte del arándano y quedando fuera gran parte de este. La inclusión del fondo de la imagen (es decir la hoja del cuaderno) como partes más importantes para el modelo, quiere decir que el modelo no es capaz de discernir de buena manera entre un arándano de buena calidad de uno que es de mala calidad.



(a) Heatmap de un arándano de buena calidad.

(b) Heatmap de un arándano de mala calidad.

FIGURA 7.30: Heatmap obtenido a partir del modelo correspondiente a la figura 7.29.

Se obtienen dos resultados importantes dentro de los experimentos realizados con BSIF y LeNet, el primero visualizado en el segundo experimento de la tabla 7.7 y el segundo experimento de la table 7.8, siendo este último el mejor de ambos.

7.1.4 Experimento 4 (BSIF - Smaller VGG)

Como se mencionó anteriormente estos filtros fueron aplicados también para un entrenamiento con Smaller VGG y comparar los resultados entre las dos redes y determinar cual logra un mejor desempeño. Cómo se mencionó en el capítulo anterior, no se generarán los mapas de calor para los resultados con mal desempeño.

TABLA 7.9: Tabla de resultados de pruebas con Smaller VGG con filtro BSIF 3x3.

Experimento	Epochs	N Imagenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	c5	Dense	Accuracy Train	Accuracy test
Experimento 1	100	20.000	$1e^{-5}$	100	96x96	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	1440	0.9634	0.9405
Experimento 2	100	20.000	$1e^{-5}$	100	64x64	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	640	0.9294	0.6914
Experimento 3	100	20.000	$1e^{-5}$	100	32x32	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	180	0.8244	0.5250

De los resultados obtenidos durante estos experimentos registrados en la tabla 7.9, sólo el primer experimento obtuvo resultados llamativos, el segundo podría mejorar cambiando los parámetros, especialmente la cantidad de épocas. Por lo tanto sólo se generará el heatmap para el primer experimento para ver el comportamiento de la red.

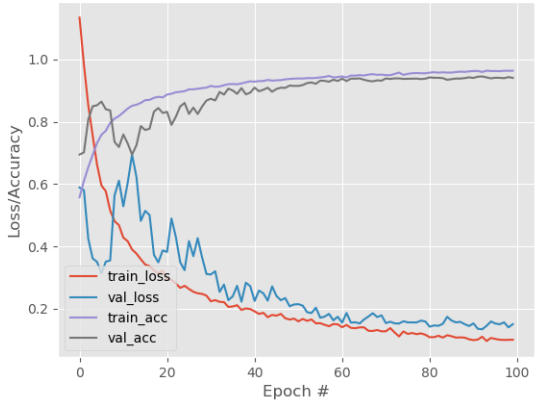
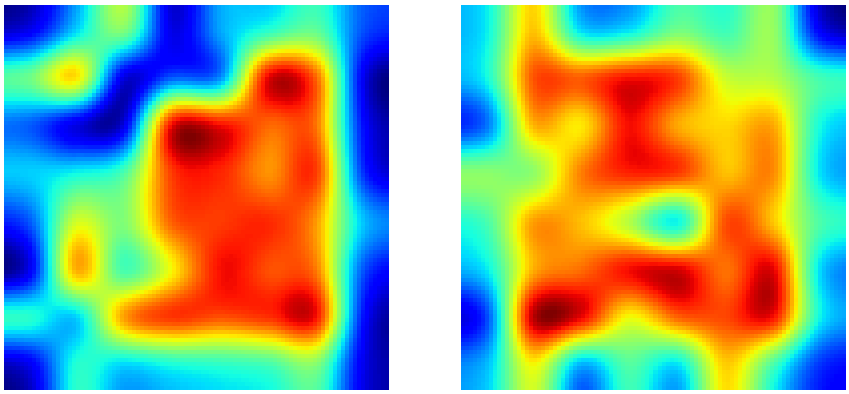


FIGURA 7.31: Curva de aprendizaje del primer experimento con la red Smaller VGG utilizando imágenes recortadas y con un filtro BSIF de 3x3 de 8 bits.

La curva de aprendizaje tiene un comportamiento llamativo, como se puede ver en la figura 7.31. Por otra parte los heatmap de la figura 7.32, se pueden apreciar una distribución de los píxeles de interés bastante dispersos, aun así no logra demostrar un enfoque orientado al fruto, a pesar de que en la imagen del arándano de buena calidad se reconoce gran parte del arándano, en la segunda imagen, se muestra un arándano donde el cual sólo muestra un leve interés en toda imagen, exceptuando algunos bordes del fruto en incluso algunas lineas de cuaderno.



(a) Heatmap de un arándano de buena calidad.

(b) Heatmap de un arándano de mala calidad.

FIGURA 7.32: Heatmap obtenido a partir del modelo correspondiente a la figura 7.31.

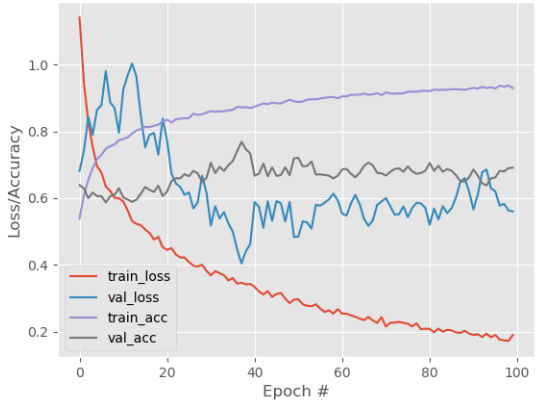


FIGURA 7.33: Curva de aprendizaje del segundo experimento con la red Smaller VGG utilizando imágenes recortadas y con un filtro BSIF de 3x3 de 8 bits.

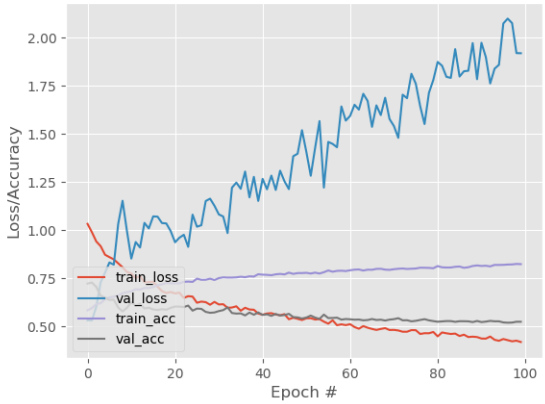


FIGURA 7.34: Curva de aprendizaje del tercer experimento con la red Smaller VGG utilizando imágenes recortadas y con un filtro BSIF de 3x3 de 8 bits.

Las curvas de aprendizaje tanto del segundo experimento, correspondiente a la figura 7.33, como el tercero correspondiente a la figura 7.34, tienen un comportamiento donde se nota que los modelos no están aprendiendo. Puede ser posible que con un ajuste en las épocas del segundo experimento pueda mejorar. Mientras que los demás con una variación del Learning Rate podría mejorar, sin embargo, esto como trabajo a futuro.

TABLA 7.10: Tabla de resultados de pruebas con Smaller VGG con filtro BSIF 5x5.

Experimento	Epochs	N Imagenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	c5	Dense	Accuracy Train	Accuracy test
Experimento 1	100	20.000	$1e^{-5}$	100	96x96	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	1440	0.9603	0.9979
Experimento 2	100	20.000	$1e^{-5}$	100	64x64	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	640	0.9268	0.9969
Experimento 3	100	20.000	$1e^{-5}$	100	32x32	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	180	0.8090	0.6667

Los experimentos en Smaller VGG utilizando filtros BSIF de 5x5, cuyos resultados se pueden visualizar en la tabla 7.10, presentan sesgo a la hora de clasificar. No sólo sus resultados

dan cuenta de esto, si no que las curvas también lo evidencian. Las curvas de aprendizaje de la figura 7.35 y 7.36 son similares, ambas evidenciando un comportamiento anómalo dadas las curvas de train en contraste a las de test, las cuales están por debajo de las de test y el comportamiento se replica en las curvas de pérdida. Sin embargo, para el tercer experimento, su desempeño podría mejorar con mayor cantidad de épocas y una variación de Learning Rate (LR), esto como trabajo a futuro.

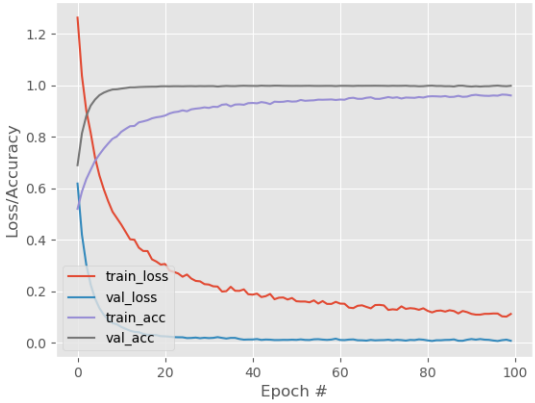


FIGURA 7.35: Curva de aprendizaje del primer experimento con la red Smaller VGG utilizando imágenes recortadas y con un filtro BSIF de 5x5 de 8 bits.

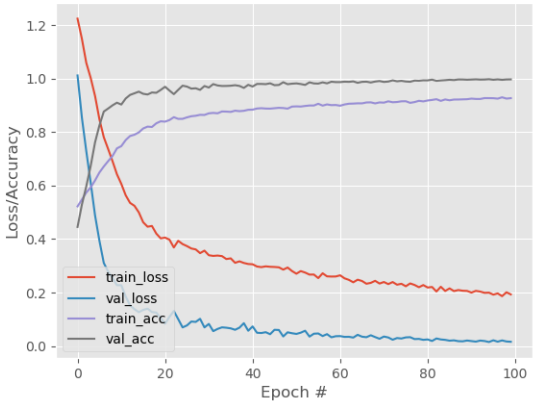


FIGURA 7.36: Curva de aprendizaje del segundo experimento con la red Smaller VGG utilizando imágenes recortadas y con un filtro BSIF de 5x5 de 8 bits.

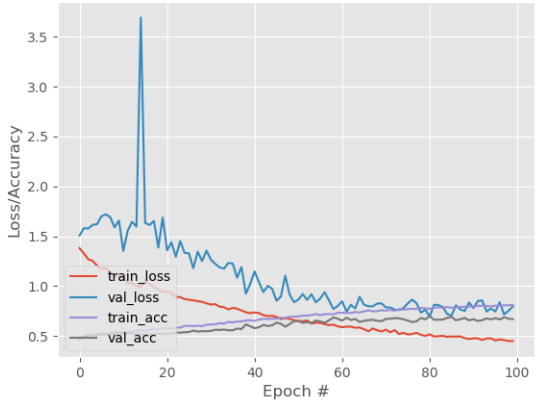


FIGURA 7.37: Curva de aprendizaje del tercer experimento con la red Smaller VGG utilizando imágenes recortadas y con un filtro BSIF de 5x5 de 8 bits.

TABLA 7.11: Tabla de resultados de pruebas con Smaller VGG con filtro BSIF 7x7.

Experimento	Epochs	N Imagenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	c5	Dense	Accuracy Train	Accuracy test
Experimento 1	100	20.000	$1e^{-5}$	100	96x96	≈ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	1440	0.9567	0.6154
Experimento 2	100	20.000	$1e^{-5}$	100	64x64	≈ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	640	0.9029	0.8005
Experimento 3	100	20.000	$1e^{-5}$	100	32x32	≈ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	180	0.7061	0.5277

Para finalizar con las pruebas, en estos últimos experimentos, cuyos desempeños pueden ser visualizados en la tabla 7.11, sólo uno de ellos presenta una posible mejora en variación de épocas, su curva de aprendizaje está plasmada en la figura 7.39. Las otras pruebas podrían mejorar con un cambio en el parámetro de Learning Rate (LR), como trabajo a futuro.

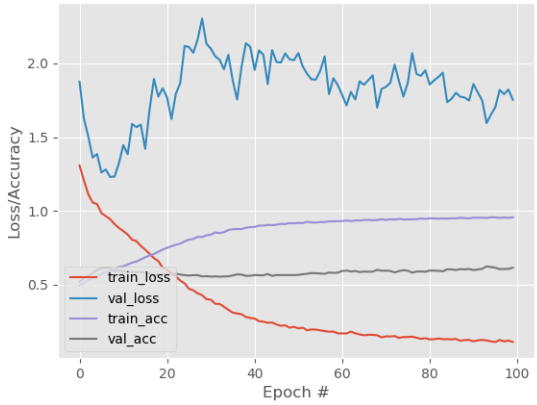


FIGURA 7.38: Curva de aprendizaje del primer experimento con la red Smaller VGG utilizando imágenes recortadas y con un filtro BSIF de 7x7 de 8 bits.

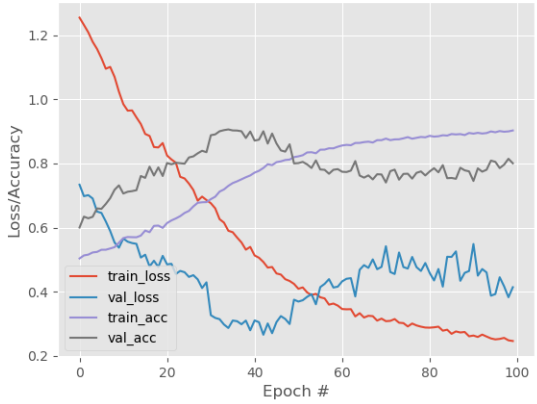


FIGURA 7.39: Curva de aprendizaje del segundo experimento con la red Smaller VGG utilizando imágenes recortadas y con un filtro BSIF de 7x7 de 8 bits.

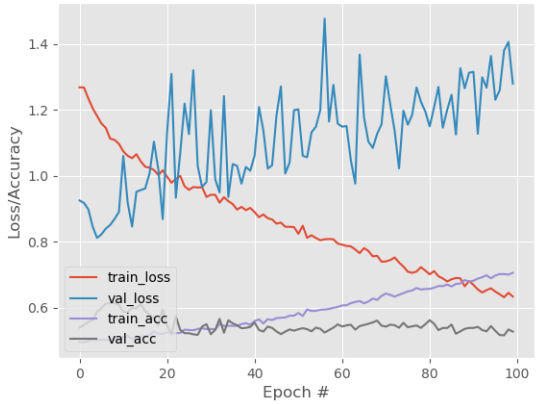


FIGURA 7.40: Curva de aprendizaje del tercer experimento con la red Smaller VGG utilizando imágenes recortadas y con un filtro BSIF de 7x7 de 8 bits.

El mejor resultado en base a la clasificación mediante la red neuronal Smaller VGG, fue en el primer experimento utilizando las imágenes con parches BSIF de 3x3 de 8 bits, en una resolución de 96x96. Mientras que en LeNet, el mejor resultado fue registrado por el tercer experimento, utilizando imágenes con parches BSIF de 5x5 de 8 bits.

Para finalizar, durante todo el proceso de pruebas, hubo patrones de comportamiento en cuanto a curvas o incluso en mapas de calor, debido a la similitud de parámetros. Existen algunos casos en los cuales se llega a resultados interesantes, sin embargo es necesario realizar más pruebas para determinar si efectivamente pueden realizar una mejor clasificación y visualización. Es por esto que se realizaron dos pruebas adicionales.

Estas pruebas tienen como fin de esclarecer el comportamiento del modelo en caso

de agregar mayor cantidad de épocas. Los parámetros son idénticos a sus pruebas anteriores, sólo se hizo una variación aumentando las épocas de estas.

En el caso registrado en el segundo experimento de la tabla 7.11, se realizó una prueba agregando 100 épocas más. Los resultados mejoraron considerablemente y pueden ser visualizados en la tabla 7.12 y corresponden al Experimento 7x7.

TABLA 7.12: Tabla de resultados de pruebas con Smaller VGG con diferentes filtros.

Experimento	Epochs	N Imagenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	c5	Dense	Accuracy Train	Accuracy test
Experimento 7x7	200	20.000	$1e^{-5}$	100	64x64	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	640	0.9396	0.9351
Experimento 3x3	200	20.000	$1e^{-5}$	100	64x64	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	640	0.9689	0.9748

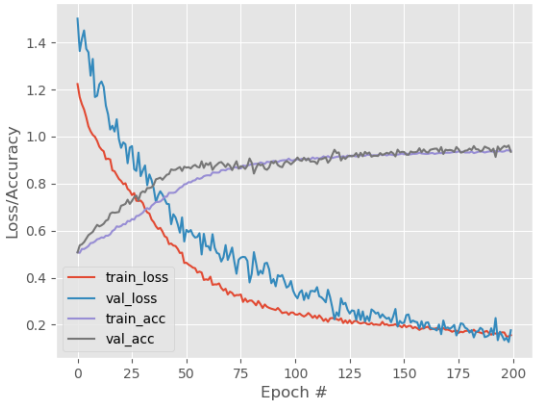
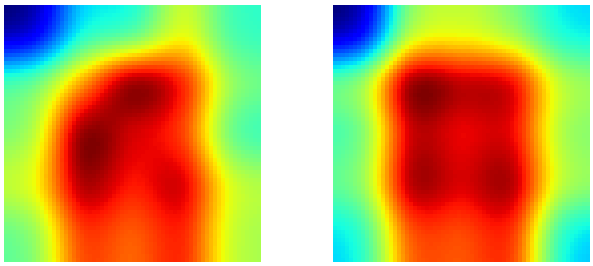


FIGURA 7.41: Curva de aprendizaje del primer experimento registrado en la tabla 7.12.

En comparación a su versión que tan sólo tiene 100 épocas, se puede visualizar en la figura 7.41 que la curva de test va aumentando hasta un 93 % aproximadamente, mientras que su error disminuye, acercándose al error registrado en train.

Si nos fijamos en el heatmap de la figura 7.42 tiene un mejor aspecto (se nota de mejor forma el arándano) que en el mejor modelo registrado, por lo tanto, es posible que también se comporte como un buen clasificador.



(a) Heatmap de un arándano de buena calidad.

(b) Heatmap de un arándano de mala calidad.

FIGURA 7.42: Heatmap obtenido a partir del modelo correspondiente a la figura 7.31.

Por otra parte, el experimento 2 de la tabla 7.12 corresponde al experimento 2 de la tabla 7.9 pero este último con mayor cantidad de épocas. Su curva de aprendizaje está registrada en la figura 7.43. Sin embargo, a partir de la época 150 aproximadamente comienza con un comportamiento donde el Accuracy de test es mayor a la de train, mostrándose así también en el Accuracy de resultado.

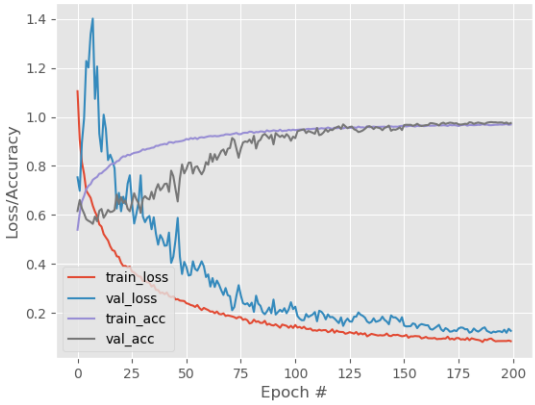


FIGURA 7.43: Curva de aprendizaje del primer experimento registrado en la tabla 7.12.

A modo de resumen, los mejores resultados son presentados a continuación en la tabla 7.13.

TABLA 7.13: Tabla resumen de los mejores resultados.

Experimento	Epochs	N Imagenes	LR	BS	Redimensión	Test size	c1	c2	c3	c4	c5	Dense	Accuracy Train	Accuracy test
Experimento Smaller VGG 7x7	200	20.000	$1e^{-5}$	100	64x64	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	640	0.9396	0.9351
Experimento Smaller VGG 3x3	100	20.000	$1e^{-5}$	100	96x96	$\approx$ 6000	10(3x3)	20(3x3)	20(3x3)	30(3x3)	30(3x3)	1440	0.9634	0.9405
Experimento LeNet 5x5	100	20.000	$1e^{-5}$	100	32x32	$\approx$ 6000	20(5x5)	40(5x5)	60(5x5)	80(5x5)	-	180	0.9800	0.9443

El resultado con mejor comportamiento es el registrado cómo Experimento Smaller VGG 3x3 de la tabla7.13 en cuanto a curvas y resultados. Se puede visualizar que el comporta-

miento de los experimentos en Smaller VGG es mejor en comparación a los de LeNet, tanto en la curva de pérdida como en la de test, en el caso de LeNet los valores van cambiando abruptamente en comparación a la de Smaller VGG. Si comparamos sus visualizaciones en Grad-CAM, los heatmap de Smaller VGG tienen un mejor reconocimiento sobre el arándano.

A pesar de que este sea el mejor resultado, no se debe ignorar el hecho de que el mapa de calor del Experimento Smaller VGG 7x7 registrado en la figura 7.42, de la tabla 7.13, marca de una más definida el arándano, sin embargo su curva registrada en la figura 7.41 mantiene siempre cambios entre train y test sobre quien tiene mayor valor.

CAPÍTULO 8. CONCLUSIONES

Respecto a la primera hipótesis es posible diseñar una base de datos de arándanos en espectro infrarrojo, específicamente en espectro SWIR. El diseño de la base de datos es un trabajo bastante arduo, requiere bastante tiempo en el etiquetado y diseño manual, dada la cantidad de imágenes que suman un total de 97084.

Respecto a la segunda hipótesis la información complementaria de dichos arándanos pudo ser destacada realizando un algoritmo que resaltara texturas, dado que la información en base a la intensidad de los píxeles no fue suficiente para que los modelos entrenados obviaran el patrón registrado en el fondo de las imágenes, por lo tanto la característica que permitió saber la calidad del arándano fue su textura.

Respecto a la tercera hipótesis se llegó a la conclusión que la mejor red neuronal convolucional Shallow (de pocas capas) fue Smaller VGG, con un accuracy registrado en la tabla 7.13 obteniendo un 96,34 % de train y 94,05 % de test, mostrando también en su visualización a modo de mapa de calor, con una imagen de arándano que no fue utilizada ni para entrenamiento ni para train, una figura que destacaba el arándano. Además en general mostró un mejor comportamiento que LeNet al ser una red más profunda y compleja.

Por otro lado a medida que se realizaron las pruebas, comenzaron a demostrarse diferentes tipos de problemas. En primer lugar el manejo de formatos de imagen es realmente importante, a veces se puede cambiar la información real de una imagen, como pasó en las primeras pruebas, por lo cual es necesario ser realmente cuidadoso con la documentación de las librerías que se están utilizando para el manejo de imágenes y evitar confusiones u/o errores. Punto importante es lo que desencadena toda esta falla en la preparación de los datos, pues, afecta directamente en el comportamiento del entrenamiento de las redes neuronales o incluso en la visualización de una imagen dado el formato en el que alguna librería funciona a diferencia de otra.

También es considerable el hecho de que se tuvieron que realizar más de las pruebas necesarias para saber que red y parámetros tenían buenos resultados, debido a la condición de sesgo que presentaban las redes neuronales, sobretodo las que utilizaban como entrada las imágenes sin filtros BSIF, dados sus diferencias en el fondo de estas.

8.1 TRABAJOS FUTUROS

Finalmente, existen diversas formas de seguir mejorando el proyecto, entre las cuales se pueden destacar:

- Hacer pruebas con absolutamente todos los filtros de BSIF.
- Realizar pruebas con redes neuronales más profundas y complejas, las propuestas en el capitulo 4 específicamente, y así aprovechar la gran cantidad de imágenes totales almacenadas en la base de datos creada.
- Tener una base de datos con la misma cantidad de imágenes para ambas clases.
- Realizar un mejor espectro de variación de parámetros, pero esta vez, con los filtros BSIF que muestren un mejor comportamiento.
- Realizar una comparación a fondo utilizando con otras métricas los mejores experimentos registrados.
- Utilizar espectrograma de las imágenes como entrada a las redes neuronales.

BIBLIOGRAFÍA

[1] Comité Arándanos Chile, *Arándanos de Chile*. dirección: http://www.comitedearandanos.cl/arandanos-de-chile/.

[2] ——, *Arándanos de Chile Compendio Estadístico Temporada 2016 - 2017*. dirección: http://www.comitedearandanos.cl/wp-content/uploads/2015/02/COMPENDIO-ESTADISTICO-2017-V3.pdf.

[3] Comité Arándanos Chile and iQonsulting, *Crop & Exports Report 2017-2018*. dirección: http://www.cbbc.iqonsulting.com/index.php?lang=es.

[4] A. González G, C. G. Morales A, J. Riquelme S, J. Hirzel C, A. France I, A. Pedreros L, H. Uribe C, B. Defilippi B, P. Robledo M y C. Becerra C, "Manual de manejo agronómico del arándano", *Boletín INIA*, vol. 06, C. G. Morales A, ed., 2017, ISSN: 0717-4829.

[5] M.-H. Hu, Y. Zhao y G.-T. Zhai, "Active learning algorithm can establish classifier of blueberry damage with very small training dataset using hyperspectral transmittance data", 2018.

[6] C. Cortes y V. Vapnik, "Support-vector networks", *Machine Learning*, vol. 20, n.º 3, págs. 273-297, sep. de 1995, ISSN: 1573-0565. DOI: 10.1007/BF00994018.

[7] S. Haykin, "Neural Networks: A Comprehensive Foundation", en *Self-organizing maps*, 2nd, Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998, cap. 9, ISBN: 0132733501.

[8] G. A. Leiva-Valenzuela y J. M. Aguilera, "Automatic detection of orientation and diseases in blueberries using image analysis to improve their postharvest storage quality", 2013.

[9] D. Mery, *BALU: A Matlab toolbox for computer vision, pattern recognition and image processing (http://dmery.ing.puc.cl/ index.php/balu)*, 2011.

[10] A. K. Jain, R. P. W. Duin y J. Mao, "Statistical pattern recognition: A review", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, n.º 1, págs. 4-37, 2000.

[11] J. Kuzy, Y. Jiang y C. Li, "Blueberry bruise detection by pulsed thermographic imaging", 2018.

[12] W. M. Gentleman y G. Sande, "Fast Fourier Transforms: For Fun and Profit", en *Proceedings of the November 7-10, 1966, Fall Joint Computer Conference*, ép. AFIPS '66 (Fall), San Francisco, California: ACM, 1966, págs. 563-578. DOI: 10.1145/1464291.1464352. dirección: http://doi.acm.org/10.1145/1464291.1464352.

[13] I. Kononenko, "Estimating Attributes: Analysis and Extensions of RELIEF", en *European Conference on Machine Learning*, F. Bergadano y L. D. Raedt, eds., Springer, 1994, págs. 171-182.

[14] R. A. Fisher, "The use of multiple measurements in taxonomic problems", *Annals of human genetics*, vol. 7, n.º 2, págs. 179-188, 1936.

[15] L. Breiman, "Random forests", *Machine learning*, vol. 45, n.º 1, págs. 5-32, 2001.

[16] P. McCullagh y J. A. Nelder, *Generalized linear models*. CRC press, 1989, vol. 37.

[17] L. E. Peterson, "K-nearest neighbor", *Scholarpedia*, vol. 4, n.º 2, pág. 1883, 2009.

[18] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis", 2018.

[19] K. Simonyan y A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[20] C. M. Bishop y col., "Pattern recognition and machine learning (information science and statistics)", 2006.

[21] T. M. Mitchell y col., "Machine learning. 1997", *Burr Ridge, IL: McGraw Hill*, vol. 45, 1997.

[22] L. Deng, D. Yu y col., "Deep Learning: Methods and Applications", *Foundations and Trends in Signal Processing*, vol. 7, n.º 3–4, págs. 197-387, 2014.

[23] Y. LeCun, L. Bottou, Y. Bengio y P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, n.º 11, págs. 2278-2324, 1998.

[24] S. Albelwi y A. Mahmood, "A Framework for Designing the Architectures of Deep Convolutional Neural Networks", *Entropy*, vol. 19, n.º 6, 2017, ISSN: 1099-4300. DOI: 10.3390/e19060242. dirección: http://www.mdpi.com/1099-4300/19/6/242.

[25] M. A. Tanner, "The Data Augmentation Algorithm", en *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions*. New York, NY: Springer US, 1993, ISBN: 978-1-4684-0192-9. DOI: 10.1007/978-1-4684-0192-9_5. dirección: https://doi.org/10.1007/978-1-4684-0192-9_5.

[26] A. K. Boyat y B. K. Joshi, "A Review Paper: Noise Models in Digital Image Processing", *CoRR*, vol. abs/1505.03489, 2015.

[27] Itseez, *Open Source Computer Vision Library*, https://github.com/itseez/opencv, 2015.

[28] J. D. Hunter, "Matplotlib: A 2D graphics environment", *Computing In Science & Engineering*, vol. 9, n.º 3, págs. 90-95, 2007. DOI: 10.1109/MCSE.2007.55.

[29] F. Chollet y col., *Keras*, https://keras.io, 2015.

[30] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu y X. Zheng, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org, 2015. dirección: https://www.tensorflow.org/.

[31] Y. Lecun, L. Bottou, Y. Bengio y P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, n.º 11, págs. 2278-2324, 1998, ISSN: 0018-9219. DOI: 10.1109/5.726791.

[32] A. Rosebrock, *Image classification with Keras and deep learning*, 2017. dirección: https://www.pyimagesearch.com/2017/12/11/image-classification-with-keras-and-deep-learning/.

[33] R. Arora, A. Basu, P. Mianjy y A. Mukherjee, "Understanding Deep Neural Networks with Rectified Linear Units", en *International Conference on Learning Representations*, 2018. dirección: https://openreview.net/forum?id=B1J_rgWRW.

[34] A. Rosebrock, *Keras and Convolutional Neural Networks (CNNs)*, 2018. dirección: https://www.pyimagesearch.com/2018/04/16/keras-and-convolutional-neural-networks-cnns/.

[35] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh y D. Batra, "Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization", *CoRR*, vol. abs/1610.02391, 2016. arXiv: 1610.02391. dirección: http://arxiv.org/abs/1610.02391.

[36] D. Bobeldyk y A. Ross, "Predicting Eye Color from Near Infrared Iris Images", en *2018 International Conference on Biometrics (ICB)*, 2018, págs. 104-110. DOI: 10.1109/ICB2018.2018.00026.

[37] K. Simonyan y A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *CoRR*, vol. abs/1409.1556, 2014. eprint: 1409.1556. dirección: http://arxiv.org/abs/1409.1556.