



Instituto Tecnológico de Hermosillo

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

**MODELADO DE VISIÓN ARTIFICIAL PARA EL ANÁLISIS
DE CAPA SUPERFICIAL DE ESPUMA EN EL PROCESO DE
FLOTACIÓN PARA LA OBTENCIÓN DE MINERALES**

TESIS

PRESENTADO POR:

LIC. NELSÓN ROMERO GARCÍA

COMO REQUISITO PARCIAL PARA
OBTENER EL GRADO DE:

MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

DIRECTOR DE TESIS: M.C. SONIA REGINA MENESES MENDOZA

CO - DIRECTOR DE TESIS: DR. JULIO WAISSMAN VILANOVA

Hermosillo, Sonora, México

Av. Tecnológico S/N Col. El Sahuaro, C.P. 83170 Hermosillo, Sonora.

Tel. (662) 2-606500 Ext. 136, e-mail: posgrado@ith.mx www.ith.mx





EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Hermosillo

"2020, Año de Leona Vicario, Benemérita Madre de la Patria"

SECCIÓN: DIV. EST. POS. E INV.
No. OFICIO: DEPI/066/20
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN
DE TESIS.

24 de Junio de 2020

**C. NELSON ROMERO GARCÍA,
PRESENTE.**

Por este conducto, y en virtud de haber concluido la revisión del trabajo de tesis que lleva por nombre **"MODELADO DE VISIÓN ARTIFICIAL PARA EL ANÁLISIS DE CAPA SUPERFICIAL DE ESPUMA EN EL PROCESO DE FLOTACIÓN PARA LA OBTENCIÓN DE MINERALES"**, que presenta para el examen de grado de la MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN, y habiéndola encontrado satisfactoria, nos permitimos comunicarle que se autoriza la impresión del mismo a efecto de que proceda el trámite de obtención de grado.

Deseándole éxito en su vida profesional, quedo de usted.

ATENTAMENTE

M.C. SONIA REGINA MENESES MENDOZA
DIRECTORA

DR. OSCAR MARIO RODRÍGUEZ ELÍAS
SECRETARIO

M.C. CÉSAR ENRIQUE ROSE GÓMEZ
VOCAL

M.C.O. ROSA IRENE SANCHEZ FERMÍN
JEFA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN



S.E.P.

RISF/fjcg*

INSTITUTO TECNOLÓGICO
DE HERMOSILLO
DIVISIÓN DE ESTUDIOS
DE POSGRADO



Av. Tecnológico S/N Col. El Sahuaro C.P. 83170 Hermosillo, Sonora
Tel. 01 (662) 2-606500, Ext. 136 e-mail: depi_hermosillo@tecnm.mx
www.tecnm.mx | www.ith.mx



ISO 9001:2015
Sistema de Gestión de Calidad Certificado

Resumen

El análisis de capa superficial de espuma ha sido clave para entender el comportamiento general de los sistemas de flotación de burbujas para la obtención de minerales en la industria minera. En el estado del arte se han desarrollado varios analizadores de capa superficial de espuma de las burbujas en las celdas de flotación con el objetivo de clasificar mejor el proceso, adición de reactivos y control del flujo de aire. El uso de la inteligencia artificial en la industria ha tenido un uso creciente y ha demostrado su aplicabilidad y buenos resultados mediante aplicación de técnicas para la clasificación de imágenes.

En la presente investigación se presenta un caso aplicado a la industria minera donde se desarrolla un modelo de inteligencia artificial para el análisis de la capa superficial de espuma en el proceso de flotación para la obtención de minerales.

Se realiza un estudio sobre la naturaleza del proceso de flotación, los métodos y herramientas de inteligencia artificial para la clasificación de imágenes, así como de los trabajos relacionados. Se detalla la metodología a seguir durante el desarrollo del modelo, como la preparación y exploración de los datos, y la implementación de algoritmos tanto de aprendizaje automático como de aprendizaje profundo para el análisis de las imágenes. Se presenta la implementación del modelo, los resultados obtenidos y el análisis de los mismos resultados.

Con el modelo desarrollado se obtuvo una precisión de 0.99 en el análisis de las burbujas en flotación, demostrando la efectividad para estimar el estado del proceso a partir de las imágenes tomadas en las celdas de flotación. Estos resultados pretenden mejorar el proceso, reducir costos, aumentar productividad y calidad en la extracción de minerales.

Abstract

The analysis of the surface foam layer has been key to understanding the general behavior of bubble flotation systems for obtaining minerals in the mining industry. In the state of the art, several bubble foam surface layer analyzers have been developed in the flotation cells to better classify the process, the addition of reagents, and control of airflow. The use of artificial intelligence in the industry has had an increasing use and has demonstrated its applicability and good results by applying techniques for image classification.

In the present investigation, a case applied to the mining industry is presented, where an artificial intelligence model is developed for the analysis of the superficial foam layer in the flotation process to obtain minerals.

A study is conducted on the nature of the flotation process, artificial intelligence methods and tools for image classification, as well as related work. The methodology to be followed during the development of the model is detailed, such as the preparation and exploration of the data, and the implementation of both machine learning and deep learning algorithms for image analysis. The implementation of the model, the results obtained and the analysis of the same results are presented.

With the developed model, a precision of 0.99 was obtained in the analysis of the floating bubbles, demonstrating the effectiveness to estimate the state of the process from the images taken in the floating cells. These results are intended to improve the process, reduce costs, increase productivity, and quality in the extraction of minerals.

Dedicatoria

Dedico esta tesis a mi esposa, Beatriz Rodríguez, por estar siempre a mi lado durante todo este tiempo y su constante apoyo. Gracias por su amor y confianza he podido llegar hasta aquí.

A mis padres, Nelsón Romero y María Elena García, y mi hermano Yulierky Romero, a quienes debo todos y cada uno de los logros alcanzados, gracias por siempre estar presentes y darme los ánimos de seguir cumpliendo mis metas y aspiraciones. Gracias por su apoyo, confianza en mí y cariño durante toda mi vida.
Sin ellos no sería la persona de hoy.

A toda mi familia, por su constante apoyo que me han dado siempre.

Agradecimientos

A mi directora de tesis, M.C. Sonia Regina Meneses, por la confianza brindada durante estos últimos años para la realización de este trabajo, por el apoyo y buena comunicación. Muchas gracias por poder contar con usted en todo momento.

A mi co-director de tesis el Dr. Julio Weissman Vilanova, por la confianza depositada en mí para la realización de esta investigación, así como por consejos y todo el conocimiento compartido, los cuales fueron determinantes para cumplir los objetivos propuestos.

A mis sinodales, Dr. Oscar Mario Rodríguez Elias y M.C. César Enrique Rose Gómez, por el valioso tiempo que dedicaron en la revisión de este trabajo de tesis y sus comentarios.

A todos mis compañeros de generación por toda la ayuda y buenos momentos compartidos dentro y fuera del laboratorio.

Al núcleo académico de la Maestría en Ciencias de la Computación del Instituto Tecnológico de Hermosillo, por su apoyo incondicional, su trato cordial y accesible en todo momento. Gracias por su invaluable labor de enseñanza y por fomentar el aprendizaje.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por su apoyo económico mediante la beca de manutención otorgada durante 2 años.

A todos les estaré eternamente agradecido por formar parte de esta maravillosa etapa que me permitió ser un mejor profesional y crecer como persona.

Índice General

1	Introducción	1
1.1.	Introducción.....	1
1.2.	Antecedentes.....	1
1.3.	Planteamiento del problema.....	3
1.4.	Preguntas de investigación	4
1.4.1.	Pregunta Principal	4
1.4.2.	Preguntas Específicas.....	4
1.5.	Objetivos	4
1.5.1.	Objetivo General	4
1.5.2.	Objetivos Específicos	4
1.6.	Justificación.....	5
1.7.	Alcances y delimitaciones	5
1.7.1.	Alcances.....	5
1.7.2.	Delimitaciones	6
1.8.	Metodología.....	6
1.9.	Organización de la tesis	7
2	Estado del arte	8
2.1.	Proceso Minero	9
2.1.1.	Proceso de extracción de cobre.....	9
2.1.2.	Proceso de flotación.....	9
2.2.	FrothMeter.....	12
2.2.1.	Ventajas y desventajas.....	16
2.2.2.	ECN.....	17
2.3.	Visión Artificial	17

2.3.1. Visión para el reconocimiento de objetos	18
2.3.2. Reconocimiento de patrones.....	18
2.3.3. Seguimiento de objetos en tiempo real	19
2.3.4. Procesamiento digital de imágenes.....	19
2.4. Inteligencia Artificial.....	20
2.4.1. Aprendizaje Automático.....	21
2.4.2. Aprendizaje Profundo	23
2.5. Trabajos relacionados	29
3 Aprendizaje Automático y Minería de Datos.....	31
3.1. Metodología.....	31
3.2 Exploración de datos	31
3.2.1. Colección de Datos	32
3.2.2. Comprensión de Datos y Preparación de Datos	33
3.2.3. Coincidencia de Histogramas.....	33
3.2.4. Reagrupamiento de Atributos	36
3.2.5. Reagrupamiento de Ejemplos Usando el Método de Mezcla de Gaussianas	38
3.2.6. Matriz de Relación Grupos Formados/Clasificación Expertos.....	41
3.3. Algoritmos de Aprendizaje.....	42
3.3.1. Gaussian Naive Bayes	42
3.3.2. Regresión Logística.....	43
3.3.3. Máquina de Soporte Vectorial con kernel Gaussiano.....	43
3.3.4. Curvas de validación para los parámetros γ y C	45
3.3.5. Máquina de Soporte Vectorial con kernel Gaussiano y parámetros óptimos	46
4.4. Resultados Parciales de Aprendizaje Automático	46

4 Aprendizaje Profundo y Transferencia de Aprendizaje	48
4.1. Entorno de Trabajo y Conjunto de Datos	48
4.2. Transferencia de Aprendizaje.....	49
4.2.1. VGGNet.....	52
4.2.2. RestNet	54
4.2.3. InceptionV3 y InceptionRestNetV2.....	58
4.2.4. Xception	60
4.3. Resultados Parciales de Aprendizaje Profundo	62
5 Resultados y discusión.....	66
6 Conclusiones y Trabajos Futuros	69
Referencias	71

Índice de Figuras

Figura 1.1: Metodología para desarrollo de la tesis.....	6
Figura 2.1: Una ilustración de una sección transversal a través de una celda de flotación, que muestra el impulsor a la izquierda y el lavado a la derecha. Las partículas rojas representan hidrófobos flotantes, los minerales valiosos y las partículas verdes representan la gana hidrófila no flotante. Tomado de [24]	12
Figura 2.2. Visualización de HMI del sistema de operación. Tomado de [25]	14
Figura 2.3: Monitoreo de Velocidad y Dirección de Burbuja. Tomado de [25].....	14
Figura 2.4: Monitoreo de tamaño y Estabilidad de Burbuja. Tomado de [25].....	15
Figura 2.5: Burbuja caracterizada como no deseable por su tamaño compacto. Tomado de [25].	15
Figura 2.6: Burbuja no deseable por su gran tamaño y alta masa de recuperación con bajo grado de pureza. Tomado de [25].	16
Figura 2.7: Burbuja dentro de los márgenes establecidos en la planta, con el mejor equilibrio entre alta recuperación con alto porcentaje de cobre. Tomado de [25].	16
Figura 2.8: Etapas de un Sistema de Visión Artificial. Tomado de [29]	18
Figura 2.9: Reconocimiento de patrones. Tomado de [29].....	19
Figura 2.10: Etapas del Procesamiento Digital de Imágenes. Tomado de [29].....	20
Figura 2.11: El aprendizaje profundo es un subconjunto del aprendizaje automático, que a su vez es un subconjunto de aprendizaje artificial. Tomado de [34]	21
Figura 2.12: Arquitectura típica de una red neuronal convolucional. Tomado de [65]	25
Figura 2.13: Operación de un kernel de convolución. Tomado de [75]	26
Figura 2.14: Convolución de una imagen de 5×5 con un filtro de 3×3 , lo que resulta en un conjunto más pequeño de características de 3×3 (solo se muestra la primera). Tomado de [65].....	27
Figura 2.15: Agrupación máxima de un conjunto 3×3 de características convueltas que resulta en un conjunto 2×2 de características agrupadas máximas. Tomado de [65]	27
Figura 3.1: Metodología para el desarrollo del modelo de IA.	31

Figura 3.2: Imágenes muestras de cada una de las clases de capa superficial de espuma	33
Figura 3.3: Resultado de aplicar la normalización de colores. Tomado de [90]	34
Figura 3.4: Coincidencia de histograma de colores trazado por cada canal RGB. Tomado de [90]	34
Figura 3.5: Imagen resultada de aplicar la normalización de colores sobre la imagen fuente	35
Figura 3.6: Resultado de histogramas de colores trazado por cada canal RGB ...	35
Figura 3.7: Resultado de aplicar t-SNE	36
Figura 3.8: PCA sobre las dos primeras variables	37
Figura 3.9: PCA sobre las dos primeras variables usando la transformada de Fourier	38
Figura 3.10: Algoritmo de K-Means para diferentes números de clústers.....	39
Figura 3.11: Método de silueta sobre K-Means.....	39
Figura 3.12: Método de Mezcla de Gaussianas para 5 clústers.....	40
Figura 3.13: Método de silueta sobre Mezcla de Gaussianas.....	41
Figura 3.14: Curva de aprendizaje para el algoritmo Gaussian Naive Bayes	42
Figura 3.15: Curva de aprendizaje para el algoritmo de Regresión Logística	43
Figura 3.16: Curva de aprendizaje para el algoritmo de Máquina de Soporte Vectorial con kernel Gaussiano.....	44
Figura 3.17: Curva de validación para el parámetro γ	45
Figura 3.18: Curva de validación para el parámetro C	45
Figura 3.19: Curva de aprendizaje para el algoritmo de Máquina de Soporte Vectorial con kernel Gaussiano y parámetros óptimos	46
Figura 4.1: Métodos definidos para el aumento de datos.....	49
Figura 4.2: Diferentes procesos de aprendizaje entre (a) máquina tradicional aprendizaje y (b) aprendizaje de transferencia. Tomado de [105].....	50
Figura 4.3: Esquema de CNN para el entrenamiento; Error! Marcador no definido.	
Figura 4.4: Representación simplificada de la arquitectura de la red neuronal convolucional VGG16, con 13 capas de convolución ($C_{11}, C_{12}, \dots C_{53}$), cinco capas de agrupación (P_1 a P_5) y tres capas totalmente conectadas (FC_1, FC_2 y FC_3). Las capas convolucional y de agrupación se organizan en cinco bloques, según lo	

indicado por los índices de las capas convolucional y de agrupación. Tomado de [65]	53
Figura 4.5: Curvas de entrenamiento y validación tanto en precisión y pérdida para VGG16	54
Figura 4.6: Curvas de entrenamiento y validación tanto en precisión y pérdida para VGG19	54
Figura 4.7: Error de entrenamiento (izquierda) y error de prueba (derecha) con redes de 20 y 56 capas. La red más profunda tiene un error de entrenamiento más alto y, por lo tanto, error de prueba. Tomado de [55]	55
Figura 4.8: Representación simplificada de la arquitectura de una red neuronal convolucional ResNet de 34 capas que muestra capas convolucionales ($C_{1,1}$, $C_{1,2}$, ... $C_{5,3}$), capas combinadas (P_1 y P_2) y una capa totalmente conectada (FC_1). Tomado de [65]	56
Figura 4.9: Curvas de entrenamiento y validación tanto en precisión y pérdida para ResNet50	57
Figura 4.10: Curvas de entrenamiento y validación tanto en precisión y pérdida para ResNet101	57
Figura 4.11: Curvas de entrenamiento y validación tanto en precisión y pérdida para ResNet152	57
Figura 4.12: Arquitecturas de las redes InceptionV3 (A) e InceptionResNetV2 (B). Tomado de [110]	58
Figura 4.13: Módulos inception para las redes Inception_V3 (A) e Inception_ResNet_V2 (B). Tomado de [110]	59
Figura 4.14: Curvas de entrenamiento y validación tanto en precisión y pérdida para InceptionV3	59
Figura 4.15: Curvas de entrenamiento y validación tanto en precisión y pérdida para InceptionResNetV2	60
Figura 4.16: Arquitectura de Xception. Tomado de [84]	61
Figura 4.17: Curvas de entrenamiento y validación tanto en precisión y pérdida para Xception	61
Figura 4.18: Estructuras definidas para la última capa de los modelos	62

Figura 5.1: Matrices de confusión para aprendizaje automático (A) y para aprendizaje profundo (B).....	67
Figura 5.2: Reportes de clasificación para aprendizaje automático (A) y aprendizaje profundo (B)	67

Índice de Tablas

Tabla 3.1: Matriz de relación entre Grupos Formados/Clasificación Expertos	41
Tabla 3.2: Precisión de clasificación de modelos en conjuntos de entrenamiento y validación.	46
Tabla 4.1: Resultados de entrenamiento y validación tanto en precisión y pérdida para los modelos de redes neuronales preentrenados	63
Tabla 5.1: Comparación en cuanto a cantidad de parámetros de los modelos DCNN	68

Capítulo 1

1 Introducción

1.1. Introducción

Sonora es una de las principales regiones mineras de México [1], siendo así que el desarrollo e implementación de nuevas tecnologías resulta de gran importancia para la mejora de los procesos de tratamiento de minerales; permitiendo ofrecer mejor calidad del producto para su empleo y comercialización. De manera constante la modernización de la industria a nivel mundial se encuentra en pleno desarrollo y la industria de minera mexicana debe mantenerse cerca de este desarrollo.

La investigación y el desarrollo en la aplicación de la visión artificial en sistemas de flotación de espuma cubre la necesidad de obtener resultados más eficientes para el análisis de capa superficial de espuma, cuyo proceso se realiza a través de la experiencia de un operador, lo que refiere a la no exactitud del análisis y pérdidas en la obtención del mineral. La visión artificial puede extraer con precisión y rapidez las características de la espuma, tanto física como dinámica, de imágenes digitales y presentar estos resultados a los operadores y / o utilizar los resultados como entradas para el control del proceso.

El modelo a desarrollar tiene como objetivo el análisis en tiempo real de capa superficial de espuma usando técnicas de procesamiento digital de imágenes, mediante imágenes obtenidas de una cámara de alta resolución y un sistema de iluminación sobre el derrame de espuma, obteniendo los parámetros deseados y enviándolos al sistema de control del proceso.

1.2. Antecedentes

Según la Secretaría de Economía de México [2], el sector minero-metalúrgico contribuye con el 4% del Producto Interno Bruto (PIB) nacional. México en el mundo se ubica entre los 10 principales productores de 16 diferentes minerales. Además,

es el 1er destino en inversión en exploración minera en América Latina y el 4to en el mundo; y es el 5to país con el mejor ambiente para hacer negocios mineros.

La comprensión de estos procesos de flotación ha sido durante mucho tiempo clave para entender el comportamiento general de los sistemas de flotación [3], [4]. Buscando resolver esta problemática, se han desarrollado analizadores de la burbuja derramada en las celdas de flotación, buscando mejorar el control de nivel, la adición de reactivos y el control de flujo de aire.

Los objetivos principales de control de las celdas de flotación son los factores metalúrgicos, es decir, recuperación y el grado de concentración del mineral. La estimación y medición de estas variables generalmente requieren instrumentos sofisticados que son caros de comprar y mantener [5]. Estudios anteriores demuestran que las características visuales de la espuma reflejan cambios en las condiciones del proceso y pueden usarse para predecir los factores metalúrgicos [5][6].

Las características de la capa superficial de espuma más importantes, incluyen el tamaño de burbuja, el color de la espuma, la velocidad de la espuma y la velocidad de colapso de la burbuja. Varios algoritmos de procesamiento de imágenes eficientes fueron desarrollados para cuantificar estas características. La distribución del tamaño y el color son variables estáticas que se calculan a partir de una sola imagen, mientras que la velocidad de la espuma y la velocidad de colapso de la burbuja son variables dinámicas que se calculan a partir de un par de imágenes.

Varias técnicas desarrolladas para la medición del tamaño de burbuja incluyen la segmentación [7][8], análisis de textura [10], enfoque de espectro de textura modificado [11] y uso de información morfológica interfacial [12]. En la práctica, cada uno de estos métodos tiene sus respectivas ventajas y desventajas.

La velocidad de la espuma se puede cuantificar siguiendo el movimiento de las burbujas en cuadros consecutivos. La comparación de bloques [13], el trazado de píxeles [14] y el rastreo de burbujas [15]; son las técnicas más utilizadas para medir la velocidad de la espuma.

El color de espuma es indicativo del grado de concentración y recuperación en las celdas de flotación [16]. El color de la espuma se puede medir mediante la extracción de los valores de rojo, verde y azul del color de las imágenes.

El colapso de las burbujas en la superficie de la espuma es un indicador de la estabilidad de la espuma, la cual se cuantifica analizando fotogramas consecutivos y detectando el cambio en la apariencia de las imágenes.

En los capítulos siguientes se desarrollan las diferentes técnicas de procesamiento de imagen más adecuadas, las cuales permitan mejorar el análisis de capa superficial de espuma en las celdas de flotación.

1.3. Planteamiento del problema

El análisis de capa superficial de espuma en el proceso de flotación minero es una tarea primordial para el desarrollo de la industria. En el mismo se dan reacciones químicas que llevan a cabo la recuperación de mineral, lo que resulta complejo de controlar ya que está fuertemente influenciado por la naturaleza caótica de las propiedades microscópicas de las partículas.

Este proceso de análisis es un problema mal definido puesto que en cada mina se realiza mediante la observación de un operador que valora el comportamiento de las burbujas y determina la calidad del proceso, por lo que la caracterización del proceso de flotación de burbujas no es exacta y por tanto la pérdida de mineral es un problema que afecta a la industria.

Actualmente se encuentra instalado un sistema en una planta concentradora de cobre en Sonora, en donde se están validando los resultados de velocidad de derrame y estabilidad. Esta variante mejora el proceso considerablemente pues aporta mejores resultados, mediante imágenes tomadas a través de una cámara colocada en la parte superior de la celda de flotación; sin embargo, aún es necesario desarrollar modelos de análisis de imágenes para calcular velocidad, dirección y estabilidad de burbujas en celdas de flotación aplicando técnicas de Visión Artificial y procesamiento en sistema embebido basado en GPU.

Por lo tanto, surge la necesidad de mejorar el algoritmo de procesamiento de imágenes, mediante el uso de técnicas de inteligencia artificial (Intelligence Artificial, IA por sus siglas en inglés) actuales que permitan caracterizar el proceso de análisis de burbujas más eficientemente de manera automática.

1.4. Preguntas de investigación

Derivado de lo anterior nos planteamos las siguientes preguntas de investigación.

1.4.1. Pregunta Principal

¿Qué elementos debe incluir el modelo para procesar imágenes correctamente durante el análisis de la capa superficial de espuma en el proceso de flotación de burbujas?

1.4.2. Preguntas Específicas

- ¿Cómo identificar una burbuja en el proceso de flotación?
- ¿Qué algoritmos de procesamiento digital de imágenes se podrían utilizar para el problema en cuestión?
- ¿Qué técnicas o algoritmos de inteligencia artificial podrían ayudar a mejorar el análisis de imágenes en el proceso de flotación?

1.5. Objetivos

Para la implementación del sistema a desarrollar, a continuación, se plantean los siguientes objetivos.

1.5.1. Objetivo General

El objetivo general de este trabajo es desarrollar un modelo de análisis de imágenes para clasificar el tamaño de burbujas en celdas de flotación aplicando técnicas de visión e inteligencia artificial.

1.5.2. Objetivos Específicos

Para satisfacer el objetivo general de este trabajo se plantean los siguientes objetivos específicos:

- Identificar la complejidad en el proceso de análisis de la capa superficial de espuma.

- Determinar una metodología para el análisis de imágenes en la capa superficial de espuma basada en inteligencia artificial.
- Seleccionar algoritmos de procesamiento digital de imágenes para la medición de parámetros en la capa superficial de burbujas.
- Evaluar la confiabilidad y exactitud de dichas mediciones para el análisis de capa superficial de espuma.
- Definir un método para caracterizar una burbuja para su análisis.
- Estudiar técnicas de inteligencia artificial para mejorar el reconocimiento de las burbujas en el proceso de análisis de capa superficial.

1.6. Justificación

El presente trabajo se plantea dada la necesidad de reducir las pérdidas de minerales y económicas, además de facilitar el análisis de la capa superficial de espuma en el proceso de flotación de burbujas; lo que en un futuro se traduce en una mayor y mejor producción en la industria minera, así como una mayor competencia en el sector industrial. Por lo que la importancia se manifiesta en el impacto económico, científico y tecnológico dada la aportación de un modelado capaz de brindar mayor control y monitoreo del proceso de flotación.

Por consiguiente, implementado este modelado se obtendrían mejores resultados en la industria minera, y además podría extenderse su alcance a otras áreas de la industria, las cuales también sufren pérdidas millonarias.

1.7. Alcances y delimitaciones

1.7.1. Alcances

- Desarrollo de algoritmos de procesamiento de imágenes para determinar parámetros como tamaño en las burbujas de la celda de flotación.
- Desarrollar modelo basado en técnicas de inteligencia artificial para relacionar las características de la burbuja con las variables del proceso de flotación.

1.7.2. Delimitaciones

- El modelo a desarrollar se hará de acuerdo al material proporcionado por ECN.
- La adquisición de variables de proceso en celda de flotación será a cargo de Froth Meter.

1.8. Metodología

La metodología a seguir para el desarrollo de la presente tesis se compone de 4 etapas, las cuales comprenden el desarrollo del proyecto de investigación y se describen brevemente a continuación (véase Fig. 1.1):

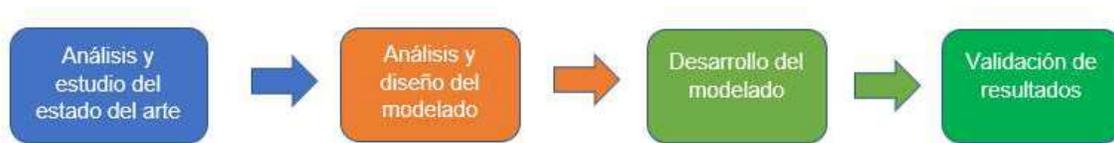


Figura 1.1: Metodología para desarrollo de la tesis

- **Análisis y estudio del estado del arte:** en esta primera etapa, se realizará una recopilación en torno a los conceptos teóricos, que existen acerca del tema de investigación, debido a que se debe formar una base del conocimiento para el análisis, diseño, desarrollo e implementación del modelo a desarrollar.
- **Análisis y diseño:** una vez comprendido el estado del arte, se continuará con el proceso de análisis de la información obtenida, lo que permitirá definir el diseño para el desarrollo del modelado.
- **Desarrollo del modelado:** esta etapa refiere al desarrollo del modelo a partir del análisis previo de las etapas anteriores, de tal manera que el algoritmo se encuentre funcional.
- **Validación de resultados:** se hará un análisis de los resultados obtenidos para la validar el modelo desarrollado.

1.9. Organización de la tesis

A continuación, se explicará por capítulos de forma detallada las etapas que se llevarán a cabo para el desarrollo del proyecto de investigación.

El capítulo 2 presenta el estado del arte en el cual se plantea y analiza los temas teóricos e investigación científica que hay entorno al desarrollo de la tesis.

El capítulo 3 refiere a la documentación del trabajo realizado para el análisis y diseño del modelo, donde se presenta a detalle las técnicas utilizadas haciendo uso de aprendizaje automático.

El capítulo 4 describe la implementación de modelos de aprendizaje profundo y los resultados obtenidos.

El capítulo 5 presenta los resultados obtenidos de acuerdo al trabajo realizado sobre el modelo desarrollado.

Finalmente, el **capítulo 6** presenta las conclusiones obtenidas en el desarrollo del trabajo de investigación y se plantea los trabajos futuros generados a partir de solución de la problemática.

Capítulo 2

2 Estado del arte

En el presente capítulo se hace un análisis documentado sobre los conceptos básicos que sirven como fundamento de este trabajo de tesis. Estas definiciones incluyen temas relacionados para comprender del trabajo desarrollado.

Se refieren temas teóricos relacionados con el proceso minero y particularmente el proceso de extracción de cobre en celdas de flotación. Con respecto a este tema se definen cómo se realiza el proceso de extracción de cobre, cómo se desarrolla el proceso de flotación y cómo es que se comportan las celdas de flotación para obtención del mineral.

Se hace un análisis del sistema de visión artificial FrothMeter instalado en una planta concentradora de cobre en Sonora, donde se describen aspectos relacionados con la implementación del sistema, sus componentes, así como las ventajas y desventajas que presenta. Añadiendo una descripción de la empresa Electro Controles del Noroeste S.A (ECN, por sus siglas) la cual junto al Centro de Investigación en Matemáticas A.C. (CIMAT, por sus siglas) desarrollaron tal aplicación.

El siguiente tema hace una revisión sobre las posibles técnicas de IA a utilizar para dar solución al problema de investigación planteado. Este tema es de gran importancia para el desarrollo de la tesis dado que pretende la mejora de obtención de las características en el proceso de flotación.

Para finalizar, se realiza un estudio de trabajos relacionados con el presente trabajo de investigación. Estos trabajos constituyen antecedentes a esta investigación, y muchos constituyen el punto de partida para mejorar el desarrollo en cuestión.

2.1. Proceso Minero

La minería es una actividad económica del sector primario representada por la explotación o extracción de minerales que se han acumulado en el suelo y subsuelo en forma de yacimientos.

La automatización en la industria minera permite obtener aumentos de productividad, disminución de costos, aumentos de continuidad operativa, mejoras en la seguridad y la salud laboral.

2.1.1. Proceso de extracción de cobre

La extracción del cobre se produce en su mayor parte de minas abiertas, donde se extrae un mineral muy pobre en contenido. Esto hace que los procesos de refinamiento posterior estén dedicados mayormente a la concentración y eliminación de impurezas.

Los procesos más utilizados para la obtención de mineral de cobre son los denominados flotación y lixiviación, que permiten separar el cobre de los minerales que lo contienen. El uso de uno u otro sistema depende de la procedencia del mineral.

Si se trata de sulfurados primarios se usa la flotación, cuya base es el agua; y para el caso de oxidados y sulfurados secundarios se utiliza la lixiviación, la que se realiza mediante la utilización de ácido sulfúrico.

También existe otras tecnologías para la obtención de cobre a través del uso de bacterias. Estas últimas hacen más limpio el proceso y reducen costos productivos.

Las etapas del proceso de producción de cobre procedente de óxidos son: extracción, chancado, lixiviación y precipitación electrolítica. En el caso de los minerales sulfurados son: extracción, chancado, flotación (concentración), fundición, refinación y moldeo, y electrorefinación.

2.1.2. Proceso de flotación

En las industrias de procesamiento de minerales, la flotación por espuma es la tecnología de separación más utilizada e importante en el procesamiento de

minerales . El control efectivo de los circuitos de flotación es una tarea difícil debido a varias variables involucradas, cambios inevitables en las características del mineral y la naturaleza no lineal y dinámica del proceso [17].

El proceso de flotación es ampliamente aplicado a la concentración de minerales y se basa en la captura de partículas hidrófobas por burbujas de aire [18]. En este proceso, la velocidad con que se recuperan los minerales depende del tamaño de las burbujas de aire. La calidad del concentrado de flotación depende de la cantidad de agua que arrastran las burbujas de aire cuando viajan de la pulpa a la cama de espuma mineralizada.

En flotación, para la producción de burbujas de aire finas y estables, se emplean agentes tensioactivos llamados *espumantes*, que también promueven la formación de una cama de espuma mineralizada, disminuyen la velocidad de ascenso de las burbujas de aire y aumentan el aire retenido en la celda de flotación[19] [20] [21]. Los espumantes mejoran notablemente la eficiencia de las celdas de flotación de tal manera que impactan tanto el grado del concentrado como la recuperación del mineral que se flota, que generalmente es el mineral valioso [22].

Desde la concepción del proceso de flotación, se han desarrollado varios tipos de equipos o celdas de flotación, las cuales se clasifican en: celdas de flujo ascendente como las celdas de flotación mecánicas y las columnas de flotación y celdas de flujo descendente como las celdas Jameson [23].

2.1.2.1. Celdas de flotación

Normalmente, en un circuito de trituración se reduce el tamaño de partícula del mineral extraído para producir una alimentación adecuada para el circuito de molienda por flotación. En el circuito de molienda, los dispositivos de molienda dividen las partículas en una distribución de tamaño de partícula deseada, la cual varía según el tipo de mineral y de sus propiedades, como la dureza y la liberación de minerales. El propósito principal del circuito de molienda es liberar los minerales y proporcionar superficies minerales limpias para el proceso de flotación.

Una vez que la suspensión sale del circuito de molienda [24], entra en un tanque de acondicionamiento donde reactivos agregados reaccionan con las partículas dentro de la suspensión. Posteriormente, la suspensión entra en la celda de flotación. Una celda de flotación es un tanque agitado, impulsado por un motor eléctrico, que contiene una entrada para aire y tiene una configuración deflectora para crear una intensidad y distribución de turbulencia específicas. Una celda de flotación tiene dos salidas, un lavado para la recogida de espuma que contiene minerales valiosos y una válvula de salida para las partículas de desecho [24].

Dentro de la celda de flotación, existen dos fases físicas muy importantes; una fase de pulpa y una fase de espuma. En la fase de pulpa, se dispersa el aire de manera efectiva a través de la agitación para promover colisiones de partículas de burbuja. Cuando una partícula con minerales hidrófobos expuestos choca con una burbuja, puede unirse a una burbuja y elevarse a través de la celda de flotación para alcanzar la interfaz pulpa-espuma y entrar en la fase de espuma [7].

Dada la presencia de reactivos espumantes, las burbujas de aire forman una capa de espuma en la parte superior de la pulpa. El papel de la fase de espuma es transportar selectivamente las partículas valiosas hacia y dentro de l lavado para la recogida de espuma. Las burbujas en la parte de la interfaz pulpa-espuma arrastran a las partículas de ganga no flotante a los espacios entre las burbujas. El grado en que estas partículas alcanzan la superficie de la espuma y se recuperan en el concentrado determina la pureza o el grado del concentrado. A continuación se muestra una celda de flotación que ilustra estas regiones (véase Fig. 2.1).

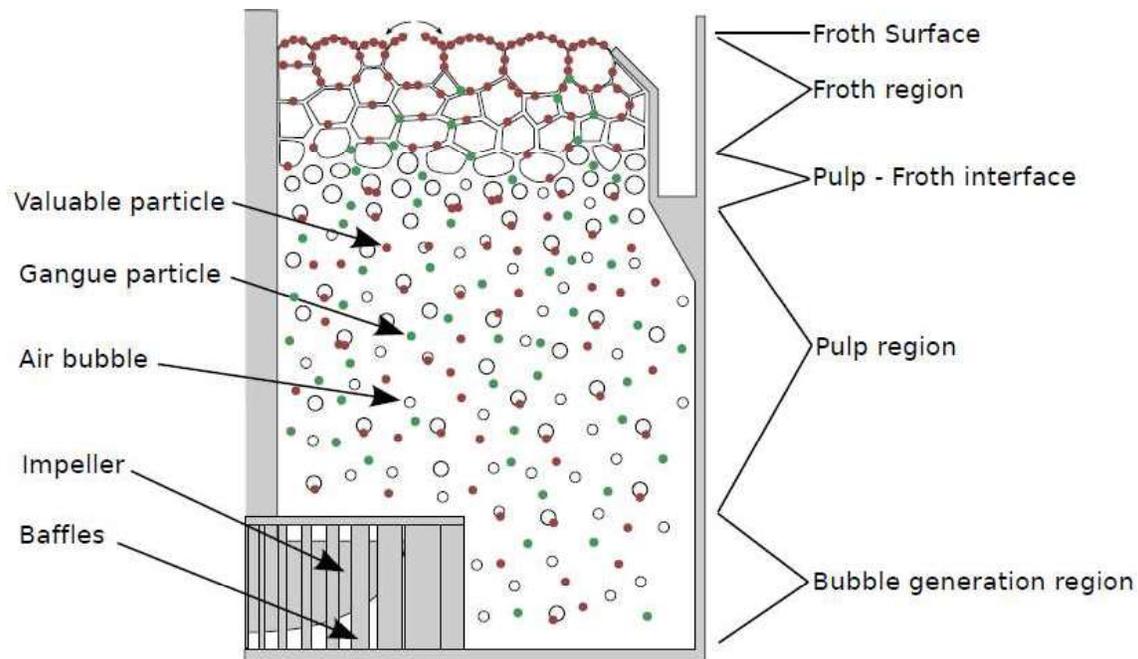


Figura 2.1: Una ilustración de una sección transversal a través de una celda de flotación, que muestra el impulsor a la izquierda y el lavado a la derecha. Las partículas rojas representan hidrófobos flotantes, los minerales valiosos y las partículas verdes representan la gana hidrófila no flotante. Tomado de [24]

En general, se entiende que a poca profundidad de la espuma, la recuperación del material arrastrado al concentrado aumenta, lo que resulta en menores grados de concentrado. En el caso de una alta profundidad de espuma, el material arrastrado tiene una mayor probabilidad de drenar la espuma, lo que resulta en mayores grados de concentrado.

2.2. FrothMeter

ECN en conjunto con el CIMAT crearon una aplicación que incluía cámaras Gigabit Ethernet (GigE), instaladas en los bancos de flotación, conectadas en una serie de sistemas embebidos de alta velocidad, las cuales efectúan los procesamientos de las imágenes y dan como resultado información relativa a la capa superficial de burbujas que se encuentra en las celdas de flotación, llamado FrothMeter.

FrothMeter es un sistema de visión artificial analizador de burbujas en celdas de flotación que utiliza sensores de visión y un sistema embebido de procesamiento de imágenes. El sistema analiza las imágenes de burbujas en tiempo real para estimar parámetros tales como velocidad, estabilidad, tamaño y color, enviando resultados

ya sea para monitoreo o alimentando a un sistema de control predictivo basado en modelos (MPC, por sus siglas en inglés).

El sistema de visión FrothMeter, fue desarrollado con el objetivo de beneficiar a los siguientes aspectos de la planta:

- Optimización del control del proceso y uso de reactivos como lo son catalizador, espumante y colector mediante las siguientes mediciones:
 - Medición de Velocidad de Espuma.
 - Medición Tamaño de Burbuja.
 - Medición de Estabilidad.
- Aumento de recuperación en área de flotación, fijando alarmas de estados no deseados de las celdas de flotación, haciendo el monitoreo de estas más fácil y preciso.
- Monitoreo continuo y remoto de las celdas de flotación.
- Almacenamiento de tendencias de estados de espuma y pantalla de análisis que permiten crear históricos de la operación del proceso.

El sistema FrothMeter consiste en: hasta 4 cámaras industriales colocados en puntos de derrame en las celdas de flotación. 1 sistema de procesamiento de alta velocidad y 1 Faceplate para monitoreo y configuración del sistema.

Especificaciones de Sistema:

- GigE / PoE: Cámara de alta definición y alta velocidad.
- Switch GigE / PoE
- Sistema de procesamiento remoto de alta velocidad para hasta 4 cámaras.
- Panel Maestro para conectar diversos sistemas de procesamiento con un Sistema de Control Distribuido (DCS, por sus siglas en inglés) y una interfaz humano-máquina (HMI, por sus siglas en inglés) (véase Fig. 2.2).

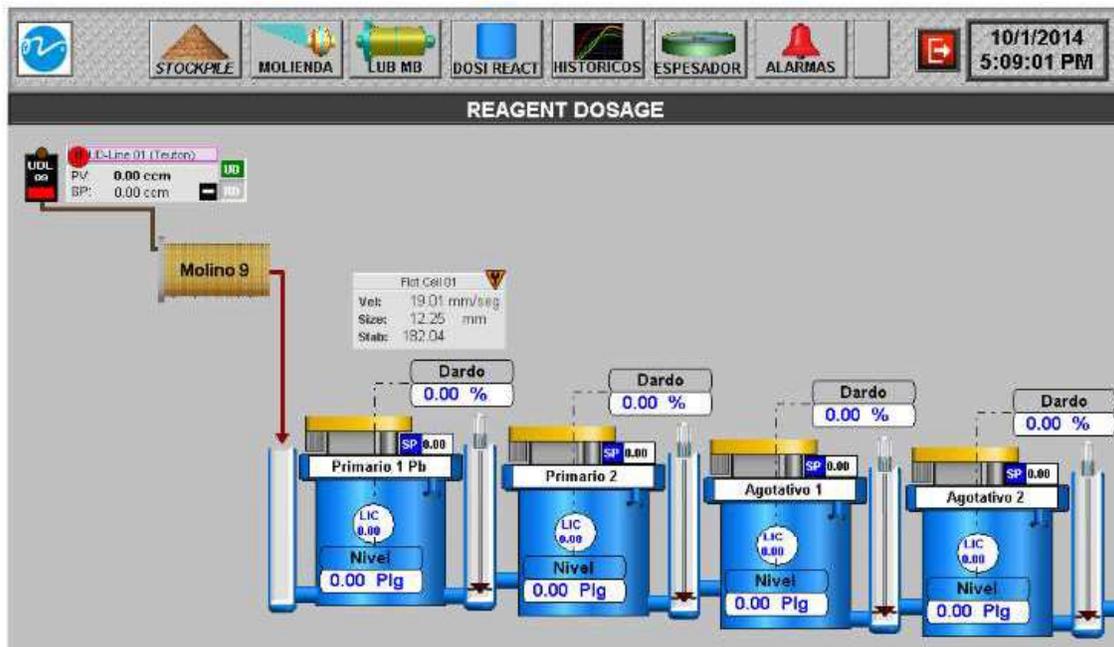


Figura 2.2: Visualización de HMI del sistema de operación. Tomado de [25]

El sistema analiza las imágenes de burbujas para estimar parámetros tales como velocidad, estabilidad y tamaño, enviando resultados al sistema de control mediante una comunicación de tipo industrial (véase Fig. 2.3 y Fig. 2.4).

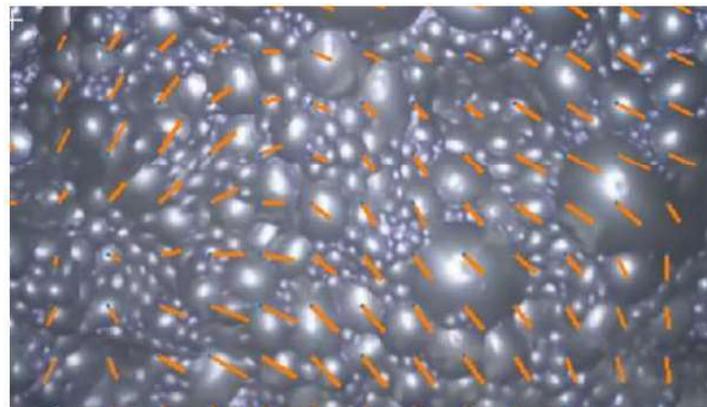


Figura 2.3: Monitoreo de Velocidad y Dirección de Burbuja. Tomado de [25].

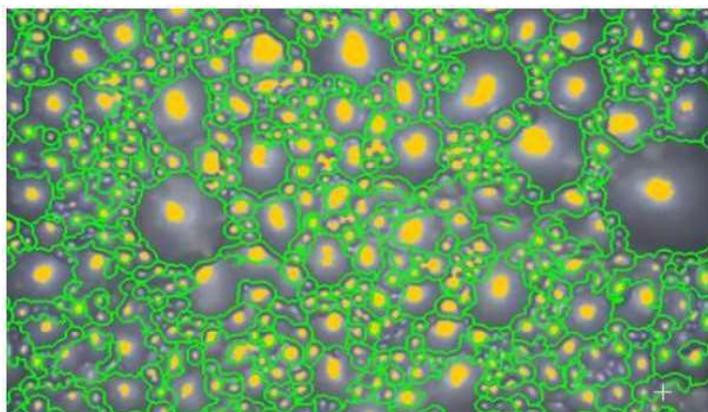


Figura 2.4: Monitoreo de tamaño y Estabilidad de Burbuja. Tomado de [25]

Basándose en la experiencia en el desarrollo de sistemas de visión, y con ayuda de expertos en el área, se identificaron tres tipos primarios de burbuja basado en diferentes estados de la flotación:

1. **Tipo 1: Espuma Acuosa:** Se caracteriza por un bajo concentrado de mineral con una burbuja acuosa, su alta movilidad, su baja estabilidad y tamaño pequeño de burbuja (véase Fig. 2.5).



Figura 2.5: Burbuja caracterizada como no deseable por su tamaño compacto. Tomado de [25].

2. **Tipo 2: Espuma Viscosa:** Se caracteriza por burbujas grandes y de forma elíptica, contiene burbujas con mucha estabilidad y por el efecto arrastre mecánico este tipo de espuma puede atraer partículas no deseables a flotar (véase Fig. 2.6).

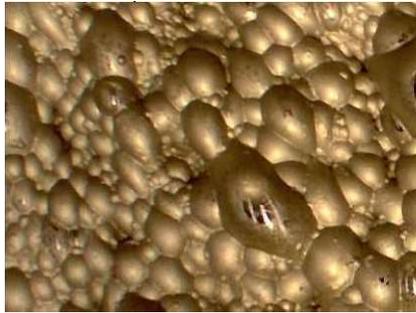


Figura 2.6: Burbuja no desable por su gran tamaño y alta masa de recuperación con bajo grado de pureza. Tomado de [25].

3. **Tipo 3: Espuma Ideal:** Este tipo espuma está dentro de una ventana donde no es muy viscosa o muy acuosa, las burbujas son más grandes que el tipo 1 pero más pequeñas que el tipo 2 (véase Fig. 2.7).

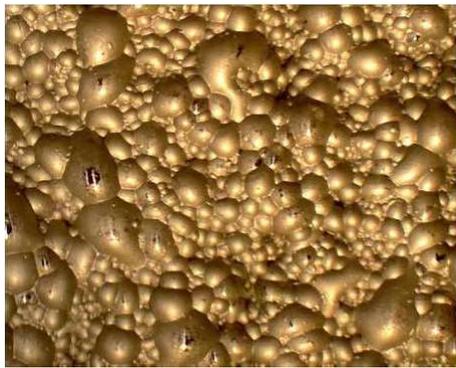


Figura 2.7: Burbuja dentro de los márgenes establecidos en la planta, con el mejor equilibrio entre alta recuperación con alto porcentaje de cobre. Tomado de [25].

2.2.1. Ventajas y desventajas

Los alcances del sistema FrothMeter están determinados por el levantamiento de datos en la planta y las funcionalidades que brinda la puesta en marcha de un sistema para el monitoreo y análisis de video en celdas de flotación.

Ventajas

1. Aumenta las capacidades de monitoreo del equipo de operadores.
2. Contribuye a la mejora en la toma de decisiones de control de parámetros críticos como son la adición de aire y nivel de celda.
3. Analiza las imágenes de burbujas para estimar parámetros tales como velocidad, estabilidad y tamaño, enviando resultados al sistema de control mediante una comunicación de tipo industrial.

Desventajas

1. La iluminación externa al sistema afecta la toma de imágenes en la identificación de burbujas.
2. El análisis de imágenes es aún insuficiente para caracterizar de manera correcta a las burbujas y por ende la estimación de parámetros.
3. No se visualiza en todas las burbujas un punto focal, se observan hasta 2 puntos focales lo que afecta la caracterización de la burbuja.

2.2.2. ECN

Electro Controles del Noroeste S.A. de C.V., ECN ®, fue fundada en la ciudad de Hermosillo, Sonora en el año de 1991 como una empresa integradora de tecnología para la automatización de procesos.

Su misión, contribuir con soluciones integrales, que incrementen el valor de los procesos industriales, a través de una empresa global e innovadora, comprometida con el desarrollo personal y profesional de su capital humano. ECN tiene más de 25 años de experiencia, especializada en la automatización de procesos industriales. Ha desarrollado procedimientos y estándares para el diseño, ingeniería y ejecución de proyectos de automatización, basados en las mejores prácticas de administración de proyectos de acuerdo al Project Management Institute (PMI, por sus siglas en inglés) y las mejores prácticas de integración de sistemas de acuerdo a la Control and System Integrators Association (CSIA, por sus siglas en inglés). Presentan certificaciones como Solution Partner de Allen Bradley y es miembro activo de la CSIA [25].

2.3. Visión Artificial

La ciencia de programar una computadora para procesar videos o imágenes e incluso entenderlos [26], se conoce como visión artificial o como le llaman varios autores visión por computadora. El reconocimiento objetos que se encuentran en el medio ambiente [27] es una función básica, primordial y compleja de la visión artificial, lo que permite que el sistema sea capaz de aprender a reconocer las formas para posteriormente clasificarlas de forma correcta [28], mientras

localización de los objetos posibilita analizar y observar el comportamiento o las trayectorias de los objetos según sea el caso.

La arquitectura y las etapas de un sistema de visión son dependientes de su aplicación, pero existen algunas etapas que se encuentran en la mayoría de las aplicaciones (véase Fig. 2.8).

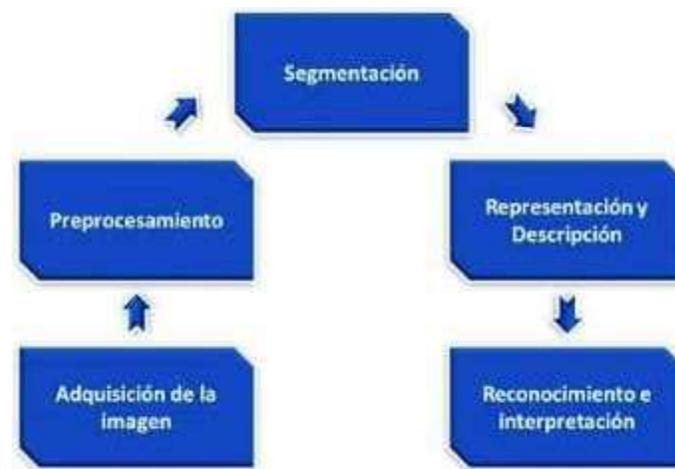


Figura 2.8: Etapas de un Sistema de Visión Artificial. Tomado de [29]

2.3.1. Visión para el reconocimiento de objetos

Se consideran dos tipos de reconocimiento [30] si en la imagen se busca un objeto particular o conocido se realiza una detección del objeto, y si se busca reconocer diferentes instancias de una categoría genérica se realiza el reconocimiento de la instancia. El segundo tipo divide los objetos en diferentes clases, por ejemplo, reconocer dos marcas y modelos de vehículos diferentes, y según ciertas características compartidas ubicarlos en la clase autos. El reconocimiento de objetos está basado en asignar dichas clases a los diferentes objetos, y la herramienta que realiza este proceso se denomina clasificador [31].

2.3.2. Reconocimiento de patrones

Una característica es la unidad elemental utilizada en la representación de los objetos. Los objetos están representados por regiones en una imagen segmentada, los que pueden ser divididos en diferentes conjuntos, que desde el punto de la clasificación, se da según ciertas características comunes que tienen dichos objetos entre sí. Algunas características son medidas de intensidad, color, esquinas, textura,

entre otros; por lo que son descripciones numéricas elementales utilizadas por la descripción de objetos estadísticos. El objeto medido según sus características es denominado *patrón*.

Un patrón no es más que un vector de características $x = (x_1, x_2, x_3, \dots, x_n)$ que describe a un objeto, y el conjunto de todas las formas de patrones posibles es denominado *espacio de características*. Los clasificadores no reconocen objetos, pero sí lo pueden hacer con los patrones que se les presenta. Seguido se muestra una imagen de los pasos a seguir en el reconocimiento de patrones (véase Fig. 2.9).

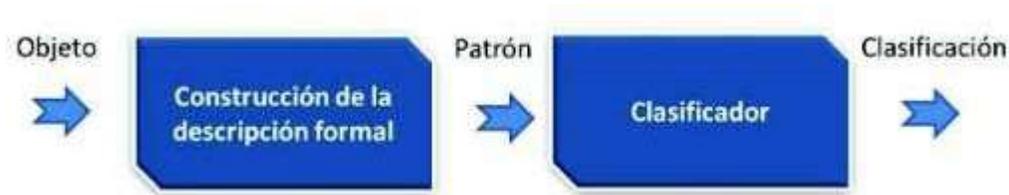


Figura 2.9: Reconocimiento de patrones. Tomado de [29]

2.3.3. Seguimiento de objetos en tiempo real

La mayoría de algoritmos de seguimiento en tiempo real están basados en la detección de cambios en una secuencia de imágenes [32]. Se puede definir que el seguimiento es el problema de estimar la trayectoria de un objeto en el plano de la imagen mientras se mueve en la escena, para eso el sistema seguidor asigna etiquetas fijas al objeto o los objetos a seguir durante la secuencia de imágenes. Las dificultades para el seguimiento incluyen [32] cambios bruscos en el movimiento, el cambio en las apariencias de patrones, tanto de la escena como del objeto mismo y obstrucciones entre objetos.

2.3.4. Procesamiento digital de imágenes

El procesamiento digital de imágenes es el conjunto de técnicas que tienen en general tres grandes objetivos [27]: mejorar, comprender y extraer medidas de una imagen digital de tal manera que facilite el proceso de interpretación visual de la misma por parte de las personas o por el proceso de interpretación de datos en una fase siguiente (véase Fig. 2.10).



Figura 2.10: Etapas del Procesamiento Digital de Imágenes. Tomado de [29]

En la fase adquisición de la imagen se captura la imagen digital, seguido del preprocesamiento que es donde se obtienen imágenes con el mínimo de ruidos no deseados de manera a facilitar el trabajo en las siguientes etapas. En esta etapa se llevan a cabo las operaciones sobre las imágenes.

La segmentación es el proceso por el cual se separan los objetos de la escena de manera a ser analizados individualmente y por último en la fase de extracción de características se convierten los datos como bordes o puntos de la imagen de manera a que sean legibles por el ordenador.

2.4. Inteligencia Artificial

La inteligencia artificial (Artificial Intelligence, AI por sus siglas en inglés) es el campo de la computación dedicado a la creación de sistemas que realizan tareas que generalmente requieren inteligencia humana [33]. En AI, las máquinas completan la tarea en función de las reglas y algoritmos estipulados. AI es un término general para cualquier programa de computadora que simule la inteligencia humana y abarque, aprendizaje automático (Machine Learning, ML por sus siglas en inglés) y aprendizaje profundo (Deep Learning, DL por sus siglas en inglés) (véase Fig. 2.11).

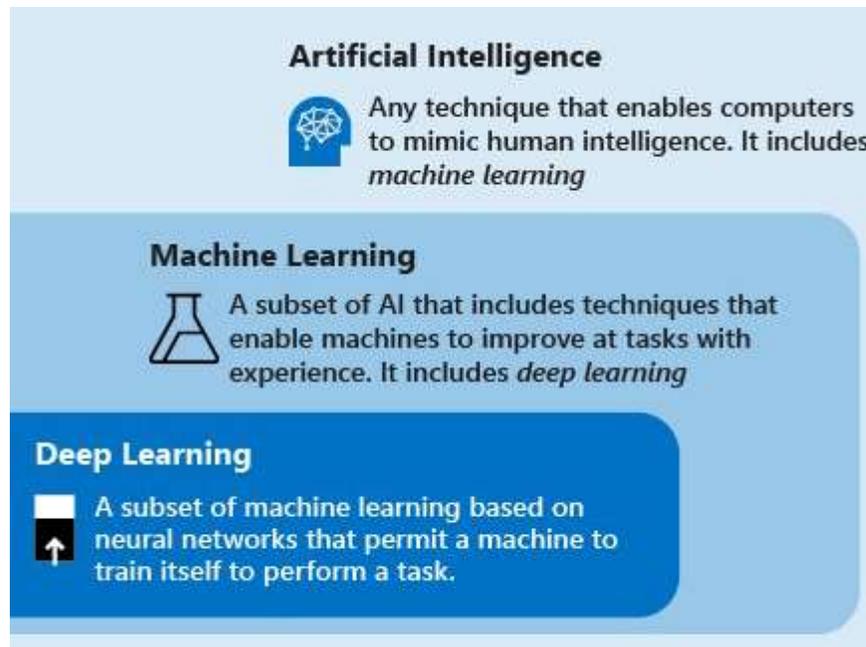


Figura 2.11: El aprendizaje profundo es un subconjunto del aprendizaje automático, que a su vez es un subconjunto de aprendizaje artificial. Tomado de [34]

ML es un subconjunto de IA, que incluye todos los enfoques que permiten a las máquinas aprender de los datos sin haber sido programados explícitamente [35]. El objetivo del ML es entrenar máquinas basadas en los datos y algoritmos proporcionados, y utilizando estos datos procesados y la información, las máquinas aprenden a tomar decisiones.

DL es un subconjunto de ML e incorpora modelos computacionales y algoritmos que imitan la arquitectura de las redes neuronales biológicas en el cerebro [36].

2.4.1. Aprendizaje Automático

En el avance tanto del análisis de datos como de la IA ha tenido un impacto fundamental el ML. Es una tendencia en la última década, el uso de algoritmos de aprendizaje automático, lo cual ha popularizado muchas posibilidades para la toma de decisiones autónomas. El aprendizaje automático permite detectar y predecir patrones [37] mostrando así ser efectivo en aplicaciones de múltiples ámbitos. Actualmente se está aplicando en muchas áreas, tareas tales como el reconocimiento de voz y la traducción entre idiomas, reconocimiento de imágenes y video, la navegación autónoma de vehículos y los sistemas de

recomendación. Algunas de estas tareas no eran factibles anteriormente, pero hoy con los recientes avances en el aprendizaje automático se han hecho posibles.

Como una de las disciplinas de la IA, se emplea bajo términos estadísticos, probabilísticos y de técnicas de optimización, para así poder comprender reglas complicadas y características de dimensión alta.

Una definición aceptada de ML ampliamente por especialistas del área es: si se aplica un algoritmo ML a un conjunto de datos y a algún conocimiento sobre estos datos, entonces el algoritmo puede aprender de los datos de entrenamiento y aplicar lo que ha aprendido para hacer una. Si se optimizan los parámetros del algoritmo de modo que su rendimiento mejore, entonces se considera que está aprendiendo esa tarea.

Los algoritmos para la capacitación de un modelo tienen como objetivo minimizar el error de predicción del modelo en el conjunto de datos de entrenamiento y, por lo tanto, en ocasiones pueden ajustarse en exceso a este conjunto de datos, lo que produce modelos que funcionan mejor en las entradas de entrenamiento que en las entradas de validación las cuales son extraídas del mismo registro de datos pero no utilizadas durante el entrenamiento. Para tratar este punto se han propuesto varias técnicas de regularización para evitar que los modelos se sobreajusten a sus conjuntos de datos de entrenamiento mientras se minimiza su error de predicción [38].

Los algoritmos de ML nos permiten comprender y analizar mejor datos complejos. En la entrenamiento de un modelo de ML existen dos tipos de aprendizaje: supervisado y no supervisado.

2.4.1.1. Aprendizaje Supervisado

En el aprendizaje supervisado los registros de entrenamiento, son las entradas del modelo, se les asignan etiquetas las cuales sería las salidas del modelo.

En la presente investigación, el aprendizaje supervisado significa obtener conocimiento al usar imágenes de ejemplos de las burbujas que contienen

información importante, siendo más específicos, etiquetas pequeñas, mediana y grande, y aplicar la conocimiento adquirido para predecir el tamaño de las burbujas en imágenes nuevas no vistas anteriormente durante el aprendizaje. Los algoritmos de aprendizaje supervisados incluyen Naive Bayes [40], Linear Regression [41], Logistic Regression [39], Decision Trees [42], K-Nearest Neighbor [43], Random Forests [44], AdaBoost, Support Vector Machine (SVM por sus siglas en inglés) [45], métodos de redes neuronales (Neural Networks, NN por sus siglas en inglés) [46], entre otros.

2.4.1.1. Aprendizaje No Supervisado

El objetivo del aprendizaje no supervisado es extraer características útiles de los datos no etiquetados y construir un modelo que explique su estructura oculta. Lo que se busca en el entrenamiento es aprender la relación entre los datos y las etiquetas, para así poder construir un modelo que pueda generalizarse a otros registros de datos más allá del conjunto de entrenamiento [38].

En el aprendizaje no supervisado, los datos se procesan con el objetivo de separar las imágenes en grupos con características similares, en nuestro caso, las imágenes que representan burbujas pequeñas, medianas y grandes, o se pueden formar además otros grupos en dependencia de la similitud de características de los datos. Esto sucede sin que al algoritmo se le proporcione información sobre los grupos, donde se determina cuántos grupos hay y cómo separarlos. Los algoritmos de aprendizaje no supervisados incluyen K-Means [47], Mean Shift [48], Hierarchical Clustering [50], DBSCAN [51], método de mezcla de Gaussianas [52], entre otros.

2.4.2. Aprendizaje Profundo

El aprendizaje profundo, también conocido como aprendizaje de redes neuronales profundas (Deep Neural Networks, DNN por sus siglas en inglés), es un área de investigación nueva y popular que está dando resultados impresionantes y está creciendo rápidamente.

Las DNN emplean arquitecturas profundas para aprender de manera automática características a partir de los datos de entrada sin necesidad de ingeniería de características [53]. Los métodos de DL se caracterizan por presentar arquitecturas

profundas con múltiples capas ocultas que les permiten aprender muchos niveles de abstracción, las cuales fueron inspiradas en cómo el cerebro de los mamíferos usa diferentes áreas de la corteza para abstraer diferentes niveles de características cuando se les da una percepción de entrada, a diferencia de las arquitecturas artificiales con 1 o 2 capas ocultas [54]. Las tecnologías creadas con DL van más allá del aprendizaje automático tradicional y los enfoques de AI que dependen de características creadas por el hombre, pues su capacidad de abstracción automática de características de alto nivel combinadas con el crecimiento exponencial de datos, ya han producido hace varios años resultados innovadores en detección automática de objetos [55], la segmentación [56] y análisis en imágenes, procesamiento de lenguaje natural (Natural Language Processing, NLP por sus siglas en inglés) y el reconocimiento automático del habla [57]. Existen herramientas de algoritmos en DL, a saber, las Máquinas de Boltzman restringidas, los Autoencoders (AE) y Redes Neuronales Convolucionales Profundas (Deep Convolutional Neural Networks, DCNN por sus siglas en inglés) [58]. Nos centraremos en los DCNN porque estos se aplican más comúnmente a las imágenes.

En el procesamiento de datos visuales, como imágenes y videos, las DCNN han demostrado ser altamente efectivas. Las mismas, toman datos de entrada sin procesar en el nivel más bajo y los transforman al procesarlos a través de una secuencia de unidades computacionales básicas para obtener representaciones que constituyen valores característicos para la clasificación en las capas superiores [59], [60].

El desarrollo con DCNN implica una gran cantidad de paralelismo en los cálculos, para esto existen varios marcos de trabajo de DL populares de código abierto, como CNTK de Microsoft [61], Theano [62], Tensorflow de Google [57], PyTorch [63], Caffe [64], Keras, por mencionar los más importantes.

2.4.2.1. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (Convolutional Neural Networks, CNN por sus siglas en inglés) pertenecen a una clase de redes neuronales artificiales

profundas y avanzadas que se han aplicado con éxito al análisis de imágenes en muchas disciplinas diferentes. Estas redes son variantes inspiradas biológicamente de perceptrones multicapa que emulan la corteza visual animal, el sistema de procesamiento visual más poderoso que existe[65].

Las CNN profundas actualmente define un nuevo enfoque en el reconocimiento de imágenes y su aplicación exitosa está creciendo rápidamente en una amplia gama de disciplinas técnicas diversas, como la identificación de tumores [66] [67], baldosas de granito [68], ganado individual en granjas [69], detección de ocupación en estacionamientos [70] y reconocimiento de marcha [71], por mencionar solo algunos ejemplos.

Estudios anteriores [72][73] han demostrado el potencial de las CNN para extraer características de las imágenes de espuma que pueden usarse para estimar los grados de platino. En [72] utilizaron una versión pre-entrenada de AlexNet [74], e identificaron cuatro estados de espuma con una confiabilidad de más del 80%; lo cual fue significativamente mejor de lo que se podría obtener con algoritmos de estado de técnicas anteriores.

2.4.2.2. Arquitectura de una Red Neuronal Convolutiva

Las arquitecturas CNN contemporáneas comprenden capas convolucionales, agrupadas y totalmente conectadas (véase Fig. 2.12).

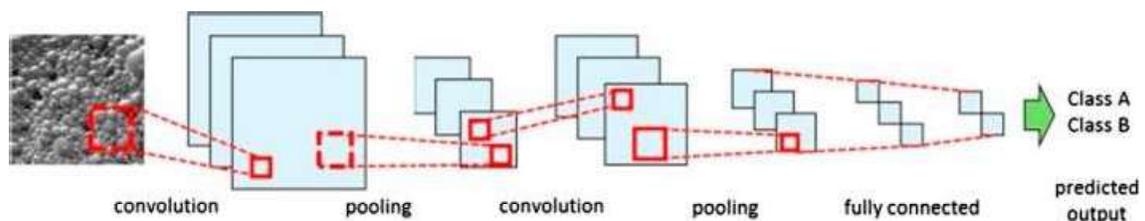


Figura 2.12: Arquitectura típica de una red neuronal convolutiva. Tomado de [65]

En esta figura se presentan los datos (imágenes) a la red en la capa de entrada. A esto le sigue la convolución de la imagen en capas convolucionales que contienen una serie de filtros de tamaño fijo o kernels convolucionales, cuya operación básica se ilustra a continuación (véase Fig. 2.13).

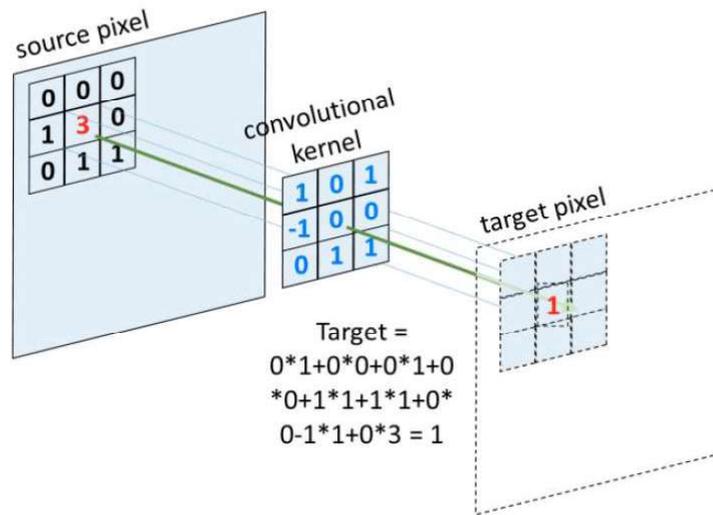


Figura 2.13: Operación de un kernel de convolución. Tomado de [75]

En esta figura, un kernel o filtro se mueve a través de una imagen de origen, un píxel a la vez, para generar una imagen de destino o un mapa de características. Esto se hace aplicando un operador de convolución a cada píxel y sus vecinos, como se indica en la Figura 14.

Cada convolución genera un mapa de características, que se reduce en la muestra mediante una capa de agrupación (véase Fig. 2.14 y Fig. 2.15).

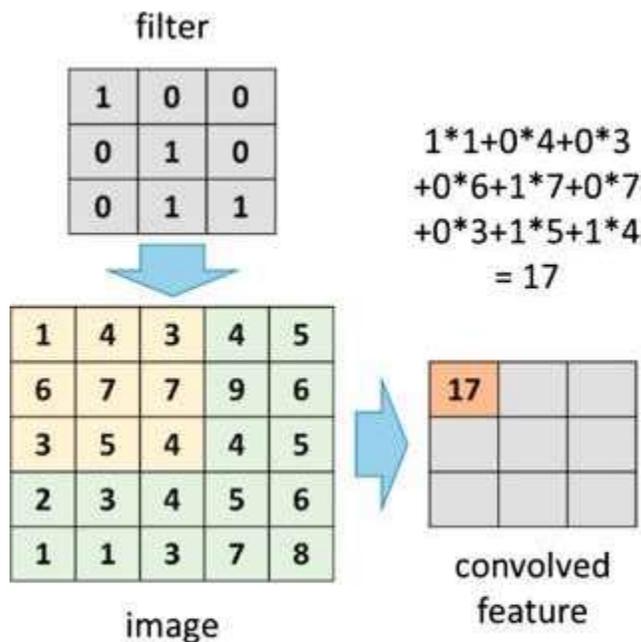


Figura 2.14: Convolución de una imagen de 5×5 con un filtro de 3×3 , lo que resulta en un conjunto más pequeño de características de 3×3 (solo se muestra la primera). Tomado de [65]

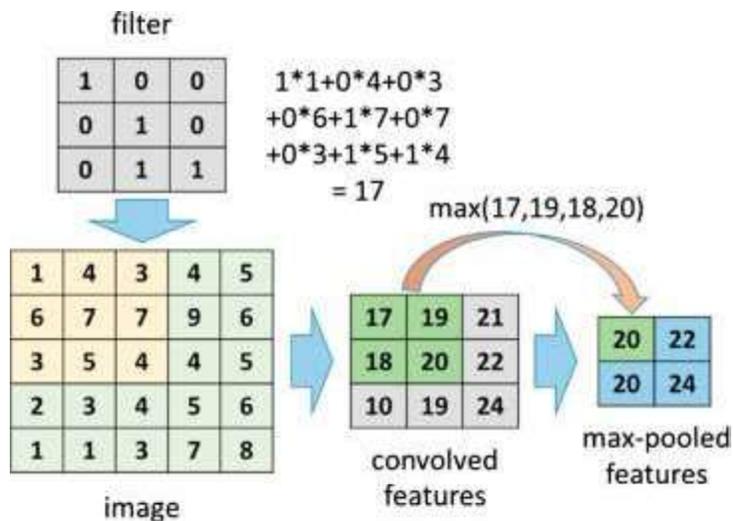


Figura 2.15: Agrupación máxima de un conjunto 3×3 de características convueltas que resulta en un conjunto 2×2 de características agrupadas máximas. Tomado de [65]

En estas figuras se observa que la agrupación máxima se usa con frecuencia, donde se conserva el valor máximo de un vecindario en el mapa de características. Esto reduce la escala de la imagen para reducir el número de pesos con cada capa.

Las capas finales completamente conectadas de la red neuronal convolucional operan de manera similar a las de un perceptrón multicapa con la capa de características precedente que sirve como predictores de la categoría o clase a la que pertenece la imagen.

La principal ventaja de las CNN sobre las redes neuronales profundas totalmente conectadas tradicionales es que tienen comparativamente menos parámetros que aprender. Las capas convolucionales con kernels pequeños son un medio eficaz para extraer características de alto nivel que se suministran a las capas totalmente conectadas. El entrenamiento de las CNN se realiza mediante el uso de la propagación hacia atrás y el descenso de gradiente estocástico [76].

Como redes neuronales profundas, las CNN tienen una gran capacidad para capturar características complejas de datos de imagen y en los últimos años, estas redes han surgido como enfoques avanzados para el reconocimiento de imágenes, a menudo superando a los enfoques tradicionales por un amplio margen [77] [78].

2.4.2.3. Modelos Redes Neuronales Convolucionales Preentrenados

La construcción y el entrenamiento efectivo de redes neuronales profundas requieren cantidades masivas de datos y recursos computacionales de alto nivel. Un enfoque para aliviar este problema es hacer uso de un proceso llamado transferencia de aprendizaje. Esto es posible, gracias a la disponibilidad pública de una serie de arquitecturas neuronales convolucionales profundas que se han capacitado en una gran base de datos de imágenes de objetos comunes. Estos modelos se pueden usar tal como están, con una eficacia notable, o se pueden usar como punto de partida para entrenamiento adicional, ya que ya han capturado características de bajo nivel, como líneas, curvas y formas.

Estos modelos han sido pre-entrenados en la base de datos de ImageNet [79] que consiste en más de 14 millones de imágenes, distribuidas en un total de 1000 categorías diferentes [80][81], algunas de estas arquitecturas son AlexNet[74], VGGNet [77], ResNet [55], InceptionV3 [82], InceptionResNetV2 [83] y Xception [84]. Como resultado, se pueden usar en otras bases de datos de imágenes sin

necesidad de reentrenamiento, excepto para la modificación de la capa final de la red de acuerdo con el problema específico en cuestión. Alternativamente, las características generadas por la penúltima capa de la red pueden usarse simplemente junto con un modelo de clasificación de elección. Ejemplos de estas aplicaciones están creciendo rápidamente en la literatura, por ejemplo en histopatología [67], teledetección [85], agricultura [86], detección de ocupación en estacionamiento [70], entre otros.

2.5. Trabajos relacionados

En esta sección se presentan diversos trabajos para tratar la problemática de visión artificial en el proceso de flotación de minerales, los cuales resultan relevantes para poder desarrollar la presente investigación a partir de propuestas y soluciones que algunos investigadores plantean para problemáticas afines.

En el trabajo *“Predicting bubble size and bubble rate data in water and in froth flotation-like slurry from computational fluid dynamics (CFD) by applying deep neural networks (DNN)”* [87], se emplean DNN para predecir el tamaño y velocidad de la burbuja utilizando datos obtenidos de cálculos de dinámica de fluidos computacional (CFD) validados. Se presentan los estudios de casos de agua pura y suspensión en condiciones similares a las empleadas en la flotación de espuma de minerales. Se demuestra que la DNN puede predecir los resultados de CFD con precisión al usar cuatro capas ocultas, mostrando errores relativos calculados entre los datos de CFD y la predicción obtenida por el DNN tan bajos como 8.8% y 1.8% para el tamaño de burbuja y la velocidad de burbuja, respectivamente.

En el artículo *“Sistema de medición basado en visión artificial para circuito rougher de laboratorio”* [88], se implementó un sistema de medición de profundidad de espuma y flujo de aire basado en visión artificial. Desarrollaron un algoritmo de visión artificial, que permite medir la profundidad de espuma y el flujo de aire, en base al procesamiento de imágenes obtenidas por medio de cámaras web ubicada en el laboratorio

En el artículo *“Using Convolutional Neural Networks to Develop State-of-the-Art Flotation Froth Image Sensors”* [75], se muestra cómo se puede utilizar una arquitectura de red neuronal pre-entrenada, a saber, VGG16, para obtener mejoras significativas en los sensores de imagen de espuma. De igual forma, se plantea como una pequeña modificación de la arquitectura de la red también puede acelerar el desarrollo de los modelos, lo que se demuestra en un estudio de caso que involucra un conjunto de datos de imágenes de una planta industrial de metales del grupo platino.

En el artículo *“Froth image analysis by use of transfer learning and convolutional neural networks”* [72], se describe el uso de AlexNet en una base de datos de imágenes de objetos comunes para extraer características de las imágenes de espuma de flotación. Se consideran dos estudios de caso, donde se analizan cómo las características de espuma en una planta de flotación industrial podrían identificarse de manera más confiable con las características generadas por AlexNet que con los enfoques de vanguardia anteriores.

En la investigación titulada, *“Flotation froth image recognition with convolutional neural networks”* [65], trata de la estimación de los grados de espuma a partir de datos de imagen industriales donde se considera el uso de tres arquitecturas de CNN, a saber, AlexNet, VGG16 y ResNet. En el mismo, AlexNet en la etapa de pre-entrenamiento superó los métodos propuestos previamente por un margen significativo, lo que podría aumentarse notablemente a través de una modificación parcial de las redes VGG16 y ResNet34.

Capítulo 3

3 Aprendizaje Automático y Minería de Datos

3.1. Metodología

Con demasiada frecuencia existe la tendencia de pasar por alto la metodología y saltar directamente a resolver un problema con ML. Sin embargo, hacerlo de esta manera no se establece una base sólida y no es posible asegurar criterios de calidad para que el proceso de ML pueda ser integrado en un proceso productivo. Por tanto, seguir una metodología más a menudo nos acerca al problema que estábamos tratando de resolver. A continuación, se muestra la metodología seguida para el desarrollo del modelo de AI (véase Fig. 3.1).

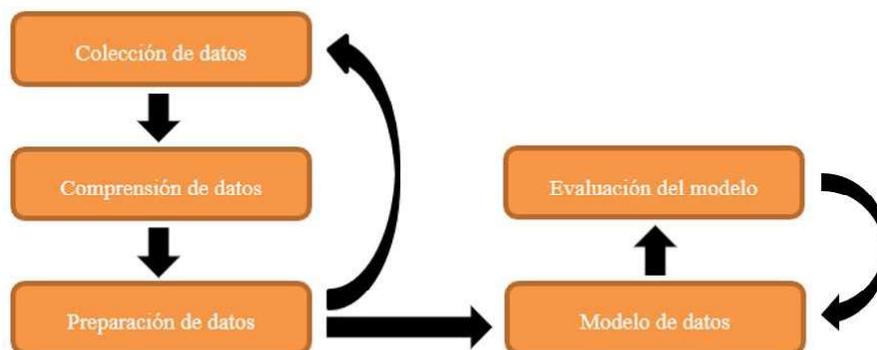


Figura 3.1: Metodología para el desarrollo del modelo de IA.

3.2 Exploración de datos

Las técnicas de ML han tenido en una amplia gama de aplicaciones, como visión por computadora, procesamiento del habla, comprensión del lenguaje natural, neurociencia, salud e internet de las cosas; dado que en la era de los grandes datos el ML ha despertado amplios intereses. Por una parte, los grandes datos proporcionan información sin precedentes para algoritmos de ML para extraer patrones subyacentes y construir modelos predictivos; por otra parte, los algoritmos

tradicionales de ML enfrentan desafíos como la escalabilidad para liberar realmente el significado de los grandes datos.

El ML surge en el área de informática, y más en concreto, de la AI. ML se basa en cómo construir un sistema informático que mejore automáticamente a través de la experiencia [89]. El problema de aprender de la experiencia con respecto a algunas tareas y medidas de desempeño, se conoce como un problema de ML; lo que permite descubrir la estructura subyacente y hacer predicciones a partir de conjuntos de datos. ML prospera con técnicas de aprendizaje eficientes (algoritmos), datos ricos y / o grandes y entornos informáticos potentes.

Dicho con otras palabras: la máquina no se programa para que responda de una determinada forma según las entradas recibidas, sino más bien para que extraiga patrones de comportamiento a partir de las entradas recibidas, y en base a dicha información aprendida o asimilada, realice la evaluación de nuevas entradas. Los algoritmos internos que constituyen la base de este aprendizaje tienen un fuerte componente estadístico y algebraico, con la consiguiente capacidad de cálculo.

3.2.1. Colección de Datos

Los datos de imagen de las burbujas fueron tomadas en una celda de flotación de una planta minera de Sonora mediante el uso de cámaras Giga Ethernet (GigE) con un lente de 16mm. Durante el período de toma de las imágenes, se hizo variación sobre la adición de reactivos y el control de flujo de aire para obtener diferentes tipos de burbujas.

Las imágenes de 499 x 499 píxeles se asociaron con tres diferentes clasificaciones de burbujas según la clasificación de los operadores, burbuja pequeña (clase 1), burbuja mediana (clase 2) y burbuja grande (clase3) (véase Fig. 18). Estas clases fueron esencialmente distinguidas por el tamaño de la burbuja en el banco de flotación. Un total de 553 imágenes en los datos fue lo obtenido al final del proceso (véase Fig. 3.2).

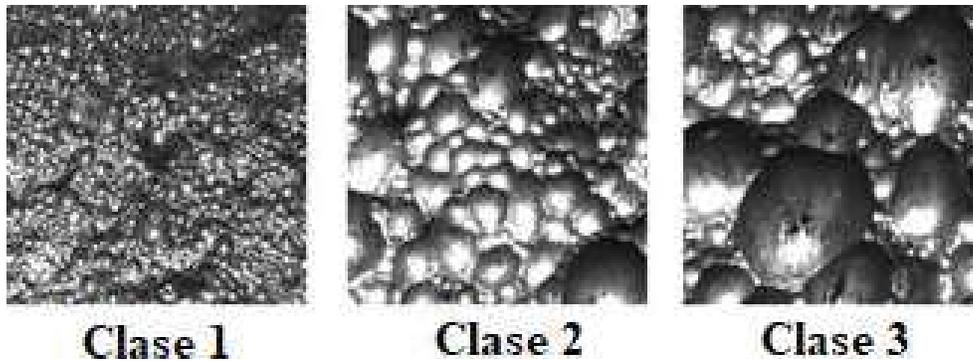


Figura 3.2: Imágenes muestras de cada una de las clases de capa superficial de espuma

Como se indica en el parte inferior de la Fig. 3.2, cada clase de burbuja está asociada con un valor, es decir, Clase 1 burbuja pequeña compuesta por 200 imágenes, Clase 2 burbuja mediana compuesta por 141 imágenes y Clase 3 burbuja grande compuesta por 212 imágenes. El conjunto de datos de entrenamiento consistió en 75% de las imágenes seleccionadas al azar, mientras que el resto sirvió como datos de prueba para validar el modelo.

3.2.2. Comprensión de Datos y Preparación de Datos

Con los datos recopilados y realizada la clasificación de las imágenes por parte de los expertos se comenzó a trabajar sobre el significado de los datos. Si bien se tenía una clasificación de las clases de las burbujas, las cuales estaban divididas en tres, nos preguntamos si se pudiese encontrar otro número de clases el cual fuera óptimo para el problema a tratar.

En el conjunto de datos se detectaron varias imágenes las cuales presentan problemas de iluminación, por tanto, se realizó un preprocesamiento de las mismas para tratar este problema.

3.2.3. Coincidencia de Histogramas

Se hace uso de la librería scikit-image [90], la cual contiene una colección de algoritmos para el procesamiento de imágenes dentro de los cuales se encuentra la coincidencia de histogramas (histogram matching, en inglés).

La coincidencia de histogramas [91] manipula los píxeles de una imagen fuente con una imagen de referencia, haciendo uso del histograma de colores de ambas imágenes. El número de canales de colores entre la imagen fuente y la imagen de

referencia deben ser igual en todo momento, por tanto, la coincidencia se realiza de forma independiente por cada canal. La coincidencia de histogramas se puede usar como una normalización ligera para el procesamiento de imágenes, como la coincidencia de características (véase Fig. 3.3), especialmente en circunstancias en las que las imágenes se han tomado de diferentes fuentes o en diferentes condiciones, por ejemplo diferente iluminación.



Figura 3.3: Resultado de aplicar la normalización de colores. Tomado de [90]

Para ver el resultado de la coincidencia de histogramas, se traza para cada canal RGB, el histograma y el histograma acumulativo (véase Fig. 3.4).

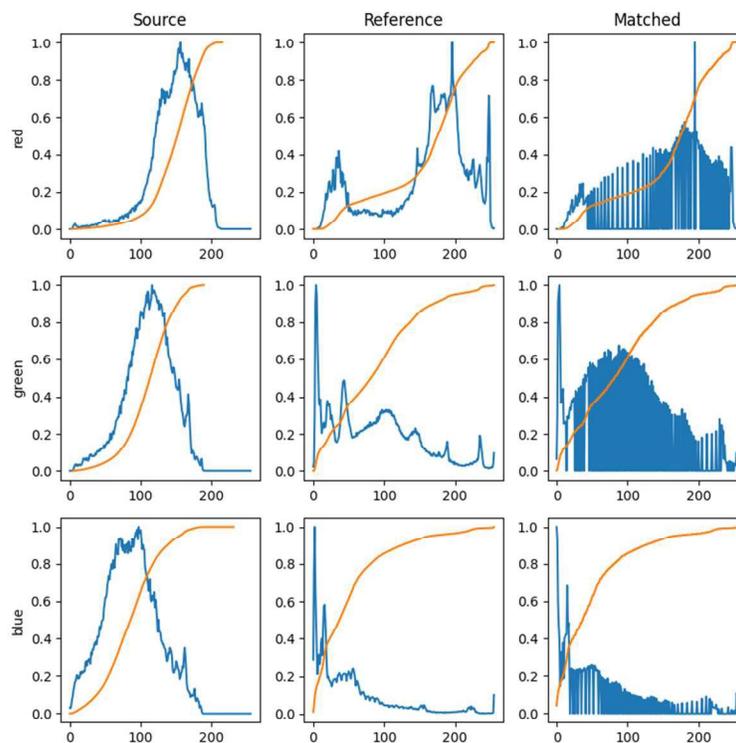


Figura 3.4: Coincidencia de histograma de colores trazado por cada canal RGB. Tomado de [90]

A continuación, se muestra el resultado de aplicar la coincidencia de histogramas a las imágenes que tomamos la decisión de normalizar (véase Fig. 3.5) y el histograma de colores trazado por cada canal RGB, donde se observa como la imagen coincidente tiene el mismo histograma acumulativo que la imagen de referencia (véase Fig. 3.6).

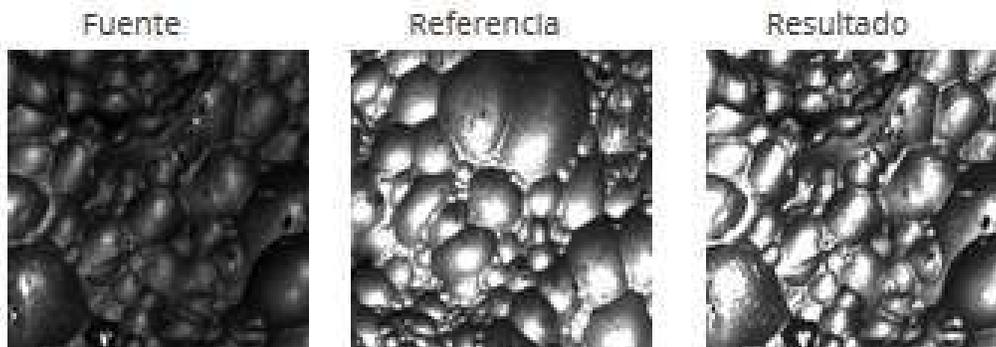


Figura 3.5: Imagen resultada de aplicar la normalización de colores sobre la imagen fuente

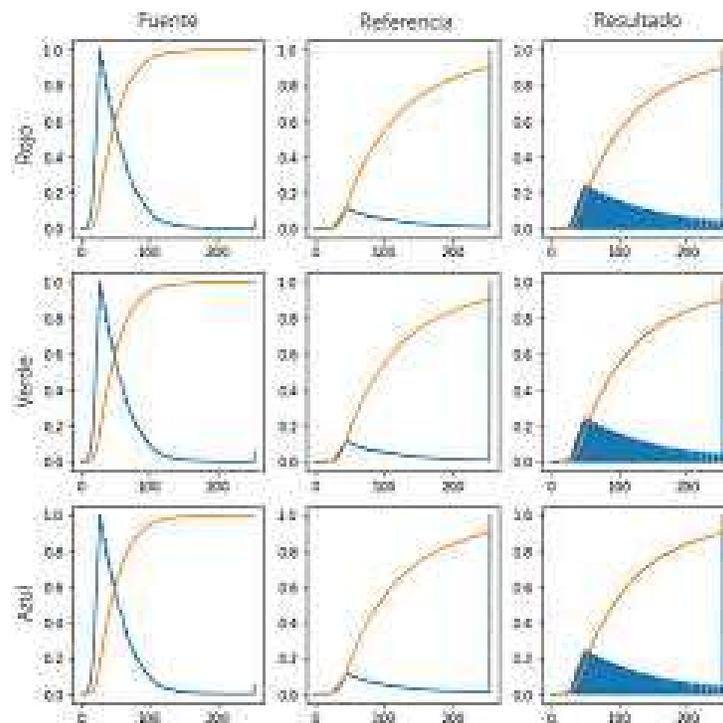


Figura 3.6: Resultado de histogramas de colores trazado por cada canal RGB

3.2.4. Reagrupamiento de Atributos

En aprendizaje automático, el reagrupamiento de atributos se ha utilizado especialmente para la visualización de los atributos de los datos, donde se ha desarrollado un subcampo conocido como reducción de dimensionalidad. En este campo se han desarrollado varios métodos tales como, Kernel PCA [92][93], Independent Component Analysis [94], análisis de componentes principales (Principal Component Analysis, PCA por sus siglas en inglés) PCA [95][96], m-SNE [97], t-Distributed Stochastic Neighbor Embedding (t-SNE, por sus siglas en inglés) [98] y otros.

Seguido a esto, nos centramos en aplicar los métodos PCA y t-SNE, dado que en el estado del arte de este campo son los algoritmos más populares de reducción de dimensionalidad.

3.2.4.1. t-SNE

t-SNE es un método para visualizar datos de alta dimensión. Convierte similitudes entre los puntos de datos en probabilidades conjuntas e intenta minimizar la divergencia de Kullback-Leibler entre las probabilidades conjuntas de la incrustación de baja dimensión y los datos de alta dimensión. t-SNE tiene una función de costo que no es convexa, es decir, con diferentes inicializaciones podemos obtener diferentes resultados [99].

Aplicando t-SNE, se obtuvo el siguiente resultado (véase Fig. 3.7).

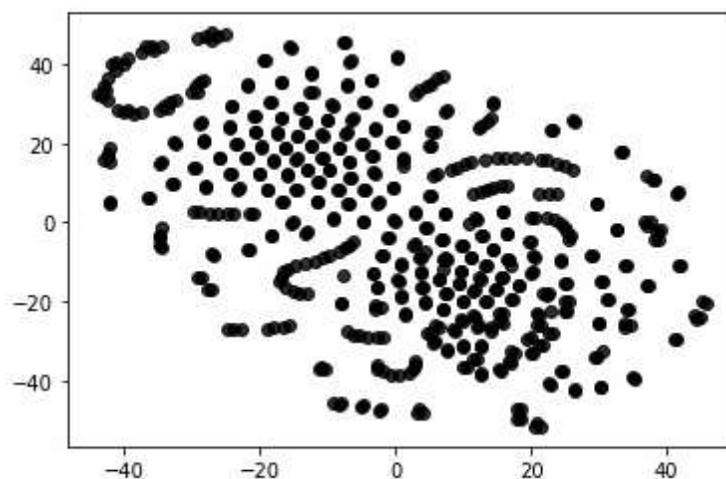


Figura 3.7: Resultado de aplicar t-SNE

3.2.4.2. PCA

El análisis de componentes principales es una de las principales técnicas para la reducción de dimensionalidad en un conjunto de datos mientras que conserva la mayor parte de la información. Se utiliza para conocer cómo están correlacionados los datos y proporciona un número mínimo de variables que mantengan la máxima cantidad de variación o información sobre cómo se distribuyen los datos originales.

Aplicando PCA a nuestro conjunto de datos y graficando las dos primeras variables que más explican la distribución de los datos, se obtuvo el siguiente resultado (véase Fig. 3.8).

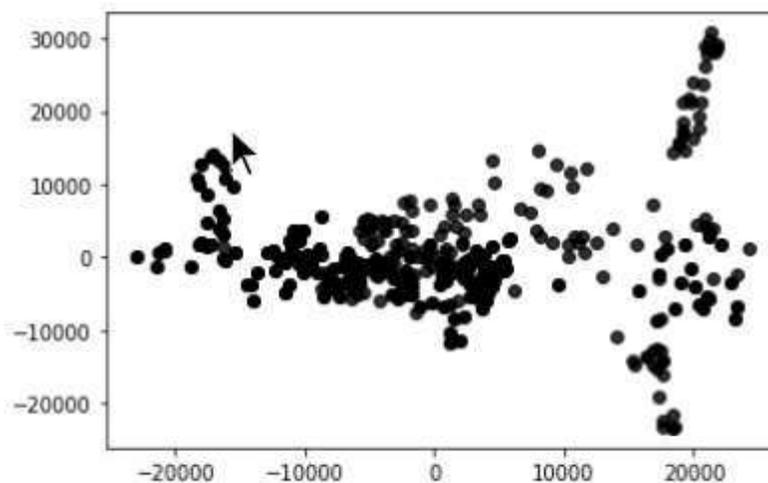


Figura 3.8: PCA sobre las dos primeras variables

Para comparar los resultados de los métodos de visualización es necesario tomar varios conjuntos de datos complejos, hacer diagramas y discutirlos. Una vez discutidos estos diagramas se tomó la decisión de continuar con los resultados mostrados por PCA.

Con este resultado del PCA no se obtuvo una separación clara de los datos, por lo que se aplicó la transformada Fourier para analizar de otra forma los datos. La transformada de Fourier [100] analiza los componentes de frecuencia en una señal bidimensional, como el sonido o la onda de radio. Toma datos bidimensionales como entrada y devuelve otros datos bidimensionales como números complejos, lo que permite encontrar variaciones en las frecuencias. En el procesamiento de

imágenes se utiliza mucho pues permite encontrar variaciones de colores. Al aplicar esta técnica previamente a nuestros datos y seguidamente aplicar PCA, obtuvimos la siguiente distribución (véase Fig. 3.9).

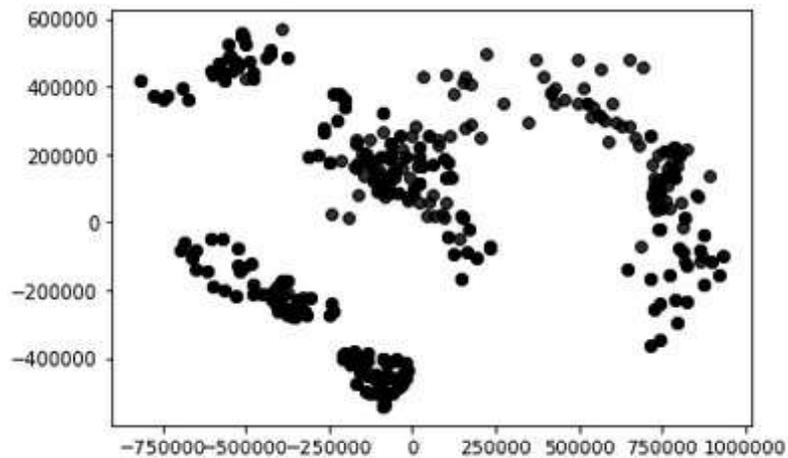


Figura 3.9: PCA sobre las dos primeras variables usando la transformada de Fourier

3.2.5. Reagrupamiento de Ejemplos Usando el Método de Mezcla de Gaussianas

El reagrupamiento de ejemplos, en términos básicos agrupación, tiene como objetivo encontrar diferentes grupos dentro de los elementos en los datos. Para esto, los algoritmos encuentran estructuras en los datos para que los elementos del mismo grupo sean más similares entre sí que con el resto de los grupos.

A continuación, se desarrollan los métodos de agrupamiento que mostraron mejores resultados después de aplicar K-Means, Hierarchical Clustering, DBSCAN y y mezcla de Gaussianas.

3.2.5.1. K-Means

El algoritmo de K-Means agrupa los datos tratando de separar muestras en n grupos o clústers de igual varianza, minimizando el criterio de suma de cuadrados dentro del grupo. Este algoritmo requiere que se especifique el número de clústeres, se adapta bien a un gran número de muestras y se ha utilizado en una amplia gama de áreas de aplicación en muchos campos diferentes [99].

Al aplicar K-Means a nuestros datos acomodados con PCA, se obtuvo el siguiente resultado para diferentes números de clústeres (véase Fig. 3.10).

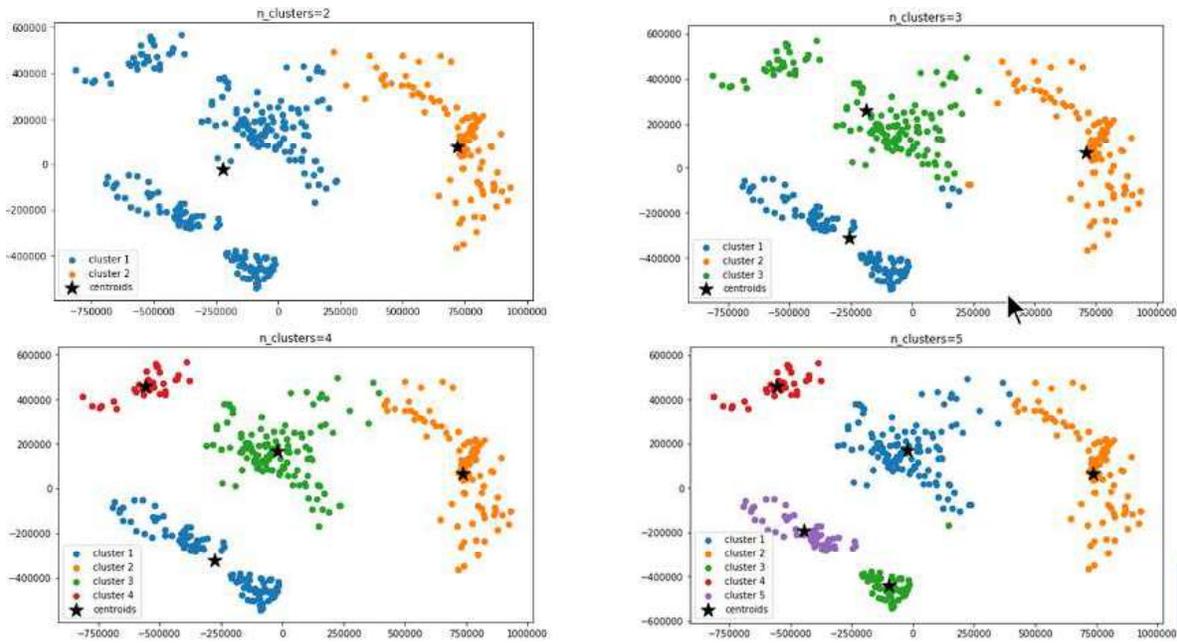


Figura 3.10: Algoritmo de K-Means para diferentes números de clústers

Para conocer el número óptimo de clústers a formar tras aplicar la técnica de K-Means para diferentes números, se utilizó el Método de Validación de Silueta [101]. Este método trabaja sobre la distancia media entre una muestra y todos los demás puntos en el mismo grupo, además de la distancia media entre una muestra y todos los demás puntos en el siguiente grupo más cercano. El método de silueta permite conocer cuán compactos y separados se encuentran los grupos, por lo que mientras más se acerque al puntaje uno, mayor es la agrupación (véase Fig. 3.11).

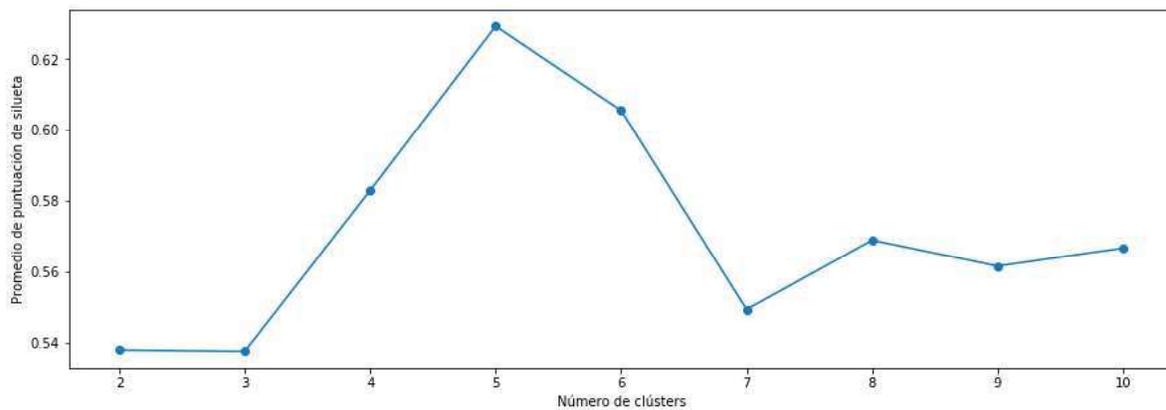


Figura 3.113: Método de silueta sobre K-Means

3.2.5.2. Mezcla de Gaussianas

El método de mezcla de Gaussianas es un modelo probabilístico que supone que todos los puntos de datos se generan a partir de una mezcla de un número finito de distribuciones gaussianas con parámetros desconocidos. Pertenecer al grupo de algoritmos de agrupación suave en los que cada punto de datos pertenecerá a cada grupo existente en el conjunto de datos, pero con diferentes niveles de pertenencia a cada grupo, donde se asigna la probabilidad, que va de 0 a 1, de cada dato de pertenecer a un determinado grupo.

A continuación, se muestra el resultado obtenido aplicando este método (véase Fig. 3.12).

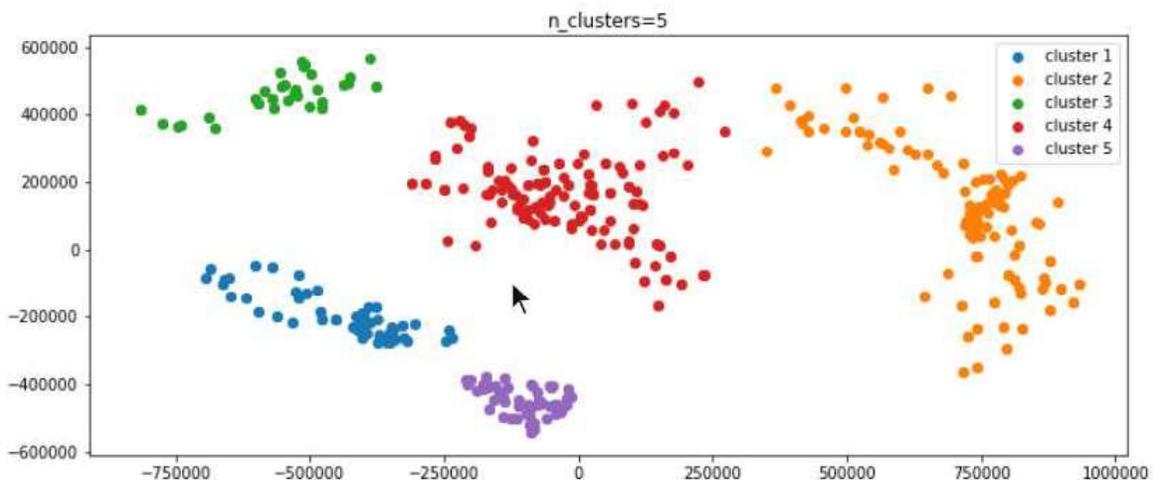


Figura 3.12: Método de Mezcla de Gaussianas para 5 clústers

Haciendo uso del método de validación de silueta [101], también validamos que para 5 clústers era el número óptimo para hacer la agrupación (véase Fig. 3.13).

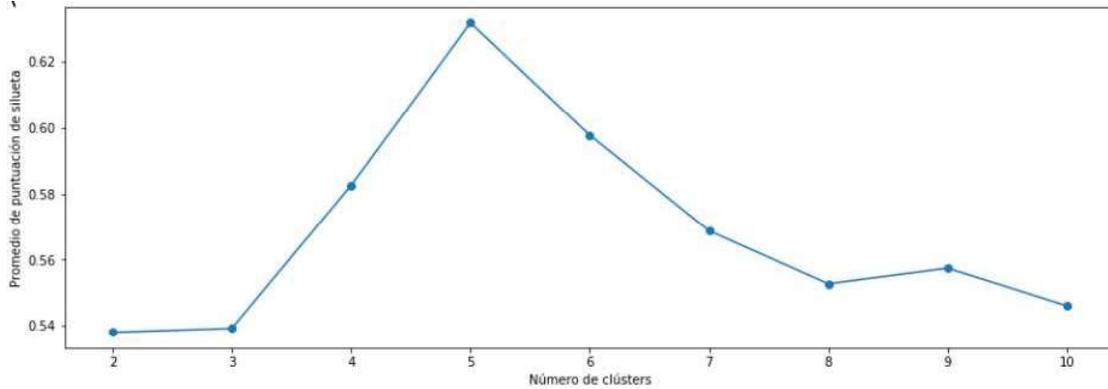


Figura 3.13: Método de silueta sobre Mezcla de Gaussianas

La gráfica nos permite observar que efectivamente para 5 clústers podemos establecer que es el número óptimo a formar durante la agrupación usando la mezcla de Gaussianas.

Una vez obtenidos los resultados de los métodos de agrupación y discutidos, se tomó la decisión de continuar con los resultados mostrados por el método de mezcla de Gaussianas.

3.2.6. Matriz de Relación Grupos Formados/Clasificación Expertos

Una vez realizado el reagrupamiento de los datos y con los resultados obtenidos por parte de la clasificación previa de los expertos, se realizó una matriz de relación entre ambos para analizar los resultados, ver Tabla 3.1.

Tabla 3.1: Matriz de relación entre Grupos Formados/Clasificación Expertos

Grupos Formados/Clasificación Expertos	1	2	3
1	100	0	0
2	0	48	80
3	0	0	60
4	0	93	72
5	100	0	0

Como se puede observar en la Tabla 3.1, la clase 1 perteneciente a las burbujas pequeñas se pueden identificar perfectamente con los grupos 1 y 5 de los grupos

formados por la agrupación. Para identificar las demás clases no es suficiente con el análisis realizado hasta el momento por lo que se puede observar, por tanto, se aplicaron algoritmos de aprendizaje automático para resolver este problema.

3.3. Algoritmos de Aprendizaje

Se muestra a continuación los resultados obtenidos de los algoritmos de aprendizaje que mostraron mejor desempeño para nuestro problema, donde se refleja, la aplicación de la metodología a los modelos datos para clasificación de imágenes.

3.3.1. Gaussian Naive Bayes

El algoritmo Naive Bayes es uno de los algoritmos más antiguos de aprendizaje automático, que se basa en el teorema de Bayes. La teoría de Bayes y los fundamentos de las estadísticas en la que se basa este algoritmo se desarrollaron en el siglo XVIII. Antes de que apareciera la primera implementación de este algoritmo por computadora, todos los cálculos se hacían manualmente.

El Gaussian Naive Bayes es un modelo clasificador basado en Naive Bayes pero haciendo una suposición de distribución normal de las verosimilitudes. Se muestra a continuación los resultados del modelo Gaussian Naive Bayes, se eligió porque es el más simple y uno de los más popular (véase Fig. 3.14).

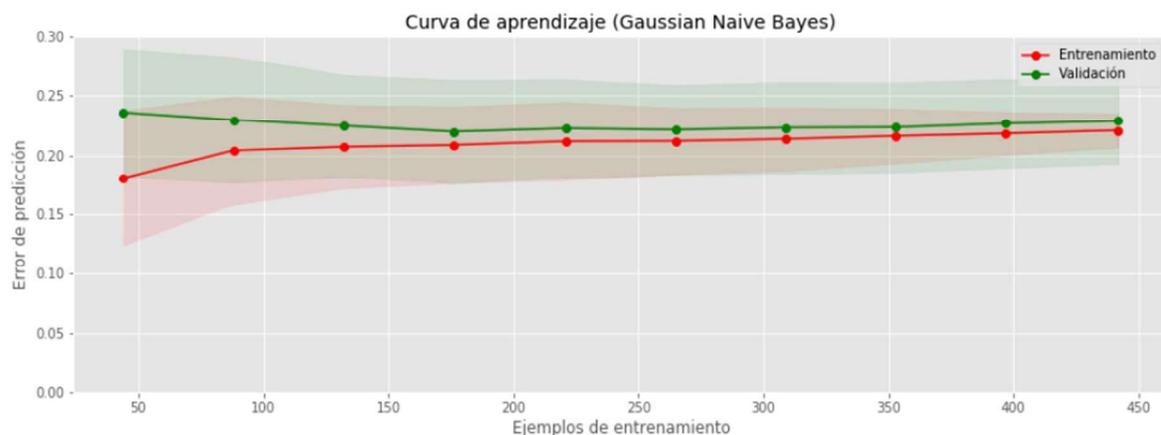


Figura 3.14: Curva de aprendizaje para el algoritmo Gaussian Naive Bayes

3.3.2. Regresión Logística

Uno de los algoritmos más utilizados actualmente en el aprendizaje automático, es el de regresión logística. Por su eficacia y simplicidad es muy utilizada por científicos de datos, pues no es necesario disponer tanto en entrenamiento como en ejecución de grandes recursos computacionales.

La regresión logística es aplicada a problemas de clasificación y no a problemas de regresión como puede indicar su propio nombre, por lo que el valor obtenido al usar esta técnica se encuentra entre 0 y 1.

Dado el propósito del presente trabajo sobre la clasificación de imágenes de burbujas, se procedió hacer uso de este algoritmo de regresión logística para analizar su rendimiento en este tipo de problema. En la siguiente imagen se muestra su desempeño (véase Fig. 3.15).

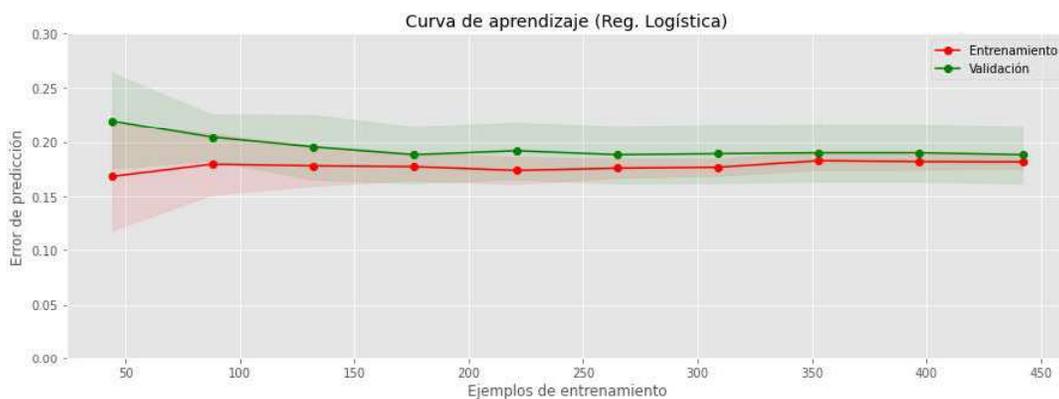


Figura 3.15: Curva de aprendizaje para el algoritmo de Regresión Logística

3.3.3. Máquina de Soporte Vectorial con kernel Gaussiano

Las SVM constituyen algoritmos de aprendizaje automático muy utilizados. SVM fue desarrollado en los años 90 [102] por Vladimir Vapnik y su equipo de trabajo. Su objetivo se centra en la separación de clases mediante un hiperplano en un espacio extendido donde maximiza el margen de clasificación, por lo que en el espacio extendido la separación entre clases es lineal mientras que no es lineal en el espacio original. Su aplicación actual se refleja en el reconocimiento de imágenes y textos, aplicaciones médicas, y muchos más; y son utilizadas tanto en problemas de clasificación como de regresión.

Para trabajar con SVM hay que tener en cuenta el kernel a utilizar, sus hiperparámetros y el parámetro de regularización. Los kernel más utilizados son el lineal, polinomial y gaussiano, donde cada uno sigue una aproximación distinta en la que se realiza una expansión de la dimensionalidad. El ajuste de una SVM con kernel gaussiano se ve determinado esencialmente por el parámetro γ (Gamma) y el parámetro de regularización C. El parámetro γ es muy importante para desarrollar una SVM con buena generalización ya que el mismo determina el ancho del kernel. Si γ es demasiado grande la SVM va tender al sobreajuste y si es pequeño habrá mayor solape entre gaussianas, y el modelo será muy pobre. El parámetro de regularización C se refiere a la penalización y establece una relación entre el error de entrenamiento y la complejidad del modelo; permite la flexibilidad de que se puedan cometer errores a la hora de clasificar ejemplos con la SVM. Si mayor es el valor de C, la penalización de errores es más rígida por lo que tiende al sobreajuste. Si es C es menor, permite mayor error en el entrenamiento por lo que tiende al subajuste.

En esta investigación se trabajó con el kernel gaussiano por sus buenos resultados obtenidos en el estado del arte, brinda una expansión de la dimensionalidad más alta y es el método kernel estándar para clasificación. El primer resultado de la SVM con valores por defecto mostró el siguiente comportamiento (véase Fig. 3.16).

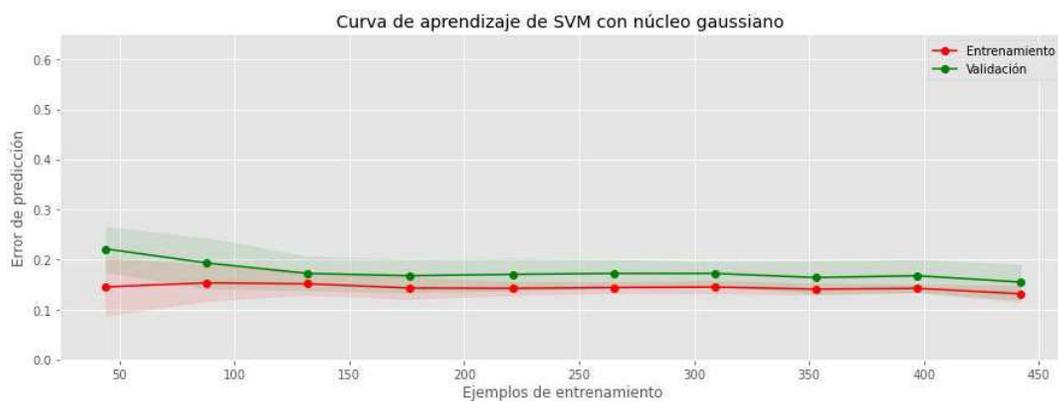


Figura 3.16: Curva de aprendizaje para el algoritmo de Máquina de Soporte Vectorial con kernel Gaussiano

Como se observa en la figura 37, la SVM con kernel gaussiano muestra mejores resultados que las anteriores técnicas aplicadas. A continuación, se trata de optimizar los parámetros γ y C para buscar una mejor predicción con la SVM.

3.3.4. Curvas de validación para los parámetros γ y C

Los valores óptimos de estos parámetros se pueden determinar por validación cruzada. Los rangos de valores que se exploran se muestran a continuación (véase Fig. 3.17 y Fig. 3.18).

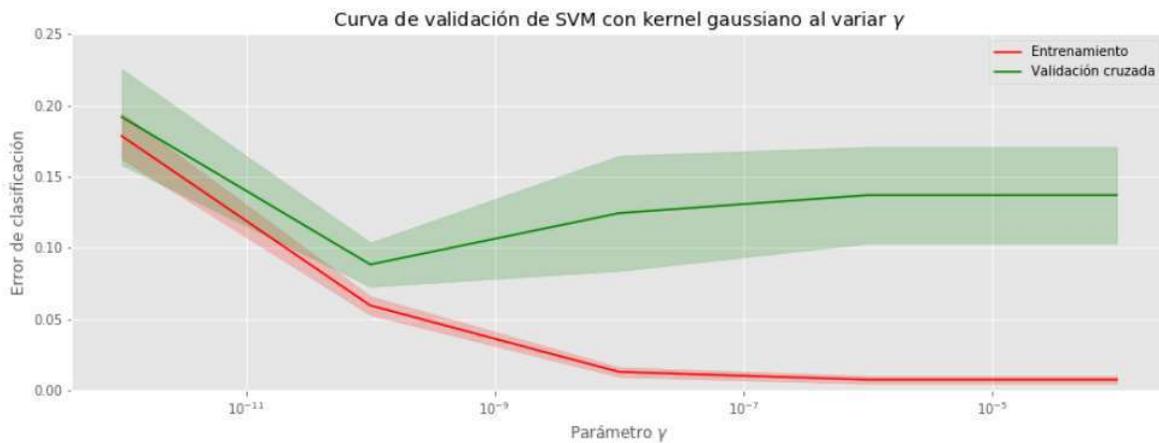


Figura 3.17: Curva de validación para el parámetro γ

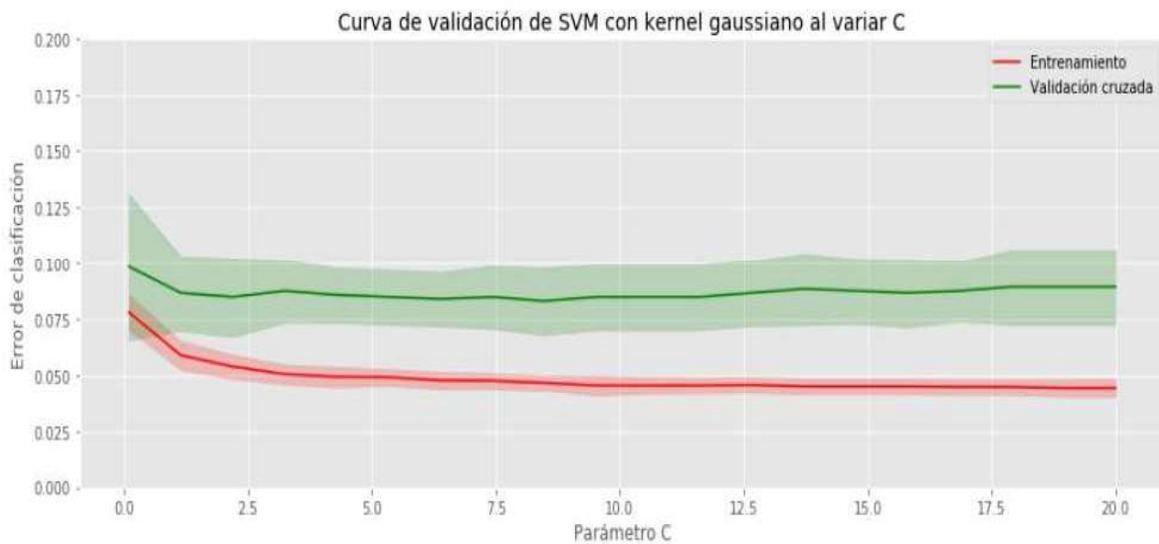


Figura 3.18: Curva de validación para el parámetro C

3.3.5. Máquina de Soporte Vectorial con kernel Gaussiano y parámetros óptimos

El rango de valores óptimos para γ se encuentran entre 10^{-11} y 10^{-9} , por lo que se toma el valor 10^{-10} como el óptimo. Para C los valores se encuentran en el rango de 3 hasta 10 y en este caso se tomó el valor 5 como el óptimo. En la figura siguiente se muestran los resultados de nuestro modelo optimizado (véase Fig. 3.19).

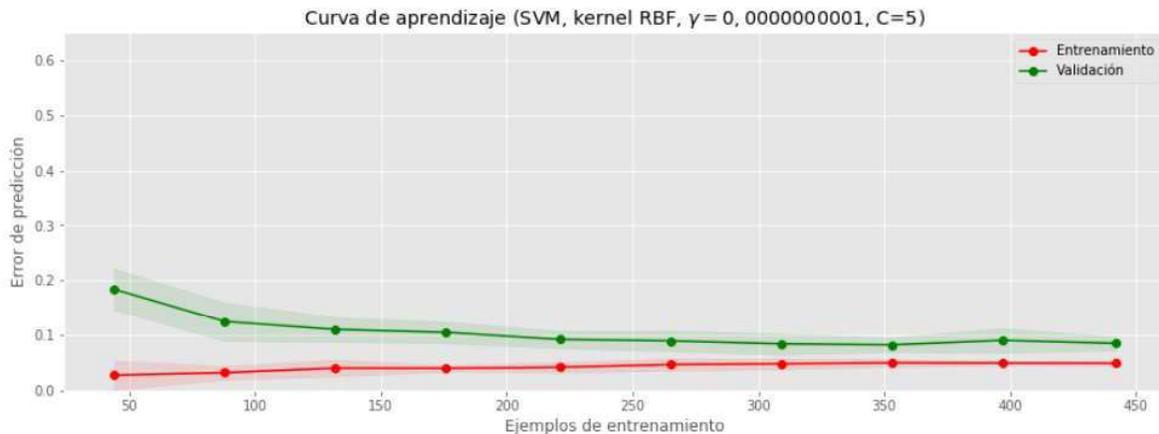


Figura 3.194: Curva de aprendizaje para el algoritmo de Máquina de Soporte Vectorial con kernel Gaussiano y parámetros óptimos

3.4 Resultados Parciales de Aprendizaje Automático

Se muestran los resultados de clasificación de los modelos en los conjuntos de datos de imágenes de entrenamiento y validación en la Tabla 3.2, para formar un punto de referencia en este conjunto de datos.

Tabla 3.2: Precisión de clasificación de modelos en conjuntos de entrenamiento y validación.

Modelo	Parámetros	Entrenamiento	Validación
Gaussian	priors=[0.1, 0.1,		
Naive	0.1], var_smoothing=1e-9	0.78	0.77
Bayes			
Regresión	penalty=l2, C=1, multi_class=auto		
Logística		0.83	0.82
SVM -	C=1, kernel=rbf, gamma=scale		
Gaussiano		0.86	0.84

SVM -	C=5, kernel=rbf, gamma=1e-5		
Gaussiano		95%	92%
Optimizado			

Como se puede observar en la Tabla 1, las precisiones de los modelos Gaussian Naive Bayes y Regresión Logística no son buenas tanto para entrenamiento como para validación. El modelo de SVM con kernel gaussiano logró resultados mucho mejores de 0.86 para entrenamiento y 0.84 para el conjunto de validación. Se obtuvieron mejoras significativas cuando se optimizaron los parámetros para la SVM con kernel gaussiano, lo que resultó en precisiones de clasificación de aproximadamente 0.95 y 0.92 para los conjuntos de entrenamiento y validación respectivamente.

Hasta el momento, se ha explorado el uso de técnicas de aprendizaje automático tanto para el procesamiento como para el análisis de datos de imágenes. Se trabajó con los modelos de aprendizaje, Gaussian Naive Bayes, Regresión Logística y SVM, para el desarrollo de la clasificación de imágenes en las celdas de flotación.

Como continuación a los resultados obtenidos, se toma la decisión de continuar trabajando en la búsqueda de mejores resultados mediante la aplicación de técnicas de aprendizaje profundo y posteriormente hacer una comparación con los resultados de la SVM.

Capítulo 4

4 Aprendizaje Profundo y Transferencia de Aprendizaje

Las técnicas de aprendizaje profundo pueden extraer características, recuperar información de datos y aprender representaciones abstractas avanzadas de datos. Resuelven los problemas de la extracción de características tradicionales y se han aplicado con éxito en visión por computadora [103], [104] y muchos otros campos.

En la última década, los métodos más interesantes para el reconocimiento de imágenes son las CNN. Con el aumento de una mayor disponibilidad de grandes cantidad de datos y unido al gran poder de cómputo desarrollado, las redes neuronales se emplean con mayor frecuencia en este tipo de tareas. Los resultados del uso de estas redes son altamente reconocidos, por lo que decidimos hacer uso de las CNN en la presente investigación buscando obtener mejores resultados que los obtenidos anteriormente con aprendizaje automático.

4.1. Entorno de Trabajo y Conjunto de Datos

Nuestro experimento con redes profundas fueron realizados en Google Collaboratory, el entorno de trabajo que brinda Google para experimentar con el aprendizaje automático y profundo. Su construcción está basada en Jupyter Notebook y permite utilizar diferentes modelos de tarjetas gráficas para realizar los cálculos; por tanto, el trabajo es conveniente y fácil de compartir con otros investigadores. Por último, nuestro proceso de entrenamiento y clasificación se desarrolló en base al marco de aprendizaje profundo Tensorflow [57].

En nuestro trabajo utilizamos el mismo conjunto de datos de la sección anterior, el cual contiene 3 clases de burbujas divididas en pequeñas, mediana y grande. Contamos con un total de 553 imágenes para entrenar nuestro modelo y la resolución de las mismas se redujo a 224x224 o 299x299 píxeles en dependencia

de la arquitectura CNN utilizada, todo esto para reducir la carga computacional en el modelo.

Además del total de imágenes que se tienen, hicimos uso de la técnica aumento de datos para poder realizar diferentes experimentos con diferentes variaciones en el conjunto de datos. A partir de las 553 imágenes se generaron 1500 imágenes más con modificaciones realizadas por métodos definidos por nosotros (véase Fig. 4.1). En adición, se utilizó también el método `ImageDataGenerator` el cual se encuentra disponible en la librería de `keras` dentro de `Tensorflow`.

```
def random_rotation(image_array: ndarray):  
    # pick a random degree of rotation between 25% on the left and 25% on the right  
    random_degree = random.choice([90, 180, 270])  
    return sk.transform.rotate(image_array, random_degree)  
  
def random_noise(image_array: ndarray):  
    # add random noise to the image  
    return sk.util.random_noise(image_array)  
  
def horizontal_flip(image_array: ndarray):  
    # horizontal flip doesn't need skimage, it's easy as flipping the image array of pixels !  
    return image_array[:, ::-1]  
  
def vertical_flip(image_array: ndarray):  
    # vertical flip doesn't need skimage, it's easy as flipping the image array of pixels !  
    return image_array[::-1, :]
```

Figura 4.1: Métodos definidos para el aumento de datos

4.2. Transferencia de Aprendizaje

El aprendizaje de transferencia (Transfer Learning, TL por sus siglas en inglés) [105] surge del DL. Normalmente resulta una tarea imposible entrenar una red profunda compleja desde cero con solo un pequeño conjunto de datos. Lo que podemos hacer es reutilizar el modelo y los parámetros obtenidos por otros investigadores a través del entrenamiento de la red, lo cual requiere mucho tiempo y computacionalmente es intensivo, en un conjunto de datos de imágenes muy grande como el de ImageNet y usar el conocimiento que ha adquirido como capacitación previa para nuestra tarea específica. Posteriormente, podemos volver a entrenar la última capa totalmente conectada definida del modelo utilizando solo

una cantidad relativamente pequeña de datos para lograr buenos resultados para nuestra tarea objetivo.

El TL tiene como objetivo extraer el conocimiento de una o más tareas de origen y aplicar este conocimiento a una tarea de destino. A diferencia del proceso de aprendizaje tradicional, conocido como aprendizaje multitarea, en lugar de aprender todas las tareas de origen y destino simultáneamente, el aprendizaje de transferencia se preocupa más por la tarea de destino [105] (véase Fig. 4.2).

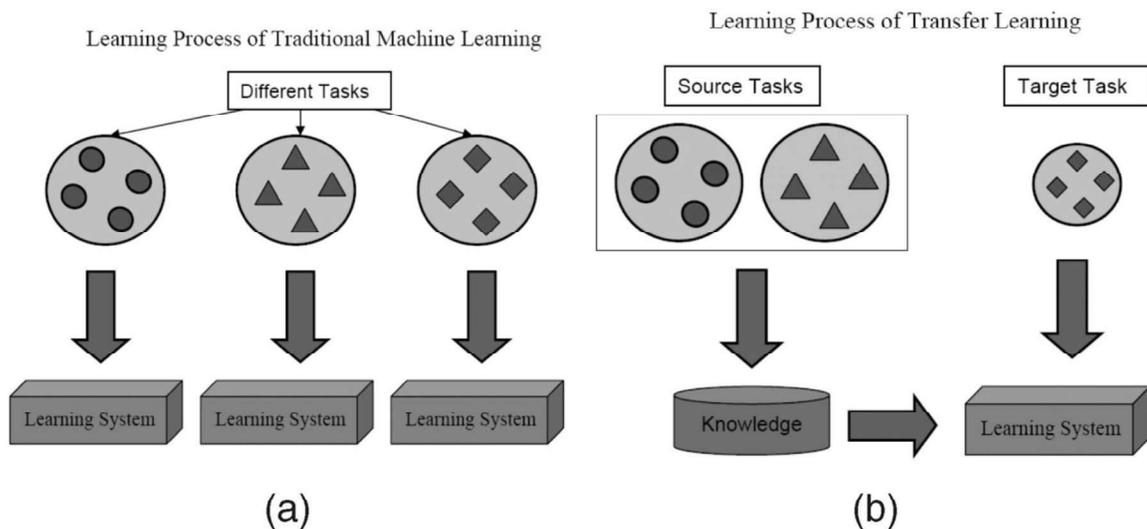


Figura 4.2: Diferentes procesos de aprendizaje entre (a) máquina tradicional aprendizaje y (b) aprendizaje de transferencia. Tomado de [105]

En la figura 37 se muestra la diferencia entre los procesos de aprendizaje de las técnicas de aprendizaje tradicionales y de transferencia. Como podemos ver, las técnicas tradicionales de aprendizaje automático intentan aprender cada tarea desde cero, mientras que las técnicas de transferencia de aprendizaje intentan transferir el conocimiento de algunas tareas anteriores a una tarea objetivo cuando esta última tiene menos datos de entrenamiento.

En la aplicación de TL existen dos estrategias principales que son ampliamente utilizadas en DL, estas incluyen:

- Estrategia 1
 1. Congelar los pesos entrenados de la red CNN desde las primeras capas.

2. Sólo entrenar las capas densas recién agregadas, que se crean a partir de la inicialización aleatoria de los pesos.
- Estrategia 2
 1. Inicializar la red CNN con los pesos pre-entrenados
 2. Luego se vuelve a capacitar a toda la red CNN mientras se establece que la tasa de aprendizaje sea muy pequeña, lo que garantiza que no se cambian drásticamente los pesos entrenados.

Como se ha mencionado, la ventaja del uso de TL es que proporciona un rápido progreso del entrenamiento ya que no estamos comenzando a entrenar desde cero. El TL es muy útil cuando se tiene un pequeño conjunto de datos de entrenamiento disponible, pero hay un gran conjunto de datos en un dominio similar (es decir, ImageNet).

Para capacitar los modelos de CNN pre-entrenados se hizo uso de la primera estrategia mencionada anteriormente, y se siguieron los siguientes pasos que incluyen:

1. Descargar los modelos y parámetros de las redes formadas en el conjunto de datos de ImageNet. Estos modelos fueron descargados sin la última capa totalmente conectada.
2. Agregar nuevas capas densas, en las cuales se modificaron el número de neuronas de la última capa completamente conectada para nuestra tarea específica con 3 neuronas para la clasificación de clases.
3. Todas las capas anteriores a las densas recién agregadas del modelo, se hacen no entrenables para no modificar los parámetros, la última capa creada si se mantiene entrenable para que aprenda sobre nuestros datos.
4. Por último, los parámetros de la capa completamente conectada se entrenan con las imágenes de burbujas en las celdas de flotación

El esquema del método de entrenamiento y diseño en nuestra CNN se presenta a continuación (véase Fig. 4.3). Se hace énfasis, que sólo se entrena la última capa, mientras que las anteriores están congeladas.

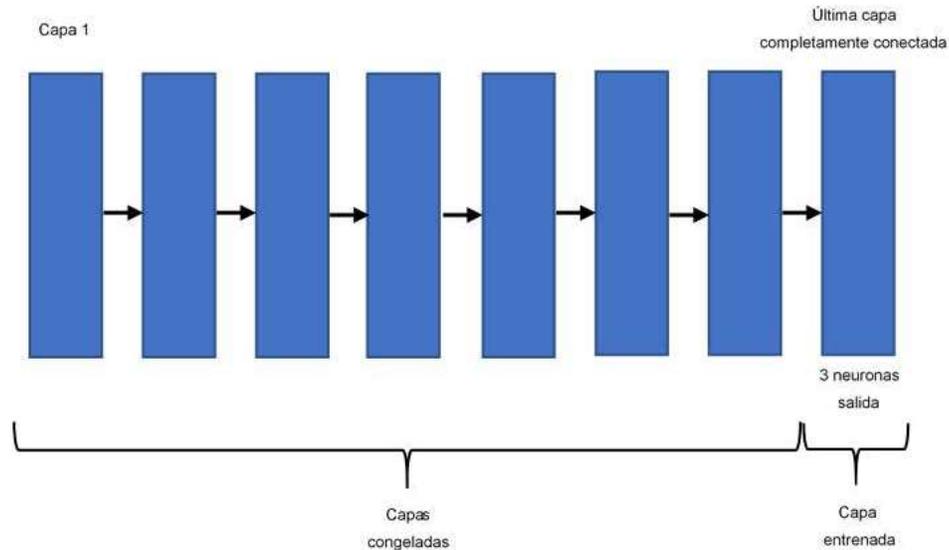


Figura 4.3: Esquema de CNN para el entrenamiento

Seguidamente se muestran los resultados obtenidos al aplicar TL con los modelos pre-entrenados mencionados en el capítulo anterior, se mostrarán los resultados de algunos modelos los cuales demuestran el trabajo de investigación realizado.

4.2.1. VGGNet

VGGNet es un modelo de CNN propuesto por investigadores del Grupo de Geometría Visual de Oxford (VGG, por sus siglas en inglés) [77]. Fue uno de los famosos modelos presentados a ILSVRC-2014 donde aseguró el primer y el segundo lugar en las tareas de localización y clasificación, respectivamente [106]. Este modelo está disponible públicamente y se ha utilizado en muchas otras aplicaciones y desafíos como un extractor de características de línea de base.

El modelo VGG se encuentra disponible en sus dos versiones VGG16 y VGG19, los cuales se diferencian en la cantidad de capas convolucionales. VGG16 consta de cinco bloques de capas convolucionales y de agrupación, es decir, los dos primeros bloques tienen dos capas convolucionales, seguidas de una capa de agrupación máxima, mientras que los tres bloques siguientes tienen tres capas convolucionales seguido de una capa de agrupación máxima. Seguidamente se muestra un diagrama simplificado (véase Fig. 4.4).

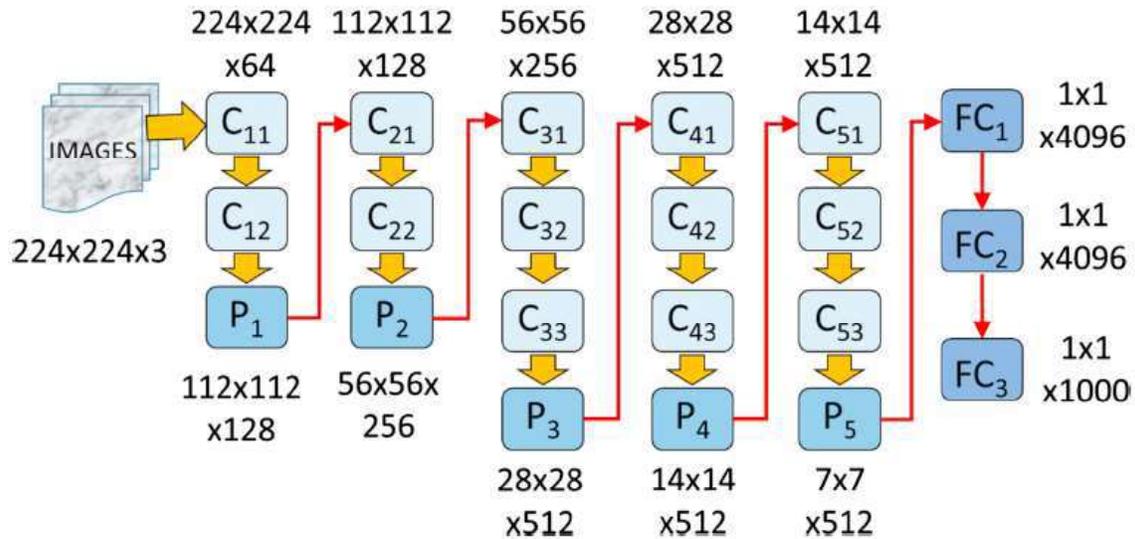


Figura 4.4: Representación simplificada de la arquitectura de la red neuronal convolucional VGG16, con 13 capas de convolución (C_{11} , C_{12} , ... C_{53}), cinco capas de agrupación (P_1 a P_5) y tres capas totalmente conectadas (FC_1 , FC_2 y FC_3). Las capas convolucional y de agrupación se organizan en cinco bloques, según lo indicado por los índices de las capas convolucional y de agrupación. Tomado de [65]

Esto le da 13 capas convolucionales, a diferencia de las cinco de AlexNet. Además, en VGG16 y VGG19, los grandes filtros convolucionales que se encuentran en AlexNet se reemplazan con múltiples filtros más pequeños de 3×3 del tamaño de un kernel, por lo que significa una mejora con respecto a AlexNet al reemplazar los filtros grandes del tamaño del kernel con múltiples filtros de tamaño más pequeños. Las pilas de filtros de tamaño más pequeño aumentan la profundidad de la red, lo que le permite aprender características más complejas [77].

Con la arquitectura de VGG se hicieron diferentes pruebas con distintas modificaciones a la red para estudiar su rendimiento en nuestro problema de investigación. Se muestra a continuación los mejores resultados de los modelos VGG16 (véase Fig. 4.5) y VGG19 (véase Fig. 4.6).

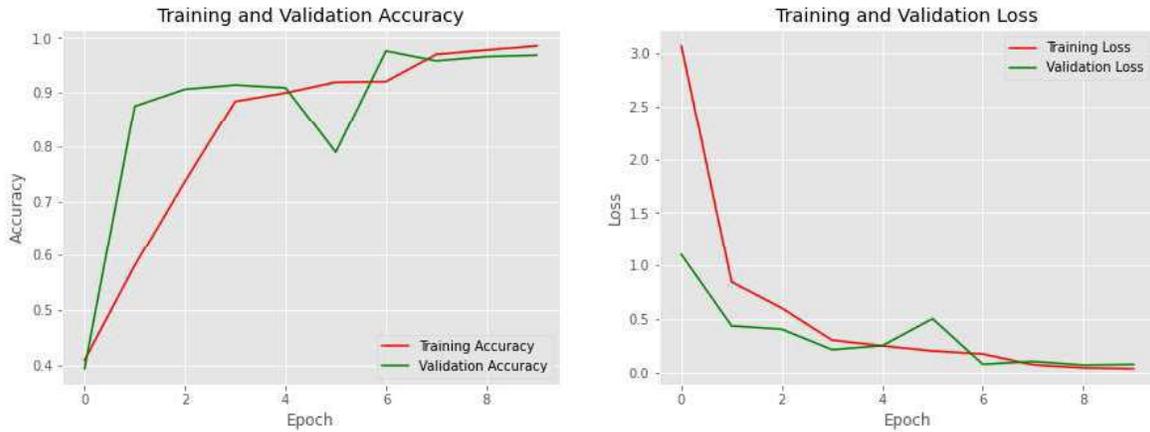


Figura 4.5: Curvas de entrenamiento y validación tanto en precisión y pérdida para VGG16

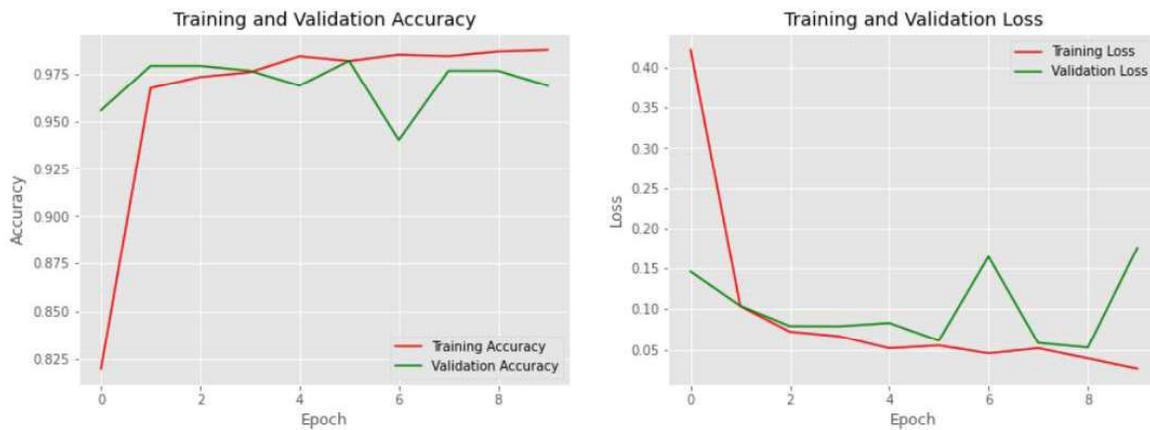


Figura 4.6: Curvas de entrenamiento y validación tanto en precisión y pérdida para VGG19

4.2.2. RestNet

Se espera que una red más profunda tenga una predicción más precisa. Cuando las redes más profundas pueden comenzar a converger, un el problema de degradación se expone: con la profundidad de la red en aumento, la precisión se satura y luego se degrada rápidamente [55]. Inesperadamente, tal degradación no es causada por un ajuste excesivo, y la adición de más capas a un modelo adecuadamente profundo conduce a un mayor error de entrenamiento, como se informa en [107][108]. A continuación se muestra un ejemplo: la red de 20 capas obtuvo un error de entrenamiento y de prueba más bajo que la red de 56 capas; se produce un problema de degradación debido a la desaparición de los gradientes (véase Fig. 4.7).

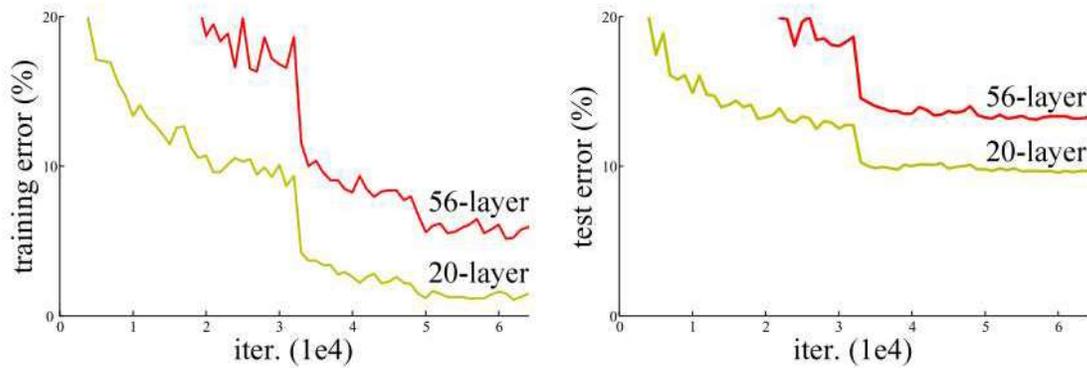


Figura 4.7: Error de entrenamiento (izquierda) y error de prueba (derecha) con redes de 20 y 56 capas. La red más profunda tiene un error de entrenamiento más alto y, por lo tanto, error de prueba. Tomado de [55]

En [55] se presentó un marco de aprendizaje residual para la capacitación de redes neuronales convolucionales mediante la reformulación explícita de las capas de la red como funciones residuales de aprendizaje con referencia a las entradas de la capa, en lugar de aprender funciones sin referencia. Estas redes residuales son relativamente fáciles de optimizar y permiten la construcción de redes con una profundidad y precisión notablemente mayores.

Por ejemplo, en el conjunto de datos de ImageNet, [55] podría aplicar redes residuales con profundidades de hasta 152 capas. A continuación, se ilustra un diagrama simplificado (véase Fig. 4.8).

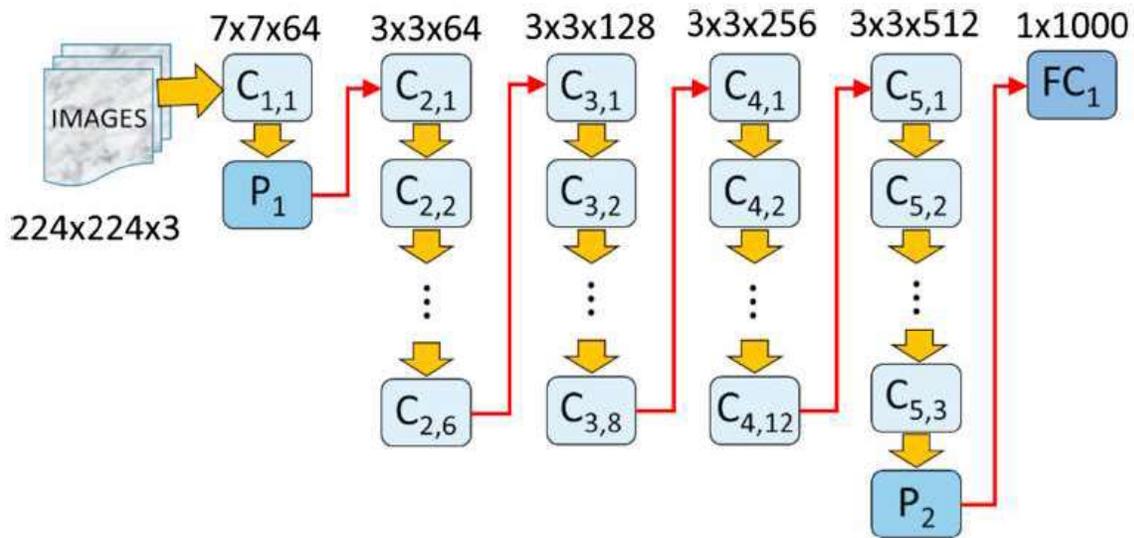


Figura 4.8: Representación simplificada de la arquitectura de una red neuronal convolucional ResNet de 34 capas que muestra capas convolucionales ($C_{1,1}$, $C_{1,2}$, ... $C_{5,3}$), capas combinadas (P_1 y P_2) y una capa totalmente conectada (FC_1). Tomado de [65]

En el ILSVRC 2015 [109], la llamada Red Neural Residual (ResNet, por sus siglas en inglés) introdujo la arquitectura de nivel con "conexiones de omisión" y presenta una normalización de lotes pesados. En el conjunto de datos de ImageNet, estas redes residuales con una profundidad de hasta 152 capas, 8x más profundas que las redes VGG obtuvieron una menor complejidad, logrando un error del 3.57%. Este resultado ganó el 1er lugar en la tarea de clasificación de ILSVRC 2015.

Haciendo uso de la arquitectura ResNet, se probaron diferentes modificaciones de la arquitectura para analizar el mejor rendimiento para nuestro problema. A continuación se muestran los resultados ResNet50 (véase Fig. 4.9), ResNet101 (véase Fig. 4.10) y ResNet152 (véase Fig. 4.11).

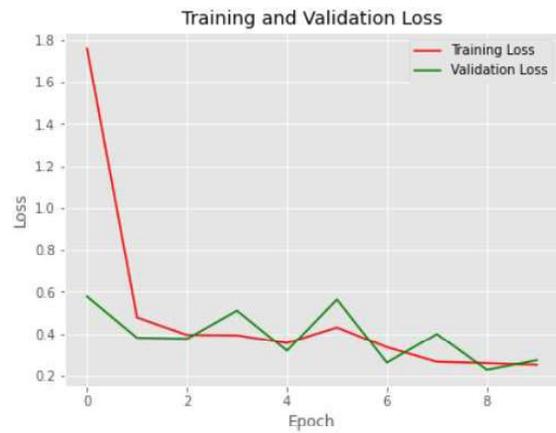
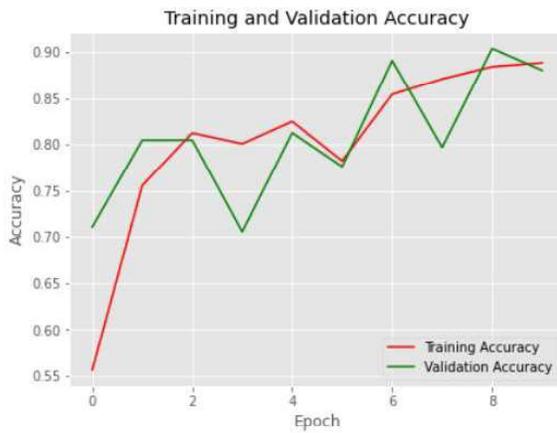


Figura 4.9: Curvas de entrenamiento y validación tanto en precisión y pérdida para ResNet50

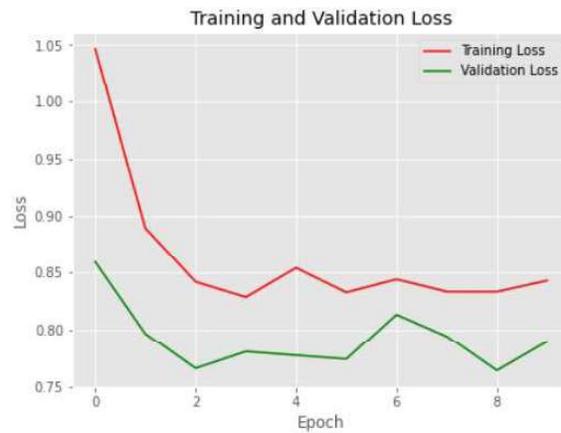
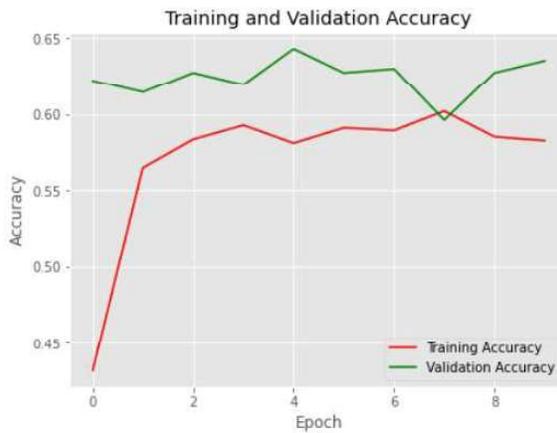


Figura 4.10: Curvas de entrenamiento y validación tanto en precisión y pérdida para ResNet101

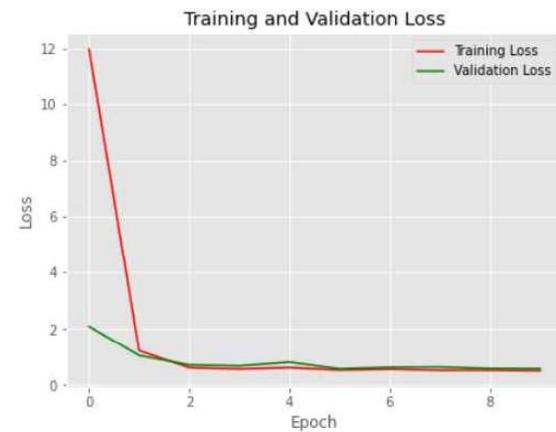
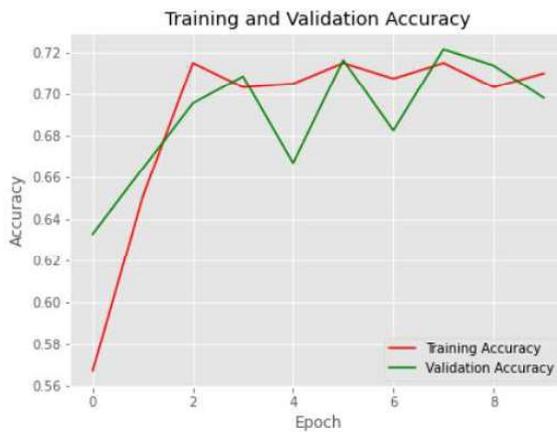


Figura 4.11: Curvas de entrenamiento y validación tanto en precisión y pérdida para ResNet152

4.2.3. InceptionV3 y InceptionResNetV2

Las arquitecturas de CNN InceptionV3 [82] e InceptionResNetV2 [83] fueron propuestas en los años 2016 y 2017 respectivamente. En la competencia ILSVRC se demostró que InceptionResNetV2 podría derrotar a la red InceptionV3 cuando se aplica a big data. Las arquitecturas de estas redes se muestran a continuación (véase Fig. 4.12).

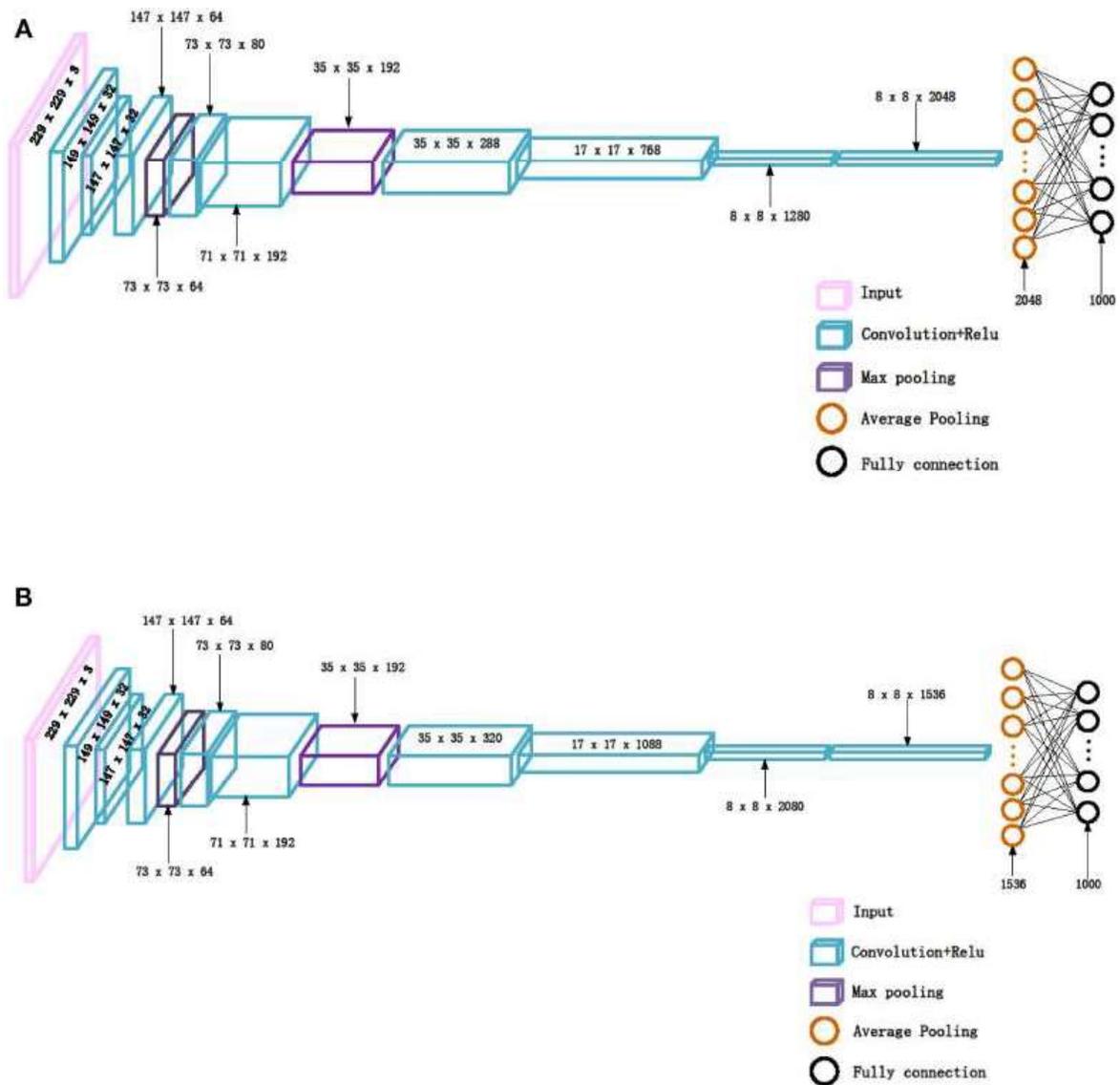


Figura 4.12: Arquitecturas de las redes InceptionV3 (A) e InceptionResNetV2 (B). Tomado de [110]

En la figura 47 se observa como las arquitecturas de ambas redes son muy similares. Las primeras capas son una transformación característica a través de las capas convolutivas tradicionales y las capas de agrupación, y la parte central está

compuesta de múltiples módulos de inicio apilados juntos. Los resultados finalmente se envían a través de la capa totalmente conectada mediante la función Softmax.

Una de las principales diferencias entre las redes InceptionV3 e InceptionResNetV2 radica en la composición diferente de los módulos Inception de las dos redes, donde InceptionResNetV2 está compuesta de módulos con conexiones residuales. Seguido se muestra las diferencias en la construcción del módulo inception con un tamaño de 8×8 entre InceptionV3 e InceptionResNetV2 (véase Fig. 4.13).

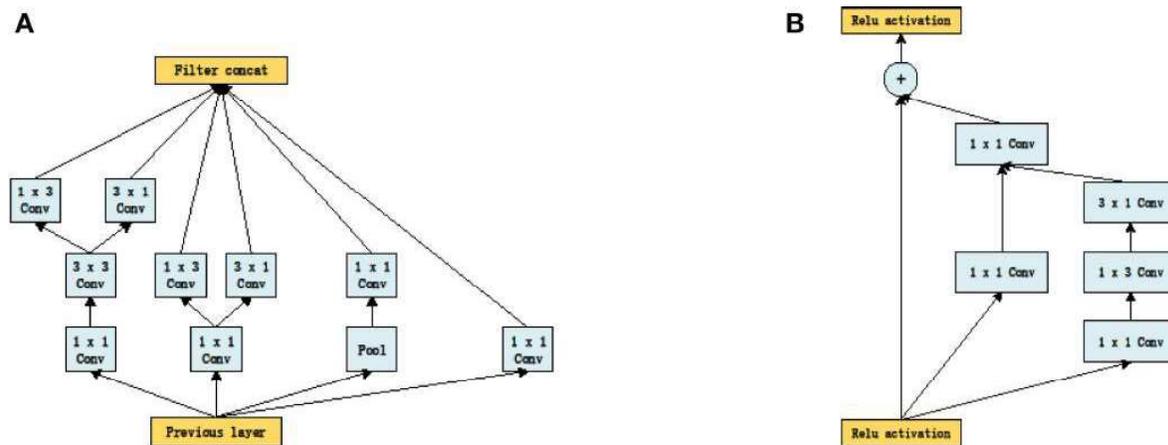


Figura 4.13: Módulos inception para las redes Inception_V3 (A) e Inception_ResNet_V2 (B). Tomado de [110]

Seguidamente se muestran los resultados obtenidos con las arquitecturas InceptionV3 (véase Fig. 4.14) e InceptionResNetV2 (véase Fig. 4.15).

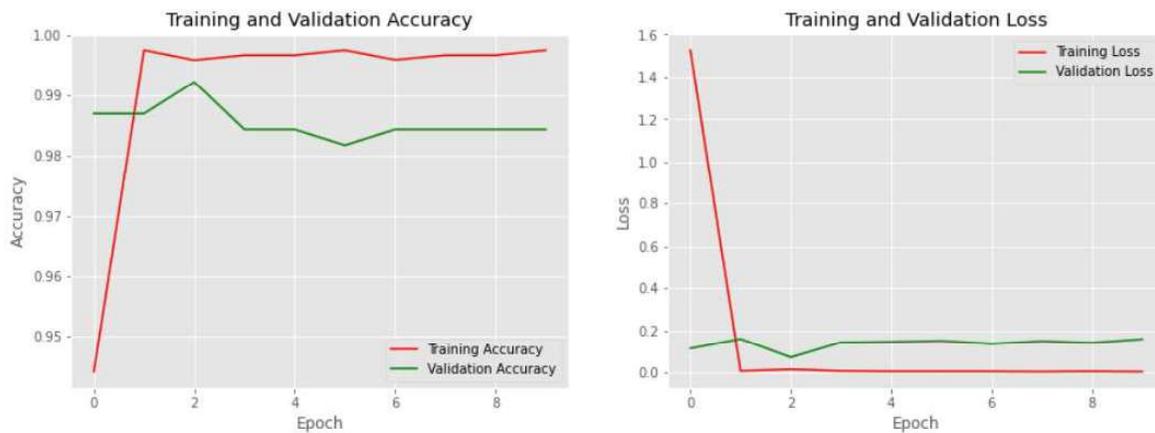


Figura 4.14: Curvas de entrenamiento y validación tanto en precisión y pérdida para InceptionV3

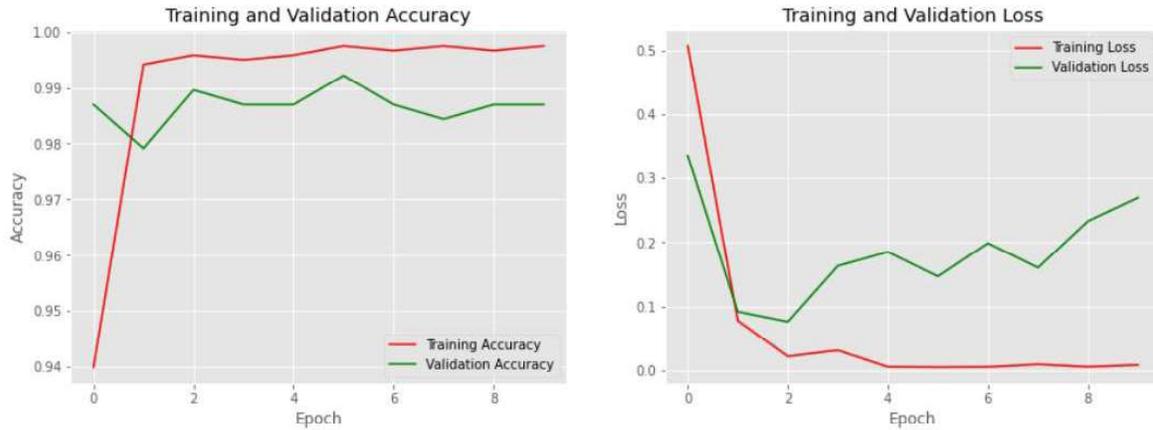


Figura 4.15: Curvas de entrenamiento y validación tanto en precisión y pérdida para InceptionResNetV2

4.2.4. Xception

La arquitectura de Xception a diferencia es que hace las convoluciones de manera óptima para que tomen menos tiempo. Separa las convoluciones de 2D en convoluciones de 1D.

En Xception los datos primero pasan por el flujo de entrada, luego a través del flujo medio que se repite ocho veces y finalmente a través del flujo de salida [84], lo que se observa a continuación (véase Fig. 4.16).

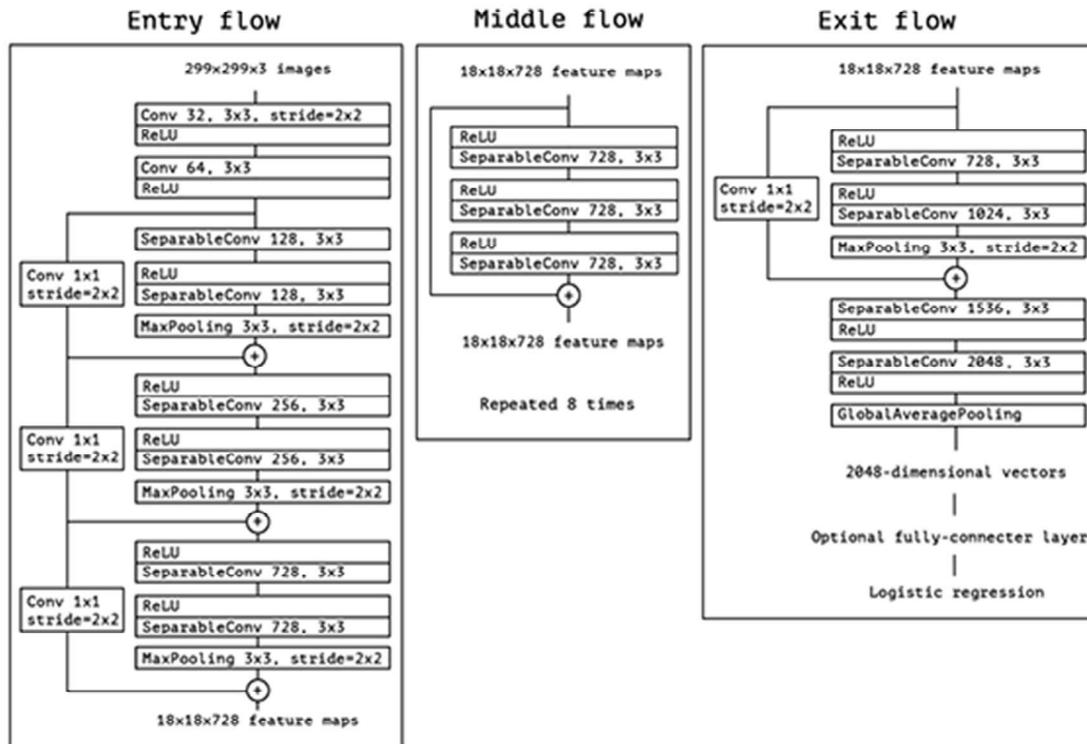


Figura 4.16: Arquitectura de Xception. Tomado de [84].

Como las arquitecturas mencionadas, con Xception se realizaron distintas pruebas con modificaciones a la red para analizar sus resultados. Se muestra a continuación los resultados del modelo Xception (véase Fig. 4.17).

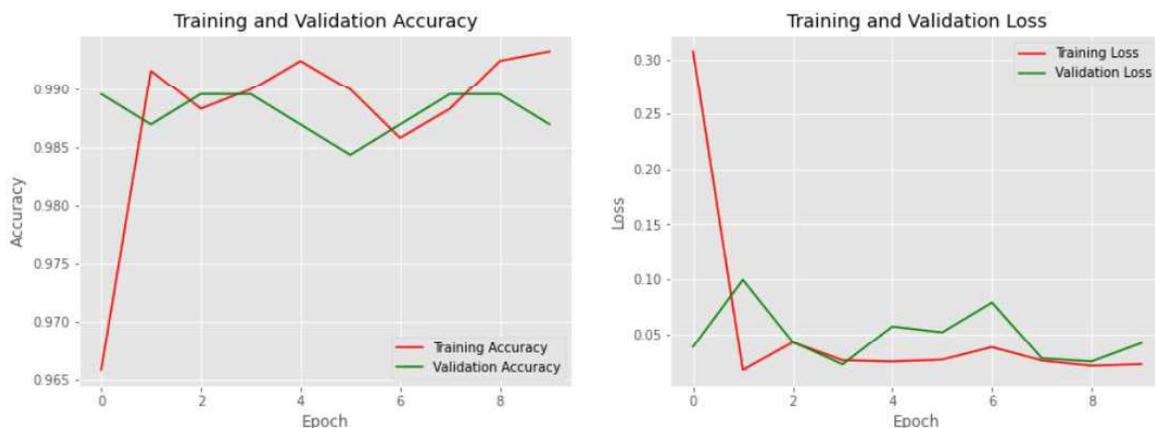


Figura 4.17: Curvas de entrenamiento y validación tanto en precisión y pérdida para Xception

4.3. Resultados Parciales de Aprendizaje Profundo

En el proceso de entrenamiento y validación se utilizó el algoritmo de Adam [111] para realizar la optimización mediante la iteración a través de 10 épocas utilizando nuestro conjunto de datos y el tamaño del lote (batch size, por su nombre en inglés) se establece en 32 en los experimentos.

Como se mencionó anteriormente, cuando usamos la transferencia de aprendizaje se trabajo con los modelos de redes neuronales convolucionales preentrenados sin su última capa totalmente conectada. Se definieron 3 estructuras o modelos para definir nuestra última capa, las cuales usamos tanto en entrenamiento como validación (véase Fig. 4.18)

- **Modelo 1**

```
x = base_model.output
x= Flatten()(x)
preds = tf.keras.layers.Dense(3, activation='softmax')(x)
```

- **Modelo 2**

```
x = base_model.output
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dense(512, activation='relu')(x)
preds = tf.keras.layers.Dense(3, activation='softmax')(x)
```

- **Modelo 3**

```
x = base_model.output
x= Flatten()(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dense(512, activation='relu')(x)
preds = tf.keras.layers.Dense(3, activation='softmax')(x)
```

Figura 4.18: Estructuras definidas para la última capa de los modelos

Haciendo uso de estas modificaciones en la estructura de la red convolucional, se observan en la Tabla 4.1 la comparativa de los resultados obtenidos durante las pruebas con aprendizaje profundo en el conjunto de datos.

Tabla 4.1: Resultados de entrenamiento y validación tanto en precisión y pérdida para los modelos de redes neuronales preentrenados

Modelo DCNN Preentrenado	Accuracy	Precision	Recall
VGG16			
TL + Modelo 1	0.73	0.65	0.66
TL + Modelo 2	0.85	0.88	0.80
TL + Modelo 3	0.97	0.96	0.97
TL + Modelo 1 + ImagenDG	0.94	0.93	0.95
TL + Modelo 2 + ImagenDG	0.95	0.95	0.96
TL + Modelo 3 + ImagenDG	0.98	0.97	0.98
VGG19			
TL + Modelo 1	0.55	0.39	0.57
TL + Modelo 2	0.98	0.98	0.98
TL + Modelo 3	0.97	0.97	0.97
TL + Modelo 1 + ImagenDG	0.95	0.95	0.94
TL + Modelo 2 + ImagenDG	0.83	0.84	0.82
TL + Modelo 3 + ImagenDG	0.93	0.93	0.93
ResNet50			
TL + Modelo 1	0.40	0.13	0.33
TL + Modelo 2	0.81	0.88	0.75
TL + Modelo 3	0.88	0.89	0.90
TL + Modelo 1 + ImagenDG	0.28	0.18	0.26
TL + Modelo 2 + ImagenDG	0.40	0.13	0.33
TL + Modelo 3 + ImagenDG	0.36	0.12	0.33
ResNet101			
TL + Modelo 1	0.40	0.13	0.33
TL + Modelo 2	0.63	0.44	0.56
TL + Modelo 3	0.63	0.43	0.56
TL + Modelo 1 + ImagenDG	0.40	0.13	0.33

TL + Modelo 2 + ImagenDG	0.58	0.39	0.52
TL + Modelo 3 + ImagenDG	0.58	0.39	0.52
ResNet152			
TL + Modelo 1	0.40	0.13	0.33
TL + Modelo 2	0.40	0.13	0.33
TL + Modelo 3	0.40	0.13	0.33
TL + Modelo 1 + ImagenDG	0.16	0.09	0.13
TL + Modelo 2 + ImagenDG	0.70	0.53	0.62
TL + Modelo 3 + ImagenDG	0.64	0.44	0.57
InceptionV3			
TL + Modelo 1	0.98	0.98	0.98
TL + Modelo 2	0.99	0.98	0.99
TL + Modelo 3	0.98	0.98	0.98
TL + Modelo 1 + ImagenDG	0.47	0.60	0.46
TL + Modelo 2 + ImagenDG	0.82	0.84	0.77
TL + Modelo 3 + ImagenDG	0.84	0.83	0.81
InceptionResNetV2			
TL + Modelo 1	0.99	0.98	0.99
TL + Modelo 2	0.99	0.99	0.99
TL + Modelo 3	0.99	0.99	0.99
TL + Modelo 1 + ImagenDG	0.96	0.96	0.96
TL + Modelo 2 + ImagenDG	0.98	0.98	0.98
TL + Modelo 3 + ImagenDG	0.98	0.98	0.99
Xception			
TL + Modelo 1	0.98	0.98	0.98
TL + Modelo 2	0.98	0.98	0.98
TL + Modelo 3	0.99	0.99	0.99
TL + Modelo 1 + ImagenDG	0.99	0.98	0.99
TL + Modelo 2 + ImagenDG	0.99	0.98	0.99
TL + Modelo 3 + ImagenDG	0.99	0.98	0.99

Como se explica anteriormente, para todos los modelos se utilizaron las implementaciones de Tensorflow además de nuestras modificaciones realizadas en la arquitectura. En la Tabla 3 se presenta una comparación de los resultados de clasificación de imágenes de burbujas en celdas de flotación. Los números en negrita indican la mejor configuración del modelo de rendimiento para cada categoría de los modelos estudiados. Las métricas de rendimiento de clasificación incluyen el accuracy, precision y recall. Los 3 modelos de mejor rendimiento seleccionados para la presente investigación adicional incluyen: InceptionResNetV2(TL + Modelo 2), InceptionResNetV2(TL + Modelo 3) y Xception(TL + Modelo 3).

Hasta este punto, se ha trabajado con diferentes modelos de DCNN los cuales se encuentran disponibles de forma abierta y capacitados en grandes conjuntos de datos, lo cual nos permite hacer uso del TL para aprovechar el trabajo realizado por otros investigadores.

Capítulo 5

5 Resultados y discusión

Se considera que los conjuntos de datos de imágenes de burbujas utilizados en este estudio son significativamente más complejos debido al trabajo realizado durante la investigación para poder obtener un mayor número de imágenes. Además de las complejidades realizadas a nuestro conjunto de datos durante la capacitación de los modelos de aprendizaje, también se introduce un orden significativamente mayor de complejidad al tratar de entrenar los modelos de DCNN preentrenados usando TL y diferentes modificaciones de la capa salida en los modelos.

Los resultados obtenidos con aprendizaje automático mediante la SVM con kernel gaussiano son satisfactorios alcanzando una precisión de 0.95 en el presente problema de investigación, lo que demuestra como en ocasiones sin recurrir a aprendizaje profundo se pueden obtener también excelentes resultados. En otro orden, los resultados con aprendizaje profundo alcanzados son muy buenos para nuestro estudio, incluso cuando no disponemos de un conjunto de datos lo suficiente completo para tratar nuestro problema de investigación, logramos alcanzar un 0.99 en todas las métricas analizadas con los modelos InceptionResNetV2(TL + Modelo 2), InceptionResNetV2(TL + Modelo 3) y Xception(TL + Modelo 3). Estos 3 modelos de DCNN arrojaron los mismos resultados para todas las evaluaciones por lo que para nuestro problema se pudiese utilizar uno de estos.

Seguido se muestra una comparación de los resultados de ML y DL, en cuanto a la matriz de confusión (véase Fig. 5.1).



Figura 5.1: Matrices de confusión para aprendizaje automático (A) y para aprendizaje profundo (B)

En la figura anterior se observa como podemos clasificar mejor las diferentes clases con DL. Si bien con ML podemos distinguir perfectamente la clase 1, para la clase 2 y 3 aún existen ruidos los cuales hace que le modelo no arroje mejores resultados. En cuanto a DL, se puede clasificar sin errores la clase 1 y la clase 3 es casi perfecta exceptuando por un valor. Mientras que para la clase 2 también se puede ver como reconocemos mejor con los modelos de DL.

Analizando más a detalle los resultados de ML y DL, se puede ver a continuación cómo haciendo uso del reporte de clasificación podemos afirmar nuevamente los resultados obtenidos con aprendizaje profundo (véase Fig. 5.2).

		precision	recall	f1-score	support
A	1.0	1.00	1.00	1.00	145
	2.0	0.91	0.90	0.90	106
	3.0	0.93	0.94	0.94	163
	accuracy			0.95	414
	macro avg	0.95	0.95	0.95	414
	weighted avg	0.95	0.95	0.95	414
		precision	recall	f1-score	support
B	1	1.00	1.00	1.00	146
	2	0.96	0.99	0.98	102
	3	0.99	0.98	0.98	162
	accuracy			0.99	410
	macro avg	0.99	0.99	0.99	410
	weighted avg	0.99	0.99	0.99	410

Figura 5.2: Reportes de clasificación para aprendizaje automático (A) y aprendizaje profundo (B)

Por último, como se ha demostrado anteriormente el uso DL en nuestra investigación significó una mayor precisión en la búsqueda del mejor modelo que se adaptara a nuestra problemática. Se obtuvieron 3 modelos de DCNN los cuales nos brindan los mismo resultados, aún así se estudió más a fondo estos modelos para poder escoger entre ellos nuestro modelo ideal. En la Tabla 5.1 a continuación se demuestra.

Tabla 5.1: Comparación en cuanto a cantidad de parámetros de los modelos DCNN

Modelo DCNN Preentrenado	Total de Parámetros	Total de Parámetros a Entrenar
InceptionResNetV2		
TL + Modelo 2	58,536,163	4,199,427
TL + Modelo 3	157,626,595	103,289,859
Xception		
TL + Modelo 3	233,203,243	212,341,763

Se puede observar cómo las modificaciones realizadas a cada uno de los modelos preentrenados, InceptionResNetV2 y Xception cambia el número de parámetros a entrenar en nuestros modelos. La cantidad de parámetros en un modelo hoy en la actualidad si bien no supone un problema para entrenar dada la capacidad tecnológica que hoy se cuenta, si refiere a una menor complejidad en la definición de hiperparámetros para el problema. Por tanto, se toma la decisión de tomar como nuestro modelo para la problemática en cuestión, el InceptionResNetV2 con la modificación del segundo modelo para la capa de salida.

Capítulo 6

6 Conclusiones y Trabajos Futuros

Los enfoques de AI en la actualidad son extremadamente robustos para las condiciones del mundo real, y los sistemas se benefician realmente del abandono forzado de algunos datos en el proceso de aprendizaje. Debido al rápido ritmo de los avances tecnológicos, las tareas que antes se consideraban limitadas a los humanos serán asumidas por los sistemas de aprendizaje. Comprender las propiedades de las herramientas de aprendizaje es fundamental para garantizar que se apliquen de la manera más segura y efectiva.

En esta investigación, se explora el uso de técnicas de aprendizaje automático y aprendizaje profundo tanto para el procesamiento como análisis de datos de imágenes. Se trabajó con los modelos de aprendizaje automático, Gaussian Naive Bayes, Regresión Logística y Máquinas de Soporte Vectorial, así como los modelos de aprendizaje profundo, VGG, ResNet, InceptionV3, InceptionResNetV2 y Xception para el desarrollo de la clasificación de imágenes en las celdas de flotación.

Se determinó una metodología de trabajo la cual nos guío durante el desarrollo de la investigación para el análisis, comprensión y entendimiento del problema de investigación. Con los resultados obtenidos podemos afirmar como un análisis detallado y una exploración sobre los datos resulta de gran relevancia para la comprensión de este tipo de problemáticas. Como se esperaba, es posible realizar mejoras adicionales mediante el ajuste a nuestros modelos, lo que resulta en un mejor reconocimiento y predicción en la clasificación de nuestros datos. La implementación de modelos para extraer características de las imágenes de burbujas para clasificar las mismas, resulta en una mejora significativa sobre la toma de decisiones en proceso industrial minero.

De acuerdo a estos resultados se puede afirmar que si bien el uso de ML puede ser aplicado de manera efectiva para el control y supervisión de procesos de flotación, los modelos de DL nos brindan una mejor precisión para esta tarea, puesto que logramos alcanzar una precisión de 0.99 para nuestro modelo en el conjunto de datos trabajados.

Actualmente el grupo de trabajo se encuentra en el proceso de recolección de imágenes de diferentes celdas de flotación con fines de validación de estos resultados. El etiquetado de datos de entrenamiento, validación y prueba en diferentes plantas impone un desafío.

Referencias

- [1] SMG, “Panorama Minero del Estado de Guerrero,” pp. 13–15, 2014.
- [2] Secretaría de Economía, “Minería,” 2020/01/17. <https://www.gob.mx/se/acciones-y-programas/mineria> (accessed Mar. 04, 2020).
- [3] D. J. McKee, “Automatic flotation control- a review of 20 years of effort,” *Miner. Eng.*, vol. 4, no. 7–11, pp. 653–666, Jan. 1991, doi: 10.1016/0892-6875(91)90055-Z.
- [4] Z. T. Mathe, M. C. Harris, C. T. O’Connor, and J. P. Franzidis, “Review of froth modelling in steady state flotation systems,” *Miner. Eng.*, vol. 11, no. 5, pp. 397–421, May 1998, doi: 10.1016/S0892-6875(98)00020-X.
- [5] J. J. Liu and J. F. MacGregor, “Froth-based modeling and control of flotation processes,” *Miner. Eng.*, vol. 21, no. 9, pp. 642–651, Aug. 2008, doi: 10.1016/j.mineng.2007.12.011.
- [6] J. Kaartinen, J. Hätönen, H. Hyötyniemi, and J. Miettunen, “Machine-vision-based control of zinc flotation-A case study,” *Control Eng. Pract.*, vol. 14, no. 12, pp. 1455–1466, Dec. 2006, doi: 10.1016/j.conengprac.2005.12.004.
- [7] S. H. Morar, M. C. Harris, and D. J. Bradshaw, “The use of machine vision to predict flotation performance,” *Miner. Eng.*, vol. 36–38, pp. 31–36, Oct. 2012, doi: 10.1016/j.mineng.2012.02.010.
- [8] N. Mehrshad and M. Massinaei, “New image-processing algorithm for measurement of bubble size distribution from flotation froth images,” *Miner. Met. Process.*, vol. 28, no. 3, pp. 146–150, 2011, doi: 10.1007/BF03402247.
- [9] W. Wang, F. Bergholm, and B. Yang, “Froth delineation based on image classification,” *Miner. Eng.*, vol. 16, no. 11, pp. 1183–1192, Nov. 2003, doi: 10.1016/j.mineng.2003.07.014.

- [10] J. J. Liu, J. F. MacGregor, C. Duchesne, and G. Bartolacci, "Flotation froth monitoring using multiresolutional multivariate image analysis," *Miner. Eng.*, vol. 18, no. 1, pp. 65–76, Jan. 2005, doi: 10.1016/j.mineng.2004.05.010.
- [11] B. Lin, B. Recke, J. K. H. Knudsen, and S. B. Jørgensen, "Bubble size estimation for flotation processes," *Miner. Eng.*, vol. 21, no. 7, pp. 539–548, Jun. 2008, doi: 10.1016/j.mineng.2007.11.004.
- [12] C. hua YANG, C. hui XU, X. min MU, and K. jun ZHOU, "Bubble size estimation using interfacial morphological information for mineral flotation process monitoring," *Trans. Nonferrous Met. Soc. China (English Ed.)*, vol. 19, no. 3, pp. 694–699, Jun. 2009, doi: 10.1016/S1003-6326(08)60335-0.
- [13] G. Forbes, "Texture and Bubble Size Measurements for Modelling Concentrate Grade in Flotation Froth Systems," 博士论文, no. August, 2007.
- [14] P. N. Holtham and K. K. Nguyen, "On-line analysis of froth surface in coal and mineral flotation using JK FrothCam," *Int. J. Miner. Process.*, vol. 64, no. 2–3, pp. 163–180, Mar. 2002, doi: 10.1016/S0301-7516(01)00070-9.
- [15] C. Botha and D. Weber, "A practical system for real-time on-plant flotation froth visual parameter extraction," in *Africon, 1999 ...*, Sep. 1999, vol. 1, pp. 103–106, doi: 10.1109/AFRCON.1999.820774.
- [16] C. Aldrich, C. Marais, B. J. Shean, and J. J. Cilliers, "Online monitoring and control of froth flotation systems with machine vision: A review," *Int. J. Miner. Process.*, vol. 96, no. 1–4, pp. 1–13, Sep. 2010, doi: 10.1016/j.minpro.2010.04.005.
- [17] G. Bonifazi, P. Massacci, and A. Meloni, "A 3D froth surface rendering and analysis technique to characterize flotation processes," *Int. J. Miner. Process.*, vol. 64, no. 2–3, pp. 153–161, Mar. 2002, doi: 10.1016/S0301-7516(01)00069-2.
- [18] S. R. Rao, *Surface Chemistry of Froth Flotation: Volume 1: Fundamentals*, vol. 1. 2013.

- [19] R. D. Crozier and R. R. Klimpel, "Frothers: Plant Practice," *Miner. Process. Extr. Metall. Rev.*, vol. 5, no. 1–4, pp. 257–279, 1989, doi: 10.1080/08827508908952652.
- [20] R. R. Klimpel and S. Isherwood, "Some industrial implications of changing frother chemical structure," *Int. J. Miner. Process.*, vol. 33, no. 1–4, pp. 369–381, 1991, doi: 10.1016/0301-7516(91)90064-P.
- [21] T. A. Wheeler, "Coal Floats by Itself - Doesn't it?," *Symp. Reagents better Metall.*, pp. 131–142, 1994.
- [22] B. A. Comley, M. A. Vera, and J.-P. Franzidis, "Interpretation of the effect of frother type and concentration on flotation performance in an OK3 cell," *Mining, Metall. Explor.*, vol. 24, no. 4, pp. 243–252, 2018, doi: 10.1007/bf03403373.
- [23] G. M. Evans and B. W. Atkinson, "The Jameson Cell," *Flotat. Sci. Eng.*, pp. 331–363, 1995.
- [24] S. H. Morar, "The use of machine vision to describe and evaluate froth phase behaviour and performance in mineral flotation systems," University of Cape Town, 2010.
- [25] ECN, "Monitoreo y análisis de Capa superficial de Espuma en celdas de Flotación FrothMeter," pp. 0–14.
- [26] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to OpenCV," in *2012 Proceedings of the 35th International Convention MIPRO*, May 2012, pp. 1725–1730.
- [27] L. E. Sucar and G. Gómez, *Vision Computacional*. 2011.
- [28] D. J. M., "Proceso de reconocimiento de objetos, asistido por computador (visión artificial), aplicando gases neuronales y técnicas de minería de datos," *Sci. Tech.*, vol. XII, pp. 385–390, 2006, [Online]. Available: <http://www.redalyc.org/articulo.oa?id=84920491049>.

- [29] A. Cantero and E. Martínez, “Visión por computadora: identificación, clasificación y seguimiento de objetos.,” *FPUNE Sci.*, no. 10, 2016, [Online]. Available:
<http://www.une.edu.py:83/fpunescientific/index.php/fpunescientific/article/view/137>.
- [30] A. Schierwagen, “Vision as Computation, or: Does a Computer Vision System Really Assign Meaning to Images?,” *Integr. Syst. Approaches to Nat. Soc. Dyn.*, pp. 579–587, 2011, doi: 10.1007/978-3-642-56585-4_37.
- [31] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis and machine vision (3. ed.)*. 2008.
- [32] A. Yilmaz, O. Javed, and M. Shah, “Object Tracking: A Survey,” *ACM Comput. Surv.*, vol. 38, no. 4, 2006, doi: 10.1145/1177352.1177355.
- [33] S. Russell and J. Bohannon, “Artificial intelligence. Fears of an AI pioneer,” *Science*, vol. 349, no. 6245, p. 252, 2015, doi: 10.1126/science.349.6245.252.
- [34] “Deep learning vs. machine learning,” 2020. <https://docs.microsoft.com/en-us/azure/machine-learning/concept-deep-learning-vs-machine-learning>.
- [35] A. L. Samuel, “Some Studies in Machine Learning Using the Game of Checkers,” *IBM J. Res. Dev.*, vol. 3, no. 3, pp. 210–229, 1959, doi: 10.1147/rd.33.0210.
- [36] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [37] S. Russell and P. Norvig, *Artificial Intelligence A Modern Approach Third Edition*. 2010.
- [38] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction,” *Math. Intell.*, vol. 27, pp. 83–85, 2004, doi: 10.1007/BF02985802.
- [39] D. W. Hosmer, S. Lemeshow, J. W. & Sons, and W. I. (Online Service), *Applied*

logistic regression, 2nd ed. New York : Wiley, 2000.

- [40] D. Lowd and P. Domingos, "Naive Bayes Models for Probability Estimation," in *Proceedings of the 22nd International Conference on Machine Learning*, 2005, pp. 529–536, doi: 10.1145/1102351.1102418.
- [41] G. A. F. Seber and A. J. Lee, "Linear regression analysis." Wiley-Interscience, Hoboken, N.J., 2003, [Online]. Available: <http://site.ebrary.com/id/10538731>.
- [42] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986, doi: 10.1007/BF00116251.
- [43] C. Y. Zhou and Y. Q. Chen, "Improving Nearest Neighbor Classification with Cam Weighted Distance," *Pattern Recogn.*, vol. 39, no. 4, pp. 635–645, 2006, doi: 10.1016/j.patcog.2005.09.004.
- [44] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [45] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.
- [46] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989, doi: [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8).
- [47] K. Krishna and M. Murty, "Genetic K-Means Algorithm," *IEEE Trans. Syst. Man. Cybern. B. Cybern.*, vol. 29, pp. 433–439, Feb. 1999, doi: 10.1109/3477.764879.
- [48] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002, doi: 10.1109/34.1000236.
- [49] D. Dueck and B. J. Frey, "Non-metric affinity propagation for unsupervised image categorization," *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, doi:

10.1109/ICCV.2007.4408853.

- [50] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967, doi: 10.1007/BF02289588.
- [51] D. Birant and A. Kut, "ST-DBSCAN: An algorithm for clustering spatial-temporal data," *Data Knowl. Eng.*, vol. 60, no. 1, pp. 208–221, 2007, doi: <https://doi.org/10.1016/j.datak.2006.01.013>.
- [52] D. A. Reynolds, "Gaussian Mixture Models," in *Encyclopedia of Biometrics*, 2009.
- [53] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006, doi: 10.1162/neco.2006.18.7.1527.
- [54] D. Xie, L. Zhang, and L. Bai, "Deep Learning in Visual Computing and Signal Processing," *Appl. Comput. Intell. Soft Comput.*, vol. 2017, p. 1320780, 2017, doi: 10.1155/2017/1320780.
- [55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Dec. 2016, vol. 2016-Decem, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [56] T. Brosch, "Efficient deep learning of 3D structural brain MRIs for manifold learning and lesion segmentation with application to multiple sclerosis," University of British Columbia, 2016.
- [57] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." 2016, [Online]. Available: <http://arxiv.org/abs/1603.04467>.
- [58] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017, doi: <https://doi.org/10.1016/j.neucom.2016.12.038>.

- [59] S. Bai, "Growing random forest on deep convolutional neural networks for scene categorization," *Expert Syst. Appl.*, vol. 71, pp. 279–287, 2017, doi: <https://doi.org/10.1016/j.eswa.2016.10.038>.
- [60] H. Goh, N. Thome, M. Cord, and J. Lim, "Learning Deep Hierarchical Visual Feature Coding," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 12, pp. 2212–2225, 2014, doi: 10.1109/TNNLS.2014.2307532.
- [61] D. Yu *et al.*, "An Introduction to Computational Networks and the Computational Network Toolkit," 2015.
- [62] T. Team *et al.*, "Theano: A Python framework for fast computation of mathematical expressions," May 2016.
- [63] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," no. NeurIPS, 2019, [Online]. Available: <http://arxiv.org/abs/1912.01703>.
- [64] Y. Jia *et al.*, "Caffe: Convolutional Architecture for Fast Feature Embedding," *MM 2014 - Proc. 2014 ACM Conf. Multimed.*, Jun. 2014, doi: 10.1145/2647868.2654889.
- [65] Y. Fu and C. Aldrich, "Flotation froth image recognition with convolutional neural networks," *Miner. Eng.*, vol. 132, no. October 2018, pp. 183–190, 2019, doi: 10.1016/j.mineng.2018.12.011.
- [66] H. Chougrad, H. Zouaki, and O. Alheyane, "Deep Convolutional Neural Networks for breast cancer screening," *Comput. Methods Programs Biomed.*, vol. 157, pp. 19–30, 2018, doi: 10.1016/j.cmpb.2018.01.011.
- [67] H. Sharma, N. Zerbe, I. Klempert, O. Hellwich, and P. Hufnagl, "Deep convolutional neural networks for automatic classification of gastric carcinoma using whole slide images in digital histopathology," *Comput. Med. Imaging Graph.*, vol. 61, pp. 2–13, 2017, doi: 10.1016/j.compmedimag.2017.06.001.
- [68] A. Ferreira and G. Giraldi, "Convolutional Neural Network approaches to granite tiles classification," *Expert Syst. Appl.*, vol. 84, no. May, pp. 1–11,

2017, doi: 10.1016/j.eswa.2017.04.053.

- [69] M. F. Hansen *et al.*, "Towards on-farm pig face recognition using convolutional neural networks Computers in Industry Towards on-farm pig face recognition using convolutional neural networks," *Comput. Ind.*, vol. 98, no. June, pp. 145–152, 2018, doi: 10.1016/j.compind.2018.02.016.
- [70] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo, "Deep learning for decentralized parking lot occupancy detection," *Expert Syst. Appl.*, vol. 72, pp. 327–334, Apr. 2017, doi: 10.1016/j.eswa.2016.10.055.
- [71] M. Alotaibi and A. Mahmood, "Improved gait recognition based on specialized deep convolutional neural network," *Comput. Vis. Image Underst.*, vol. 164, no. May 2016, pp. 103–110, 2017, doi: 10.1016/j.cviu.2017.10.004.
- [72] Y. Fu and C. Aldrich, "Froth image analysis by use of transfer learning and convolutional neural networks," *Miner. Eng.*, vol. 115, pp. 68–78, 2018, doi: 10.1016/j.mineng.2017.10.005.
- [73] Z. C. Horn, L. Auret, J. T. McCoy, C. Aldrich, and B. M. Herbst, "Performance of Convolutional Neural Networks for Feature Extraction in Froth Flotation Sensing," *IFAC-PapersOnLine*, vol. 50, no. 2, pp. 13–18, 2017, doi: 10.1016/j.ifacol.2017.12.003.
- [74] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Neural Inf. Process. Syst.*, vol. 25, 2012, doi: 10.1145/3065386.
- [75] Y. Fu and C. Aldrich, "Using Convolutional Neural Networks to Develop State-of-the-Art Flotation Froth Image Sensors," *IFAC-PapersOnLine*, vol. 51, no. 21, pp. 152–157, 2018, doi: 10.1016/j.ifacol.2018.09.408.
- [76] D. E. Rumelhart, G. E. Hinton, R. J. Williams, and others, "Learning representations by back-propagating errors," *Cogn. Model.*, vol. 5, no. 3, p. 1, 1988.
- [77] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-

Scale Image Recognition,” pp. 1–14, 2014, [Online]. Available: <http://arxiv.org/abs/1409.1556>.

- [78] S. R. Kheradpisheh, M. Ghodrati, M. Ganjtabesh, and T. Masquelier, “Deep Networks Can Resemble Human Feed-forward Vision in Invariant Object Recognition,” *Sci. Rep.*, vol. 6, pp. 1–24, 2016, doi: 10.1038/srep32672.
- [79] P. U. Stanford Vision Lab, Stanford University, “ImageNet.” <http://imagenet.org/> (accessed May 25, 2019).
- [80] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Sep. 2015, doi: 10.1007/s11263-015-0816-y.
- [81] D. Mishkin, N. Sergievskiy, and J. Matas, “Systematic evaluation of convolution neural network advances on the Imagenet,” *Comput. Vis. Image Underst.*, vol. 161, pp. 11–19, Aug. 2017, doi: 10.1016/j.cviu.2017.05.007.
- [82] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. B. Wojna, *Rethinking the Inception Architecture for Computer Vision*. 2016.
- [83] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017*, pp. 4278–4284.
- [84] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017*, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.
- [85] H. Xing, Y. Meng, Z. Wang, K. Fan, and D. Hou, “Exploring geo-tagged photos for land cover validation with deep learning,” *ISPRS J. Photogramm. Remote Sens.*, vol. 141, pp. 237–251, Jul. 2018, doi: 10.1016/j.isprsjprs.2018.04.025.
- [86] X. Cheng, Y. Zhang, Y. Chen, Y. Wu, and Y. Yue, “Pest identification via deep residual learning in complex background,” *Comput. Electron. Agric.*, vol. 141, pp. 351–356, 2017, doi: 10.1016/j.compag.2017.08.005.

- [87] G. Montes-Atenas *et al.*, “Predicting bubble size and bubble rate data in water and in froth flotation-like slurry from computational fluid dynamics (CFD) by applying deep neural networks (DNN),” *Int. Commun. Heat Mass Transf.*, vol. 76, pp. 197–201, 2016, doi: 10.1016/j.icheatmasstransfer.2016.05.031.
- [88] R. Unzueta Arce, “Sistema de medición basado en visión artificial para circuito rougher de laboratorio,” 2016.
- [89] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” vol. 349, no. 6245, pp. 255–260, 2015, doi: 10.1126/science.aaa8415.
- [90] S. van der Walt *et al.*, “scikit-image: image processing in Python,” *PeerJ*, vol. 2, p. e453, 2014, doi: 10.7717/peerj.453.
- [91] P. Bourke, “Histogram Matching,” 2011. <http://paulbourke.net/miscellaneous/equalisation/>.
- [92] B. Schölkopf, A. J. Smola, and K.-R. Müller, “Kernel Principal Component Analysis,” in *Proceedings of the 7th International Conference on Artificial Neural Networks*, 1997, pp. 583–588.
- [93] B. Schölkopf, A. J. Smola, and K.-R. Müller, “Kernel Principal Component Analysis,” in *Advances in Kernel Methods: Support Vector Learning*, Cambridge, MA, USA: MIT Press, 1999, pp. 327–352.
- [94] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, vol. 26. 2001.
- [95] S. Wold, K. H. Esbensen, and P. Geladi, “Principal Component Analysis,” *Chemom. Intell. Lab. Syst.*, vol. 2, pp. 37–52, 1987.
- [96] I. Jolliffe, “Principal component analysis. 2nd ed,” [http://ist-iiiep.iiiep-unesco.org/cgi-bin/wwwi32.exe/\[in=epidoc1.in\]/?t2000=017716/\(100\)](http://ist-iiiep.iiiep-unesco.org/cgi-bin/wwwi32.exe/[in=epidoc1.in]/?t2000=017716/(100)), vol. 98, Oct. 2005, doi: 10.1002/0470013192.bsa501.
- [97] B. Xie, Y. Mu, and D. Tao, “m-SNE: Multiview Stochastic Neighbor Embedding

- BT - Neural Information Processing. Theory and Algorithms,” 2010, pp. 338–346.
- [98] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
- [99] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [100] J. W. Cooley and J. W. Tukey, “An Algorithm for the Machine Calculation of Complex Fourier Series,” *Math. Comput.*, vol. 19, pp. 297–301, 1965, doi: 10.1090/S0025-5718-1965-0178586-1.
- [101] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987, doi: [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7).
- [102] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.
- [103] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034, doi: 10.1109/ICCV.2015.123.
- [104] J. Xie *et al.*, “The Automatic Identification of Butterfly Species,” *ArXiv*, vol. abs/1803.0, 2018.
- [105] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.
- [106] “ImageNet Large Scale Visual Recognition Challenge 2014 (ILSVRC2014).” 2014, [Online]. Available: <http://www.image-net.org/challenges/LSVRC/2014/results>.
- [107] K. He and J. Sun, “Convolutional neural networks at constrained time cost,” in

Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Dec. 2015, vol. 07-12-June, pp. 5353–5360, doi: 10.1109/CVPR.2015.7299173.

- [108] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Highway Networks,” May 2015, Accessed: May 25, 2019. [Online]. Available: <http://arxiv.org/abs/1505.00387>.
- [109] “ImageNet Large Scale Visual Recognition Challenge 2015 (ILSVRC2015).” 2015, [Online]. Available: <http://www.image-net.org/challenges/LSVRC/2015/results>.
- [110] J. Xie, R. Liu, J. Luttrell 4th, and C. Zhang, “Deep Learning Based Analysis of Histopathological Images of Breast Cancer,” *Front. Genet.*, vol. 10, p. 80, Feb. 2019, doi: 10.3389/fgene.2019.00080.
- [111] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *CoRR*, vol. abs/1412.6, 2015.