



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales No

Renovables

Carrera de Computación

Fitting of the “vit-base-patch16-224-in21k” model for pneumonia classification in chest radiographic images using transformer vision networks.

Ajuste del modelo “vitbase-patch16-224-in21k” para la clasificación de neumonía en imágenes radiográficas de tórax utilizando redes de visión transformer.

Trabajo de Integración Curricular, previo a la obtención del título de Ingeniero en Ciencias de la Computación.

AUTOR:

Eduardo José Paccha Rojas

DIRECTOR:

Ing. Oscar Miguel Cumbicus Pineda, Mg.

Loja – Ecuador

2024

Certificación

Loja, 10 de septiembre del 2024

Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc.

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICO:

Que he revisado y orientado todo el proceso de la elaboración del Trabajo de Integración Curricular denominado: **Fitting of the “vit-base-patch16-224-in21k” model for pneumonia classification in chest radiographic images using transformer vision networks**, previo a la obtención del título de **Ingeniero en Ciencias de la Computación**, de autoría del estudiante: **Eduardo José Paccha Rojas**, con **cédula de identidad Nro. 1105706319**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc.

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Autoría

Yo, **Eduardo José Paccha Rojas**, declaro ser autor del presente Trabajo de Integración Curricular eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Integración Curricular, en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:

Cédula de identidad: 1105706319

Fecha: 10 de septiembre del 2024

Correo electrónico: eduardo.paccha@unl.edu.ec

Teléfono: 0964099664

Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Integración Curricular

Yo, **Eduardo José Paccha Rojas**, declaro ser el autor del Trabajo de Integración Curricular denominado: **Fitting of the “vit-base-patch16-224-in21k” model for pneumonia classification in chest radiographic images using transformer vision networks**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad. La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de integración Curricular que realice un tercero.

Para constancia de esta autorización suscribo, en la ciudad de Loja, a los diez días del mes de septiembre de dos mil veinticuatro.

Firma:

Autor: Eduardo José Paccha Rojas

Cédula de identidad: 1105706319

Dirección: Loja - Ecuador

Correo electrónico: eduardo.paccha@unl.edu.ec

Teléfono: 0964099664

DATOS COMPLEMENTARIOS:

Director del Trabajo de Integración Curricular: Ing. Oscar Miguel Cumbicus Pineda, Mg. Sc.

Dedicatoria

Quiero dedicar con profundo cariño este trabajo de integración curricular a mi hermanito Santy, quien ha sido mi compañero a lo largo de este viaje. Su apoyo constante y su presencia han sido pilares fundamentales que me han impulsado a dar siempre un paso más, sin importar las dificultades. Su capacidad para escucharme y su ánimo sincero han sido cruciales para que este trabajo culmine con éxito. No existen palabras suficientes para expresar lo agradecido que estoy por su apoyo incondicional y su confianza en mí durante todo este proceso.

Eduardo José Paccha Rojas

Agradecimiento

Primeramente, quiero expresar al más grande de los agradecimientos a mi familia que son las personas por las que tuve la voluntad y la fuerza para realizar cada esfuerzo. A mis padres José Miguel y Karla Elizabeth y a mi hermana Karen Lizbeth, por ser ustedes quienes motivaron y alentaron el trabajo realizado durante todo este importante periodo y por ser quienes establecieron los valores de la persona que soy y por brindar el apoyo para el desarrollo de las habilidades a las que recurrí para la culminación de esta importante investigación.

De manera similar quisiera agradecer a mis amigos, compañeros y docentes que formaron parte de este largo camino hacia el desarrollo profesional y por compartir conmigo sus conocimientos y enseñanzas. Agradezco especialmente a mi director del trabajo de integración curricular, el Ingeniero Oscar Cumbicus, por su orientación constante durante todo el arduo proceso de investigación y su invaluable mentoría. Sus conocimientos y su consejo han sido pilares fundamentales en la dirección de esta investigación.

"Este trabajo no hubiera sido posible sin el compromiso y el apoyo de todos ustedes, mis más sinceros agradecimientos a cada persona que ha sido parte fundamental de este proceso."

Eduardo José Paccha Rojas

Índice de Contenidos

Portada	i
Certificación	ii
Autoría	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento	vi
Índice de Contenidos	vii
Índice de Tablas.....	x
Índice de Figuras.....	xii
Índice de ecuaciones.....	xv
Índice de anexos	xvi
1. Título	1
2. Resumen	2
Abstract.....	3
3. Introducción	4
4. Marco teórico	6
4.1. Antecedentes.....	6
4.2. Fundamentación Teórica	6
4.2.1. Neumonía.....	6
4.2.2. Radiología	7
4.2.3. Aspectos radiológicos de la detección de neumonía	7
4.2.4. Transfer Learning (Transferencia de aprendizaje)	8
4.2.5. Fine Tuning (Ajuste fino)	8
4.2.6. Balanceo de clases	8
4.2.7. Aumento de datos	9
4.2.8. Redes neuronales	9
4.2.9. Redes de visión transformer.....	10

4.2.10.	Funcionamiento de la arquitectura.....	10
4.2.11.	Zero-Shot learning.....	11
4.2.12.	Conjuntos de datos.....	11
4.2.13.	Métricas de evaluación.....	14
4.2.14.	Herramientas de desarrollo y Tecnologías.....	16
4.3.	Trabajos Relacionados.....	20
5.	Metodología.....	26
5.1.	Área de estudio.....	26
5.2.	Procedimiento.....	26
5.2.1.	Objetivo 1: Ajustar el modelo “vit-base-patch16-224-in21k” de Google, para la clasificación de neumonía en imágenes radiográficas de tórax utilizando el conjunto de datos “Chest X-Ray Images (Pneumonia)”.....	27
5.2.2.	Objetivo 2: Evaluar el ajuste del modelo, por medio de la métrica de sensibilidad, para la clasificación neumonía en imágenes radiográficas de tórax, utilizando el conjunto de datos para validación “Chest X-Ray Images (Pneumonia)”.....	30
5.3.	Recursos.....	31
5.3.1.	Recursos metodológicos.....	31
5.3.2.	Recursos técnicos.....	32
5.3.3.	Participantes.....	32
6.	Resultados.....	33
6.1.	Objetivo 1: Ajustar el modelo “vit-base-patch16-224-in21k” de Google, para la clasificación de neumonía en imágenes radiográficas de tórax utilizando el conjunto de datos “Chest X-Ray Images (Pneumonia)”.....	33
6.1.1.	Tarea 1: Recolección y selección de los datos.....	33
6.1.2.	Tarea 2: Exploración de los datos.....	35
6.1.3.	Tarea 3: Balanceo de clases.....	38
6.1.4.	Tarea 4: Ajuste del modelo “vit-base-patch16-224-in21k”.....	48
6.1.5.	Tarea 5: Entrenamiento del modelo “vit-base-patch16-224-in21k”.....	53
6.2.	Objetivo 2: Evaluar el ajuste del modelo, por medio de la métrica de sensibilidad, para la clasificación neumonía en imágenes radiográficas de tórax, utilizando el conjunto de datos para validación “Chest X-Ray Images (Pneumonia)”.....	55

6.2.1. Tarea 6: Evaluación del modelo ajustado con el conjunto de entrenamiento 55

6.2.2. Tarea 7: Evaluación del modelo ajustado con el conjunto de prueba57

7. Discusión.....65

7.1 Primer objetivo: Ajustar el modelo “vit-base-patch16-224-in21k” de Google, para la clasificación de neumonía en imágenes radiográficas de tórax utilizando el conjunto de datos “Chest X-Ray Images (Pneumonia)”.....65

7.2 Segundo objetivo: Evaluar el ajuste del modelo, por medio de la métrica de sensibilidad, para la clasificación neumonía en imágenes radiográficas de tórax, utilizando el conjunto de datos para validación “Chest X-Ray Images (Pneumonia)”.....66

8. Conclusiones68

9. Recomendaciones69

9.1. Trabajos Futuros:69

9.2. Limitaciones:69

10. Bibliografía70

11. Anexos.....77

Índice de Tablas

Tabla 1. Cantidad de imágenes en el conjunto de entrenamiento.	12
Tabla 2. Cantidad de imágenes en el conjunto de prueba.....	12
Tabla 3. Cantidad de imágenes en el conjunto de validación	12
Tabla 4. Trabajos relacionados	20
Tabla 5 Cantidad de imágenes por clase en los conjuntos de datos en entrenamiento, prueba y validación	34
Tabla 6. Cantidad de imágenes por clase en el segundo conjunto de datos.	35
Tabla 7. Cantidad de datos resultantes de la división del conjunto de datos de la clase "NORMAL"	39
Tabla 8. Cantidad de imágenes totales después de la aplicación de filtros en el subconjunto de datos de entrenamiento para la clase "NORMAL"	43
Tabla 9. Cantidad de imágenes totales para el conjunto de entrenamiento. Representa la suma de la cantidad de imágenes en el conjunto de entrenamiento del conjunto original y el resultado de las técnicas de aumento de datos para la clase "NORMAL" en el subconjunto de imágenes seleccionadas para el aumento de datos.	44
Tabla 10. Total de imágenes radiográficas del conjunto de entrenamiento en la clase "NORMAL" luego de técnicas de aumento de datos.....	45
Tabla 11. Número total de imágenes radiográficas destinadas al conjunto de entrenamiento	45
Tabla 12. Cantidad de imágenes totales después del aumento de datos en el conjunto de prueba de la clase "NORMAL".....	46
Tabla 13. Cantidad de imágenes totales en el conjunto de prueba por clase	47
Tabla 14. Total de imágenes en el conjunto de datos que se va a utilizar sumando la cantidad del conjunto inicial y el total de imágenes generadas mediante técnicas de aumento de datos	47
Tabla 15. Cantidad de imágenes por conjunto de datos en la nueva distribución.....	51
Tabla 16. Valores de hiperparámetros utilizados en la función de entrenamiento del modelo	54
Tabla 17. Resultados de métricas pérdidas y precisión por épocas en la ejecución del entrenamiento del modelo.	55
Tabla 18. Valores de sensibilidad y especificidad obtenidos de la matriz de confusión en el entrenamiento del modelo.	57
Tabla 19. Distribución de imágenes por clase en el dataloader de prueba.....	58
Tabla 20. Valores de sensibilidad y especificidad obtenidos de la matriz de confusión en la prueba del modelo.....	61

Tabla 21. Valores de sensibilidad y especificidad obtenidos de la matriz de confusión en la prueba del modelo con el conjunto recolectado.....64

Índice de Figuras

Figura 1. Arquitectura de la red Visión Transformer.....	10
Figura 2. Mapa de la carrera de ingeniería en computación de la Universidad Nacional de Loja	26
Figura 3. Repositorio del conjunto de datos seleccionado “Chest X-Ray Images (Pneumonia)”	33
Figura 4. Repositorio del segundo conjunto de datos seleccionado “COVID-QU-Ex Dataset”	34
Figura 5. Distribución y cantidad de las imágenes por clase y subconjunto de datos en el conjunto “Chest X-Ray Images (Pneumonia)”	35
Figura 6. Distribución de clases del primer conjunto de datos para el entrenamiento del modelo	36
Figura 7. Distribución de clases del primer conjunto de datos para la prueba del modelo ...	37
Figura 8. Distribución de clases del primer conjunto de datos para la validación del modelo	37
Figura 9. Código para la división y selección del subconjunto para la aplicación de filtros...38	
Figura 10. Aplicación del filtro rotación aleatoria en imagen radiográfica. La imagen (A) muestra la imagen original de la radiografía, la imagen (B) muestra la imagen luego de ser aplicado el filtro.	39
Figura 11. Código para la aplicación de rotación aleatoria.....	40
Figura 12. Aplicación del filtro de giro horizontal en imagen radiográfica. La imagen A muestra la imagen original, la imagen B muestra el resultado de la imagen aplicado un giro horizontal.	40
Figura 13. Código para la aplicación de giro horizontal.....	41
Figura 14. Aplicación de Desenfoque Gaussiano en imagen radiográfica. La imagen (A) representa la radiografía normal y La imagen (B) representa el filtro aplicado. Establece un número aleatorio de bloques para el desenfoque en regiones aleatorias de la imagen (Los bloques están delimitados por líneas color verde solamente para fines explicativos).....	41
Figura 15. Código para la aplicación de Desenfoque Gaussiano.....	42
Figura 16. Aplicación de filtro para la intensidad de brillo en imágenes radiográficas. El filtro aplica una intensidad de brillo aleatoria, En la imagen A se muestra la aplicación con una intensidad de brillo baja, en la imagen B muestra una intensidad más alta que la normal, mientras que la imagen C se muestra la intensidad de brillo normal en las radiografías del conjunto	42
Figura 17. Código para la aplicación de intensidad de brillo aleatoria.....	43
Figura 18. Distribución de clases del conjunto de entrenamiento después de realizar la técnica de aplicación de filtros.....	44

Figura 19. Distribución de clases balanceadas del conjunto de datos para el entrenamiento del modelo.....	45
Figura 20. Código para la selección aleatoria del conjunto de prueba seleccionado.....	46
Figura 21. Distribución balanceada de las clases para el conjunto de prueba	47
Figura 22. Comparación entre el total de imágenes de los conjuntos de entrenamiento, prueba y el total de imágenes obtenidas en el conjunto original y el conjunto aumentado.....	48
Figura 23. Arquitectura del modelo “vit-base-patch16-224-in21k” para la clasificación de neumonía en imágenes radiográficas. (Arquitectura adaptada del documento “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale” [3])	49
Figura 24. Código de las transformaciones definidas para los datos de entrada al modelo	49
Figura 25. Aplicación de la redimensión a 224 x 244 píxeles para las imágenes de los conjuntos de datos	50
Figura 26. Código creación de los conjuntos de datos Entrenamiento, Prueba y Validación.....	50
Figura 27. Estructura de los directorios de los conjuntos de datos. La imagen A muestra los directorios separados en conjuntos de validación, entrenamiento y prueba. La imagen B muestra la estructura interna de cada directorio, conteniendo carpetas separadas por tipo de clase.....	50
Figura 28. Código para establecer la nueva distribución de los datos en el conjunto de imágenes.....	51
Figura 29. Comparación de distribución de datos entre el conjunto inicial y el conjunto balanceado. La imagen A representa la distribución del conjunto de datos inicial, la imagen B muestra la distribución de los datos tras la aplicación de la nueva distribución de los datos.	51
Figura 30. Código para la creación y configuración de los dataloaders de entrenamiento, prueba y validación	52
Figura 31. Carga y configuración del modelo para la clasificación de 2 clases.....	52
Figura 32. Instancia del modelo “vit-base-patch16-224-in21k” configurado	52
Figura 33. Código de la asignación de los hiperparámetros, Optimizador y función de pérdida	53
Figura 34. Código configuración de dispositivo para el entrenamiento del modelo.	53
Figura 35. Código de llamado a la función de entrenamiento del modelo	53
Figura 36. Función de empaquetado de modelo ajustado entrenado.....	54
Figura 37. Matriz de confusión del entrenamiento del modelo. Donde se presenta la cantidad de verdaderos positivos (TP), falsos negativos (FN), verdaderos negativos (TN) y falsos positivos (FP) por cada clase en el entrenamiento del modelo para el cálculo de sensibilidad y especificidad.....	56
Figura 38. Código para cargar el modelo ajustado en la fase de entrenamiento.....	57

Figura 39. Código para la asignación de la configuración del modelo ajustado.	57
Figura 40. Parámetros de la función de evaluación.	58
Figura 41. Matriz de confusión de la evaluación del modelo con el conjunto de prueba. Donde se presenta la cantidad de verdaderos positivos (TP), falsos negativos (FN), verdaderos negativos (TN) y falsos positivos (FP) por cada clase.	58
Figura 42. Imágenes clasificadas incorrectamente (Falsos negativos) de la clase NORMAL, obtenidas de la matriz de confusión en la prueba del modelo ajustado.	59
Figura 43. Imagen clasificada incorrectamente (Falso negativo) de la clase PNEUMONIA, obtenida de la matriz de confusión en la prueba del modelo ajustado.	60
Figura 44. Distribución de clases dentro del conjunto recolectado para la evaluación con la técnica de Zero-Shot.....	62
Figura 45. Configuración del conjunto de imágenes recolectado para la evaluación con el modelo ajustado.....	63
Figura 46. Matriz de confusión resultante de la evaluación del modelo con la técnica Zero-Shot. Donde se presenta la cantidad de verdaderos positivos (TP), falsos negativos (FN), verdaderos negativos (TN) y falsos positivos (FP) por cada clase.....	63

Índice de ecuaciones

Ecuación 1. Fórmula de cálculo para la precisión.....	14
Ecuación 2. Fórmula de cálculo para la sensibilidad	15
Ecuación 3. Cálculo de sensibilidad de la clase NORMAL en la matriz de confusión de entrenamiento.	56
Ecuación 4. Cálculo de sensibilidad de la clase PNEUMONIA en la matriz de confusión de entrenamiento.	56
Ecuación 5. Cálculo de especificidad de la clase NORMAL en la matriz de confusión de entrenamiento.	56
Ecuación 6. Cálculo de especificidad de la clase PNEUMONIA en la matriz de confusión de entrenamiento.	56
Ecuación 7. Cálculo de sensibilidad de la clase NORMAL en la matriz de confusión de prueba.	60
Ecuación 8. Cálculo de sensibilidad de la clase PNEUMONIA en la matriz de confusión de prueba.....	61
Ecuación 9. Cálculo de especificidad de la clase NORMAL en la matriz de confusión de prueba.....	61
Ecuación 10. Cálculo de especificidad de la clase PNEUMONIA en la matriz de confusión de prueba.....	61
Ecuación 11. Cálculo de sensibilidad de la clase NORMAL en la matriz de confusión de la evaluación con el conjunto recolectado.	63
Ecuación 12. Cálculo de sensibilidad de la clase PNEUMONIA en la matriz de confusión de la evaluación con el conjunto recolectado.	63
Ecuación 13. Cálculo de especificidad de la clase NORMAL en la matriz de confusión de la evaluación con el conjunto recolectado.	64
Ecuación 14. Cálculo de especificidad de la clase PNEUMONIA en la matriz de confusión de la evaluación con el conjunto recolectado.	64

Índice de anexos

Anexo 1: Transcripción entrevista especialista77

Anexo 2: Certificación de traducción del resumen del TIC.....84

1. Título

Fitting of the “vit-base-patch16-224-in21k” model for pneumonia classification in chest radiographic images using transformer vision networks.

Ajuste del modelo “vitbase-patch16-224-in21k” para la clasificación de neumonía en imágenes radiográficas de tórax utilizando redes de visión transformer.

2. Resumen

La detección de neumonía es una tarea de alto impacto dentro del campo médico en general, y el análisis de la enfermedad mediante imágenes radiográficas se establece como prioridad para un diagnóstico oportuno ante la afección médica. En este contexto, la investigación planteó como objetivo principal utilizar las redes de vision transformers (ViT) para la clasificación de neumonía en imágenes radiograficas de tórax. Con el propósito de cumplir el objetivo se realizó un ajuste fino al modelo “vitbase-patch16-224-in21k” utilizando como marco de trabajo la metodología CRISP-ML(Q). El marco de trabajo ayudó al desarrollo de la investigación mediante 3 fases principales: Ingeniería de datos, donde se realizó la elección del conjunto de imágenes y en posteriormente se realizó el balanceo de las clases. La fase de ingeniería de modelos, ayudó al ajuste del modelo pre entrenado para la necesidad de la clasificación de neumonía según las clases del conjunto de datos “PNEUMONIA” y “NORMAL”. La tercera y última fase ayudó a la evaluación del modelo ajustado y que los resultados de la evaluación sean analizados en base a la métrica de la sensibilidad. Tras la finalización de las fases, se logró obtener los resultados de la investigación, que son: El modelo ajustado con un porcentaje del 99.34% de precisión en la clasificación de neumonía, y el análisis de clasificación basado en la métrica de sensibilidad, dónde el modelo ajustado obtuvo un porcentaje del 99.88% para clasificación de la clase “PNEUMONÍA” y un 98.76% de sensibilidad para la clasificación de la clase “NORMAL” en la evaluación. Adicionalmente se utilizó la técnica de Zero-Shot donde se obtuvo un resultado del 96.67% en la precisión del modelo y una sensibilidad total de 93.33% en la clasificación de neumonía. Finalmente se concluye que la solución propuesta para el ajuste del modelo es eficaz para la clasificación de neumonía, dados los altos porcentajes obtenidos en las métricas de evaluación. Por lo tanto, se establece como un aporte valioso en el estudio del uso de las redes ViT para la clasificación eficiente de enfermedades en imágenes.

Palabras clave: *Visión transformer, Visión por computador, Neumonía, vitbase-patch16-224-in21k, CRISP-ML(Q), radiografía de tórax.*

Abstract

Detecting pneumonia is a high-impact task in the medical sector in general, and illness analysis utilizing radiographic images has been designated as a priority for rapid diagnosis of the medical condition. In this context, the primary goal of the research was to use vision transformer (ViT) networks to classify pneumonia in chest radiography pictures. To achieve this goal, the "vit base-patch16-224-in21k" model was fine-tuned, utilizing the CRISP-ML (Q) methodology as a foundation of reference. The framework facilitated the advancement of the study in three primary stages: data engineering, which involved choosing the chest radiography image collection, and class balancing, which was then completed. The pre-trained model was refined during the model engineering phase to satisfy the requirements of pneumonia classification based on the classes of the "PNEUMONIA" and "NORMAL" datasets. The last stage helped assess the refined model, and the sensitivity metric was used to examine the assessment findings. Upon completion of these phases, the research achieved results showing that the fine-tuned model reached 99.34% accuracy in pneumonia classification. The classification analysis, based on the sensitivity metric, showed that the fine-tuned model obtained a 99.88% sensitivity for the "PNEUMONIA" class and a 98.76% sensitivity for the "NORMAL" class. Additionally, the zero-shot technique was used, yielding a 96.67% model accuracy and an overall sensitivity of 93.33% in pneumonia classification. Finally, it is concluded that the proposed solution for model fitting is effective for pneumonia classification, given the high percentages obtained in the evaluation metrics. Furthermore, it is established as a valuable contribution to the study of the use of ViT networks for the efficient classification of diseases in images.

Keywords: Vision transformer, vision computer, Pneumonía, vitbase-patch16-224- in21k, CRISP-ML(Q), Chest X-Ray.

3. Introducción

La utilización de la inteligencia artificial y las aplicaciones en el campo médico, han tomado gran relevancia en los últimos años. La clasificación de neumonía mediante el procesamiento de imágenes radiográficas, es una tarea que actualmente tiene una gran relevancia dentro de los campos de la visión por computadora y el médico en general. Considerando que, según lo expuesto en [1] la neumonía es la tercera causa de muerte a nivel global, y dentro del Ecuador representa una total de 3000 muertes cada año dentro de diversos grupos sociales. Por lo tanto, la detección temprana y precisa de la enfermedad no solo beneficia individualmente al paciente con la enfermedad, sino que también tiene un impacto positivo a nivel social. Por consiguiente, el presente trabajo de integración curricular plantea utilizar las tecnologías como vías, para solucionar los problemas médicos y resolver los obstáculos en la visión médica [2]. Dado que, la implementación correcta y eficiente de la clasificación de neumonía se ha visto limitada por su implementación en arquitecturas de aprendizaje automático más tradicionales, como las redes neuronales convolucionales (CNN), donde se ha demostrado que su nivel de eficiencia es considerablemente menor en comparación de las redes de Visión Transformers, que, aunque es considerada una arquitectura de aprendizaje automático emergente, se ha comprobado que tiene la capacidad de abstraer y generalizar de manera más efectiva grandes volúmenes de información sin aumentar el costo computacional [3]. Además de que los modelos existentes de clasificación de neumonía mayormente solo abordan la métrica de precisión para realizar sus hipótesis y conclusiones, pero, no contemplan el rendimiento de sus propuestas respecto a métricas con mayor impacto en el campo médico como la sensibilidad [4], que es la capacidad de un modelo para clasificar una enfermedad o identificar los verdaderos positivos. Por lo tanto, surge como pregunta de investigación: ¿Qué porcentaje de sensibilidad se obtendrá en el afinamiento del modelo “vit-base-patch16-224-in21k” para la clasificación de neumonía en imágenes radiográficas de tórax, utilizando la arquitectura de red visión transformer (ViT)?

Para responder a la pregunta de investigación, se establece como objetivo principal analizar en base a la métrica de sensibilidad, el afinamiento del modelo “vitbase-patch16-224-in21k”, con el propósito de utilizar las configuraciones y pesos del modelo en el contexto de la clasificación de radiografías que contengan la enfermedad de neumonía. Además, se establecen como objetivos específicos el ajuste del modelo con el conjunto de datos “Chest X-Ray Images (Pneumonia)” aplicando como marco de trabajo la metodología CRISP ML(Q) y como segundo objetivo específico, la evaluación de los resultados del modelo ajustado, en base a la sensibilidad de clasificación entre clases utilizando el conjunto de datos de prueba.

El desarrollo del trabajo de integración curricular (TIC) se enfocó en cumplir con todas las fases y actividades definidas por el marco de trabajo para el ajuste, entrenamiento, evaluación y análisis de un modelo de aprendizaje automático.

El presente TIC se destaca principalmente por la obtención de un conjunto de datos único obtenido a través de las técnicas de balanceo de las clases dentro del conjunto de datos, junto a la aplicación de técnicas de aumento de datos para mejorar la calidad de información inicial. Además, la realización eficaz del ajuste del modelo que permitió evaluar los resultados en relación con trabajos relacionados. Además, este trabajo en comparación a otros implementó nuevos análisis de clasificación para el modelo ajustado, como el análisis de verdaderos positivos o la prueba del modelo con la técnica Zero-Shot con imágenes nuevas, obtenidas de centros médicos.

Los beneficios que se pueden resaltar en el presente trabajo son: la creación de un conjunto de imágenes radiográficas de tórax estandarizado, que brinda la posibilidad de ser utilizado en investigaciones futuras sin la limitante de buscar el ajuste óptimo de balanceo de clases o técnicas de aumento adecuadas. De igual forma, se destaca la obtención del modelo empaquetado completamente funcional para la clasificación de neumonía en imágenes radiográficas de tórax. El aporte del trabajo destaca de igual forma, la utilización de la metodología CRISP-ML(Q) como marco de trabajo para el desarrollo de modelos de aprendizaje automático y además, se establece como un aporte valioso en el estudio del uso de las redes ViT para la clasificación de enfermedades en imágenes

4. Marco teórico

4.1. Antecedentes

La clasificación de neumonía mediante el procesamiento de imágenes radiográficas, es una tarea que actualmente tiene un gran enfoque dentro del campo de la visión por computadora, ya que plantea utilizar las tecnologías como vías para solucionar los problemas médicos y resolver los obstáculos en la visión médica [5].

Sin embargo, la eficiente clasificación de neumonía se ha visto limitada por su implementación en modelos que utilizan arquitecturas de red como las redes neuronales convolucionales (CNN), que son poco eficientes, a comparación de las redes de ViT, que son redes capaces de abstraer mayor información en un conjunto de imágenes sin aumentar el costo computacional [3].

Por lo tanto, los modelos que se desarrollan a partir de la red ViT, son mejores para la clasificación de imágenes y son la opción más conveniente para la realización de estudios científicos. Las mejoras de las redes ViT se reflejan en métricas como la precisión, tiempo de ejecución y costo computacional [3], esto en las tareas de clasificación de imágenes en general. Pero, al ser una tecnología que se viene desarrollando en un lapso menor a una década, las aplicaciones específicas de este tipo de redes se restringen a que generalmente se utilizan modelos más convencionales como las CNN. Por ende, se establece utilizar la efectividad de los modelos ViT en la clasificación de imágenes, para utilizarla en el contexto de la detección de neumonía, y que, mediante técnicas de afinamiento, se evalué los resultados porcentuales en la métrica de sensibilidad en las predicciones realizadas por el modelo. Para la investigación propuesta, es necesario contar con un modelo ViT especializado en la clasificación de imágenes. Por ello, el modelo “vit-base-patch16-224-in21k” de Google, se señala como la opción para el desarrollo del afinamiento. Ya que, su efectividad en la clasificación de más de 21 mil clases, su arquitectura compuesta por 12 capas de transformadores y sus 144 cabezas de atención, son justificativos para utilizar y abstraer la estructura del modelo, con el fin de que resulte en un modelo especializado en el caso establecido de la clasificación de neumonía en imágenes radiográficas de tórax.

4.2. Fundamentación Teórica

4.2.1. Neumonía

Las Infecciones Respiratorias Agudas (IRAG) son un grupo de enfermedades que afectan el aparato respiratorio y son causadas por diferentes agentes, como virus y bacterias [6]. La neumonía es una forma grave de IRAG y se caracteriza por la inflamación de los pulmones debido a una infección [7]. La neumonía se define de dos maneras dependiendo

del enfoque. Desde el enfoque anatómico-patológico, es una infección de los alvéolos, en la porción distal de las vías respiratorias y el intersticio del pulmón, Por otro lado, el enfoque clínico define la neumonía como los síntomas y signos propios de la enfermedad en combinación con una opacidad de los campos pulmonares en la radiografía de tórax [8].

4.2.2. Radiología

La radiología es una rama de la medicina que utiliza la tecnología imagenológica para diagnosticar y tratar una enfermedad, enfocada principalmente en el uso de imágenes generadas por la radiación para brindar diagnósticos y tratamientos médicos. La especialidad es abordada por médicos especialistas que poseen habilidades para la interpretabilidad de las imágenes [9].

Principalmente la radiología diagnóstica es el área competente respecto al problema planteado ya que ese tipo de radiología permite a los profesionales médicos observar algún tipo de estructuras dentro del cuerpo reflejado en la imagen [10], las ventajas de la radiología son: la detección y diagnóstico de la causa de síntomas de alguna enfermedad, vigilancia o monitorización sobre la respuesta del cuerpo respecto a un tratamiento frente a una afección [11].

4.2.3. Aspectos radiológicos de la detección de neumonía

Según el artículo llamado "Neumonía viral grave en adultos" [9]. Un diagnóstico positivo de neumonía se realiza necesariamente con pruebas de diagnóstico por imagen, lo que puede comprender técnicas de radiología como la radiografía de tórax o tomografía computarizada torácica con el motivo de evaluar las lesiones en el paciente.

La radiografía simple de tórax es el estándar predilecto en el diagnóstico de neumonía, se recalca en el estudio llamado "Diagnóstico por imagen de neumonía" [12], que es importante obtener la posición postero-anterior del tórax en la radiografía para un correcto análisis del paciente, ya que permite una mejor perspectiva frente a la extensión de la enfermedad y sus complicaciones [13]. También se resalta que la radiografía es la opción idónea para el diagnóstico médico ya que referente a la enfermedad las alteraciones en la radiografía pueden ser visibles a las 12 horas del inicio de los síntomas [5].

Aunque existe el diagnóstico microbiológico, las técnicas de imagen tienen un papel importante para apoyar la gravedad de la neumonía debido a que la afectación es principalmente pulmonar, por lo cual el primer método que suele emplearse es la radiografía de tórax, permitiendo una identificación de los pacientes en quienes se evidencia un empeoramiento del estado respiratorio [14].

4.2.4. Transfer Learning (Transferencia de aprendizaje)

La transferencia de aprendizaje surge como una solución a todos los problemas que pueden llegar a surgir frente al desarrollo de modelos de aprendizaje automático [15], como el entrenamiento de redes neuronales artificiales en modelos con tamaños de capas y parámetros extensos, que en la ejecución pueden llegar a ser poco accesibles económicamente [16]. La transferencia de aprendizaje está formada por un conjunto de métodos del aprendizaje automático, donde su funcionamiento se basa en el uso de los pesos de los modelos pre entrenados en determinada tarea, para que en un futuro puedan ser un punto de desarrollo en la adaptación del mismo modelo, pero para una tarea de Computer Vision o Natural Language Processing NLP [17].

4.2.5. Fine Tuning (Ajuste fino)

La técnica de fine Tuning o de afinamiento surgió como una opción a las problemáticas netamente centradas en el diseño de mejores modelos, ya que su función se centra específicamente en buscar la configuración óptima de diversos parámetros de un modelo base para mejorar su rendimiento en una tarea determinada [18]. La técnica se basa sobre una idea que consta de conocer un modelo que ya se ha establecido como solución a un problema y que para ello hayan aprendido características generales que pueden ser útiles para el problema [19]. Esto sirve que para evitar el entrenamiento sea completamente desde cero y que en base a la referencia de los pesos se puedan mejorar para el abordamiento del problema inicial. El fine Tuning posee cierto tipo de ventajas [20] relacionadas al:

- **Reducción del tiempo de entrenamiento:** Al aprovechar un modelo preentrenado, no es necesario entrenar el nuevo modelo desde cero.
- **Mejoras en el rendimiento:** El Fine Tuning puede mejorar el rendimiento del modelo final en comparación con un modelo base.
- **Requiere menos datos:** El Fine Tuning puede ser útil cuando se tiene una cantidad limitada de datos para la nueva tarea.

4.2.6. Balanceo de clases

El desbalance de clases en un conjunto de datos es un problema recurrente en el análisis exploratorio de datos. Se presenta como la predominancia de una cantidad mayor de datos sobre otros, el desbalance también se presenta como la ausencia de datos en las diferentes clases [21]. El problema del desbalanceo de las clases se presenta en el entrenamiento y prueba de modelos como la incapacidad de abstraer la información de los

datos correctamente o con dificultad. La aplicación de técnicas de balanceo de datos puede resultar en la construcción del mejor modelo por la calidad de los datos que se llegan a tener [22]. La aplicación de técnicas de balanceo resulta ser útiles cuando se espera una respuesta dentro de entradas binarias o dos tipos de clases y alguna de las dos clases es muy pequeña en comparación de la otra [23].

4.2.7. Aumento de datos

Las técnicas de aumento de datos se presentan como una solución para los trabajos relacionados al entrenamiento de modelo de aprendizaje automático [24], ya que actualmente el mayor problema para el desarrollo de modelos es la accesibilidad a la limitada información de datos específicos [25]. Por ende, las técnicas de aumento se presentan como una forma de potenciar y maximizar los beneficios que se pueden obtener de los conjuntos de datos iniciales aumentando la cantidad de información cuidando no dañar o afectar a la calidad de los datos. La aplicación de cada técnica de aumento representa la creación de datos nuevos y completamente útiles para los procedimientos de Deep Learning [26].

4.2.8. Redes neuronales

Dentro del campo de la inteligencia artificial, el concepto de las redes neuronales dentro del machine Learning es un aspecto esencial que se debe conocer para adentrarse más dentro de esta área de conocimiento. Los modelos computacionales han mejorado significativamente en las últimas décadas, brindando a la sociedad la capacidad de poder realizar una gama extensa de aplicaciones relacionadas al reconocimiento de voz, generación de texto e imagen, predicción de enfermedades, entre muchas más [27]. Para el campo del machine Learning, existen las llamadas redes neuronales artificiales (ANN), que su concepto fue dado a una analogía respecto al concepto biológico de una neurona. y que para el desarrollo y entrenamiento de modelos en sistemas inteligentes se pueden acoplar los distintos tipos de redes neuronales junto a las técnicas de optimización [28]. La mayoría de los tipos de redes neuronales artificiales cuentan con una estructura similar, constituida por [29]:

- **Capas de entrada:** Es la primera capa de la red, y debe contener el tamaño correcto acorde con el vector de entrada.
- **Capas escondidas:** Son denominadas las capas intermedias, el número de estas no es trivial y dependerá del problema a abordar o el tipo de salida que se requiera.
- **Capa de salida:** Es la última capa de la red y es la encargada de devolver el resultado del problema tratado.

4.2.9. Redes de visión transformer

Las redes de visión transformer (ViT) son una técnica de aprendizaje profundo en desarrollo desde el año 2021, mencionada por primera vez en el artículo llamado “An image is worth 16X16 words: transformers for image recognition at scale” [3]. Las redes de ViT principalmente se basaron en el uso de un modelo de atención para el procesamiento visual proporcionado por las redes Transformers, que inicialmente fueron creadas para funciones de procesamiento de lenguaje natural [30]. El que sea un modelo de atención permite reforzar las partes más sensibles de una imagen de entrada y extraer información importante, evitando el procesamiento de toda la información, dando como resultado la reducción del tiempo de cómputo y un procesamiento de manera paralela. Algo igual de resaltante de las redes ViT es su gran capacidad para ser escalables y la flexibilidad que prestan para la adición de capas en la red sin tener que perder la precisión y la eficiencia [31]. Como se visualiza en la Figura 1, la entrada en la arquitectura es una imagen o una secuencia de imágenes y la salida es una representación vectorial que se utiliza para realizar tareas específicas, que pueden ser la detección de objetos, clasificación de imágenes y procesamiento visual en general [3].

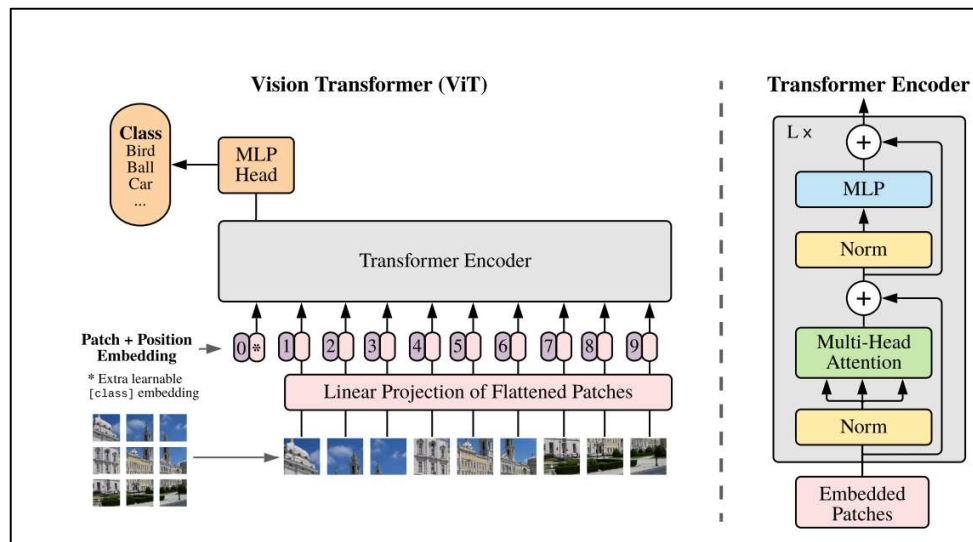


Figura 1. Arquitectura de la red Visión Transformer

4.2.10. Funcionamiento de la arquitectura

La red adopta un enfoque basado en la transformación de imágenes en una serie de pequeños segmentos o “parches” de un determinado tamaño, que representan a una parte de la imagen [3]. Cada parche aplanado y transforman la secuencia de los tokens que es el equivalente a un parche de la imagen. Cada uno del token codifica una parte de la imagen, pero al aplanar la imagen original, se pierde información sobre la jerarquía visual de la imagen.

Por ello, a estos vectores se les añade información sobre el orden en el que se presentaban en la imagen, el llamado embedding posicional [32]. Los vectores se introducen entonces al codificador de la arquitectura Transformer, donde se realiza la self-attention o auto-atención, igual que como si se estuviese tratando de un problema de lenguaje natural. Y gracias a la auto-atención, no se analiza cada parche de la imagen por separado de forma aislada, sino que el propio modelo evalúa la importancia relativa de cada parche en relación con los demás, lo que permite que los ViT capten relaciones a largo alcance entre diferentes partes de la imagen [33].

4.2.11. Zero-Shot learning

La técnica de Zero-Shot Learning o por sus siglas (ZSL) propone la identificación y categorización de elementos, conceptos o clases sin haber sido vistas previamente por el modelo en las diferentes fases de entrenamiento, validación o prueba [34] La aplicación de la técnica se destaca por ser útil en los casos donde los datos etiquetados representan una cantidad minoritaria o son inexistentes para las clases de un conjunto de datos. [35]

4.2.12. Conjuntos de datos

4.2.12.1. Chest X-Ray Images (Pneumonia)

Dentro del proyecto de investigación se planteó utilizar el conjunto de datos llamado “Chest X-Ray Images (Pneumonia)”, dado que se detalla dentro de la información del conjunto, que dos médicos expertos y especialistas calificaron los diagnósticos de las imágenes antes de recibir autorización para entrenar sistemas de inteligencia artificial [36], garantizando así la calidad médica de las imágenes para el uso en la visión por computadora y modelos de inteligencia artificial. El conjunto de datos se utilizó en las diferentes tareas de desarrollo, por lo tanto, se necesitó realizar una exploración sobre el conjunto de datos. Con el fin de controlar la calidad de la información [37].

4.2.12.1.1. Detalles sobre el conjunto de datos

Es necesario realizar una exploración sobre el conjunto de datos y la información de su contenido, ya que será una parte fundamental para el proceso de investigación y desarrollo del ajuste de un modelo [38]:

- **Formato de imágenes:** El conjunto de imágenes contiene las imágenes en formato .jpeg.

- **Número de imágenes:** El conjunto de datos cuenta con 5856 imágenes radiográficas de tórax.
- **Categorías del conjunto de datos:** Cuenta con 2 clases que son: Normal y Neumonía.
- **Distribución de imágenes dentro del conjunto:** En las siguientes tablas se presentan las cantidades de imágenes presentes dentro del conjunto de datos completo, separados por conjuntos de entrenamiento, prueba y validación por cada clase de “Neumonía” o “Normal”.

1. **Conjunto de entrenamiento:** Son un total de 5216 imágenes distribuidas como lo describe la **Tabla 1**:

Tabla 1. Cantidad de imágenes en el conjunto de entrenamiento.

Clases	Nro. imágenes
Normal	1341
Neumonía	3875
Total	5216

2. **Conjunto de Prueba:** Son un total de 624 imágenes distribuidas como lo describe la *Tabla 2*:

Tabla 2. Cantidad de imágenes en el conjunto de prueba

Clases	Nro. imágenes
Normal	234
Neumonía	390
Total	624

3. **Validación:** Cuenta con un total de 16 imágenes distribuidas como lo describe la *Tabla 3*:

Tabla 3. Cantidad de imágenes en el conjunto de validación

Clases	Nro. imágenes
Normal	8
Neumonía	8
Total	16

4.2.12.1.2. Licencia de uso

El conjunto de datos posee la licencia de uso “CC BY 4.0” que es una licencia de Creative Commons que permite a cualquier persona compartir, copiar y redistribuir el material en cualquier medio o formato, y adaptarlo, remezclarlo, transformarlo y construir sobre él para cualquier propósito, incluso comercialmente, siempre y cuando se dé crédito al autor original y se indique si se han realizado cambios. Y se utiliza comúnmente para obras académicas, científicas y de investigación. [39]

4.2.12.2. COVID-QU-Ex Dataset

Para la realización del proyecto de investigación se utilizó un segundo conjunto de datos llamado “COVID-QU-Ex Dataset”, con la finalidad de apoyar el proceso de aumento de datos resultado del balanceo de clases. Por lo tanto, se determinó importante una revisión de la estructura del conjunto de datos. Dentro del conjunto seleccionado se informa que es el resultado de una investigación realizada por investigadores de la Qatar University [40]. La creación del conjunto de datos nació como una investigación resultante de la necesidad de aplicación del diagnóstico asistido por computadora para la enfermedad de COVID-19 [41] El conjunto de datos cuenta con imágenes que resultaron de la aplicación de 5 técnicas diferentes para la mejora de imágenes. El conjunto seleccionado fue utilizado en diversos estudios para la visión médica donde se abordó el análisis de la información en diferentes redes neuronales convolucionales [42], [43]

4.2.12.2.1. Detalles sobre el conjunto de datos

A continuación, se detalla la información del conjunto:

- **Formato de imágenes:** El conjunto de imágenes contiene los datos en formato .png.
- **Número de imágenes:** El conjunto de datos cuenta con 33.920 imágenes radiográficas de tórax.
- **Categorías del conjunto de datos:** Cuenta con 3 clases que son: COVID-19, Infecciones de Neumonía viral o bacteriana y Normal.
- **Distribución de imágenes dentro del conjunto:** Las imágenes del conjunto de datos están por defecto divididas por conjuntos de entrenamiento, prueba y validación por cada clase de “COVID-19”, “Non-COVID” y “Normal”.

4.2.12.2.2. Licencia de uso

El conjunto de datos seleccionado cuenta con la licencia de uso “CC BY-SA 4.0” que es una licencia de Creative Commons que permite a cualquier persona compartir, adaptar,

transformar y construir sobre el material en cualquier medio o formato, y para cualquier propósito, incluso comercialmente, siempre y cuando se le notifique al autor original y se licencien de igual manera todos los resultados de la utilización de la información. [44]

4.2.13. Métricas de evaluación

Dentro del desarrollo del proyecto de investigación, las métricas son herramientas esenciales que permiten valorizar los resultados y el rendimiento de un modelo de aprendizaje automático en una tarea específica. Y que a partir de este tipo valorizaciones se pueda decir si el modelo desarrollado es el mejor o no.

4.2.13.1. Métrica de precisión (accuracy)

La precisión es una métrica principalmente utilizada para evaluar el rendimiento de un modelo de clasificación [45]. En donde su estimación se realiza con la relación entre los verdaderos positivos y el total de elementos que el modelo clasificó como positivos [46].

Ecuación 1. Fórmula de cálculo para la precisión

$$P = \frac{VP}{VP + FP}$$

- **Precisión (P):** Verdaderos Positivos dividido entre la suma de los verdaderos positivos y los falsos positivos
- **Verdaderos positivos (VP):** Son los elementos que el modelo clasificó como positivos y que realmente son positivos.
- **Falsos positivos (FP):** Son los elementos que el modelo clasificó como positivos pero que en realidad son negativos.

4.2.13.2. Métrica de sensibilidad (recall)

Esta es una de las métricas más importantes para la estimación de la clasificación correcta en casos positivos y casos negativos, “La sensibilidad se representa como la fracción de verdaderos positivos, mientras que la especificidad, es la fracción de verdaderos negativos” [47]. Además, dentro del campo médico se describe que la sensibilidad es la capacidad que tiene un sistema para la correcta detección de una enfermedad. Conocida como tasa de verdaderos positivos, la sensibilidad es la contabilización total de los casos correctamente predichos como positivos [48].

Ecuación 2. Fórmula de cálculo para la sensibilidad

$$S = \frac{VP}{VP + FN}$$

- **Sensibilidad (S):** Verdaderos Positivos dividido entre la suma de los verdaderos positivos y los falsos negativos
- **Verdaderos positivos (VP):** Son los elementos que el modelo clasificó como positivos y que realmente son positivos.
- **Falsos positivos (FN):** Son los elementos que el modelo clasificó como negativos pero que en realidad son positivos.

4.2.13.2.1. Elección de métrica para el proyecto

Para la evaluación de modelos en inteligencia artificial existen diversas métricas como la precisión, especificidad, Puntuación F1, Puntuación ROC-AUC, entre otras métricas, sobre las que se puede establecer una conclusión respecto al rendimiento de un modelo [48]. En el contexto de la problemática se establece que la métrica de Sensibilidad es la más adecuada, ya como lo menciona el especialista en inteligencia artificial ing. Oscar Cumbicus Pineda en el anexo A "... la métrica de sensibilidad en el contexto médico es la métrica de mayor impacto y la que utiliza recurrentemente". Además, se añade dentro de su argumentación de ejemplo los estudios realizados para la detección de covid-19 (que se relaciona con la neumonía por COVID) en donde los resultados de efectividad de los modelos se basaban en la sensibilidad hacia los casos positivos, facilitando los diagnósticos correctos [49]. El argumento se respalda en el artículo "Revisión crítica sobre la identificación de COVID-19 a partir de imágenes de rayos x de tórax usando técnicas de inteligencia artificial " [50]. Donde la métrica principal de los estudios es la sensibilidad. Sucede algo similar dentro de la presente investigación, la métrica de sensibilidad servirá para realizar un correcto análisis y evaluación de los resultados frente a los verdaderos positivos en las predicciones realizadas por el modelo que se plantea afinar, puesto que, el enfoque principal de la detección de neumonía es de conocer las imágenes radiográficas donde si se tenga la enfermedad, además se debe a que el uso de la métrica en la clasificación de imágenes es el predilecto para los estudios científicos en el campo de la visión por computadora aplicado a la medicina.

4.2.14. Herramientas de desarrollo y Tecnologías

Dentro del desarrollo de la presente investigación, se necesitan esencialmente varias herramientas y tecnologías que con el uso de las mismas se pueden cumplir los objetivos planteados.

4.2.14.1. Modelo google/ vit-base-patch16-224-in21k

Es un modelo de visión transformer (ViT) que utiliza el codificador de transformador (tipo BERT) pre entrenado con un amplio conjunto de imágenes de forma supervisada, concretamente de ImageNet-21k, con una resolución de 224x224 píxeles [51]. Dentro del modelo, las imágenes se presentan al modelo como una secuencia de parches de tamaño fijo con una resolución de 16x16, los parches están incrustados linealmente. También se agrega un token [CLS] al comienzo de una secuencia para usarlo en tareas de clasificación utilizando la regresión [52]. También se agregan incrustaciones de posición absoluta antes de alimentar la secuencia a las capas del codificador Transformer. El modelo incluye el agrupador previamente entrenado (pre-trained pooler) que es un componente, también conocido como “encoder”, que es responsable de convertir la entrada del modelo (imágenes) en una representación vectorial que pueda ser utilizada por el resto del modelo para realizar la tarea específica como la clasificación de imágenes [3].

4.2.14.2. Estructura del modelo

- **Procesamiento de imagen:** El procesamiento de imagen implica la redimensión de las imágenes de entrada a 224x224 píxeles antes de su procesamiento, lo cual se logra mediante el parámetro “image_size” establecido en 224 [52]. Además, el modelo procesa imágenes en color con 3 canales (RGB), lo que se indica mediante el parámetro “num_channels” establecido en 3. Por último, las imágenes se dividen en parches de 16x16 píxeles para su procesamiento posterior, lo cual se logra mediante el parámetro “patch_size” establecido en “16” [3], [51].
- **Arquitectura transformer:** El modelo pertenece al tipo de arquitecturas conocidas como Vision Transformers, cuya configuración incluye las siguientes características:
 - model_type: “vit”: Confirma que este es un modelo Vision Transformer.
 - num_hidden_layers: 12: El modelo tiene 12 capas de transformadores para procesar la información.

- num_attention_heads: 12: Cada capa de transformador utiliza 12 “cabezas de atención” para centrarse en diferentes aspectos de la imagen.
- hidden_size: 768: La dimensión de los vectores ocultos utilizados en las capas de transformador es 768.
- intermediate_size: 3072: Dimensión intermedia utilizada en las capas de transformador para proyectar los vectores ocultos.
- hidden_act: “gelu”: La función de activación utilizada en las capas de transformador es la GELU (Gaussian Error Linear Unit).
- hidden_dropout_prob: 0.0: No se aplica abandono a los vectores ocultos.
- attention_probs_dropout_prob: 0.0: No se aplica abandono a las probabilidades de atención.
- qkv_bias: True: Se utilizan valores de polarización en las consultas, claves y valores de las capas de atención.
- layer_norm_eps: 1e-12: Parámetro pequeño utilizado para la normalización de capas.
- initializer_range: 0.02: Rango utilizado para la inicialización de los pesos del modelo.
- transformers_version: "4.13.0. dev0": Indica la versión de la librería Transformers utilizada en el modelo

Además, es importante resaltar los permisos que el modelo utiliza, específicamente hace uso de la licencia Apache 2.0, que es una licencia permisiva cuyas condiciones principales requieren la preservación de los derechos de autor y los avisos de licencia. Los contribuyentes proporcionan una concesión expresa de derechos de patente. Las obras bajo licencia, las modificaciones y las obras más grandes se pueden distribuir bajo diferentes términos y sin código fuente. [25]

4.2.14.3. Elección del modelo para el trabajo de integración curricular

Los modelos de ViT ya se han utilizado dentro del contexto de la clasificación de enfermedades en el campo médico, como el estudio “Clasificación de imágenes mediante Redes Neuronales Convolucionales y técnicas de Deep Learning avanzadas: Transformers”, 2021 [53]. Que realiza el análisis de imágenes radiográficas de tórax para la clasificación enfermedad de neumonía. Dentro del estudio de establecer que los factores más determinantes para obtener los mejores resultados en los modelos ViT es la información de entrada, en este caso las imágenes. Por ello, se ha realizado un análisis de modelos ViT entrenados en la clasificación de imágenes en general, que logren

procesar con eficiencia estas entradas. Por ende, se ha llegado a establecido que el modelo “vit-base-patch16-224-in21k” [52], es el más adecuado para la presente investigación. Principalmente se analizaron modelos ViT para la clasificación de imágenes ya que a partir de la estructura de estos modelos se facilita en mayor medida la realización de técnicas de ajuste y afinamiento, para lograr que la arquitectura sea funcional en el caso específico de clasificación de neumonía en imágenes radiográficas de tórax. La estructura en la que el modelo seleccionado “vit-base-patch16-224-in21k” ha sido configurado, refleja su efectividad, ya que, gracias a su entrenamiento con 14 millones de imágenes permite la clasificación de más de 21 mil clases, este notable rendimiento fue posible ya que el modelo fue desarrollado por la empresa Google que ha establecido fuertes bases en la arquitectura del modelo gracias a su accesibilidad en los recursos de computacionales y de información. Entonces se plantea utilizar estas bases de arquitectura del modelo para llevar la eficiencia a casos de clasificación específicos como lo es la presente problemática.

4.2.14.4. Lenguaje de programación Python

Python es un lenguaje de programación de alto nivel, interpretado, de propósito general y multiparadigma. Es ampliamente utilizado en una variedad de campos, pero principalmente es el lenguaje de programación predilecto en el área de la inteligencia artificial y al aprendizaje automático, se ha establecido como la herramienta de desarrollo de sistemas inteligentes por varios motivos, en los cuales está la amplia cantidad de librerías para el uso libre, las más importantes son: TensorFlow, PyTorch, Scikit learn, Keras y Pandas. Estas herramientas facilitan el desarrollo de modelos de IA y ML, y permiten a los científicos de datos centrarse en la investigación y el análisis. [54], [55], [56].

4.2.14.5. Tensorflow

Tensorflow es una biblioteca de código abierto para el aprendizaje automático desarrollada por la empresa de Google. Actualmente es una de las plataformas más líderes para el desarrollo de modelos de aprendizaje automático, gracias a los grandes beneficios que trae su utilización como la flexibilidad en su uso dentro del desarrollo de varios tipos de modelos de ML, también la escalabilidad de su código permite ejecutar diversos modelos en una variedad de dispositivos [57]. Dentro del aprendizaje automático tensorflow es utilizado en aplicaciones de visión artificial para el reconocimiento de objetos, detección de rostros, clasificación de imágenes, también para el análisis de

datos, toma de decisiones para recomendaciones, optimización, automatización, entre otros [55].

4.2.14.6. Pytorch

Es una biblioteca de código abierto para el aprendizaje automático basada en Python. Es una herramienta popular para el desarrollo de modelos de aprendizaje automático dado a que ofrece una API intuitiva, flexible, escalable y una comunidad activa para el constante desarrollo de soluciones tecnológicas [58]. PyTorch se utiliza para una amplia variedad de aplicaciones, como visión artificial, procesamiento del lenguaje natural, robótica, análisis de datos y toma de decisiones [56].

4.2.14.7. Google Colab

También conocido como Google Colaboratory, es un servicio gratuito en la nube que permite a cualquier usuario escribir y ejecutar código Python en el navegador web. Es una herramienta popular para el aprendizaje automático, la ciencia de datos y la educación [59]. El servicio proporciona una amplia gama de herramientas accesibles, pero frente a al desarrollo de modelos de ML, Google Colab ofrece el acceso gratuito a GPUs de alto rendimiento, lo que permite ejecutar modelos de aprendizaje automático con mayor capacidad de cómputo [60].

4.2.14.8. Metodología CRISP-ML(Q)

En el marco del proyecto de investigación se adoptará como referencia una metodología estructurada y sistemática dentro del campo de minería de datos, que es conocida como “CRISP-ML(Q)”, que es, la base de referencia para el desarrollo del proyecto que se llevará a cabo [61]. Se utilizarán ciertos principios referentes a 4 de sus 6 fases que constituyen la estructura de la metodología, permitiendo establecer un proceso riguroso y efectivo para el desarrollo de la presente investigación. Las fases se describen a continuación [62]:

- **Comprensión de datos y negocios:** Comienza con la definición del alcance, los criterios de éxito y la verificación de la calidad de los datos. Se busca asegurar la viabilidad del proyecto. [62], [63] Se recopilan criterios de éxito tanto en negocio, ML y económicos, y estos deben ser medibles. Además, permite a todos los stakeholders especificar la disponibilidad de datos, restricciones regulatorias y requisitos de aplicación, como robustez, escalabilidad, explicabilidad y demanda de recursos [62], [63].

- **Ingeniería de datos:** Se enfoca en preparar los datos para la fase de modelado. Durante esta fase, se realizan tareas de selección de datos, limpieza de datos, ingeniería de características y estandarización de datos [64]. Se identifican características valiosas y necesarias para el futuro entrenamiento del modelo mediante métodos de selección de datos, y se seleccionan los datos descartando muestras que no cumplen con los requisitos de calidad de los datos. También se pueden abordar problemas de clases desequilibradas mediante estrategias de sobremuestreo o submuestreo [65].
- **Ingeniería de modelos de aprendizaje automático:** Se centra en definir uno o varios modelos de aprendizaje automático y depende del problema empresarial que se está tratando de resolver. Las restricciones y requisitos de la fase de entendimiento de negocio y de datos afectan esta fase. Esta fase incluye la selección del modelo, la especialización del modelo y el entrenamiento del modelo. Además, según la aplicación, se puede utilizar un modelo preentrenado, comprimir el modelo o aplicar métodos de aprendizaje por ensamble para obtener el modelo final de ML [66].
- **Evaluación del modelo de aprendizaje automático:** En esta fase, se valida la precisión del modelo entrenado utilizando un conjunto de datos de prueba. Además, se evalúa la robustez del modelo con datos de entrada noisy o incorrectos [66], [67]. El desarrollo del modelo se basa automáticamente en los criterios de éxito o manualmente por expertos en dominio y en aprendizaje automático. Todos los resultados de la fase de evaluación deben ser documentados [64].

4.3. Trabajos Relacionados

En la revisión bibliografía representada en la **Tabla 4**, se logró realizar un análisis del problema planteado en el TIC y las formas en las que fue abordado anteriormente por otros estudios. El análisis de la literatura funcionó como sustento para el desarrollo de una solución.

Tabla 4. Trabajos relacionados

Código	Título	Resumen	Cita
TR01	Clasificación de imágenes mediante Redes Neuronales Convolucionales y técnicas de	El documento está centrado en la clasificación de imágenes médicas, profundizando en modelos de las arquitecturas vision Transformer (ViT). En	[53]

Código	Título	Resumen	Cita
	Deep Learning avanzadas: Transformers	el estudio se analizan varios modelos desde el aspecto teórico y matemático de las partes más importantes que los componen. Además, de cada modelo se especifica un ejemplo en python de los conceptos abordados en el resto del documento, este estudio utiliza un conjunto de datos de imágenes médicas clasificándolas según la patología (normal-neumonía), permitiendo así realizar una comparativa entre varios modelos ViT y CNN.	
TR02	Transfer Learning With Adaptive Fine-Tuning	Este estudio trata y se enfoca en el problema determinado por factores clave en varios tipos de conjuntos de datos, ya que establece que el procesamiento de datos es complicado especialmente en el campo de la medicina. El estudio resalta el afinamiento de modelos ya que no existe un serie de pasos específicos para determinar la estructura correcta y más eficiente dentro de un modelo, el estudio lleva a cabo el método de ajuste basado en la técnica llamada "evolución diferencial (DEFT)" que sirve para la selección de capas para un conjunto de datos objetivo bajo ciertas restricciones. El método se evaluó frente al problema de identificar la enfermedad de osteosarcoma a partir de las imágenes médicas procesadas.	[68]
TR03	Analyzing Transfer Learning of Vision Transformers for Interpreting Chest Radiography	El estudio tiene como objetivo explorar la eficacia de los modelos ViT entrenados en imágenes médicas. Se analizó principalmente un modelo entrenado con el	[4]

Código	Título	Resumen	Cita
		<p>conjunto CheXpert y un conjunto de datos de neumonía pediátrica. se utilizaron modelos de redes neuronales convolucionales, incluidos VGGNet y ResNet, para los que servirán de referencias en el estudio. El estudio concluye que el aprendizaje por transferencia de los ViT pre entrenados muestra resultados mejorados en comparación con la CNN, los modelos ViT demuestran una mayor capacidad de transferencia de aprendizaje en el contexto de imágenes médicas.</p>	
TR04	<p>Comparación y Evaluación de Modelos transformer en la Clasificación de Imágenes Médicas del Dataset NIH Chest X-rays</p>	<p>El estudio tiene el propósito de comparar varios modelos Transformer: ViT, Swin y MaxViT, los modelos fueron entrenados en diferentes conjuntos de datos NIH Chest X-rays y imagesnet. Su objetivo principal fue el analizar la precisión de las arquitecturas en la clasificación enfermedades en radiografías de tórax. exploraron los resultados en diferentes métricas de evaluación, la cantidad de hiper parámetros, entre otros.</p>	[2]
TR05	<p>Vision Transformers in medical computer vision—A contemplative retrospection</p>	<p>El estudio aborda las redes ViT y las imágenes médicas utilizan varios modelos de redes ViT que se utilizan en el estudio de las redes aplicadas a la visión médica. Analiza y estudia las aplicaciones prácticas de las redes, y resalta entre ellas la clasificación de enfermedades en imágenes, segmentación de de elementos anómalos, entre otros. se analizaron los modelos aplicables en cada tarea y discute</p>	[69]

Código	Título	Resumen	Cita
		de los desafíos que aun presenta la presente tecnología.	
TR06	PneuNet: deep learning for COVID-19 pneumonia diagnosis on chest X-ray image analysis using Vision Transformer	El estudio aborda el problema de detección de neumonía. Ya que establece que existen limitaciones en la abstracción de información sobre las imágenes. El estudio aborda el modelo PneuNet para realizar un análisis de imágenes de pulmón, las cabezas de atención de las redes ViT son utilizadas dentro de los parches y capas de los modelos ViT. El estudio plantea soluciones técnicas enfocadas en la medicina, modelos ViT y redes neuronales profundas.	[49]
TR07	Pneumonia Detection on Chest X-ray Images Using Ensemble of Deep Convolutional Neural Networks	Este artículo presenta la clasificación de neumonía a través de la visión por computadora, utiliza y presenta una técnica llamada Ensemble Learning (EL), para la simplificación del análisis de imágenes radiográficas de tórax. La propuesta del estudio está inicialmente planteada para las redes CNN pre entrenados en tarea específicas en el campo médico. Se utilizan varios modelos que se desprenden de las CNN como DenseNet169, MobileNetV2 y principalmente Vision Transformer que es el modelo entrenado en base al conjunto de imágenes de imagesnet.	[70]
TR08	Vision Transformer for Pneumonia Classification in X-ray Images	El estudio plantea un nuevo algoritmo para la clasificación de la enfermedad de la neumonía. El método propuesto por el estudio utiliza la arquitectura de red ViT	[71]

Código	Título	Resumen	Cita
		<p>para la abstracción de información sobre las imágenes y clasificar la enfermedad. En el estudio también se plantea una comparativa entre arquitecturas convencionales: ViT y las CNN. Específicamente se evaluaron los modelos de ViT-B/16y los modelos de CNN de MobileNetV2, VGG16, ResNet-50. En el estudio los resultados de las redes ViT muestran relativamente mejores resultados con un porcentaje de precisión del 94</p>	

La revisión de los trabajos relacionados al problema de investigación del presente proyecto, permiten establecer un punto de referencia sobre el estado actual de la problemática generalmente en el uso de las redes de vision transformer para la clasificación de imágenes en el campo médico, los trabajos se centran específicamente en la enfermedad de neumonía, y en su gran mayoría abordan aspectos de desde un enfoque comparativo entre otros tipos de redes neuronales y arquitecturas más convencionales. En varios de los estudios analizados en la tabla de trabajos relacionados, las redes de visión transformer se establecen como un tipo de arquitectura de red emergente con la promesa de ser mejor a las redes convolucionales en la actividad de clasificación de imágenes médicas, entonces, el uso de las redes ViT es la opción más adecuada para el desarrollo de la investigación. En los trabajos relacionados también se abordan los desafíos que este tipo de redes pueden afrontar, las redes ViT al ser una técnica relativamente nueva en los estudios que las detallan, abordan varias perspectivas a considerar para realizar un diseño óptimo y eficiente, una perspectiva que se ha identificado en los trabajos es en el uso de los conjuntos de datos, ya que se ha encontrado una fuerte tendencia de la precisión del modelo respecto al conjunto de datos de entrada que se utilizó, la red depende en mayor parte de la estructuración de los parches en los que las imágenes de entrada se pueden configurar. Como se analiza en la documentación relacionada, el abordamiento del conjunto de datos es una parte esencial que se debe llevar a cabo dentro de la presente investigación.

Otro enfoque importante, identificado en el estado del arte del problema es la viabilidad del proceso de ajuste de un modelo de visión transformer, en los estudios se establecen que los modelos basados en transformadores muestran mejores resultados frente a técnicas de

transferencia de aprendizaje y de ajuste fino en modelos pre entrenados. Estos estudios establecen una base de investigación para el estudio de técnicas y procedimientos correctos que sean apropiados para el problema de investigación sobre el ajuste de un modelo de clasificación de neumonía en imágenes radiográficas utilizando redes de visión transformers.

5. Metodología

En la presente sección del documento se describe cada una de las tareas y procesos de las etapas metodológicas, que guiaron el desarrollo de la investigación para el proyecto de integración curricular. La sección brinda información completa sobre el área de estudio (*Sección 5.1*), las tareas que formaron parte del procedimiento (*Sección 5.2*) y los métodos y herramientas utilizadas para poder completar cada objetivo planteado (*Sección 5.3*).

5.1. Área de estudio

El escenario en el cual se desarrolló el proyecto de integración curricular titulado “Fitting of the “vit-base-patch16-224-in21k” model for pneumonia classification in chest radiographic images using transformer vision networks.” se encuentra en la ciudad de Loja, Ecuador con dirección Av. Pío Jaramillo Alvarado y Reinaldo Espinosa, en la Universidad Nacional de Loja. Específicamente en facultad de la Energía, las Industrias y los Recursos Naturales no Renovables, para la carrera de computación, como se visualiza en la Figura 2.

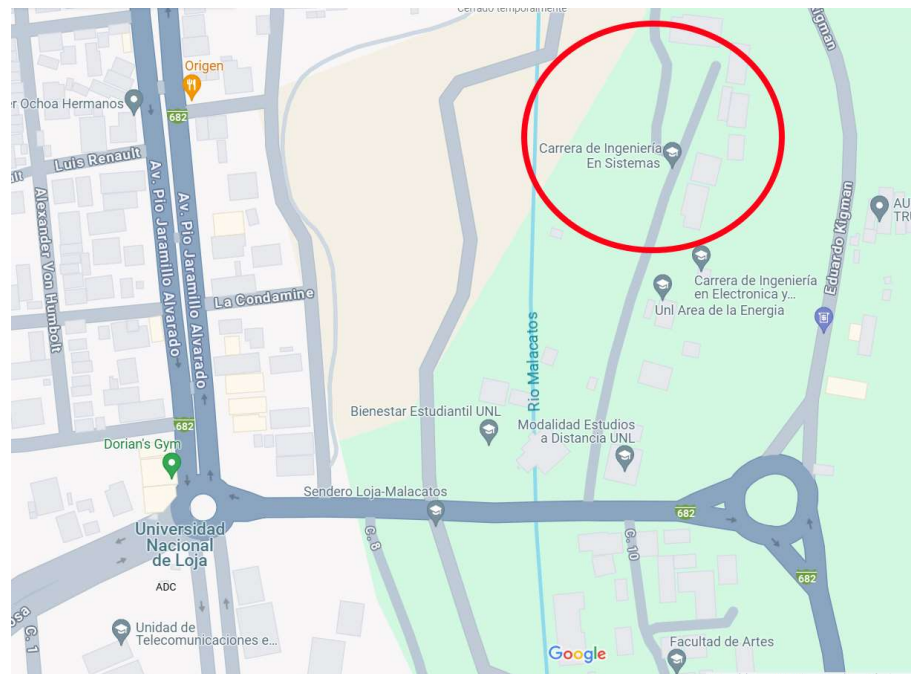


Figura 2. Mapa de la carrera de ingeniería en computación de la Universidad Nacional de Loja

5.2. Procedimiento

Con el fin de poder cumplir con el objetivo general del presente proyecto de integración curricular, se tomó como referencia determinadas fases y tareas de la metodología para el aseguramiento de la calidad en el desarrollo de modelos de machine learning “CRISP-ML(Q)”,

Entre las fases están: la fase de Ingeniería de datos, ingeniería de modelos de aprendizaje automático y la evaluación de los modelos de aprendizaje automático. Finalmente, para el establecimiento de las tareas se utilizó como referencia las recomendaciones de la metodología.

5.2.1. Objetivo 1: Ajustar el modelo “vit-base-patch16-224-in21k” de Google, para la clasificación de neumonía en imágenes radiográficas de tórax utilizando el conjunto de datos “Chest X-Ray Images (Pneumonia)”.

Para el alcance del objetivo 1, se utilizaron las fases de la metodología propuesta donde se incluye la primera fase de ingeniería de datos, la cual está enfocada completamente en la preparación de los datos para trabajar con los modelos de aprendizaje automático y que se compone de ciertas tareas que se acoplan a la investigación, como lo son:

• Tarea 1. Recolección y selección de datos

Inicialmente se realizó la búsqueda en el repositorio Kaggle¹, para un conjunto de datos que contenga con radiografías de tórax. Inicialmente se seleccionó un conjunto de datos con determinadas características, como que, las imágenes radiográficas deben ser tomadas desde una perspectiva anterior-posterior de tórax. Además, debe contener dos tipos de clases “Neumonía” y “Normal”, para la clase de neumonía no sé tomo a consideración los diferentes agentes causales de la enfermedad. Finalmente, se consideró las veces que el conjunto de datos fue citado, utilizado en estudios relacionados y la fuente de obtención de los datos.

Para el desarrollo de las siguientes fases del proyecto, de igual forma se tuvo que seleccionar un segundo conjunto de datos. Los criterios de selección para el segundo conjunto de datos son similares al primer conjunto con la adición de realizar un análisis en que, el conjunto no sea un derivado, modificación o que la fuente de los datos sea la misma del primer conjunto, para evitar que los datos lleguen duplicarse y así dañar la información, cómo se presenta en la sección 6.1.1.

• Tarea 2. Exploración de los datos

Se realizó un análisis exploratorio de los datos con las imágenes que conforman el conjunto seleccionado. El procedimiento se llevó a cabo en cada una de las clases y los subconjuntos destinados al entrenamiento, validación y prueba, como se presenta en la

¹ <https://www.kaggle.com>

sección 6.1.2. se describió la distribución de las clases y la cantidad de imágenes por subconjunto.

- **Tarea 3. Balanceo de clases**

Una vez realizado el análisis de los datos, se determinó necesario un balanceo de clases en el conjunto de datos seleccionado. Como se presenta en la sección 6.1.3, se eligió el aumento de datos como la técnica para el balanceo de datos. Para seleccionar las técnicas de aumento de datos (DA), se utilizaron las recomendaciones encontradas en trabajos relacionados [72], [73][74], [75]. Permitiendo así determinar que las técnicas de aplicación de filtros y adición de imágenes son las técnicas más adecuadas en las redes ViT y en modelos que utilicen imágenes médicas como lo son las radiografías. Estas técnicas fueron seleccionadas y aplicadas a la clase NORMAL de los subconjuntos de datos desbalanceados de entrenamiento y prueba. La aplicación de las técnicas se llevó a cabo mediante la ejecución de las subtareas:

- **Tarea 3.1. Técnica de aumento de datos 1: Aplicación de filtros en el conjunto de entrenamiento**

Para el balanceo del conjunto de entrenamiento, se seleccionó una cantidad de imágenes del conjunto “Chest X-Ray images (Pneumonia)” para la aplicación de los filtros. Para establecer el número de imágenes a las que se aplicarían los filtros seleccionados, fue necesario dividir el conjunto de datos de entrenamiento original en 4 partes iguales, y luego seleccionar la cantidad de imágenes que represente $\frac{1}{4}$ del conjunto de datos. Para la selección del subconjunto al que se aplicarían la técnica de aumento de datos, se seleccionó aquel con mayor cantidad de imágenes en comparación a los demás conjuntos resultantes. A este subconjunto, se empleó la técnica de aumento de datos elegida, que consistió en la aplicación de los filtros, como se presenta en la sección 6.1.3.1.1 se implementó la rotación aleatoria, giro horizontal, intensidad de brillo aleatorio y aplicación de desenfoque Gaussiano en regiones aleatorias de la imagen.

- **Tarea 3.2. Técnica de aumento de datos 2: Adición de imágenes desde el segundo conjunto de datos seleccionado hacia el conjunto de entrenamiento**

Dada la aplicación del primer balanceo de clases, se aplicó la segunda técnica de aumento de datos seleccionada, que fue la adición de imágenes. Como se presenta en la sección 6.1.3.1.2 las imágenes son inicialmente extraídas del segundo conjunto de datos seleccionado “COVID-QU-Ex dataset” para posteriormente ser transformadas del formato *PNG* a *JPEG*, que es el formato de todas las imágenes del conjunto de

entrenamiento y prueba. Finalmente, las imágenes se añaden al subconjunto de datos de la clase NORMAL que resultó de la aplicación de la primera técnica de aumento de datos.

- **Tarea 3.3. Técnica de aumento de datos 2: Adición de imágenes desde el segundo conjunto de datos seleccionado hacia el conjunto de prueba**

La segunda técnica de aumento de datos se aplicó para equilibrar la clase "NORMAL" en el conjunto de prueba. Como se presenta en la sección 6.1.3.2 , las imágenes inicialmente fueron extraídas aleatoriamente del segundo conjunto de datos "COVID-QU-Ex dataset" y posteriormente transformadas del formato *PNG* a *JPEG* para la correcta adición a la cantidad total de imágenes destinadas a pruebas en el conjunto "Chest X-Ray images (Pneumonia)".

La siguiente fase en el proceso de desarrollo, es la de ingeniería de modelos, que comprende actividades enfocadas al desarrollo y construcción de modelos ML. La fase se desarrolló con la utilización de las siguientes tareas establecidas por la metodología:

- **Tarea 4. Ajuste fino del modelo (fine tuning)**

Para el ajuste del modelo se utilizaron los conjuntos de datos de validación y los conjuntos de datos aumentados de entrenamiento y prueba, además para la delimitación de clasificación de clases y la abstracción de pesos se utilizaron las funciones proporcionadas por el sitio de Hugging Face² para la correcta configuración del modelo para la clasificación en 2 clases, como se presenta en la sección 6.1.4.

- **Tarea 5. Entrenamiento del modelo**

Para realizar el entrenamiento del modelo, inicialmente se configuró los hiperparámetros que corresponden a la función de pérdida y la función de activación para controlar la clasificación de las clases y la actualización de los pesos del modelo, posteriormente se definió una cantidad de épocas con el propósito de poder manejar la duración y el proceso de aprendizaje del modelo, en la ejecución de la función de entrenamiento. El entrenamiento del modelo se llevó a cabo en el clúster de GPU's proporcionado por la Universidad Nacional de Loja, como se presenta en la sección 6.1.5.

² <https://huggingface.co/google/vit-base-patch16-224-in21k>

- **Tarea 6. Documentación de resultados**

Se documentó los resultados obtenidos en la fase de entrenamiento del modelo, con las configuraciones y los valores de los hiperparámetros. Además, como se presenta en la sección 6.1.5, se realizó un análisis en base a los resultados de la fase de entrenamiento. Según los resultados en las métricas de pérdida, precisión y sensibilidad, junto al análisis de la matriz de confusión.

5.2.2. Objetivo 2: Evaluar el ajuste del modelo, por medio de la métrica de sensibilidad, para la clasificación neumonía en imágenes radiográficas de tórax, utilizando el conjunto de datos para validación “Chest X-Ray Images (Pneumonia)”.

- **Tarea 6. Evaluación del modelo ajustado con el conjunto de entrenamiento**

Se realizó un análisis dentro los resultados obtenidos en el entrenamiento del modelo ajustado. Con la utilización de la matriz de confusión se realizó el cálculo de los resultados en base a las métricas de precisión, sensibilidad y especificidad. Como se detalla en la sección 6.2.1

- **Tarea 7. Evaluación del modelo ajustado con el conjunto de prueba**

Se codificó una función de evaluación para el modelo ajustado con el conjunto de imágenes de prueba y permitir obtener el porcentaje de precisión del modelo y la matriz de confusión junto a la cantidad de los verdaderos positivos y falsos negativos, como se detalla en la sección 6.2.2.

- **Tarea 8. Análisis de clasificación de verdaderos positivos y falsos negativos**

Se realizó un análisis dentro de las imágenes que fueron incorrectamente clasificadas con la función de evaluación, mediante la utilización de la matriz de confusión y la obtención del archivo de imagen con la información de la etiqueta real y la etiqueta predicha, como se detalla en la sección 6.2.2.1.

- **Tarea 9. Análisis de sensibilidad en la prueba del modelo ajustado**

Se realizó el análisis de los porcentajes de la métrica de sensibilidad y especificidad del modelo con la utilización de los datos obtenidos de la matriz de confusión y la fórmula de la métrica de sensibilidad, como se indica en la sección 6.2.2.2.

• Tarea 10. Análisis del modelo ajustado mediante la técnica de Zero-Shot Learning

Se aplicó la evaluación del modelo con la utilización de la técnica Zero-Shot, en la cual se requirió crear un nuevo conjunto de datos que cuente con imágenes completamente desconocidas para el modelo, o que no hayan participado en las fases de evaluación, prueba o validación. Como se presenta en la sección 6.2.2.3 , para la recolección de las imágenes radiográficas de tórax se tuvo acceso a centros médicos y radiológicos en la localidad, donde se apoyó con el criterio de profesionales para la selección de las imágenes radiográficas de tórax con la enfermedad de neumonía y las imágenes que no presentaban la afección médica. Al finalizar la recolección de las imágenes se valoró los resultados de precisión del modelo y el porcentaje de sensibilidad mediante la utilización del matiz de confusión.

• Tarea 11. Documentación de resultados

Se documentó los resultados obtenidos en la fase de evaluación del modelo, con el modelo resultante del entrenamiento (sección 6.2.2). Además, se realizó un análisis en base a los resultados de la fase de evaluación según las métricas de precisión y sensibilidad, junto al análisis de la matriz de confusión.

5.3. Recursos

5.3.1. Recursos metodológicos

- **Método experimental**

El recurso experimental permitió la adaptación de las fases de la metodología CRISP ML(Q) para el tratamiento de los datos en el conjunto original, entrenamiento del modelo y su respectiva experimentación con hiperparámetros y finalmente la evaluación y análisis de los resultados.

- **Técnicas de aumento de datos**

Las técnicas de aumento de datos fueron fundamentales, ya que dada su implementación se logró completar el balanceo del conjunto de datos. Como se presenta en la sección 6.1.3.1, las técnicas utilizadas fueron:

- Aplicación de filtros como el giro aleatorio, rotación horizontal, brillo aleatorio y el desenfoque gaussiano aleatorio
- Adición de imágenes de conjuntos externos, permitieron el aumento de imágenes válidas para el entrenamiento del modelo. La selección de la cantidad de imágenes por agregar, es definido por el número de imágenes disponibles en el conjunto

- **Ajuste fino (Fine Tuning)**

La técnica permitió el ajuste y la adaptación del modelo “vit-base-patch16-224-in21k” para la utilización en la clasificación de la enfermedad de neumonía, gracias a la posibilidad de establecer diferentes configuraciones de hiperparámetros con el propósito de potenciar los resultados del modelo en la clasificación de las clases

- **Sensibilidad**

Métrica la cual permitió la evaluación del modelo ajustado para conocer la tasa porcentual de la clasificación de los verdaderos positivos con el conjunto de datos balanceado. Los datos para el cálculo de la métrica se obtuvieron utilizando la matriz de confusión.

5.3.2. Recursos técnicos

- **Clúster de GPU's**

Se utilizó un clúster de GPU's brindado por la Universidad Nacional de Loja, para el entrenamiento del modelo y la experimentación de los diferentes hiperparámetros, se seleccionó utilizar este recurso gracias a la disponibilidad en términos de tiempo y recursos computacionales, además de la potencia de cómputo.

- **Jupyter Lab**

Es el entorno de desarrollo web utilizado para el tratamiento de las imágenes de los conjuntos de datos, además de, la codificación de la arquitectura del modelo y pruebas en si entorno de ejecución.

- **Kaggle**

Refiere al repositorio digital de información que permitió el acceso al conjunto de datos original sobre el cual se aplicaron las técnicas de balanceo y al conjunto utilizado para el aumento de datos, como se presenta en la sección 6.1.3.

5.3.3. Participantes

El proyecto de integración curricular fue llevado a cabo, dada la participación de las siguientes personas involucradas en el desarrollo:

- El estudiante Eduardo José Paccha Rojas, en el rol de autor para el desarrollo del proyecto de integración curricular conforme a los objetivos planteados en las secciones 5.2.1 y 5.2.2.
- Ing. Óscar Miguel Cumbicus Pineda, como director a cargo de la supervisión del desarrollo de las diferentes actividades del proyecto de integración curricular y correcciones del mismo.

6. Resultados

En la siguiente sección, se presentan y describen los resultados obtenidos durante la ejecución de los objetivos planteados en el proyecto de integración curricular, con el fin de alcanzar el objetivo general establecido. La investigación involucró las actividades ya establecidas en la sección 5, y que estuvieron centradas en la preparación de datos, desarrollo, entrenamiento y evaluación del modelo. Todas estas actividades se completaron, y los resultados se exponen en las siguientes subsecciones.

6.1. Objetivo 1: Ajustar el modelo “vit-base-patch16-224-in21k” de Google, para la clasificación de neumonía en imágenes radiográficas de tórax utilizando el conjunto de datos “Chest X-Ray Images (Pneumonia)”.

Los siguientes resultados corresponden a la fase: **Ingeniería de datos**

6.1.1. Tarea 1: Recolección y selección de los datos

Antes de haber comenzado el proceso de ajuste del modelo “vit-base-patch16-224-in21k” para la clasificación de neumonía. El conjunto seleccionado, se obtuvo desde el repositorio digital Kaggle, como se visualiza en la *Figura 3*, el conjunto ya ha sido utilizado anteriormente en varios estudios relacionados a la visión médica.

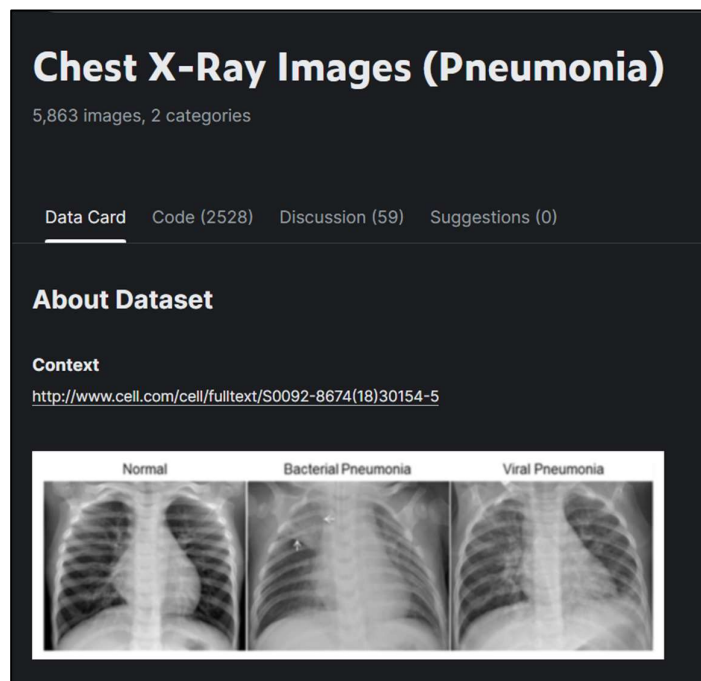


Figura 3. Repositorio del conjunto de datos seleccionado “Chest X-Ray Images (Pneumonia)”

La información completa del conjunto de datos se detalla en la *Tabla 5* donde se aborda la información respecto a cada una de las dos clases de las imágenes radiográficas, en los conjuntos de datos divididos para el entrenamiento, pruebas y validación. La selección de datos resultó en la utilización de las 5.856 imágenes totales de las clases de “PNEUMONIA” Y “NORMAL” para el desarrollo del primer objetivo del proyecto. Por lo tanto, se empleó el conjunto total de datos “Chest X-Ray Images (Pneumonia)”.

Tabla 5. Cantidad de imágenes por clase en los conjuntos de datos en entrenamiento, prueba y validación

Clase	Entrenamiento	Prueba	Validación
NORMAL	1341	234	8
PNEUMONIA	3875	390	8
Total	5216	624	16

El conjunto de datos inicial, conformado por 5856 imágenes totales divididas en conjuntos de imágenes destinadas a las fases entrenamiento, prueba y validación.

Para el desarrollo de las próximas fases del proyecto, se seleccionó un segundo conjunto de datos. El conjunto fue obtenido desde el mismo repositorio que el primer conjunto. Pero con la característica que la selección de los datos fue enfocada especialmente en la búsqueda de imágenes radiográficas de tórax de la clase “NORMAL”.

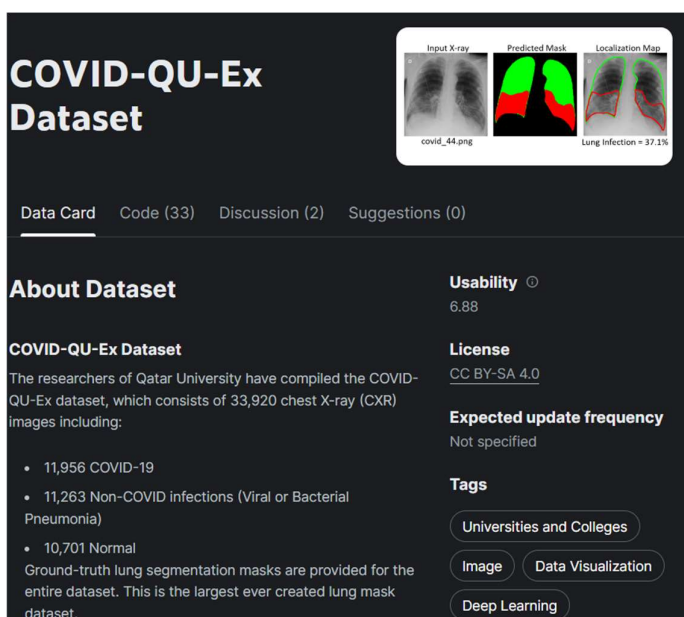


Figura 4. Repositorio del segundo conjunto de datos seleccionado “COVID-QU-Ex Dataset”

La información del segundo conjunto de datos se detalla en la *Tabla 6*. Donde aborda la información respecto a cada una de las clases y los subconjuntos de datos divididos que lo

conforman. Como resultado de la selección de datos, el segundo conjunto de datos sirve para la extracción de las imágenes específicamente de la clase “Normal”. Lo que corresponde a 932 imágenes del conjunto de entrenamiento y 291 del conjunto de prueba.

Tabla 6. Cantidad de imágenes por clase en el segundo conjunto de datos.

Clase	Entrenamiento	Validación	Prueba
COVID	1.864	466	583
Non-COVID	932	233	292
Normal	932	233	291
Total:	3.728	932	1.166

El segundo conjunto de datos seleccionado, está conformado 3 clases COVID, Non-COVID y Normal, que en total suman 5.826 imágenes totales divididas en conjuntos de imágenes destinadas a las fases entrenamiento, prueba y validación.

6.1.2. Tarea 2: Exploración de los datos

Una vez que se obtuvo el primer conjunto de datos seleccionado “Chest X-Ray Images (Pneumonia)”, se realizó un análisis exploratorio EDA para poder conocer el estado y la distribución de las clases del conjunto. Cada uno de los 3 subconjuntos de entrenamiento, validación y prueba, cuentan con imágenes que corresponden a las clases “PNEUMONIA” y “NORMAL”. En el análisis del primer conjunto de datos se observó que la mayoría de información está distribuida mayormente en el subconjunto de entrenamiento, a comparación de los demás subconjuntos de validación y prueba, como se puede visualizar en la Figura 5.

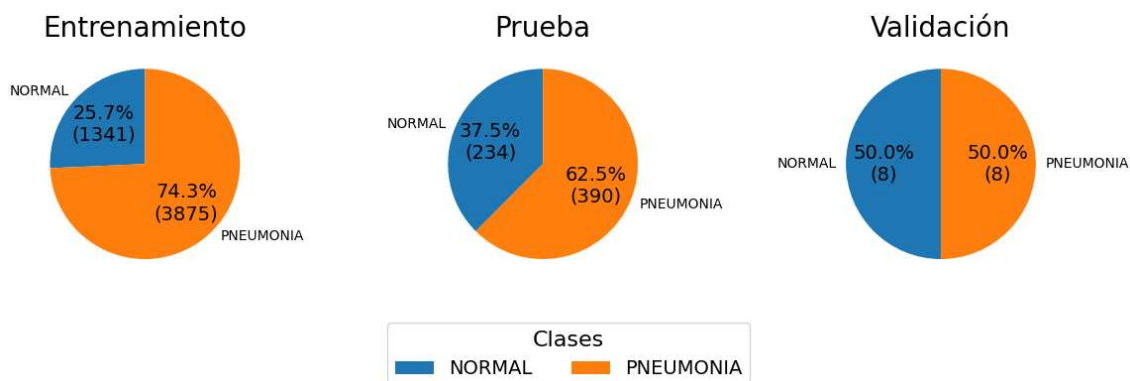


Figura 5. Distribución y cantidad de las imágenes por clase y subconjunto de datos en el conjunto “Chest X-Ray Images (Pneumonia)”

El análisis exploratorio de los datos, resultó en la siguiente conclusión sobre la distribución de las clases: En el primer conjunto de datos “Chest X-Ray Images (Pneumonia)”, las clases de “PNEUMONIA” y “NORMAL” se encontraban desbalanceadas a favor de la clase

de "PNEUMONIA" siendo la clase "NORMAL" la que contaba con menos cantidad de imágenes en los conjuntos de entrenamiento y prueba, el conjunto de validación se encontraba perfectamente balanceado. Se pueden describir los conjuntos desbalanceados de la siguiente manera: El conjunto de entrenamiento cuenta con 5.216 imágenes totales en donde 1.341 son imágenes radiográficas de la clase "NORMAL", mientras que la clase "PNEUMONIA" contaba con 3.875, resultando en un desequilibrio de 2.534 imágenes en comparación con la clase "NORMAL" como se muestra en la Figura 6.

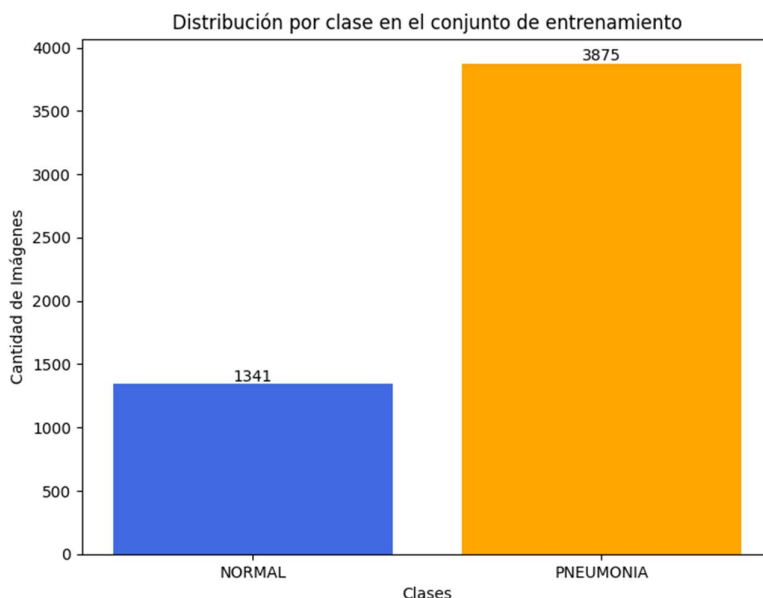


Figura 6. Distribución de clases del primer conjunto de datos para el entrenamiento del modelo

Durante el análisis exploratorio de datos, se determinó que el conjunto de prueba del modelo también tenía un desbalance en la clase "NORMAL". El conjunto contaba con 234 datos en la clase "NORMAL" y 390 en la clase "PNEUMONIA", como se puede visualizar en la Figura 7, así de esta manera se tenía un balance de 156 imágenes entre clases.

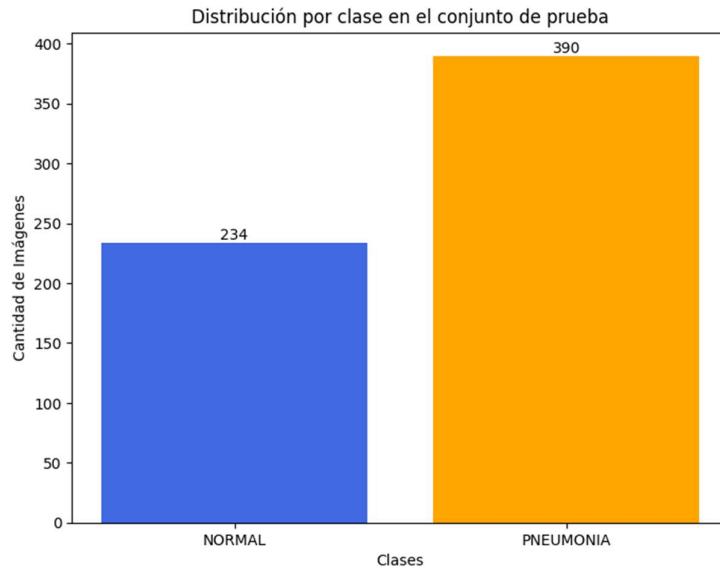


Figura 7. Distribución de clases del primer conjunto de datos para la prueba del modelo

Finalmente, durante el procedimiento del análisis exploratorio de datos, se analizó la distribución de las clases del conjunto de datos destinado a la validación del modelo. Donde el análisis concluyó que las 2 clases del conjunto estaban perfectamente balanceadas, con un número total de 8 imágenes por clase como se muestra en la Figura 8. Sin embargo, es necesario resaltar que la escasa cantidad de imágenes limita la realización de una correcta evaluación del modelo.

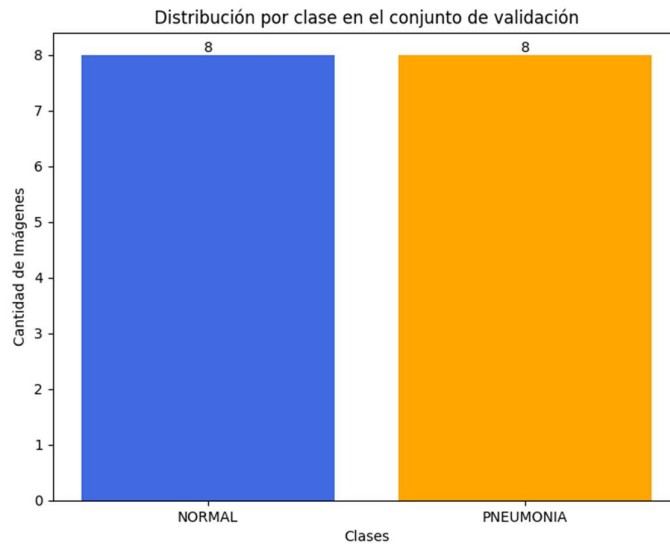


Figura 8. Distribución de clases del primer conjunto de datos para la validación del modelo

6.1.3. Tarea 3: Balanceo de clases

En la sección actual se detallarán las tareas ejecutadas después de la selección y análisis del primer conjunto de datos “Chest X-Ray Images (Pneumonia)” para ajustar la distribución de los datos en los conjuntos de entrenamiento y prueba. Para el balance de las clases en el conjunto de entrenamiento, se aplicaron cuatro filtros a un subconjunto de datos como se detalla en las secciones 5.2.1. y 6.1.3.1. Además, se añadió una segunda técnica de aumento de datos que es la adición de imágenes desde el conjunto de datos seleccionado “COVID-QU-Ex dataset”. Esta técnica complementó el balanceo del conjunto de entrenamiento y ayudo al balanceo del conjunto de prueba.

6.1.3.1. Balanceo del conjunto de entrenamiento

Con la aplicación de filtros a la clase “NORMAL”, se logró balancear los datos el conjunto de entrenamiento al seleccionar y procesar $\frac{1}{4}$ del total de imágenes del conjunto original. Sin embargo, para poder cumplir con el planteamiento de selección de los datos correctamente, se utilizó el siguiente código en Python representado en la Figura 9.

```
imagenes = os.listdir(image_folder)
random.shuffle(imagenes)
num_imagenes_por_conjunto = len(imagenes) // 4

# Seleccionar el ultimo conjunto de la división
conjunto_seleccionado = imagenes[num_imagenes_por_conjunto * 3:]
```

Figura 9. Código para la división y selección del subconjunto para la aplicación de filtros

El código se utilizó para mezclar las imágenes de la clase “NORMAL” con la función *random.suffle()* de la librería *random* de Python. Con el propósito de dividir el conjunto de imágenes a las que se les aplicará los filtros de manera aleatoria. Para la división del conjunto, se utilizó el operador de división entera de Python para descartar la parte decimal y obtener un número entero resultante considerando que la división de 1.341 imágenes de la clase “NORMAL” en el conjunto de entrenamiento, no resulta en un valor entero, si no que resulta en el número decimal de 335.35. En razón de los expuesto, se utilizó el operador de división para la obtención de 4 subconjuntos distribuidos equitativamente, representados en la Tabla 7. Finalmente se seleccionó el subconjunto al que se aplicarían la técnica de aumento de datos, siendo este el subconjunto 4 que posee 337 imágenes.

Tabla 7. Cantidad de datos resultantes de la división del conjunto de datos de la clase “NORMAL”

Conjunto dividido	Cantidad de imágenes
Subconjunto 1	334
Subconjunto 2	335
Subconjunto 3	335
Subconjunto 4	337
Total	1.341

6.1.3.1.1. Tarea 3.1. Aumento de datos al conjunto de entrenamiento: Aplicación de Filtros

Una vez obtenido el subconjunto de imágenes, se llevó a cabo la aplicación de la primera técnica de aumento de datos en las 337 imágenes seleccionadas. Los filtros utilizados en la técnica de aumento de datos incluyen

- **Rotación Aleatoria**

Se empleó la siguiente técnica de aumento de datos en las imágenes seleccionadas. El filtro consiste en establecer giro aleatorio de entre 0 y 180 grados en sentido antihorario utilizando la función *RandomRotation* de la librería *transforms* de *torchvision*. El resultado de la aplicación de este filtro genera una nueva imagen con un grado de rotación aleatorio, el filtro ayuda a generar una mayor diversificación en el conjunto de datos. Las imágenes que resultan de esta técnica están representadas en la Figura 10.

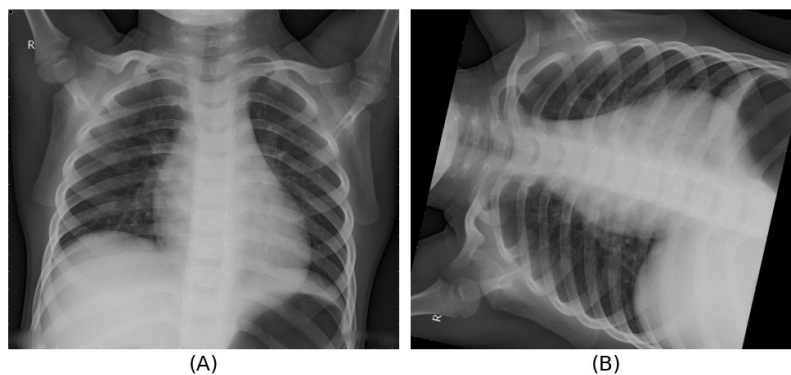


Figura 10. Aplicación del filtro rotación aleatoria en imagen radiográfica. La imagen (A) muestra la imagen original de la radiografía, la imagen (B) muestra la imagen luego de ser aplicado el filtro.

Para la aplicación del filtro se utilizó del código representado en la Figura 11. Donde, la utilización de la librería *OpenCV* sirvió para la lectura de las imágenes de la clase “NORMAL”,

una vez realizada la lectura de las imágenes se utiliza *cv2* para la transformación del formato de la imagen de *BGR* a formato *RGB*, que es el formato adecuado para *Pytorch*. Luego se utilizó la librería *transforms* para convertir las entradas (imágenes) en tensores para la utilización de la función de *RandomRotation*. La función fue definida con los parámetros de 0 y 180 en el rango de rotación de la imagen.

```
# Obtener lista de archivos en la carpeta
image_files = os.listdir(conjunto_da_dir)

# Transformación para rotación aleatoria
rotater = transforms.RandomRotation(degrees=(0, 180))

for image_file in image_files:
    # Cargar la imagen
    image_path = os.path.join(conjunto_da_dir, image_file)
    image = cv2.imread(image_path)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # Convertir la imagen a tensor
    tensor_image = transforms.functional.to_tensor(image)

    # Aplicar rotación aleatoria a la imagen
    rotated_img = rotater(tensor_image)
```

Figura 11. Código para la aplicación de rotación aleatoria.

- **Giro horizontal (Horizontal flip)**

Se empleó la técnica de giro horizontal o también conocida Horizontal flip en todo el conjunto de imágenes seleccionadas como se visualiza en la Figura 12. Para la aplicación de este filtro se usó el código de la Figura 13. Se utilizó la librería de *OpenCv* para la lectura de las imágenes y para realizar el giro con su función *flip*. La función *flip* recibe como entrada la imagen que se quiere realizar el giro y el parámetro del eje en el cual se deseó realizar el giro, en este caso 1 significa que el giro debe realizarse en el eje Y.

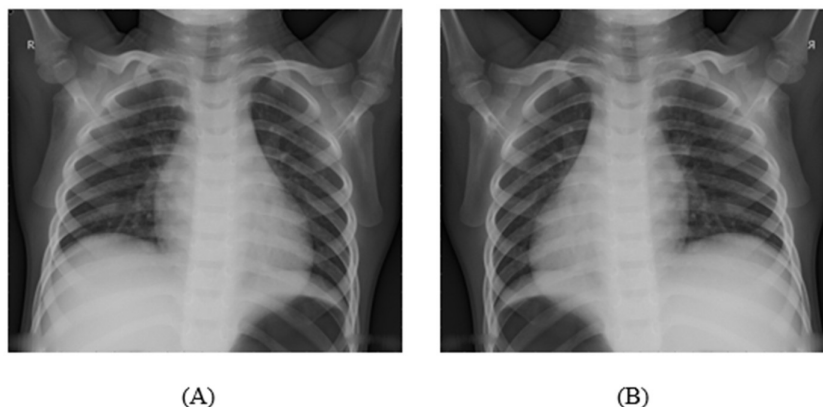


Figura 12. Aplicación del filtro de giro horizontal en imagen radiográfica. La imagen A muestra la imagen original, la imagen B muestra el resultado de la imagen aplicado un giro horizontal.


```

# Obtener lista de archivos en la carpeta
image_files = os.listdir(conjunto_da_dir)

for image_file in image_files:
    # Cargar la imagen
    image_path = os.path.join(conjunto_da_dir, image_file)
    image = cv2.imread(image_path)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # Voltar horizontalmente la imagen
    flipped_image = cv2.flip(image, 1)

```

Figura 13. Código para la aplicación de giro horizontal

- **Desenfoque Gaussiano**

Para la aplicación de este filtro se establecieron 2 factores específicos para la implementación de esta técnica en el conjunto de imágenes, que son: la determinación del número de zonas o regiones a las que se les aplicará el desenfoque y las coordenadas en las que se presentarán como se representan en la Figura 14. La codificación de estos elementos se presenta en la Figura 15 donde se detalla el código realizado para la aplicación del filtro en el conjunto de imágenes. Con el uso de *OpenCV* se pudo leer el conjunto seleccionado y obtener las dimensiones de cada imagen con la función *shape*. Para determinar las coordenadas en las que se presentara el desenfoque, se utilizó la función *randint* de la librería *random* de *Python*, se utilizó de forma que las formas de las regiones de desenfoque no sean siempre las mismas y tampoco se repita el lugar donde se presenten dentro de la imagen. Se especificó que el número de bloques de desenfoque sea un número aleatorio entre 5 y 10 para definir esta cantidad. Con el uso de *cv2* y su función *GaussianBlur* se aplicó el desenfoque con los parámetros por defecto del kernel (25,25) y un valor de 0 en la desviación estándar significando que la aplicación del filtro es un desenfoque más suave.

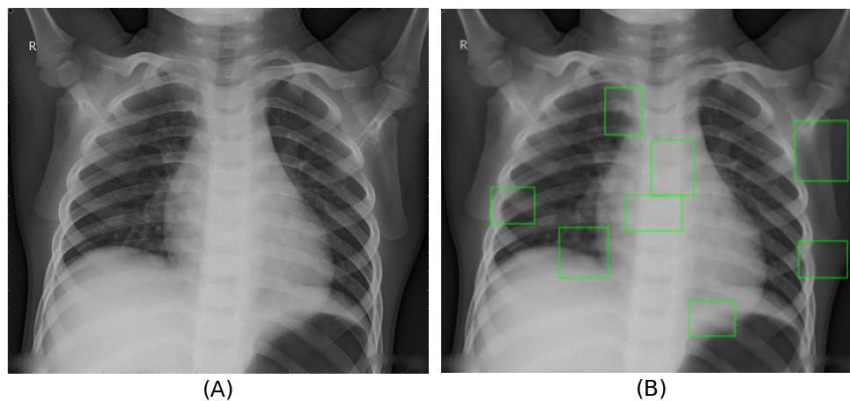


Figura 14. Aplicación de Desenfoque Gaussiano en imagen radiográfica. La imagen (A) representa la radiografía normal y La imagen (B) representa el filtro aplicado. Establece un número aleatorio de bloques para el desenfoque en regiones aleatorias de la imagen (Los bloques están delimitados por líneas color verde solamente para fines explicativos)

```

# Número de regiones a desenfocar
num_regions = random.randint(5, 10)

for image_file in image_files:
    # Cargar la imagen
    image_path = os.path.join(conjunto_da_dir, image_file)
    image = cv2.imread(image_path)

    # Obtener dimensiones de la imagen
    height, width, _ = image.shape

    for _ in range(num_regions):
        # Generar coordenadas aleatorias para la región a desenfocar
        x = random.randint(0, width - 1)
        y = random.randint(0, height - 1)
        w = random.randint(50, 200) # Ancho de la región
        h = random.randint(50, 200) # Alto de la región

        # Aplicar el desenfoco gaussiano a la región seleccionada
        region = image[y:y+h, x:x+w]
        blurred_region = cv2.GaussianBlur(region, (25, 25), 0)
        image[y:y+h, x:x+w] = blurred_region

```

Figura 15. Código para la aplicación de Desenfoco Gaussiano

- **Intensidad de brillo aleatorio**

La aplicación de la intensidad de brillo aleatoria, es la última aplicación de filtros en esta fase de aumento de datos, el resultado del filtro en las imágenes radiográficas se presenta en la Figura 16. La aplicación del filtro se centró en establecer un rango en la que se aplicaría más o menos brillo a las imágenes seleccionadas. Como se presenta en la Figura 17, se realizó con el uso de la función *ColorJitter* de la librería *transformers* donde utiliza el parámetro de la variable *brightness_adjuster*, que es un rango de 0.4 a 1.6 para la disminución del brillo original hasta un 60% y el aumento de brillo igualmente hasta un 60% más que el original.

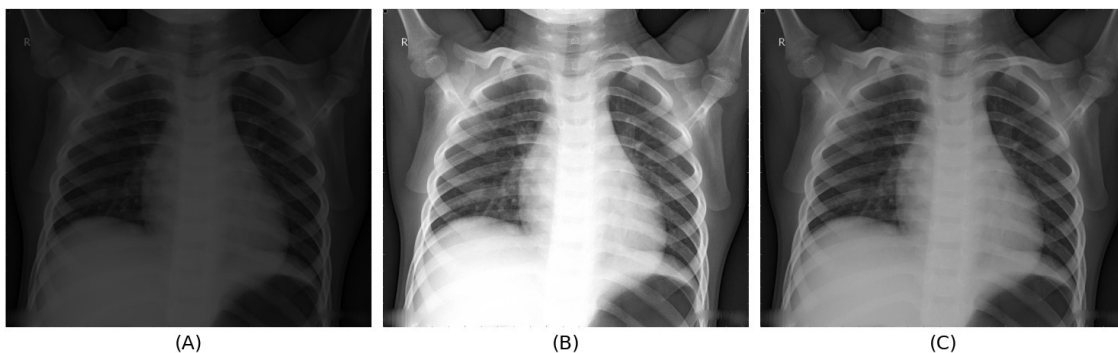


Figura 16. Aplicación de filtro para la intensidad de brillo en imágenes radiográficas. El filtro aplica una intensidad de brillo aleatoria. En la imagen **A** se muestra la aplicación con una intensidad de brillo baja, en la imagen **B** muestra una intensidad más alta que la normal, mientras que la imagen **C** se muestra la intensidad de brillo normal en las radiografías del conjunto

```

# Rango de brillo aleatorio (0.4 disminuye el brillo 60%, 1.6 aumenta el brillo en un 60%)
brightness_adjuster = transforms.ColorJitter(brightness=(0.4, 1.6))

for image_file in image_files:
    # Cargar la imagen
    image_path = os.path.join(conjunto_da_dir, image_file)
    image = cv2.imread(image_path)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    # Convertir la imagen a PIL Image
    pil_image = transforms.functional.to_pil_image(image)

    # Aplicar el ajuste de brillo aleatorio
    adjusted_image = brightness_adjuster(pil_image)

    # Crear la carpeta de destino si no existe
    if not os.path.exists(brightness_dir):
        os.makedirs(brightness_dir)

    # Guardar la imagen ajustada
    output_path = os.path.join(brightness_dir, 'brightness_' + image_file)
    adjusted_image.save(output_path)

print("Proceso completado.")

```

Figura 17. Código para la aplicación de intensidad de brillo aleatoria

La aplicación de los 4 filtros seleccionados dentro del subconjunto de imágenes resultó en 1.348 imágenes que se pueden añadir al conjunto de entrenamiento representados en la Tabla 8. Si se suma esta cantidad de imágenes con la cantidad inicial en el conjunto de datos original se obtiene como resultado un total de 2.689 imágenes totales en la clase "NORMAL" como se detalla en la Tabla 9. Sin embargo, la cantidad resultante de imágenes es inferior a la cantidad de imágenes de la clase "PNEUMONIA". Exactamente en el conjunto, la clase "NORMAL" cuenta con 1.186 imágenes menos en relación a su segunda clase como se presenta en la Figura 18, así reduciendo el desbalance inicial, pero, no solucionándolo por completo, por lo que fue necesario utilizar la segunda técnica de balanceo, el aumento de datos desde el segundo conjunto seleccionado.

Tabla 8. Cantidad de imágenes totales después de la aplicación de filtros en el subconjunto de datos de entrenamiento para la clase "NORMAL"

Filtro	Nro. Imágenes
Rotación aleatoria	337
Giro Horizontal	337
Intensidad de brillo aleatorio	337
Desenfoque Gaussiano aleatorio	337
Total	1.348

Tabla 9. Cantidad de imágenes totales para el conjunto de entrenamiento. Representa la suma de la cantidad de imágenes en el conjunto de entrenamiento del conjunto original y el resultado de las técnicas de aumento de datos para la clase "NORMAL" en el subconjunto de imágenes seleccionadas para el aumento de datos.

Conjunto	Nro. imágenes
Conjunto original	1.341
Subconjunto seleccionado aumentado	1.348
Total	2.689

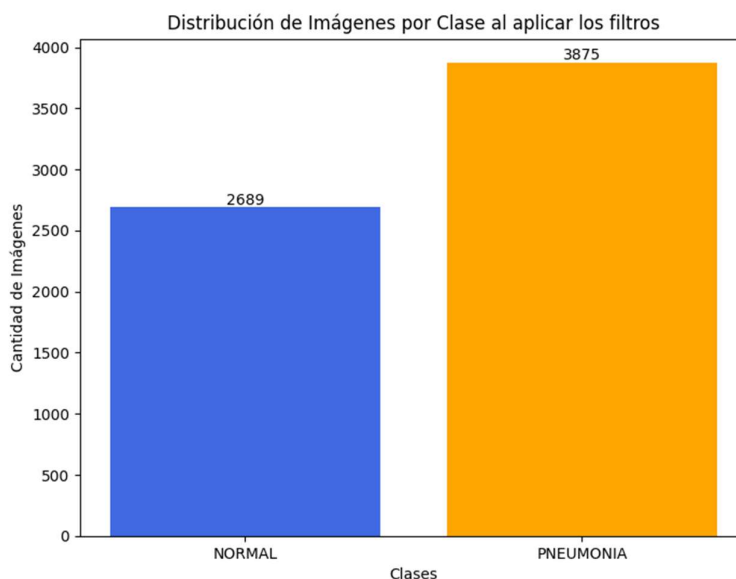


Figura 18. Distribución de clases del conjunto de entrenamiento después de realizar la técnica de aplicación de filtros

6.1.3.1.2. Tarea 3.2. Aumento de datos al conjunto de entrenamiento: Adición de imágenes desde el segundo conjunto de datos "COVID-QU-Ex dataset"

Para la adición de imágenes se centró en la clase que seguía manteniendo el desbalance de los datos, que es la clase "NORMAL". Para lograr el balance óptimo se debe conseguir un número de datos igual o cercano al de ambas clases, por lo tanto, se determinó aumentar 932 imágenes de la clase "NORMAL" obtenidas del conjunto de datos seleccionado. El número de imágenes seleccionadas del nuevo conjunto de datos, hace referencia al total de imágenes radiográficas del subconjunto de entrenamiento de clase "NORMAL" en el conjunto de datos del que se extrae las imágenes, como se detalla en la Tabla 6. Para lograr una correcta adición de las imágenes seleccionadas fue necesario, establecer inicialmente una transformación de formato a las imágenes, específicamente se necesitó transformar de *PNG* que es formato de las imágenes del segundo conjunto de datos, al formato *JPEG* el cual es del conjunto de datos inicial.

Como resultado de la tarea, se obtuvo un total de 3.621 imágenes en la clase “NORMAL” representadas en la Tabla 10, a partir de la suma de las 932 imágenes del conjunto de datos “COVID-QU-Ex” y las 2.689 imágenes obtenidas en el primer aumento de datos como se presenta en la Tabla 10.

Tabla 10. Total de imágenes radiográficas del conjunto de entrenamiento en la clase “NORMAL” luego de técnicas de aumento de datos

Fuente	Nro. imágenes
Total imágenes primer DA	2.689
Total imágenes segundo conjunto de datos	932
Total imágenes	3.621

Como resultado final se obtuvo un total de 7.496 imágenes radiográficas en todo el conjunto de entrenamiento luego del balanceo de la clase “NORMAL” (Tabla 11). Están divididas en 3.875 radiografías de la clase “PNEUMONIA” y 3.621 de la clase “NORMAL” como se representa en la Figura 19. De esta manera minimizando el desbalance dentro del conjunto de datos a 254 imágenes de diferencia, que resulta una cantidad de datos poco relevante de desbalanceo en la fase de entrenamiento del modelo.

Tabla 11. Número total de imágenes radiográficas destinadas al conjunto de entrenamiento

Clase	Nro. imágenes
NORMAL	3.621
PNEUMONIA	3.875
Total	7.496

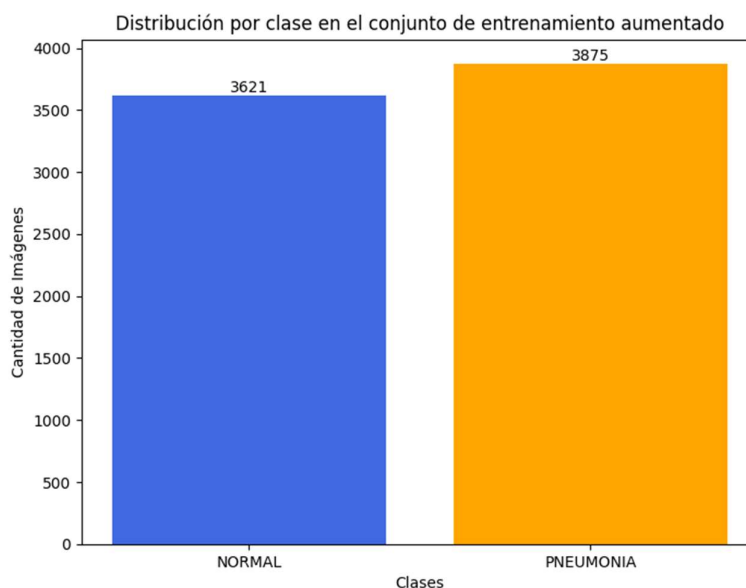


Figura 19. Distribución de clases balanceadas del conjunto de datos para el entrenamiento del modelo

6.1.3.2. Balanceo del conjunto de prueba

6.1.3.2.1. Tarea 3.3. Aumento de datos al conjunto de prueba: Adición de imágenes desde el segundo conjunto de datos “COVID-QU-Ex dataset”

Para completar el proceso de balanceo del conjunto de datos seleccionado “Chest X-Ray images Pneumonia” fue necesario aplicar técnicas de aumento en el conjunto desbalanceado restante, el cual es el conjunto destinado a pruebas. Para el balanceo de las clases, se utilizó la segunda técnica de aumento de datos que se aplicó al conjunto de entrenamiento. La adición de imágenes fue desde el segundo conjunto de datos “COVID-QU-Ex dataset”. Para lograr el balance óptimo dentro del conjunto a utilizar, inicialmente se realizó el proceso de transformación de formato de las imágenes del conjunto “Normal”, la transformación consistió en cambiar el formato PNG de las imágenes del conjunto del que se van a extraer, al formato del conjunto inicial JPEG. Una vez realizado este proceso se pudo ejecutar el código de la *Figura 20* para la selección aleatoria de las 156 del conjunto de prueba de la clase “Normal” (*Tabla 6*) mediante el uso de la función *sample()* de la librería *random* de python.

```
num_images_to_move = 156

# Obtener la lista de imágenes disponibles
all_images = glob.glob(os.path.join(in_dir, '*.jpeg'))

# Seleccionar 156 imágenes al azar
selected_images = random.sample(all_images, num_images_to_move)
```

Figura 20. Código para la selección aleatoria del conjunto de prueba seleccionado

Como resultado de la aplicación de la técnica de aumento se tuvo un total de 390 imágenes en la clase “NORMAL”. Como se presenta en la *Tabla 12*, a partir de la suma de las 156 imágenes añadidas y las 234 iniciales dentro del conjunto. Por lo tanto, como se puede visualizar en la *Figura 21* las clases se encuentran perfectamente balanceadas con una cantidad de datos igual por cada clase, obteniendo así un total de 780 imágenes en el conjunto de prueba como se detalla en la *Tabla 13*.

Tabla 12. Cantidad de imágenes totales después del aumento de datos en el conjunto de prueba de la clase “NORMAL”

Conjunto	Cantidad imágenes
Imágenes originales	234
Imágenes seleccionadas	156
Total	390

Tabla 13. Cantidad de imágenes totales en el conjunto de prueba por clase

Clase	Cantidad de imágenes
NORMAL (aumentada)	390
PNEUMONIA	390
Total	780

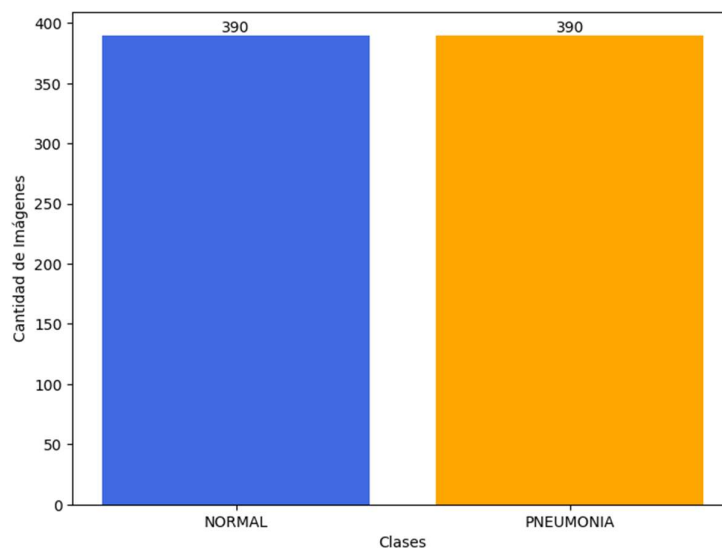


Figura 21. Distribución balanceada de las clases para el conjunto de prueba

Como resultado de la tarea de balanceo de clases, se obtuvo un conjunto aumentado de 8.292 imágenes totales contando los subconjuntos de entrenamiento, prueba y validación como se detalla en la Tabla 14, y que posteriormente fue publicado en el repositorio digital Kaggle bajo el nombre “Redistributed and balanced Chest x-ray pneumonia”³.

Dados los resultados se pudo conocer que, las técnicas de aumento de datos permitieron la generación de 2.436 imágenes entre los conjuntos de entrenamiento y prueba de la clase “NORMAL” para el correcto balanceo de las clases en los conjuntos. Las imágenes se añadieron a las 5.856 imágenes del conjunto inicial como se visualiza en la Figura 22.

Tabla 14. Total de imágenes en el conjunto de datos que se va a utilizar sumando la cantidad del conjunto inicial y el total de imágenes generadas mediante técnicas de aumento de datos

Conjunto	Total de imágenes
Conjunto inicial	5.856
Aumento de datos	2.436
Total conjunto amentado	8.292

³<https://www.kaggle.com/datasets/edrdopcch41/redistributed-and-balanced-chest-x-ray-pneumonia>

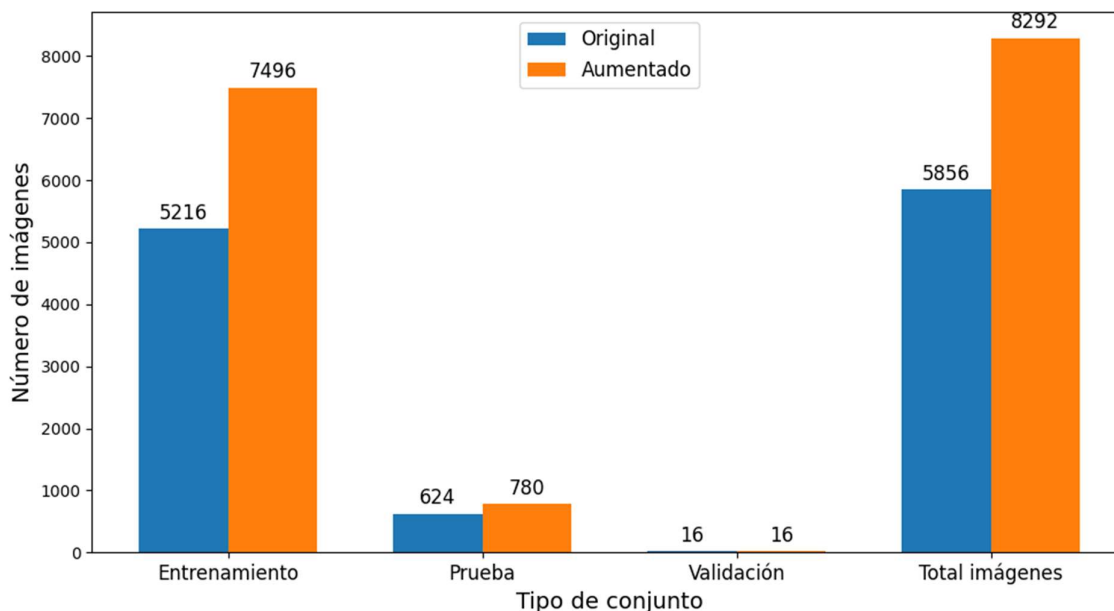


Figura 22. Comparación entre el total de imágenes de los conjuntos de entrenamiento, prueba y el total de imágenes obtenidas en el conjunto original y el conjunto aumentado.

Los siguientes resultados corresponden a la fase: **Ingeniería de modelos**

6.1.4. Tarea 4: Ajuste del modelo “vit-base-patch16-224-in21k”

6.1.4.1. Configuración del modelo

Para la realización de la tarea que comprende ajuste del modelo, se utilizó la guía proporcionada por el sitio oficial Hugging Face⁴ para la utilización de los modelos de Visión Transformes (ViT). El ajuste del modelo se realizó utilizando los recursos del clúster de GPU’s de la Universidad Nacional de Loja y el entorno de Jupyter Lab. Para iniciar con el ajuste del modelo fue necesario establecer una configuración adecuada en base a las entradas establecidas, que son las imágenes del conjunto de datos aumentado y las salidas del modelo que es la etiqueta clasificada de “PNEUMONIA” o “NORMAL” como se detalla en la Figura 23, donde se presenta la estructura de la arquitectura utilizada por el modelo “vit-base-patch16-224-in21k”, y el funcionamiento del proceso de las entradas para la generalización de características en imágenes radiográficas de tórax para la clasificación de neumonía (salidas).

⁴ <https://huggingface.co>

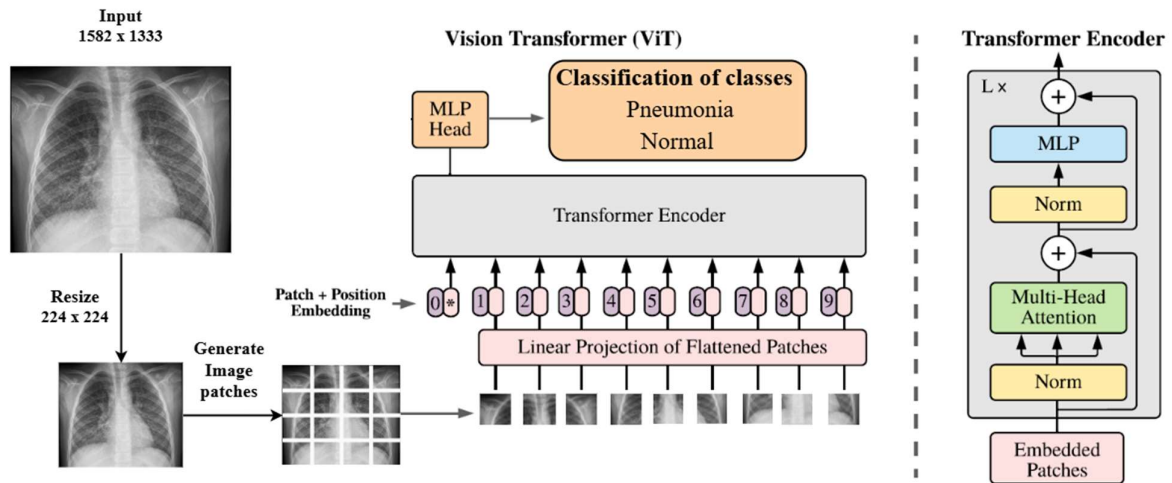


Figura 23. Arquitectura del modelo “vit-base-patch16-224-in21k” para la clasificación de neumonía en imágenes radiográficas. (Arquitectura adaptada del documento “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale” [3])

Para la configuración de las entradas del modelo, fue necesario definir 2 transformaciones fundamentales antes de la creación de los dataloaders de los conjuntos de entrenamiento, prueba y validación. Como se puede visualizar en la Figura 24 se utilizó la clase `torchvision.transforms` y la función `compose()` la cual permitió establecer las 2 transformaciones concatenadas que se aplicarían a los dataloaders. Las transformaciones son: El redimensionado de las imágenes a la resolución de 224 píxeles de alto y 224 píxeles de ancho, se utilizó la función `transforms.Resize()` para la redimensión de las imágenes como se visualiza en la Figura 25. La segunda transformación establecida es la conversión de las imágenes redimensionadas a tensores con la función `transforms.ToTensor()` que es el formato adecuado para trabajar con modelos *Pytorch*.

```

train_transforms = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
])

val_test_transforms = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
])

```

Figura 24. Código de las transformaciones definidas para los datos de entrada al modelo

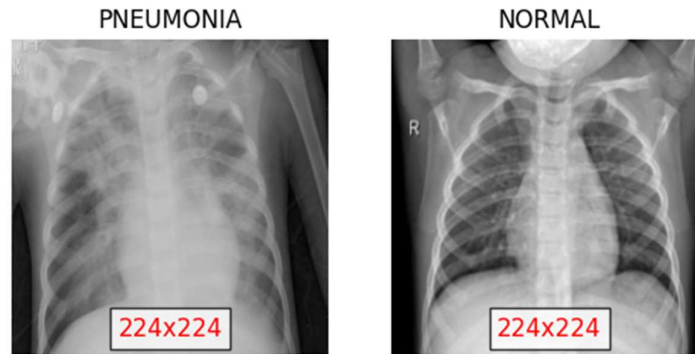


Figura 25. Aplicación de la redimensión a 224 x 224 píxeles para las imágenes de los conjuntos de datos

El segundo proceso necesario dentro de la configuración del modelo fue la creación del conjunto de datos y de los dataloaders. Inicialmente se creó el conjunto de datos con la utilización de la función `ImageFolder()` de la clase `torchvision.datasets` la cual permitió cargar las imágenes desde los directorios en las que se encontraban clasificadas las imágenes, como se representa en la Figura 27. Como se presenta en la Figura 24, la función utilizada recibió de entrada la dirección de los directorios y las transformaciones definidas anteriormente para la aplicación en cada imagen de los conjuntos de entrenamiento, prueba y validación como representa la Figura 26.

```
full_dataset = ImageFolder(data_dir, transform=train_transforms)
```

Figura 26. Código creación de los conjuntos de datos Entrenamiento, Prueba y Validación

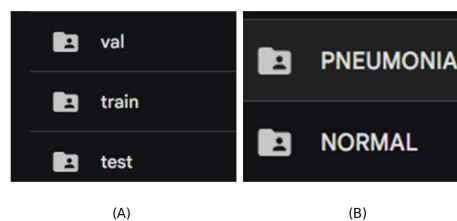


Figura 27. Estructura de los directorios de los conjuntos de datos. La imagen A muestra los directorios separados en conjuntos de validación, entrenamiento y prueba. La imagen B muestra la estructura interna de cada directorio, conteniendo carpetas separadas por tipo de clase.

Una vez creado el conjunto de datos, se estableció crear una distribución propia para los conjuntos de entrenamiento, prueba y validación. Inicialmente el conjunto ya poseía una distribución inicial de los datos, pero para la nueva distribución de los subconjuntos se definió establecer que el 70% de los datos sean destinados al entrenamiento del modelo, el 20% de los datos destinados a la prueba del modelo y el 10% de los datos destinados a la validación

como se visualiza en la Figura 28, los datos del conjunto balanceado es distribuido de manera aleatoria según los porcentajes establecidos con la *funcion random_split()* que recibe como parámetros el directorio del conjunto y los tamaños destinados para los conjuntos. Seguido a la creación de los conjuntos se le aplica a cada uno las transformaciones definidas en la Figura 24.

```

train_size = int(0.7 * len(full_dataset))
test_size = int(0.2 * len(full_dataset))
val_size = len(full_dataset) - train_size - test_size

train_dataset, test_dataset, val_dataset = random_split(
    full_dataset, [train_size, test_size, val_size]
)

train_dataset.dataset.transform = train_transforms
test_dataset.dataset.transform = val_test_transforms
val_dataset.dataset.transform = val_test_transforms

```

Figura 28. Código para establecer la nueva distribución de los datos en el conjunto de imágenes

Como se visualiza en la Figura 29 la distribución de las imágenes en los subconjuntos resulta en un conjunto de datos más balanceado y correctamente distribuido entre los subconjuntos respecto a la división de los datos en el conjunto inicial. Como se ve en la Tabla 15 se tuvo una nueva distribución de imágenes por conjunto de datos con un total de 8.292 imágenes.

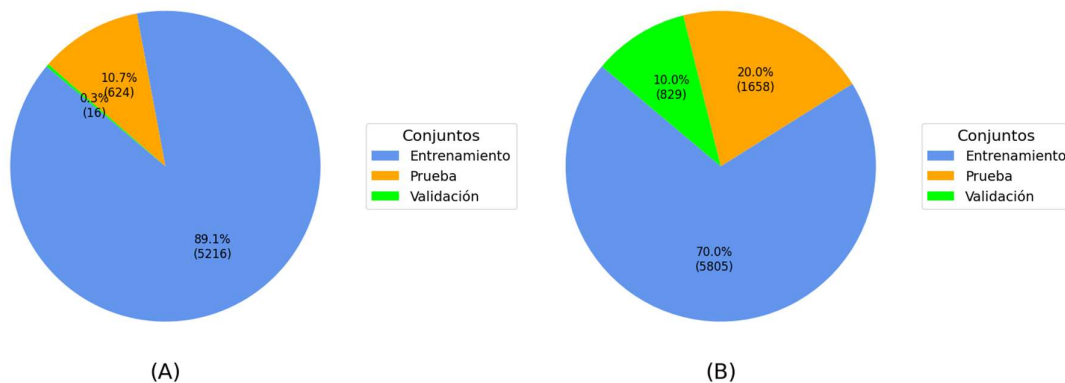


Figura 29. Comparación de distribución de datos entre el conjunto inicial y el conjunto balanceado. La imagen A representa la distribución del conjunto de datos inicial, la imagen B muestra la distribución de los datos tras la aplicación de la nueva distribución de los datos.

Tabla 15. Cantidad de imágenes por conjunto de datos en la nueva distribución

Clases	Entrenamiento	Prueba	Validación
NORMAL	2785	805	394
PNEUMONIA	3020	853	435
Total	5805	1658	829

Una vez creados los subconjuntos de datos se procedió a la creación de los dataloaders con los subconjuntos de datos transformados. Como presenta la Figura 30, se utilizó la función *DataLoader* de la clase *torch.utils.data*. Dentro de la función recibe el conjunto de datos para el dataloader respectivo. La función requiere asignar un valor para el tamaño del lote de las imágenes, en este caso el parámetro *batch_size* se definió con un valor de 64 para cada uno de los dataloaders creados. El último parámetro que recibe la función es *shuffle*, que indica si los datos del dataloader deben ser mezclados de manera aleatoria en cada época de las fases de entrenamiento, prueba o validación.

```
train_loader = DataLoader(train_dataset, batch_size=64, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=64, shuffle=True)
test_loader = DataLoader(test_dataset, batch_size=64, shuffle=True)
```

Figura 30. Código para la creación y configuración de los dataloaders de entrenamiento, prueba y validación

Finalmente, para la utilización del modelo “vit-base-patch16-224-in21k” en la clasificación de las dos clases “NORMAL” y “PNEUMONIA” fue necesario definir determinadas características del modelo pre entrenado. Inicialmente se cargó la configuración por defecto del modelo utilizando la función *ViTConfig.from_pretrained*. Luego se delimitó la configuración del modelo para que clasifique entre dos tipos de etiquetas o clases con la función *.num_labels*, como se puede observar en la Figura 31.

```
config = ViTConfig.from_pretrained("google/vit-base-patch16-224-in21k")
config.num_labels = 2 # Normal and Pneumonia
```

Figura 31. Carga y configuración del modelo para la clasificación de 2 clases.

Una vez definida la configuración necesaria para el modelo, se utiliza la función *ViTForImageClassification.from_pretrained* para cargar el modelo “vit-base-patch16-224-in21k”. Como se presenta en la Figura 31, la función requiere de otro parámetro que sirve para la determinar la configuración del modelo, en este caso se carga la configuración realizada con anterioridad. La ejecución de la función especificada, crea una instancia del modelo con la configuración requerida, como se presenta en la Figura 32. Finalmente, se utiliza la función *.DataParallel()* para permitir el entrenamiento y prueba del modelo en múltiples GPU's.

```
model = ViTForImageClassification.from_pretrained(
    "google/vit-base-patch16-224-in21k",
    config=config
)
model = nn.DataParallel(model)
```

Figura 32. Instancia del modelo “vit-base-patch16-224-in21k” configurado

6.1.5. Tarea 5: Entrenamiento del modelo “vit-base-patch16-224-in21k”

Para poder realizar la tarea correspondiente al entrenamiento del modelo, primeramente, se estableció la configuración de los hiper parámetros que utilizaría la función de entrenamiento. Como se visualiza en la Figura 33 se definió 2 variables. La variable *criterion* se estableció con la función de pérdida *CrossEntropyLoss()* seleccionada para la clasificación multiclase en el modelo. De igual forma se estableció la variable *optimizador* donde el valor se establece en con dos criterios: el optimizador para los pesos del modelo se definió *Adam* y la tasa de aprendizaje o learning rate se estableció con un valor de $1e^{-6}$.

```
 criterion = nn.CrossEntropyLoss()
 optimizer = optim.Adam(model.parameters(), lr=1e-6)
```

Figura 33. Código de la asignación de los hiperparámetros, Optimizador y función de pérdida

Para garantizar la ejecución del entrenamiento dentro de las GPU's se estableció el código de la Figura 34, donde con el uso de la variable *device* se pudo gestionar la ubicación el modelo y asignar el modelo a las GPU's disponibles mediante *model.to(device)*.

```
 device = torch.device(
     "cuda" if torch.cuda.is_available() else "cpu"
 )
 model.to(device)
```

Figura 34. Código configuración de dispositivo para el entrenamiento del modelo.

Finalmente, para entrenar el modelo se ejecutó la función *train_model()* desde el entorno de Jupyter Lab para la utilización del cluster de GPU's en el entrenamiento del modelo. Como se visualiza en la Figura 35, la llamada a la función recibe 6 parámetros que son: el modelo que se configuró previamente, las variables de los hiperparámetros de la tasa de pérdida y la función de optimización, los Dataloaders de entrenamiento y validación, y el número de épocas que durará el entrenamiento, en este caso se definió un total de 20 épocas.

```
 trained_model, train_losses, val_accuracies, best_epoch = train_model(
     model,
     criterion,
     optimizer,
     train_loader,
     val_loader,
     num_epochs=20
 )
```

Figura 35. Código de llamado a la función de entrenamiento del modelo

Una vez establecidos los hiperparámetros detallados en la Tabla 16 logró completar el proceso de entrenamiento del modelo “vit-base-patch16-224-in21k” obteniendo en este procedimiento los siguientes resultados: El modelo fue entrenado durante 20 épocas, la Tabla 17 presenta la información obtenida sobre la ejecución de las 5 primeras épocas y las últimas 2. Cada época presenta información sobre la validación de precisión y la pérdida obtenida. Como se puede apreciar en la Tabla 17 el modelo en la fase de entrenamiento llegó a un 97.71% de precisión y un total de 0.0283 en la pérdida durante la ejecución de la última época. Lo que permite inferir que el modelo cuenta con una muy buena capacidad para la clasificación de imágenes radiográficas. Finalmente el mejor modelo entrenado fue empaquetado en formato “.pth” con la función mostrada en la Figura 36 para posteriormente ser publicado junto al código expuesto dentro del repositorio “Fitting Vitbase-patch16-224-in21k for pneumonia”⁵ en la plataforma kaggle, para garantizar el acceso en las siguientes fases de desarrollo.

```
best_model_filename = (
    f"modelo_train_accuracy_{best_val_accuracy:.2f}_ "
    f"train_loss_{avg_train_loss:.4f}_epochs_{num_epochs}.pth"
)

torch.save(model.state_dict(), best_model_filename)
print(f"Best model saved as {best_model_filename}")
```

Figura 36. Función de empaquetado de modelo ajustado entrenado

Tabla 16. Valores de hiperparámetros utilizados en la función de entrenamiento del modelo

Hiperparámetro	Valor
Función de pérdida	CrossEntropyLoss
Optimizador	Adam
Tasa de aprendizaje	1×10^{-6}
Épocas	20
Batch size	64

⁵ <https://www.kaggle.com/models/edrdopcch41/vitbase-patch16-224-in21k-fine-tuning>

Tabla 17. Resultados de métricas pérdidas y precisión por épocas en la ejecución del entrenamiento del modelo.

Época	Pérdida	Precisión
1/20	0.6490	88.41%
2/20	0.4993	91.69%
3/20	0.3505	93.49%
4/20	0.2602	95.18%
5/20	0.2065	96.27%
...
19/20	0.0345	97.71%
20/20	0.0283	97.71%

6.2. Objetivo 2: Evaluar el ajuste del modelo, por medio de la métrica de sensibilidad, para la clasificación neumonía en imágenes radiográficas de tórax, utilizando el conjunto de datos para validación “Chest X-Ray Images (Pneumonia)”.

6.2.1. Tarea 6: Evaluación del modelo ajustado con el conjunto de entrenamiento

Gracias al conjunto afinado por la aplicación de las técnicas de balanceo de clases, la tarea de entrenamiento resultó en el modelo completamente ajustado para la detección de neumonía en imágenes radiográficas de tórax. Además, el proceso de entrenamiento permitió obtener la matriz de confusión, la cual permitió analizar al modelo desde la métrica de sensibilidad, como se expone en la Figura 37 donde se detalla que el mejor modelo cuenta con una precisión de 97.71% significando que clasificó correctamente 2.772 de 2.785 imágenes de la clase “NORMAL”, siendo estos la cantidad de verdaderos positivos, contando solamente con 13 imágenes clasificadas incorrectamente que representan la cantidad de falsos negativos en la clase “NORMAL”. Para el análisis de sensibilidad de la clase “PNEUMONIA”, se analizó que el modelo logró clasificar correctamente 3.018 de 3.020 imágenes como verdaderos positivos y una cantidad de 2 imágenes clasificadas incorrectamente como falsos negativos de la clase “PNEUMONIA”.

Con la aplicación de la fórmula de la sensibilidad se logró obtener los siguientes resultados: Como se puede ver en la Ecuación 3 la clase “NORMAL” obtuvo un 99.53% de sensibilidad en la clasificación de imágenes pertenecientes a esta clase. En la Ecuación 4 se detalla el cálculo de la sensibilidad para la clase “PNEUMONIA” obteniendo un total de 99.03% de sensibilidad para la clasificación de imágenes pertenecientes a esta clase. En la Tabla 18 se fueron analizados los valores según la métrica de sensibilidad, donde se puede establecer que el modelo posee una gran capacidad para la clasificación correcta de ambas clases obteniendo una cantidad muy pequeña de errores o falsos negativos. Además, con el análisis

de especificidad en la Ecuación 5 y Ecuación 6 se puede confirmar que el modelo logra distinguir correctamente las imágenes que no pertenecen a una clase, es decir, los verdaderos negativos obteniendo un total de 99.93% en la clase “NORMAL” y 99.53% en la clase “PNEUMONIA”.

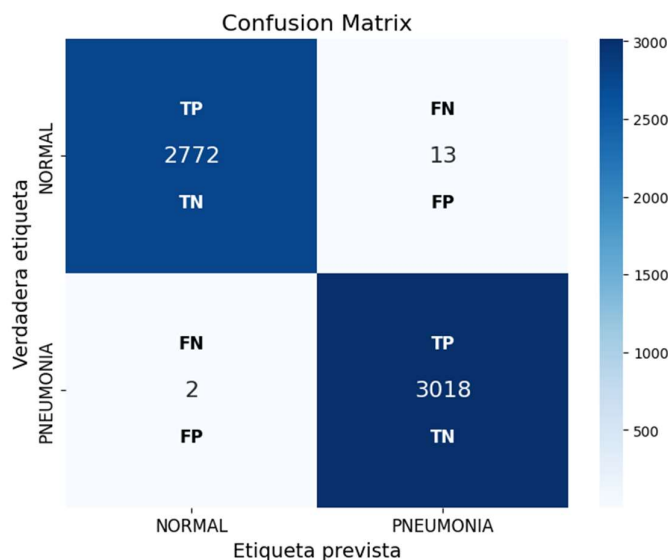


Figura 37. Matriz de confusión del entrenamiento del modelo. Donde se presenta la cantidad de verdaderos positivos (TP), falsos negativos (FN), verdaderos negativos (TN) y falsos positivos (FP) por cada clase en el entrenamiento del modelo para el cálculo de sensibilidad y especificidad.

Ecuación 3. Cálculo de sensibilidad de la clase NORMAL en la matriz de confusión de entrenamiento.

$$\frac{TP}{TP + FN} = \frac{2772}{2772 + 13} = \frac{2772}{2785} = 0.9953 = 99.53\%$$

Ecuación 4. Cálculo de sensibilidad de la clase PNEUMONIA en la matriz de confusión de entrenamiento.

$$\frac{TP}{TP + FN} = \frac{3018}{3018 + 2} = \frac{3018}{3020} = 0.9993 = 99.93\%$$

Ecuación 5. Cálculo de especificidad de la clase NORMAL en la matriz de confusión de entrenamiento.

$$\frac{TN}{TN + FP} = \frac{3018}{3018 + 2} = \frac{3018}{3020} = 0.9993 = 99.93\%$$

Ecuación 6. Cálculo de especificidad de la clase PNEUMONIA en la matriz de confusión de entrenamiento.

$$\frac{TN}{TN + FP} = \frac{2772}{2772 + 13} = \frac{2772}{2785} = 0.9953 = 99.53\%$$

Tabla 18. Valores de sensibilidad y especificidad obtenidos de la matriz de confusión en el entrenamiento del modelo.

Clase	Sensibilidad	Especificidad
NORMAL	99.53%	99.93%
PNEUMONIA	99.93%	99.53%

6.2.2. Tarea 7: Evaluación del modelo ajustado con el conjunto de prueba

Como se visualiza en la Figura 38 para la evaluación del modelo ajustado se tuvo que utilizar una función para cargar el modelo empaquetado en formato “.pth” obtenido de la fase de entrenamiento. Se utilizó la función `torch.load()` que permitió importar un modelo desde un archivo con la extensión de Pytorch. La utilización de la función permitió guardar el modelo junto a sus pesos y sesgos en la variable `model_state_dict`.

```
model_state_dict = torch.load(  
    "modelo_train_accuracy_97.71_train_loss_0.0283_epochs_20.pth"  
)
```

Figura 38. Código para cargar el modelo ajustado en la fase de entrenamiento

Similar a la fase de entrenamiento, la evaluación del modelo se realizó con el uso del clúster de GPU's. En donde, para poder utilizar la configuración del modelo ajustado, se tuvo que asignar los pesos en una instancia del modelo seleccionado utilizando la función `load_state_dict()` que permite la asignación segura de la configuración de los pesos y sesgos del modelo entrenado, como se representa en la Figura 39.

```
if torch.cuda.device_count() > 1 :  
    config = ViTConfig.from_pretrained("google/vit-base-patch16-224-in21k")  
    config.num_labels = 2  
  
    model_ent = ViTForImageClassification(config)  
    model_ent = nn.DataParallel(model_ent)  
    # Establecer los pesos cargados al modelo  
    model_ent.load_state_dict(model_state_dict)  
  
model_ent = model_ent.cuda()
```

Figura 39. Código para la asignación de la configuración del modelo ajustado.

Finalmente, para la evaluación del modelo se ejecutó la función de evaluación `evaluate_model()` desde el entorno de Jupyter Lab. Como se visualiza en la Figura 40, la función recibe 3 argumentos importantes, que son: La instancia del modelo con la configuración de pesos del modelo ajustando, el dispositivo para la utilización de las GPU's y el dataloader con las imágenes del conjunto de prueba. Como se ve en la Tabla 19, el

dataloader está conformado por 1.658 imágenes radiográficas distribuidas en 805 imágenes por la clase “NORMAL” y 853 por la clase “PNEUMONIA”.

```

evaluate_model(
    model_ent,
    test_loader,
    device
)

```

Figura 40. Parámetros de la función de evaluación.

Tabla 19. Distribución de imágenes por clase en el dataloader de prueba.

Clase	Cantidad de imágenes
NORMAL	805
PNEUMONIA	853
Total	1.685

Una vez ejecutada la función *evaluate_model()* se logró completar el proceso de evaluación del modelo ajustado, obteniendo así los siguientes resultados: El modelo evaluado presenta como resultado un porcentaje del 99.34% de precisión en la clasificación de neumonía mediante imágenes radiográficas de tórax. Permitiendo describir al modelo con una muy buena capacidad para la clasificación de las imágenes radiográficas. Además, como se puede ver en la Figura 41 se obtuvo la matriz de confusión de la prueba del modelo ajustado, la cual posteriormente fue publicada junto al código expuesto dentro del repositorio “Fitting Vitbase-patch16-224-in21k for pneumonia”⁶ en la plataforma Kaggle.

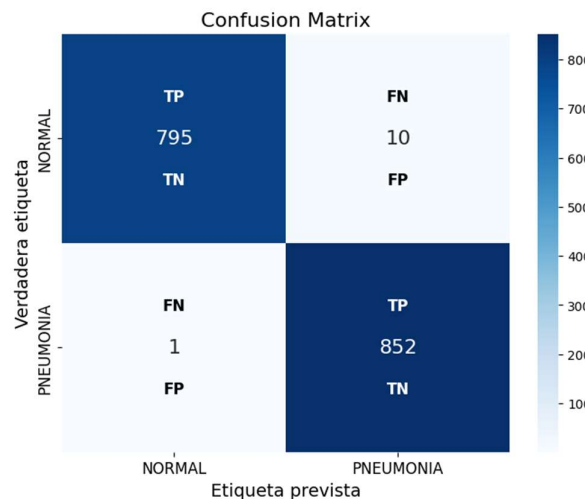


Figura 41. Matriz de confusión de la evaluación del modelo con el conjunto de prueba. Donde se presenta la cantidad de verdaderos positivos (TP), falsos negativos (FN), verdaderos negativos (TN) y falsos positivos (FP) por cada clase.

⁶ <https://www.kaggle.com/models/edrdopcch41/vitbase-patch16-224-in21k-fine-tuning>

6.2.2.1. Análisis de clasificación de verdaderos positivos TP y falsos negativos FN

Una vez ejecutada la prueba el modelo ajustado y obtenida la matriz de confusión de la evaluación del modelo, como se visualiza en la Figura 41. Se procedió a evaluar los resultados de la clasificación, analizando los verdaderos positivos y falsos negativos. Inicialmente se pudo analizar que el modelo logró clasificar correctamente 795 de las 805 imágenes de la clase "NORMAL" en el conjunto de prueba. Las 795 imágenes representan la cantidad de verdaderos positivos TP, mientras que, 10 imágenes fueron clasificadas incorrectamente como "PNEUMONIA", representando a los falsos negativos FN de la clase "NORMAL". En la revisión de las imágenes clasificadas incorrectamente, es decir, los falsos negativos. Se logró observar que, en 7 de las 10 imágenes se presentan alteraciones en comparación a una radiografía normal. Las alteraciones se presentan en las imágenes 1, 5, 6, 7, 8, 9 y 10 representadas en la Figura 42. Las imágenes mencionadas presentan la característica común que, muestran alteraciones en la intensidad del brillo, es decir, 7 de las 10 imágenes clasificadas incorrectamente forman parte del conjunto aumentado resultado de la aplicación del filtro de intensidad de brillo aleatorio (Sección 6.1.3.1.1). Específicamente, las imágenes 1, 6, 7, 8, 9 y 10 presentan una intensidad de brillo alto a comparación de la imagen 5 que presenta una intensidad de brillo bajo.

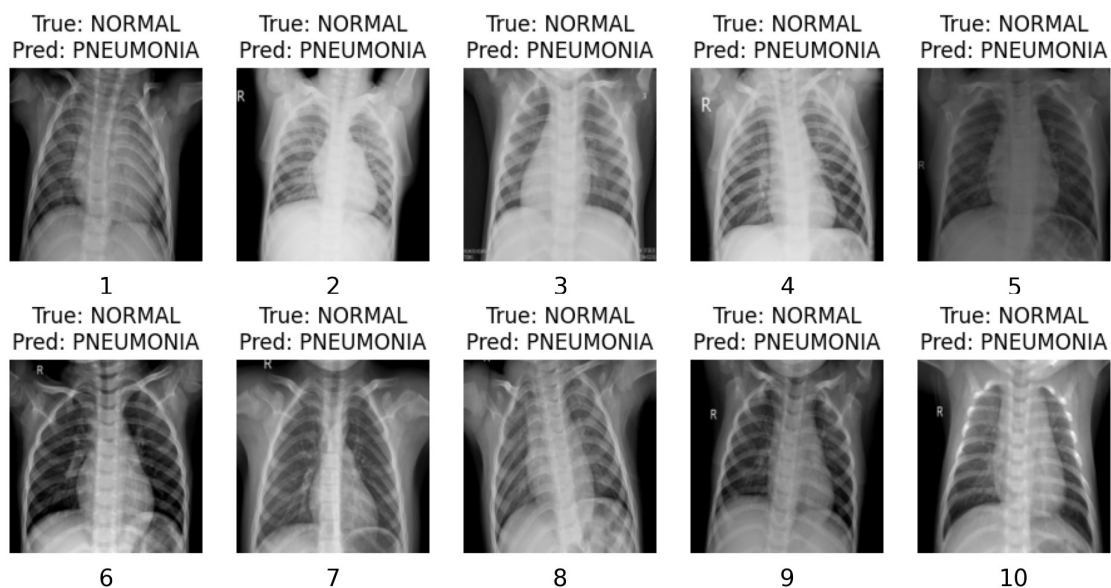


Figura 42. Imágenes clasificadas incorrectamente (Falsos negativos) de la clase NORMAL, obtenidas de la matriz de confusión en la prueba del modelo ajustado.

De igual forma, se analizó que el modelo logró clasificar correctamente 852 de las 853 imágenes de la clase "PNEUMONIA". Las 852 imágenes representan a los verdaderos positivos, mientras que 1 imagen fue clasificada incorrectamente, la cual representa la

cantidad de falsos negativos en la clasificación de imágenes de la clase “PNEUMONIA”. En la revisión de la imagen clasificada incorrectamente, presentada en la Figura 43, se logró observar que, presenta una alteración en la intensidad de brillo de la imagen, específicamente una intensidad de brillo más alta que en una radiografía normal. Es decir, la imagen clasificada incorrectamente en la clase “PNEUMONIA” de igual forma pertenece al conjunto de imágenes resultantes de la aplicación del filtro de intensidad de brillo aleatorio (Sección 6.1.3.1.1).

True: PNEUMONIA
Pred: NORMAL



Figura 43. Imagen clasificada incorrectamente (Falso negativo) de la clase PNEUMONIA, obtenida de la matriz de confusión en la prueba del modelo ajustado.

6.2.2.2. Análisis de sensibilidad en la prueba del modelo ajustado

Una vez realizó el análisis de los verdaderos positivos y falsos negativos en los resultados de clasificación del modelo ajustado, se pudo utilizar la fórmula de la sensibilidad y de especificidad para obtener los resultados de la evaluación en base a la métrica: Como se puede ver en la Ecuación 7 la clase “NORMAL” obtuvo un 98.76% de sensibilidad en la clasificación de imágenes pertenecientes a esta clase. Como se puede ver en la Ecuación 8 se detalla que el modelo obtuvo un total de 99.88% de sensibilidad en la clasificación de imágenes radiográficas de la clase “PNEUMONIA”. Luego del análisis de la sensibilidad por clase se obtuvo la Tabla 20, donde a partir de los resultados se puede concluir que, el modelo posee una muy buena capacidad para la clasificación correcta de ambas clases obteniendo una cantidad muy pequeña de falsos negativos en la clasificación de neumonía. Finalmente se pudo analizar la especificidad por cada clase, dando como resultado un total de 99.88% en la clase “NORMAL” y un total de 98.76% de especificidad en la clase “PNEUMONIA” como se detalla en la Ecuación 9 y Ecuación 10. Las ecuaciones permitieron concluir que la capacidad del modelo en detectar correctamente las imágenes que no pertenecen a una clase es muy alta.

Ecuación 7. Cálculo de sensibilidad de la clase NORMAL en la matriz de confusión de prueba.

$$\frac{TP}{TP + FN} = \frac{795}{795 + 10} = \frac{795}{805} = 0.9876 = 98.76\%$$

Ecuación 8. Cálculo de sensibilidad de la clase PNEUMONIA en la matriz de confusión de prueba.

$$\frac{TP}{TP + FN} = \frac{852}{852 + 1} = \frac{852}{853} = 0.9988 = 99.88\%$$

Ecuación 9. Cálculo de especificidad de la clase NORMAL en la matriz de confusión de prueba.

$$\frac{TN}{TN + FP} = \frac{852}{852 + 1} = \frac{852}{853} = 0.9988 = 99.88\%$$

Ecuación 10. Cálculo de especificidad de la clase PNEUMONIA en la matriz de confusión de prueba.

$$\frac{TN}{TN + FP} = \frac{795}{795 + 10} = \frac{795}{805} = 0.9876 = 98.76\%$$

Tabla 20. Valores de sensibilidad y especificidad obtenidos de la matriz de confusión en la prueba del modelo.

Clase	Sensibilidad	Especificidad
NORMAL	98.76%	99.88%
PNEUMONIA	99.88%	98.76%

6.2.2.3. Análisis del modelo ajustado mediante la técnica de Zero-Shot Learning

Como parte de la evaluación del modelo se utilizó la técnica de Zero Shot Learning para el análisis de resultados con la utilización de imágenes no vistas por el modelo en ninguna de sus fases anteriores, ya sea entrenamiento, prueba o validación con el fin de conocer el rendimiento del modelo ajustado frente a datos sin previo procesamiento y principalmente en un entorno real. Para ello se realizó una recolección de imágenes radiográficas de tórax en instituciones médicas y radiológicas en la ciudad de Loja, contando con el apoyo de profesionales médicos para la identificación de la enfermedad de neumonía y la asignación de la clase en las radiografías seleccionadas. Obteniendo así los siguientes resultados: El conjunto de imágenes recolectado se encuentra distribuido de manera equitativa como se ve en la Figura 44 contando con 15 imágenes radiográficas de tórax por cada clase en formato jpeg. Además, como se visualiza en la Figura 45, el conjunto recolectado fue utilizado para la creación de zero_data_loader, el cual contenía la información de todas las imágenes y al que se establecieron las transformaciones necesarias para que la información pueda ser ingresada al modelo ajustado. De manera similar al proceso de evaluación de la sección 0, se utilizó la función de evaluación del modelo con la diferencia que como argumento principal recibió el data_loader creado a partir del conjunto recolectado. Finalmente, se ejecutó la función de evaluación del modelo obteniendo como resultados el porcentaje de precisión obtenido en la evaluación y la matriz de confusión, la cual fue

publicada en el repositorio “Fitting Vitbase-patch16-224-in21k for pneumonia”⁷ dentro de la plataforma kaggle junto al código de ejecución para la evaluación.

Dentro del análisis de resultados se puede apreciar que la precisión del modelo que se obtuvo es del 96.67%, permitiendo describir al modelo con una elevada capacidad para la predicción de las nuevas imágenes radiográficas, es decir, que no se hayan utilizado anteriormente en las fases de entrenamiento y prueba. Mediante la matriz de confusión se permitió analizar los resultados en base a la métrica de sensibilidad, como se visualiza en la Figura 46, se clasificaron correctamente las 15 imágenes correspondientes a la clase “NORMAL” representando los verdaderos positivos de la clase, en cuanto a la clase “PNEUMONIA” se obtuvo un total de 14 imágenes clasificadas correctamente, siendo estas la cantidad de verdaderos positivos, mientras que 1 imagen fue clasificada incorrectamente, representando así la cantidad de falsos negativos, Una vez obtenidas las cantidades de verdaderos positivos y falsos negativos se pudo obtener los resultados en cuanto a la métrica. Como se puede ver en la Ecuación 11, la evaluación del modelo obtuvo un total del 100% de sensibilidad en la clase “NORMAL” y en la clase de “PNEUMONIA” se obtuvo un total de 93.33% para la clasificación correcta de la enfermedad. Por lo tanto, el modelo presenta altos e importantes porcentajes de sensibilidad en las clases para la correcta clasificación de nuevas imágenes.

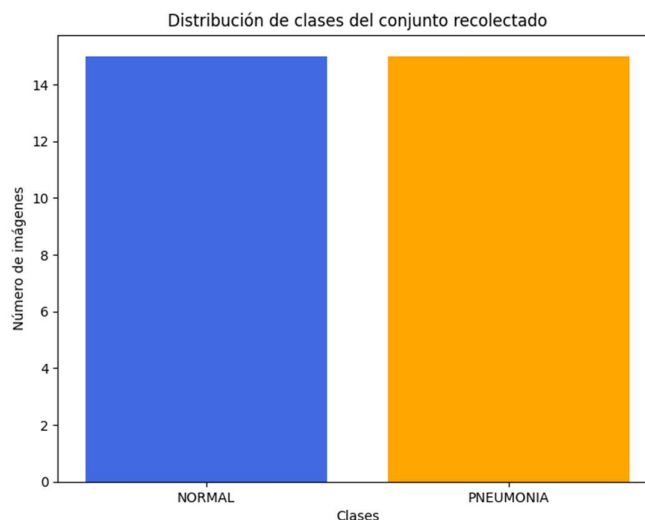


Figura 44. Distribución de clases dentro del conjunto recolectado para la evaluación con la técnica de Zero-Shot

⁷ <https://www.kaggle.com/models/edrdopcch41/vitbase-patch16-224-in21k-fine-tuning>

```

zero_transforms = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
])

zero_dataset = ImageFolder(
    data_dir, transform=zero_transforms
)

zero_dataloader = DataLoader(
    zero_dataset,
    batch_size=64,
    shuffle=True
)

```

Figura 45. Configuración del conjunto de imágenes recolectado para la evaluación con el modelo ajustado

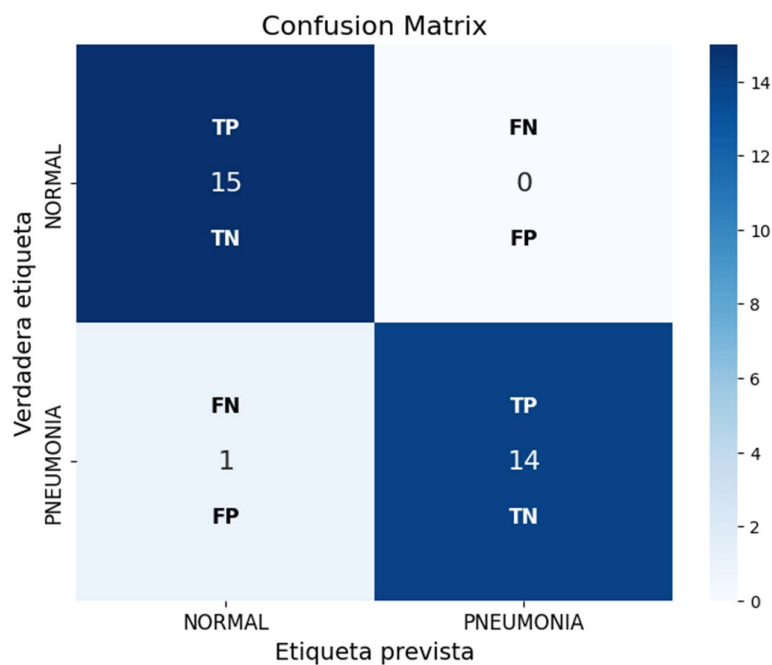


Figura 46. Matriz de confusión resultante de la evaluación del modelo con la técnica Zero-Shot. Donde se presenta la cantidad de verdaderos positivos (TP), falsos negativos (FN), verdaderos negativos (TN) y falsos positivos (FP) por cada clase.

Ecuación 11. Cálculo de sensibilidad de la clase NORMAL en la matriz de confusión de la evaluación con el conjunto recolectado.

$$\frac{TP}{TP + FN} = \frac{15}{15 + 0} = \frac{15}{15} = 1 = 100\%$$

Ecuación 12. Cálculo de sensibilidad de la clase PNEUMONIA en la matriz de confusión de la evaluación con el conjunto recolectado.

$$\frac{TP}{TP + FN} = \frac{14}{14 + 1} = \frac{14}{15} = 0.933 = 93.33\%$$

Ecuación 13. Cálculo de especificidad de la clase NORMAL en la matriz de confusión de la evaluación con el conjunto recolectado.

$$\frac{TN}{TN + FP} = \frac{14}{14 + 1} = \frac{14}{15} = \mathbf{0.933} = \mathbf{93.33\%}$$

Ecuación 14. Cálculo de especificidad de la clase PNEUMONIA en la matriz de confusión de la evaluación con el conjunto recolectado.

$$\frac{TP}{TP + FN} = \frac{15}{15 + 0} = \frac{15}{15} = \mathbf{1} = \mathbf{100\%}$$

Tabla 21. Valores de sensibilidad y especificidad obtenidos de la matriz de confusión en la prueba del modelo con el conjunto recolectado.

Clase	Sensibilidad	Especificidad
NORMAL	100%	93.33%
PNEUMONIA	93.33%	100%

7. Discusión

El objetivo principal del trabajo de integración curricular (TIC) se alcanzó mediante el desarrollo de 2 objetivos específicos, y los resultados obtenidos se discutirán en la siguiente sección:

7.1 Primer objetivo: Ajustar el modelo “vit-base-patch16-224-in21k” de Google, para la clasificación de neumonía en imágenes radiográficas de tórax utilizando el conjunto de datos “Chest X-Ray Images (Pneumonia)”.

Para poder realizar el ajuste del modelo “vit-base-patch16-224-in21k” se establecieron fases acordes al marco de trabajo de la metodología CRISP-ML(Q), que demostró ser un pilar fundamental para desarrollar de manera flexible las tareas que se acoplaron dentro de cada fase para el desarrollo del TIC. El desarrollo de las tareas para el ajuste del modelo requirió de un conjunto de datos que cuente con imágenes anterior-posterior de tórax. Por lo tanto, la selección del conjunto de datos fue realizada en referencia a trabajos relacionados [29], [49], [70], [71], [76] resultando en la selección del mismo conjunto de datos utilizado en los estudios. Para conocer las características del conjunto, se realizó un análisis exploratorio del conjunto seleccionado, evidenciando que los subconjuntos de entrenamiento y prueba se encontraban desbalanceados y mal distribuidos afectando a la calidad de la información. Para solucionar los problemas se optó por la aplicación de técnicas de balanceo y aumento de datos basado en los trabajos relacionados [72], [73][74], [75] resultando en un conjunto conformado por un total de 8.292 imágenes radiográficas (Tabla 14), y se realizó una redistribución de las imágenes totales para los subconjuntos de entrenamiento, prueba y validación (Tabla 15). El conjunto balanceado resultante se establece como la versión más adecuada para el entrenamiento del modelo, ya que, a diferencia del estado inicial, el conjunto balanceado presenta una mejor uniformidad entre el balance de sus clases y diversificación en las características de las imágenes gracias a las técnicas de balanceo y aumento de datos utilizadas. El ajuste del modelo “vit-base-patch16-224-in21k” se realizó con Pytorch y dentro del entorno de JupyterLab con la utilización de un clúster de GPU's que ayudó con los recursos computacionales. El modelo ajustado para el TIC, demostró una gran capacidad para la clasificación de neumonía en imágenes de rayos-x presentando un total de 97.71% de precisión, que es un porcentaje de precisión con mayor relevancia sobre el 92.45% descrito en [70] y el 87% reportado en [77]. Sin embargo, como parte del análisis del modelo ajustado fue necesario evaluar los resultados en base a la métrica de sensibilidad y la técnica de matriz de confusión. Como mencionó el especialista Ing. Óscar Miguel Cumbicus Pineda en la entrevista realizada para el análisis de modelos de IA en el ámbito médico. (Anexo 1), obteniendo como resultados del entrenamiento un total de 99.93% de sensibilidad para la

correcta clasificación de las imágenes pertenecientes a la clase “PNEUMONIA” (Tabla 18), representando así que el modelo posee una gran capacidad para el reconocimiento correcto de las radiografías pertenecientes a la enfermedad, en comparación a [70] que documentó obtener un total de 92.99% de sensibilidad en el entrenamiento de su modelo e igual que [49] obtuvo un 98.84% de sensibilidad en su propuesta.

7.2 Segundo objetivo: Evaluar el ajuste del modelo, por medio de la métrica de sensibilidad, para la clasificación neumonía en imágenes radiográficas de tórax, utilizando el conjunto de datos para validación “Chest X-Ray Images (Pneumonia)”.

La fase de evaluación del modelo es fundamental para la valoración de los resultados obtenidos en las fases anteriores de ajuste y entrenamiento. Por ende, en base al marco de trabajo se elaboró una función de evaluación donde se pueda obtener resultados de la clasificación de imágenes con el modelo ajustado, utilizando el conjunto de prueba. Se obtuvo como resultado de la evaluación un porcentaje total de 99.34% en la métrica de precisión del modelo, demostrando tener una mejor capacidad en la predicción de las clases, a comparación del 94% de precisión reportado por [71] y el 97.61% obtenido en los resultados de evaluación de [76]. Una vez ejecutada la evaluación del modelo, se realizó un análisis de clasificación de verdaderos positivos y falsos negativos con el propósito de conocer las características de las imágenes que se clasificaron correcta e incorrectamente en la evaluación. Se destaca que los trabajos relacionados no cuentan con el análisis propuesto, ya que, ayudó para la realización de un análisis más profundo de los resultados obtenidos de la matriz de confusión de la evaluación. En el análisis se destacó que el modelo tuvo conflictos para la clasificación de los verdaderos positivos mayormente de la clase “NORMAL” ya que 10 imágenes de las imágenes de esa clase fueron clasificadas incorrectamente, mientras que en la clase “PNEUMONIA” el modelo falló en la clasificación de 1 imagen (Figura 41). El análisis, logró como resultado identificar los patrones en las imágenes clasificadas incorrectamente (Figura 42 y Figura 43) las cuales en su mayoría contaban con alteraciones en la intensidad de su brillo, demostrando que la intensidad de brillo en la imagen puede afectar negativamente los resultados del modelo ajustado. Una vez realizado el análisis se obtuvo los porcentajes de sensibilidad con un total de 99.88% para la correcta clasificación de las imágenes de la clase “PNEUMONIA” en comparación a [76] que obtuvo un total de 94.9%. Los resultados de la métrica permitieron evidenciar que el modelo posee un alto rendimiento respecto a la fase de entrenamiento, en las métricas de precisión y en la sensibilidad de la clase “PNEUMONIA”. Al concluir la fase de evaluación del modelo con el conjunto de prueba, se establece que los resultados obtenidos son excelentes para la clasificación de neumonía en imágenes radiográficas de tórax. Finalmente se realizó la evaluación del modelo con la

utilización de la técnica llamada “Zero-Shot” que no ha sido utilizada en los trabajos relacionados. Y se la aplicó con el propósito de analizar los resultados del modelo en la clasificación de nuevas imágenes radiográficas, obteniendo los siguientes resultados: La precisión del modelo fue de 96.67%, con un porcentaje de sensibilidad total de 93.33% en la clase “PNEUMONIA” (Tabla 21). Al analizar los resultados del modelo en el contexto de la evaluación y con las nuevas imágenes, se observó que el desempeño del modelo no fue similar al obtenido en sus fases de evaluación y de entrenamiento, presentando menor sensibilidad en la clasificación de imágenes con la enfermedad, sin embargo, los porcentajes obtenidos, representan una capacidad muy buena para la clasificación de la enfermedad en imágenes nuevas. Esto demuestra que los procesos y tareas realizadas en base al marco de trabajo de la metodología CRISP-ML(Q) en los dos objetivos específicos, son adecuados para obtener resultados efectivos en el desarrollo del ajuste del modelo propuesto en el presente TIC.

8. Conclusiones

Una vez concluido el trabajo de integración curricular y analizado los resultados obtenidos se pueden establecer las siguientes conclusiones.

- El ajuste del modelo fue efectivo y satisface los alcances de los objetivos planteados. Dado que, fue realizado a partir del modelo de visión transformer “vit-base-patch16-224-in21k” ajustando las últimas capas de la red para la clasificación de las 2 clases y modificando los hiper parámetros en la fase de entrenamiento, donde su intervención fue importante para la obtención de resultados satisfactorios en la evaluación final reflejados en el 99.34% de precisión obtenido del modelo y el 99.88% de sensibilidad para la clasificación de neumonía a comparación de los trabajos relacionados, donde sus resultados son inferiores a los expuestos.
- En conclusión, el proceso de ajuste del modelo “vit-base-patch16-224-in21k” cumplió con los resultados esperados en cada una de las tareas establecidas por el marco de trabajo CRISP-ML(Q), donde resaltó su utilidad frente al desarrollo del ajuste ya que, funcionó como eje principal para la adaptación de fases clave como la ingeniería de datos que apporto en la redistribución y balanceo del conjunto de datos “Chest X-Ray Images (Pneumonia)” para que sea una fuente de información con mayor calidad referente a la brindada por el estado inicial del conjunto, y la fase de ingeniería de modelos que guio el proceso de ajuste con la experimentación con el fin de realizar la selección óptima de los hiperparámetros y creación de subconjuntos de datos funcionales para el entrenamiento del modelo.
- Finalmente, el proceso de evaluación del modelo ajustado fue una parte fundamental para la valoración del desempeño final, ya que al realizar un análisis profundo sobre los resultados en la clasificación se pudieron obtener características de los datos que pueden influir directamente en el rendimiento del modelo, como lo es el factor de intensidad de brillo. Además, la evaluación del modelo con el conjunto de prueba establece un elevado desempeño para la clasificación de neumonía dado el 99.34% de precisión obtenido y el 99.88% de sensibilidad en la clase de la enfermedad. De igual forma la evaluación del modelo con la aplicación de la técnica Zero-Shot permitió analizar el modelo, en la eficiencia de clasificación de nuevas imágenes de la enfermedad, dentro de un entorno real.

9. Recomendaciones

Una vez finalizado el trabajo de integración curricular, se pueden realizar las siguientes recomendaciones:

- Hacer uso del presente trabajo de integración curricular como referente a estudios de visión por computadora y la clasificación eficiente de imágenes médicas.
- Usar equipos con alta capacidad de procesamiento para la ejecución segura de la fase de entrenamiento del modelo, específicamente para la experimentación con la configuración de los hiperparámetros.
- Utilizar un entorno de desarrollo accesible, donde se pueda garantizar el acceso y vista de los resultados de las tareas en las fases de la metodología durante la investigación.

9.1. Trabajos Futuros:

- Se plantea el incremento en el número de clases dentro del modelo ajustado para la detección de varios tipos de neumonía. De esta manera se lograría poseer un modelo con una robustez extendida en la clasificación de la enfermedad.
- Considerar la implementación del modelo ajustado dentro de un aplicativo web o móvil, permitiendo que el modelo sea evaluado dentro del campo médico.
- Utilizar el modelo ajustado como referente en la clasificación de neumonía para nuevas investigaciones y realizar una comparativa contra otras arquitecturas de red ViT y para lograr comparar el desempeño del modelo propuesto.

9.2. Limitaciones:

- La cantidad de imágenes radiográficas de tórax recolectadas para la aplicación de la técnica de Zero-Shot Learning, representa una cantidad muy reducida de información limitando de esta manera que se pueda abstraer las características o patrones de las imágenes mediante el análisis de clasificación de verdaderos positivos y falsos negativos.

10. Bibliografía

- [1] L. F. Pisco-Acebo, A. J. Mora-Chancay, y J. A. Baque-Pin, “Infecciones por neumonía en individuos con Virus de Inmunodeficiencia Humana”, *MQRInvestigar*, vol. 8, núm. 1, 2024, doi: 10.56048/mqr20225.8.1.2024.2078-2099.
- [2] Y. V. Huallpa Vargas, “Comparación y Evaluación de Modelos Transformer en la Clasificación de Imágenes Médicas del Dataset NIH Chest X-rays”, *Universitat Politècnica de València*, oct. 2023, Consultado: el 6 de mayo de 2024. [En línea]. Disponible en: <https://riunet.upv.es:443/handle/10251/198542>
- [3] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, oct. 2020, [En línea]. Disponible en: <http://arxiv.org/abs/2010.11929>
- [4] M. Usman, T. Zia, y A. Tariq, “Analyzing Transfer Learning of Vision Transformers for Interpreting Chest Radiography”, *J Digit Imaging*, vol. 35, núm. 6, pp. 1445–1462, dic. 2022, doi: 10.1007/s10278-022-00666-z.
- [5] G. Jaime, A. Redactó, : Carranza-Cruz, y V. Sofia, “INFORMES DE CONFERENCIA Diagnóstico por imagen de neumonía Radiografía simple de tórax”, 2022, Consultado: el 23 de abril de 2024. [En línea]. Disponible en: https://remus.unison.mx/index.php/remus_unison/article/view/161/140
- [6] L. G. Silva-Guayasamín, D. Callejas, C. A. Silva-Sarabia, y G. S. Silva-Orozco, “PERFIL EPIDEMIOLÓGICO DE INFECCIONES RESPIRATORIAS AGUDAS EN PACIENTES PEDIÁTRICOS EN ECUADOR”, *Enfermería Investiga*, vol. 7, núm. 2, 2022, doi: 10.31243/ei.uta.v7i2.1620.2022.
- [7] Ministerio de Salud, “Situación epidemiológica de las infecciones respiratorias agudas (IRA) en el Perú.”, *Boletín Epidemiológico del Perú*, vol. 28, núm. 34, 2019.
- [8] MINSA, “Plan de Comunicaciones - Prevención de Infecciones Respiratorias Agudas (IRA) y Neumonía 2015”, *Boletín Informativo N° 09*, 2015.
- [9] P.-R. Delmotte y A. Monsel, “Neumonía viral grave en adultos”, *EMC - Anestesia-Reanimación*, vol. 50, núm. 1, pp. 1–15, feb. 2024, doi: 10.1016/S1280-4703(23)48663-X.
- [10] A. Rodríguez, L. Martínez, y S. Reyes Alvarado, “Uso de nuevas tecnologías en Radiología e imágenes diagnósticas y su relación con las competencias profesionales y/o perfil de egreso del Licenciado en Radiología de Panamá y Latinoamérica en los últimos 15 años”, *Ciencia Latina Revista Científica Multidisciplinar*, vol. 7, núm. 1, 2023, doi: 10.37811/cl_rcm.v7i1.4929.

- [11] MedlinePlus y Biblioteca Nacional de Medicina de EE. UU., "Imagenología y radiología: MedlinePlus enciclopedia médica", MedlinePlus. Consultado: el 7 de mayo de 2024. [En línea]. Disponible en: <https://medlineplus.gov/spanish/ency/article/007451.htm>
- [12] M. Garzón, L. Morales, y N. Tinoco, "Hallazgos radiológicos en radiografía y tomografía de tórax en pacientes diagnosticados de neumonía por sars-cov-2 Revisión bibliográfica", *Mediciencias UTA*, vol. 6, núm. 2, 2022, doi: 10.31243/mdc.uta.v6i2.1635.2022.
- [13] M. A. Amaro Garrido, Y. T. Solenzal Alvarez, T. Hernández González, y G. A. Orellana Meneses, "Diagnóstico imagenológico de neumonía por SARS-CoV-2 en pacientes con la Covid-19", *Gaceta Médica Espirituana*, vol. 22, núm. 3, 2020.
- [14] S. A. Douglas Clayton, D. Rodríguez Jarquín, y S. Zumbado Víquez, "Aspectos radiográficos de la neumonía por SARS-CoV-2", *Revista Médica Sinergia*, ISSN 2215-4523, ISSN-e 2215-5279, Vol. 8, N°. 5 (Mayo), 2023, vol. 8, núm. 5, p. 9, 2023, doi: 10.31434/rms.v8i5.1039.
- [15] M. Salina, J. Osio, M. Cappelletti, y M. Morales, "Aprendizaje automático aplicado al procesamiento de imágenes para la clasificación de objetos reciclables", en *Memorias del Congreso Argentino en Ciencias de la Computación - CACIC 2021*, 2021.
- [16] S. M. Rezaei, M. Ghorvei, R. Abedi-Firouzjah, H. Mojtahedi, y H. Entezari Zarch, "Detecting COVID-19 in chest images based on deep transfer learning and machine learning algorithms", *Egyptian Journal of Radiology and Nuclear Medicine*, vol. 52, núm. 1, 2021, doi: 10.1186/s43055-021-00524-y.
- [17] M. Roshni Thanka *et al.*, "A hybrid approach for melanoma classification using ensemble machine learning techniques with deep transfer learning", *Computer Methods and Programs in Biomedicine Update*, vol. 3, 2023, doi: 10.1016/j.cmpbup.2023.100103.
- [18] D. Khaledyan, T. J. Marini, T. M. Baran, A. O'Connell, y K. Parker, "Enhancing breast ultrasound segmentation through fine-tuning and optimization techniques: Sharp attention UNet", *PLoS One*, vol. 18, núm. 12 December, 2023, doi: 10.1371/journal.pone.0289195.
- [19] Putrama Alkhairi y A. P. Windarto, "Classification Analysis of Back propagation-Optimized CNN Performance in Image Processing", *Journal of Systems Engineering and Information Technology (JOSEIT)*, vol. 2, núm. 1, 2023, doi: 10.29207/joseit.v2i1.5015.
- [20] Tensorflow, "Transferencia de aprendizaje y ajuste | TensorFlow Core", TensorFlow. Consultado: el 7 de mayo de 2024. [En línea]. Disponible en: https://www.tensorflow.org/tutorials/images/transfer_learning?hl=es-419

- [21] J. Moreno, D. Rodríguez, M. A. Sicilia, J. C. Riquelme, y R. Ruiz, "SMOTE-I: mejora del algoritmo SMOTE para balanceo de clases minoritarias", *Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos*, vol. 3, núm. 1, 2009.
- [22] O. Spositto, G. Blanco, L. Matteo, y M. Levi, "SMOTE , Algoritmo para balanceo de clases en un estudio aplicado a la ganadería .", en *XXVI Congreso Argentino de Ciencias de la Computación - CACIC*, 2020.
- [23] V. Morales Oñate, L. Moreta, y B. Morales-Oñate, "SMOTEMD: UN ALGORITMO DE BALANCEO DE DATOS MIXTOS PARA BIG DATA EN R.", *Perfiles*, vol. 1, núm. 24, 2020, doi: 10.47187/perf.v1i24.75.
- [24] Á. Galán-Cuenca, R. Vázquez-Martín, A. Mandow, J. Morales, y A. García-Cerezo, "Análisis de técnicas de aumento de datos y entrenamiento en YOLOv3 para detección de objetos en imágenes RGB y TIR del UMA-SAR Dataset", en *XLII JORNADAS DE AUTOMÁTICA : LIBRO DE ACTAS*, 2021. doi: 10.17979/spudc.9788497498043.686.
- [25] C. Shorten y T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning", *J Big Data*, vol. 6, núm. 1, 2019, doi: 10.1186/s40537-019-0197-0.
- [26] C. C. Gómez Sarasa y J. D. Ortega Pabón, "Técnicas de aumento de datos para imágenes aéreas y evaluación de rendimiento en modelos de deep learning", *Revista Universidad Católica de Oriente*, vol. 31, núm. 45, 2020, doi: 10.47286/01211463.285.
- [27] D. J. Matich, "Redes Neuronales: Conceptos Básicos y Aplicaciones.", *Historia Santiago*, 2001.
- [28] H. Galán y A. Martínez, "Inteligencia artificial . Redes neuronales y Aplicaciones", *Universidad Carlos III de Madrid, Journal*, 2015.
- [29] Matas González Iván, "Clasificación de imágenes mediante Redes Neuronales Convolucionales y técnicas de Deep Learning avanzadas: Transformers", Ingeniería, Universidad de Sevilla, Sevilla, 2021. Consultado: el 7 de mayo de 2024. [En línea]. Disponible en: <https://idus.us.es/bitstream/handle/11441/126503/TFG-3609-MATAS%20GONZALEZ.pdf?sequence=1&isAllowed=y>
- [30] H. Yan, E. Zhang, J. Wang, C. Leng, A. Basu, y J. Peng, "Hybrid Conv-ViT Network for Hyperspectral Image Classification", *IEEE Geoscience and Remote Sensing Letters*, vol. 20, 2023, doi: 10.1109/LGRS.2023.3287277.
- [31] S. Sha *et al.*, "Image Classification and Restoration of Ancient Textiles Based on Convolutional Neural Network", *International Journal of Computational Intelligence Systems*, vol. 17, núm. 1, 2024, doi: 10.1007/s44196-023-00381-9.
- [32] H. Wu *et al.*, "CvT: Introducing Convolutions to Vision Transformers", en *Proceedings of the IEEE International Conference on Computer Vision*, 2021. doi: 10.1109/ICCV48922.2021.00009.

- [33] W. Wang *et al.*, “Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions”, en *Proceedings of the IEEE International Conference on Computer Vision*, 2021. doi: 10.1109/ICCV48922.2021.00061.
- [34] F. Pourpanah *et al.*, “A Review of Generalized Zero-Shot Learning Methods”, *IEEE Trans Pattern Anal Mach Intell*, vol. 45, núm. 4, 2023, doi: 10.1109/TPAMI.2022.3191696.
- [35] W. Wang, V. W. Zheng, H. Yu, y C. Miao, “A survey of zero-shot learning: Settings, methods, and applications”, 2019. doi: 10.1145/3293318.
- [36] D. Kermany, K. Zhang, y M. Goldbaum, “Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification”, *Mendeley Data*, vol. 2, 2018, doi: 10.17632/RSCBJBR9SJ.2.
- [37] R. Kundu, R. Das, Z. W. Geem, G. T. Han, y R. Sarkar, “Pneumonia detection in chest X-ray images using an ensemble of deep learning models”, *PLoS One*, vol. 16, núm. 9 September, sep. 2021, doi: 10.1371/journal.pone.0256630.
- [38] D. S. Kermany *et al.*, “Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning”, *Cell*, vol. 172, núm. 5, pp. 1122-1131.e9, feb. 2018, doi: 10.1016/J.CELL.2018.02.010.
- [39] S. Arifin, D. Wijonarko, y E. Kristianto Sijabat, “CC-BY) 4.0 license”, *Journal of Computer Science*, 2024.
- [40] T. Rahman *et al.*, “Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images”, *Comput Biol Med*, vol. 132, p. 104319, may 2021, doi: 10.1016/J.COMPBIOMED.2021.104319.
- [41] D. Kuzinkovas y S. Clement, “The Detection of COVID-19 in Chest X-rays Using Ensemble CNN Techniques”, *Information (Switzerland)*, vol. 14, núm. 7, 2023, doi: 10.3390/info14070370.
- [42] A. Degerli *et al.*, “COVID-19 infection map generation and detection from chest X-ray images”, doi: 10.1007/s13755-021-00146-8.
- [43] M. E. H Chowdhury *et al.*, “Can AI Help in Screening Viral and COVID-19 Pneumonia?”, doi: 10.1109/ACCESS.2020.3010287.
- [44] D. María y C. Delgadoillo Álvarez, “Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)”, *Review Revista Fesahancccal*, vol. 7, 2021.
- [45] M. Waseem Sabir, M. Farhan, N. S. Almalki, M. M. Alnfiai, y G. A. Sampedro, “FibroVit—Vision transformer-based framework for detection and classification of pulmonary fibrosis from chest CT images”, *Front Med (Lausanne)*, vol. 10, 2023, doi: 10.3389/fmed.2023.1282200.

- [46] IBM, “Métricas - Documentación de IBM”, IBM. Consultado: el 7 de mayo de 2024. [En línea]. Disponible en: <https://www.ibm.com/docs/es/mas-cd/maximo-vi/continuous-delivery?topic=configuring-understanding-metrics>
- [47] Barrios Arce Juan Ignacio, “La matriz de confusión y sus métricas – Inteligencia Artificial –”, Health Big Data. Consultado: el 7 de mayo de 2024. [En línea]. Disponible en: <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>
- [48] R. Diaz, “Métricas de Clasificación”. Consultado: el 25 de junio de 2024. [En línea]. Disponible en: <https://www.themachinelearners.com/metricas-de-clasificacion/>
- [49] T. Wang *et al.*, “PneuNet: deep learning for COVID-19 pneumonia diagnosis on chest X-ray image analysis using Vision Transformer”, *Med Biol Eng Comput*, vol. 61, núm. 6, pp. 1395–1408, jun. 2023, doi: 10.1007/s11517-022-02746-2.
- [50] J. Daniel y L. Cabrera, “Revisión crítica sobre la identificación de covid-19 a partir de imágenes de rayos x de tórax usando técnicas de inteligencia Artificial Critical Review on Covid-19 identification from Chest X-Ray images using Artificial intelligence Techniques”, *Revista cubana de transformación digital*, 2020, [En línea]. Disponible en: <https://www.worldometers.info/coronavirus/>
- [51] B. Wu *et al.*, “Visual Transformers: Token-based Image Representation and Processing for Computer Vision”.
- [52] Hugging Face, “google/vit-base-patch16-224-in21k · Hugging Face”, Hugging Face. Consultado: el 25 de junio de 2024. [En línea]. Disponible en: <https://huggingface.co/google/vit-base-patch16-224-in21k>
- [53] I. Matas González, “Clasificación de imágenes mediante Redes Neuronales Convolucionales y técnicas de Deep Learning avanzadas: Transformers”, 2021, Consultado: el 6 de mayo de 2024. [En línea]. Disponible en: <https://hdl.handle.net/11441/126503>
- [54] W. McKinney y P. D. Team, “Pandas-Powerful python data analysis toolkit”, *Pandas—Powerful Python Data Analysis Toolkit*, vol. 1625, 2015.
- [55] N. Silaparasetty, *Machine Learning Concepts with Python and the Jupyter Notebook Environment: Using Tensorflow 2.0*. 2020. doi: 10.1007/978-1-4842-5967-2.
- [56] S. Imambi, K. B. Prakash, y G. R. Kanagachidambaresan, “PyTorch”, en *EAI/Springer Innovations in Communication and Computing*, 2021. doi: 10.1007/978-3-030-57077-4_10.
- [57] A. Yadav, I. Rais, M. Kumar, A. Sharma, y A. Kushwaha, “Image Classification using Deep Learning and Tensorflow”, *Int J Res Appl Sci Eng Technol*, vol. 10, núm. 5, 2022, doi: 10.22214/ijraset.2022.43385.
- [58] Jalolov Sadriddinovich Tursunbek, “Artificial intelligence python (PYTORCH)”, *Oriental Journal of Academic and Multidisciplinary Research*, vol. 1, núm. 3, 2023.

- [59] W. Vallejo, C. Díaz-Urbe, y C. Fajardo, “Google Colab and Virtual Simulations: Practical e-Learning Tools to Support the Teaching of Thermodynamics and to Introduce Coding to Students”, *ACS Omega*, vol. 7, núm. 8, 2022, doi: 10.1021/acsomega.2c00362.
- [60] P. G. J. y N. K. V., “Google Colaboratory : Tool for Deep Learning and Machine Learning Applications”, *Indian Journal of Computer Science*, vol. 6, núm. 3–4, 2021, doi: 10.17010/ijcs/2021/v6/i3-4/165408.
- [61] S. Studer *et al.*, “Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology”, *Mach Learn Knowl Extr*, vol. 3, núm. 2, 2021, doi: 10.3390/make3020020.
- [62] G. Alexey, “CRISP-ML(Q)”. Consultado: el 25 de junio de 2024. [En línea]. Disponible en: <https://ml-ops.org/content/crisp-ml>
- [63] R. Brungel, B. Bracke, J. Ruckert, y C. M. Friedrich, “Teaching Machine Learning with Industrial Projects in a Joint Computer Science Master Course: Experiences, Challenges, Perspectives”, en *2023 IEEE 2nd German Education Conference, GECon 2023*, 2023. doi: 10.1109/GECon58119.2023.10295130.
- [64] K. Jeong, “The effect of audit quality on analyst following”, *Cogent Business and Management*, vol. 7, núm. 1, 2020, doi: 10.1080/23311975.2020.1798068.
- [65] M. Tyler y J. McCeney, “Assured AI reference architecture”, en *Proceedings of SPIE - The International Society for Optical Engineering*, 2024. doi: 10.1117/12.3027227.
- [66] S. Studer *et al.*, “Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology”, *Mach Learn Knowl Extr*, vol. 3, núm. 2, pp. 392–413, 2021, doi: 10.3390/make3020020.
- [67] Y. Eliwa, J. Haslam, y S. Abraham, “Earnings quality and analysts’ information environment: Evidence from the EU market”, *Journal of International Accounting, Auditing and Taxation*, vol. 42, 2021, doi: 10.1016/j.intaccaudtax.2020.100373.
- [68] G. Vrbančič y V. Podgorelec, “Transfer learning with adaptive fine-tuning”, *IEEE Access*, vol. 8, pp. 196197–196211, 2020, doi: 10.1109/ACCESS.2020.3034343.
- [69] A. Parvaiz, M. A. Khalid, R. Zafar, H. Ameer, M. Ali, y M. M. Fraz, “Vision Transformers in medical computer vision—A contemplative retrospection”, *Eng Appl Artif Intell*, vol. 122, p. 106126, jun. 2023, doi: 10.1016/J.ENGAPPAI.2023.106126.
- [70] A. Mabrouk, R. P. D. Redondo, A. Dahou, M. A. Elaziz, y M. Kayed, “Pneumonia Detection on Chest X-ray Images Using Ensemble of Deep Convolutional Neural Networks”, *Applied Sciences (Switzerland)*, vol. 12, núm. 13, jul. 2022, doi: 10.3390/app12136448.
- [71] P. N. Ha, A. Doucet, y G. S. Tran, “Vision Transformer for Pneumonia Classification in X-ray Images”, *ACM International Conference Proceeding Series*, pp. 185–192, feb. 2023, doi: 10.1145/3591569.3591602.

- [72] A. T. Abdulahi, R. O. Ogundokun, A. R. Adenike, M. A. Shah, y Y. K. Ahmed, "PulmoNet: a novel deep learning based pulmonary diseases detection model", *BMC Med Imaging*, vol. 24, núm. 1, 2024, doi: 10.1186/s12880-024-01227-2.
- [73] G. L. E. Maquen-Niño, J. G. Nuñez-Fernandez, F. Y. Taquila-Calderon, I. Adrianzén-Olano, P. De-La-cruz-vdv, y G. Carrión-Barco, "Classification Model Using Transfer Learning for the Detection of Pneumonia in Chest X-Ray Images", *International journal of online and biomedical engineering*, vol. 20, núm. 5, pp. 150–161, 2024, doi: 10.3991/ijoe.v20i05.45277.
- [74] WASUNAN CHOKCHAITHANAKUL, PROADPRAN PUNYABUKKANA, y EKAPOL CHUANGSUWANICH, "Adaptive Image Preprocessing and Augmentation for Tuberculosis Screening on Out-of-Domain Chest X-Ray Dataset", IEE explore. Consultado: el 11 de julio de 2024. [En línea]. Disponible en: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9987474>
- [75] M. S. Alam, D. Wang, y A. Sowmya, "Image data augmentation for improving performance of deep learning-based model in pathological lung segmentation", *DICTA 2021 - 2021 International Conference on Digital Image Computing: Techniques and Applications*, 2021, doi: 10.1109/DICTA52665.2021.9647209.
- [76] S. Singh, M. Kumar, A. Kumar, B. K. Verma, K. Abhishek, y S. Selvarajan, "Efficient pneumonia detection using Vision Transformers on chest X-rays", *Sci Rep*, vol. 14, núm. 1, 2024, doi: 10.1038/s41598-024-52703-2.
- [77] K. Pinasthika, E. Y. Sofyanda, S. Ulumiyah, y L. Muflikhah, "Deep Learning Approach for High Recall Pneumonia Classification with Swin Transformer and L2 Regularization", en *ACM International Conference Proceeding Series*, 2023, pp. 190–195. doi: 10.1145/3626641.3626670.

11. Anexos

Anexo 1: Transcripción entrevista especialista

Entrevistado: Ing. Oscar Cumbicus Pineda

Enlace de entrevista realizada:

https://drive.google.com/drive/folders/1nJP56jj5wY3dHDWB9cTi9cS_EPF8aEmH?usp=sharing

Transcripción de la entrevista realizada:

Tesista: Ingeniero Oscar Cumbicus Pineda, muy buenas tardes, la siguiente entrevista surge como parte integral de un estudio dirigido a evaluar la viabilidad de mejorar un modelo de transformer visión, enfocado específicamente en la detección de neumonía, basado en el procesamiento de imágenes radiográficas.

Tesista: Voy a hacer las siguientes preguntas. Voy a empezar con la entrevista, con la siguiente pregunta. ¿Conoce usted Transformer Vision? ¿Conoce aspectos sobre el avance de la tecnología? ¿Y las áreas en las que se está utilizando actualmente?

Entrevistado: Ya. Bueno, Vision Transformer o Transformer Vision en inglés es una variante de la red Neuronal Transformer que nació por allá en 2017. Entonces, la red Neuronal Transformer evolucionó al procedimiento del lenguaje natural por ser una nueva arquitectura que se basaba en la atención. Después del éxito que tuvo con el procedimiento del lenguaje natural, las personas que trabajaban en visión por computadora vieron que se podría extrapolar este conocimiento o esta arquitectura a la visión por computadora. Así que empezaron sus pruebas ya por allá en 2019 o 2018 mismo. Y empezaron a obtener muy buenos resultados. A tal punto que en la actualidad se dice que Vision Transformer va a ser el reemplazo de la red Neuronal Convolutiva. De hecho, en muchos estudios está obteniendo mejores resultados. Existen muchas redes que nos pusieron a utilizar justamente por el año 2019 cuando fue la pandemia para llegar al 2020. Los primeros usos que se le dieron a Vision Transformer eran para la clasificación de COVID o para la detección de COVID en radiografía. Esos fueron los primeros usos. Muy bien. De ahí en la actualidad se está utilizando Vision Transformer para muchas otras áreas como el cuidado de medicina, la detección de objetos, la segmentación de imágenes. Entonces, Vision Transformer en la actualidad es, se podría decir, la punta de la flecha que está dirigiendo la visión por computadora.

Tesista: Digamos, para el contexto del problema, sería el desarrollo de un Fine Tuning respecto a un modelo que esté basado en Transformer, ¿no? Entonces, para la siguiente

pregunta, este, le voy a decir, ¿cree que el Fight Tuning es una técnica que pueda aportar soluciones al problema de mejorar la detección de imágenes utilizando redes Transformer?

Entrevistado: Ya, muchas técnicas como el Fine Tuning, como el Transfer Learning, la gran mayoría de las veces mejoran la detección, la precisión de los modelos. Pero siempre este tipo de técnicas hay que manejarlas con cautela, no porque yo haga un Fine Tuning. Estoy asegurando de que va a funcionar. No es que, porque yo haga un Fight Tuning a una red, ya me va a mejorar el porcentaje de acierto o de precisión, sino que todo va a depender justamente de la calidad del Fine Tuning que yo le haga, si le agrego nuevos datos a eso, a esa red. Si los pasos que doy, por ejemplo, a la cuenta de la tarea de aprendizaje que hemos aprendido en clases, son más pequeños de los que se probaron con anterioridad. Entonces todo hay que manejar con cautela, pero tiene mucha probabilidad de mejorar los modelos. El Fine Tuning y el Transfer Learning tienen mucha probabilidad o una probabilidad exitosa de mejorar los modelos. Con eso, no estoy diciendo que no fallen. Sí fallan muchas de las veces, pero son muy pocas esas veces que fallan. Pero en la gran mayoría mejoran los modelos.

Tesista: Pero igual sí se puede considerar como éxito un avance de, digamos, un 1% o de décimas de por ciento.

Entrevistado: De hecho, en el Machine Learning o en cualquier área de la inteligencia artificial, la gran mayoría de estudios, o el estado del arte se da por décimas. Yo he tenido casos en que el estado del arte se ha dado por tres décimas. Es decir, que un nuevo estudio ha dicho que ha mejorado en tres décimas. Pero siempre hay que tener cuidado, porque esas tres décimas pueden ser un poco... No sé cómo decirlo. Tal vez ilusorias o divisorias, porque esas tres décimas pueden ser la variabilidad. El porcentaje, ¿no? Pero si estamos hablando de un 1%, ese ya es un número grande. De mejora. De un número aún más grande que uno. Pero cuando estamos hablando de décimas, de 0.2 a 0.5, ese es tema de la actualidad. Porque un modelo puede obtener eso en un rango, ¿no? De 0 a 0.5. Si hablamos de un punto, es mucho. Un punto porcentual es bastante mejor.

Tesista: Entonces, digamos, ahí llegaría un problema. El problema de no poder replicar los resultados en el rango de las décimas de ganancia.

Entrevistado: Sí, justamente existen muchos problemas cuando un estudio dice que ha llegado al estado del arte por décimas. Cuando se lo replica, cuando se lo replica, ha bajado esas décimas y ha obtenido peores resultados que el anterior. Justamente por décimas. Porque es un rango de variabilidad de 0 a 0.5, por ejemplo. Entonces, hay que determinar ese rango de variabilidad. Pero ya superar un modelo por un 1% se considera como un caso de éxito.

Tesista: Bueno, me comentaba que hay técnicas para llegar a esos resultados, ¿no? Entonces, digamos, ¿existen técnicas específicas para encontrar los valores óptimos en la configuración de los hiper parámetros?

Entrevistado: Las técnicas específicas son el ensayo y el error. Es decir, aumentando progresivamente, por ejemplo, la tasa de aprendizaje. Pones una tasa de aprendizaje y gradualmente vas aumentando o disminuyendo esa tasa de aprendizaje. Otra de las técnicas, una de las técnicas, entonces, es aumentar o disminuir gradualmente la tasa de aprendizaje. Otra de las técnicas es aumentar más épocas. Otra de las técnicas es mejorar el dataset. Es decir, meter imágenes de mayor calidad. Pues ahí estás finetuneando, porque le estás dando mejores datos. Se supone que, si tiene mejores datos, debería dar mejores resultados. Así mismo, evaluarlo a través de los pasos y no de épocas. Porque una cosa es que te lo evalúe al final de la época y otra es que te lo evalúe. Cada cierto paso, por ejemplo, en una época. Por ejemplo, si la época se demora en ejecutar 10.000 pasos, podrías evaluarlo cada 5.000 pasos o cada 2.000 pasos. Podrías ir viendo el avance. Puede ser que entre los pasos vas a encontrar el mejor modelo.

Tesista: Y partiendo, digamos, de las pruebas, vemos indicios de los resultados

Entrevistado: Mientras haces las pruebas, tú tienes que ir monitoreando. Si es que cuando haces las pruebas o estás entrenando el modelo, el porcentaje de aciertos va aumentando, pues está funcionando. Pero si en cambio, cuando está entrenando, si el porcentaje de aciertos no está aumentando, es porque no va a funcionar.

Tesista: Entonces, ¿nos alejamos de la perspectiva?

Entrevistado: Nos alejamos de la perspectiva o buscamos otra. Bueno, modificamos los parámetros y volvemos a entrenar. Si es que entiende mejor antes, lo dejamos hasta que mejore su punto más alto.

Tesista: Bueno, la siguiente pregunta era si era un, digamos, ¿estos ensayos de error era una técnica óptima para sacar el resultado o conocer un buen resultado en el ajuste del modelo? (¿Las técnicas de ensayo y error era una técnica óptima para desarrollo de ajuste sobre un modelo de visión transformer?).

Entrevistado: No, en verdad es la única. No hay otro. A pesar de que se ha automatizado, por ejemplo, la prueba y error. Por ejemplo, tú le puedes decir a tu modelo que te entrenen 10 épocas, pero si en las primeras 3 épocas no ha mejorado el modelo, mejor modelo, pues se corta. Pero al final son prueba y error. Es que sigue probando si los parámetros son los óptimos, ¿no? Al final tú lo paras automáticamente, pero al final es una prueba y error.

Tesista: Ya. Digamos acá lo que serán las pruebas. También tenemos que evaluar, ¿no? Nuestros resultados. Entonces, ¿Cree que la métrica de sensibilidad es relevante para evaluar el rendimiento de un modelo enfocado en detección de radiografías basado en transformers?

Entrevistado: Si. En la medicina se utiliza mucho el tema de la sensibilidad. Las pruebas de sensibilidad, tú escuchas la prueba de COVID, es más sensible a los casos

positivos que los casos negativos. Entonces, la sensibilidad en el tema médico es una medida muy alta. Entonces, se debería considerar evaluar el modelo por la sensibilidad que tenga. Más allá de la precisión global. Ustedes han aprendido en clase. La precisión global es una, pero la sensibilidad es el solo. Entonces, deberías, en el contexto médico de desarrollo de un modelo ViT para la detección de enfermedades, se debería enfocar la evaluación, a métricas de mayor impacto como la sensibilidad, ya que en la mayoría de modelos de ViT la métrica en la que más se enfocan es la precisión, pero no es la más importante ya que no solo con esa métrica se puede evaluar la eficiencia real de un modelo en casos específicos.

Tesista: ¿Conoce métricas relevantes para evaluar el rendimiento de un modelo de Visión Transformers en el contexto médico?

Entrevistado: Sensibilidad, sí. Lo que se conoce como TFP rate, o el rayo de los verdaderos positivos, el rayo de los verdaderos negativos, TFPR. Esas también son métricas que sirven para ti poder evaluar. Entonces, tú vas con esas métricas. Si mi modelo es más sensible para encontrar los casos positivos, o más sensible para encontrar los casos negativos. ¿Qué estás diciendo? ¿Cómo? Porque si tu modelo, si tu modelo es más sensible para encontrar los casos positivos, entonces encontrará, efectivamente, o será mejor, prediciendo cuando tú le metas una radiografía que efectivamente tenía pulmonía. Y si tu modelo es más efectivo para encontrar los casos negativos, será más efectivo cuando tú le metas una radiografía sana. Entonces, va a ser más efectivo. Y va a decir, esta efectivamente no tiene pulmonía. Entonces, esas son también las métricas que puedes usar. ¿De dónde sale eso? De la matriz de confusión. Entonces, para que no se complique en la vida, uno saca la matriz de confusión y manualmente puede calcular esto. La matriz de confusión está por cuántas clases se clasificaron bien y cuántas clases se clasificaron mal. De ahí, de la matriz de confusión, tú sacas las demás métricas que estamos hablando. La precisión, la sensibilidad, las clases. El score F1, el TFP rate. Entonces, la medida en tu caso sería la sensibilidad, pero en realidad todo esto sale de la matriz de confusión. Si tú tienes la matriz de confusión, analizas la matriz de confusión y sacas la sensibilidad, la precisión y todo lo demás.

Tesista: Bien. Igualmente, los datos. Igual, una parte relevante, ¿no?

Entrevistado: Yo te decía que una parte del fine tuning no solamente es los hiper parámetros como así, sino también ajustar las imágenes. Te dan imágenes de mejor calidad. Imágenes más variadas. O dataset equilibrado. Entonces, ¿a qué me refiero con imágenes más variadas o que existan imágenes de todos los casos? Si ibas a detectar tres casos, digamos, neumonía A, B y C, deberían haber de todos esos casos imágenes. Porque si solo tienes más del A, más del B y menos del C, tu modelo va a

funcionar peor en el C. Eso es por una ley. Claro. Ya. Y si tienes menos, menos muestras de los otros a aprender. Entonces, la calidad de los datos, la homogeneidad de los

datos o que estén presentes todas las muestras dentro de los datos o todos los individuos en este caso. Si vas a detectar neumonía y no neumonía, deberías tener tantos casos iguales de neumonía como una neumonía en su mejor de los casos. Y si es que la tienes desequilibrada, ese desequilibrio no debería ser tan alto. Es decir, no pasar que la una le sobrepase con más de la mitad que la otra clase. Sino que debería tener algo de equilibrio. Sí.

Tesista: Entonces, en resumen, también los datos son una parte esencial para lo que es el ajuste

Entrevistado: Los datos son la parte más esencial en todo el proceso de inteligencia artificial, machine learning o de deep learning, por lo tanto, deben estar correctamente procesados, en el caso de las redes ViT el tamaño del parche es la parte más importante de estos modelos, entonces la imagen de entrada o el conjunto de imágenes deben tener un correcto análisis y pre procesamiento que por lo general no se lo realiza y se utilizan directamente los conjuntos de datos de internet. Porque sin datos, no es que porque tengas más datos vas a tener mejor resultado. Tus datos deben ser de calidad para que tú tengas mejores resultados. No porque yo tenga 10 mil fotos de un problema, yo voy a obtener mejores resultados. Esos 10 mil datos deberían ser mejores que los datos que anteriormente se probaron. Si es que no son mejores, no vas a obtener mejores resultados.

Tesista: Y digamos ya como que entrando aquí el tema del procesamiento de los datos. ¿Cree que las técnicas de aumentación como la rotación, hacerles zoom a las imágenes, recortes, son un aspecto esencial que se debe analizar y utilizar para mejorar el modelo en transformación?

Entrevistado: Sí. Sobre todo, si te faltan datos. Si las clases han desequilibrado, si necesitas equilibrar una clase con otra, una recomendación es utilizar las técnicas de aumento de datos. En este caso rotación, desenfoque también te sirve. Porque tienes que tomar en cuenta que no todas las imágenes van a estar tomando las mismas condiciones. Ponerle un fondo, ponerle ruido, eso se llama ponerle ruido, ¿no? Ponerle un fondo, desenfoclarlas, rotarlas. Entonces esas técnicas son válidas y se han probado dentro de visión. No solo de visión transforma, sino en sí de la parte que es visión por computador. En visión transforma también funciona bien. ¿Por qué? Porque visión transformer es basado en transformer. La base de transformers la atención. Entonces cuando tú haces un tipo de esas técnicas, en visión transforma lo que hace es enfocarse en un área específica de la fotografía. Así como se enfocaba al inicio en el texto, en una palabra, específica del texto, en visión transforma se va a enfocar en un área específica de la imagen. Entonces cuando tú le das una imagen de una forma y después le das de otra forma, él va a tratar de enfocarse justamente en una parte esencial de esa fotografía. Entonces sí, son técnicas que se deberían considerar, pero que también hay que manejar con cautela. No puedes, por ejemplo, rotar todito el conjunto de datos para tener el doble. Si no, se deberían aplicar las técnicas, qué te digo, un 25% de

rotación, un 25% de desenfoque, otro 25% de ponerle ruido a las fotos y cosas así, ¿no? Todo hay que manejar con cautela porque recuerda que podríamos hacer que nuestro modelo se sobre-entrene. ¿Qué quiere decir eso? Que eso no detecte las fotos que yo le doy y que cuando le mando una nueva ya no me voy a ver. Entonces eso hay que manejar con cautela. Además, encuentras esa información también en todos los trabajos relacionados que se han hecho, en todas las técnicas cuando hemos utilizado estas técnicas, en todos los papers cuando hemos utilizado estas técnicas. Él te dice que más o menos el 25% se debería aplicar ese tipo de técnicas, una de las recomendaciones, pero tendrías que seguir leyendo para que puedas saber cuál es el porcentaje, ¿no?

Tesista: Igual también creo que es válido considerar eso de que no se pueden meter, digamos, técnicas de aumento, digamos, meter las que más se puedan, ¿no?

Entrevistado: Sí, Tampoco es que yo tengo que tener las clases equilibradas y decimos, es decir, 100 de la una y 100 de la otra. No, yo puedo tener 100 y 80 y no tengo un desequilibrio muy grande. Si tengo 100 y 20, entonces el desequilibrio entre las clases es muy grande. Entonces va... ahí sí necesito una técnica de aumento de las clases. Pero si yo tengo entre 100 muestras de una y 80 de la otra, yo necesitaría hacer un equilibrio de las clases, entonces necesitaría aplicar esas técnicas.

Tesista: Pero digamos acá en el caso del enfoque de radiografías, sí se podría considerar, digamos, utilizar el zoom, ¿no? Por la naturaleza de la imagen

Entrevistado: ¿El enfoque o el desenfoque? ¿Qué te decía? ¿El zoom? ¿Enfocas o desenfocas? Por ejemplo, rotarla... también se podría hacer, ¿no? Cambiar... ahora, si los pones de cabeza, por ejemplo, si las rotas son 45, 90... Si las rotas son 180 grados, no te serviría de nada, porque nunca creo que vayas a mandar una radiografía de cabeza, por ejemplo. Los pies para arriba y la cabeza para abajo, entonces... ¿Si o no? Los pulmones están al revés. Entonces, no creo que tengas un tipo de esas radiografías. O sea que por eso te digo hay que hacerlas con cuidado y viendo qué es lo que me serviría y qué no me serviría

Tesista: la última pregunta. ¿Usted cuál puede considerar que puedan ser los mayores impedimentos o problemas que se pueden presentar dentro del proceso de fine Tuning o ajuste?

Entrevistado: El problema que te decía al inicio. Hacerlo con cuidado, porque puede ser que al hacer el fine Tuning, empeoremos la clasificación. ¿Sí? Y entonces pensemos que no sirve el fine Tuning. Hay que hacerlo de a poco, no hay que hacer saltos tan grandes, ¿no? Es decir, los profesores decían que no hay cómo... Hay que hacer saltos pequeños para encontrar, llegar a la solución óptima. Porque si hacemos la modificación del Learning Rate de una forma muy abrupta o con parámetros... Pasar del menos cinco, del menos uno al menos cinco es un paso muy grande. Deberías pasar por el menos tres y después al menos cinco. Porque ahí estamos haciendo pasos más pequeños. Si me entienden, ¿no?

Tesista: Sí.

Entrevistado: Entonces, hacer pasos pequeños para poder encontrar la solución. De hecho, esa es la base del fine tuning. Hacer pasos bien pequeños para poder llegar a la solución óptima. No hacer pasos abruptos. Entonces, ese es uno de los problemas que podría tener. Otro de los problemas es que, efectivamente, fine tuning con los datos iniciales no obtengan mejores resultados. Pero puede ser por la calidad de los datos. Si tienes datos de mala calidad, lo que decíamos hace un momento. Por más que le metas fine tuning, puede ser que no mejores. Pero si tú le metes nuevos datos o le aumentas datos de mejor calidad, seguramente te va a funcionar. Ya. Entonces, esos son grandes, unos, dos grandes desafíos. Tener datos de calidad, hacerlo de manera progresiva. Y tal. Podría ser el tema de la infraestructura tecnológica o el hardware que tengas. Porque al utilizar ya un fine tuning más grande. O un número de, perdón, un learning rate más grande. O un número de épocas de mayor tamaño. O un batch size más grande. Necesitarás más hardware para procesar. No es lo mismo procesar cien épocas que procesar mil épocas. O procesar datos en batches de 16 que batches de 64. Ustedes ya lo han visto en clase. Las computadoras normales no te aguantan un batch size de 64. Pero una computadora con una GPU de 16 gigas sí te va a avanzar. Entonces, ese también podría ser un desafío. O un problema que pueda ser el hardware.

Tesista: Entonces, digamos, esos serían los tres principales enfoques en los que se podrían presentar algunas limitaciones.

Entrevistado: Sí. No, no creo que haya más limitantes. Tal vez la versión en la que se hizo la primera arquitectura. Porque tal vez utilizaran un, qué te digo, un framework. Transformer Vision 1.0. Y ahora ya no hay soporte para 1.0. Y no sean compatibles las librerías con Python de ahora, como en la anterior. Esa tal vez podría ser uno también que habría que considerar. Ver si es que el Python se va a hacer sobre una arquitectura nueva. No muy vieja. Porque debería funcionar en las condiciones actuales de los paquetes que pagamos.

Tesista: Listo ingeniero Oscar, esa sería toda la parte de la entrevista respecto a la problemática. yo le agradezco, muchas gracias por su tiempo.

Anexo 2: Certificación de traducción del resumen del TIC por parte de un profesional



Loja, 02 de agosto de 2024

Lic. Pedro Geovanny Calva Jiménez
LICENCIADO EN PEDAGOGÍA DEL IDIOMA INGLÉS

CERTIFICO:

Que el resumen del Trabajo de Integración Curricular cuyo título es: **Fitting of the "vit-base-patch16-224-in21k" model for pneumonia classification in chest radiographic images using transformer vision networks**, con cédula de identidad Nro.1105706319, de la Carrera de Computación de la Universidad Nacional de Loja, ha sido traducido al inglés y cumple con las características propias del idioma extranjero.

Lo certifico en honor a la verdad y autorizo hacer uso del presente en lo que a sus intereses convenga.

Lic. Pedro Geovanny Calva Jiménez

1150428496

Nro. Reg. Senecyt: 1031-2022-2421774

LICENCIADO EN PEDAGOGIA DEL IDIOMA INGLÉS

