

UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI



FACULTAD DE COMERCIO INTERNACIONAL, INTEGRACIÓN, ADMINISTRACIÓN Y
ECONOMÍA EMPRESARIAL

CARRERA DE LOGÍSTICA Y TRANSPORTE

Tema: “Diseño de un programa para el pesaje de vehículos de carga pesada a partir de imágenes con Inteligencia Artificial”

Trabajo de Integración Curricular previo a la obtención del
título de Ingeniero en Logística y Transporte

AUTOR: Herrera Montenegro Diego Joel

TUTOR: Mafla Bolaños Iván Gabriel MSc.

Tulcán, 2025.

CERTIFICADO DEL TUTOR

Certifico que el estudiante Herrera Montenegro Diego Joel con el número de cédula 0450167978 ha desarrollado el Trabajo de Integración Curricular: "Diseño de un programa para el pesaje de vehículos de carga pesada a partir de imágenes con Inteligencia Artificial"

Este trabajo se sujeta a las normas y metodología dispuesta en la Codificación del Reglamento de Régimen Académico y de Estudiantes de la UPEC, por lo tanto, autorizo la presentación de la sustentación para la calificación respectiva.

Mafla Bolaños Iván Gabriel MSc.

TUTOR

Tulcán, septiembre de 2025

AUTORÍA DE TRABAJO

El presente Trabajo de Integración Curricular constituye un requisito previo para la obtención del título de Ingeniero en la Carrera de logística y transporte de la Facultad de Comercio Internacional, Integración, Administración y Economía Empresarial

Yo, Herrera Montenegro Diego Joel con cédula de identidad número 0450167978 declaro que la investigación es absolutamente original, auténtica, personal y los resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.



Herrera Montenegro Diego Joel

AUTOR

Tulcán, septiembre de 2025

ACTA DE CESIÓN DE DERECHOS DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Yo Herrera Montenegro Diego Joel declaro ser autor de los criterios emitidos en el Trabajo de Integración Curricular: "Diseño de un programa para el pesaje de vehículos de carga pesada a partir de imágenes con Inteligencia Artificial" y eximo expresamente a la Universidad Politécnica Estatal del Carchi y a sus representantes de posibles reclamos o acciones legales.



Herrera Montenegro Diego Joel

AUTOR

Tulcán, septiembre de 2025

AGRADECIMIENTO

Agradezco a Dios por haberme permitido culminar con esta etapa de mi vida, a mis padres, Diego Herrera y Betty Montenegro, por el apoyo constante que me han brindado, su sacrificio por verme cumplir mis metas, por sus palabras de motivación, sus consejos y porque siempre hacen que crea en mí mismo, también a mi hermana, Diana Herrera, por ser parte importante en mi vida, a mis familiares, por estar pendientes en cada paso, y, Alison Molina, quien me ha acompañado en este camino universitario, con quien he compartido muchos momentos memorables.

Así mismo, quiero expresar mi más sincero agradecimiento a mi tutor, MSc. Iván Mafla, quien ha sido un guía para el desarrollo de este proceso, con su apoyo y entusiasmo por la docencia, que ha generado en mí un gusto por el mundo académico, influenciando mi formación académica.

DEDICATORIA

Este proyecto, se lo dedico a mis padres, pues son el motivo por el cual logré culminarlo, por la paciencia y el sacrificio que han puesto en mí, han sido la base para mi formación, tanto académica como personal, igualmente, la dedico a mi persona, por el tiempo y esfuerzo entregado en cada actividad, demostrándome que puedo hacer posible todo lo que me proponga.

ÍNDICE

RESUMEN.....	12
ABSTRACT	13
INTRODUCCIÓN	14
I. EL PROBLEMA.....	15
1.1. PLANTEAMIENTO DEL PROBLEMA	15
1.2. FORMULACIÓN DEL PROBLEMA.....	17
1.3. JUSTIFICACIÓN.....	17
1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN.....	19
1.4.1. Objetivo General.....	19
1.4.2. Objetivos Específicos	19
1.4.3. Preguntas de Investigación.....	19
II. FUNDAMENTACIÓN TEÓRICA	20
2.1. ANTECEDENTES DE LA INVESTIGACIÓN.....	20
2.2. MARCO TEÓRICO	23
2.2.1. Teoría General de Sistemas.....	23
2.2.2. Teoría de la Información	24
2.2.3. Inteligencia Artificial	25
2.2.4. Aprendizaje automático	26
2.2.5. Aprendizaje Profundo.....	27
2.2.6. Métricas de rendimiento	29
2.2.7. Lenguajes de programación	31
2.2.8. Pesaje en el transporte de mercancías.....	31
2.2.9. Vehículos de transporte de carga	32
III. METODOLOGÍA	36
3.1. ENFOQUE METODOLÓGICO	36

3.1.1. Enfoque	36
3.1.2. Tipo de Investigación.....	36
3.2. HIPÓTESIS	37
3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES.....	37
3.3.1. Variables	37
3.3.2. Operacionalización de Variables	38
3.4. MÉTODOS UTILIZADOS	39
3.5. ANÁLISIS ESTADÍSTICO	41
3.5.1. Prueba T de <i>Student</i>	41
IV. RESULTADOS Y DISCUSIÓN	45
4.1. RESULTADOS	45
4.1.1. Proceso de pesaje de los vehículos de carga pesada.....	45
4.1.2. Métodos de IA.....	58
4.1.3. Diseño del sistema de pesaje con Inteligencia Artificial.	95
4.1.4. Evaluación del rendimiento del programa	103
4.1.5. Cotización de equipos computacionales.....	118
4.1.6. Comparación entre sistema por báscula y programa de pesaje.	120
4.1.7. Hipótesis	122
4.2. DISCUSIÓN	127
V. CONCLUSIONES Y RECOMENDACIONES	133
5.1. CONCLUSIONES	133
5.2. RECOMENDACIONES.....	134
VI. REFERENCIAS BIBLIOGRÁFICAS.....	136
VII. ANEXOS.....	143

ÍNDICE DE TABLAS

Tabla 1. Definiciones utilizadas sobre las características de los vehículos de carga pesada.	32
Tabla 2. Operacionalización de las variables para el tema" Diseño de un programa para el pesaje de vehículos de carga pesada a partir de imágenes con Inteligencia Artificial"	38
Tabla 3. Especificaciones de Básculas tradicionales	48
Tabla 4. Ensayo de excentricidad y repetibilidad en báscula referencial.	49
Tabla 5. Ensayo de carga de báscula referencia.....	50
Tabla 6. Costos asociados con el uso de báscula.	52
Tabla 7. Errores comunes en las básculas tradicionales	57
Tabla 8. Tasas de error (%) de los resultados de single-model en el conjunto de validación de ImageNet.....	60
Tabla 9. Ventajas y limitaciones de ResNet, ResNext, DenseNet y EfficientNet.....	63
Tabla 10. Librerías usadas para el desarrollo del programa.....	65
Tabla 11. Especificaciones del computador utilizado en los entrenamientos de IA... ..	67
Tabla 12. Condiciones en las que fueron tomadas las fotografías,.....	70
Tabla 13. Distancia para la captura de imágenes entre el un ángulo del vehículo y la cámara, y por bodegas.....	72
Tabla 14. Resumen de base de Excel con la información de las imágenes.	74
Tabla 15. Forma de la base de Excel con la información de las imágenes.	75
Tabla 16. Número de vehículos por cada tipo de vehículo,.....	76
Tabla 17. Número de vehículos por clasificación de camión y tractocamión	77
Tabla 18. Código para unificar las imágenes por lados.	78
Tabla 19. Código para unificar las imágenes por sus 4 ángulos.	81
Tabla 20. Cantidad de imágenes por <i>dataset</i>	92
Tabla 21. <i>Datasets</i> en formato <i>TFRecord</i> y su tamaño de almacenamiento.....	94
Tabla 22. Configuración por modelo a generar	99
Tabla 23. Características del entrenamiento por modelo.	102
Tabla 24. Excel con datos de la imagen y predicciones.....	104
Tabla 25. Estimación del peso de los vehículos por tipo de modelo.....	105
Tabla 26. Resultados con métricas MAE y MSE, del modelo 2 por el tipo de ángulo.	109

Tabla 27. Resultados con métricas MAE y MSE, del modelo 3 por el tipo de unión..	109
Tabla 28. Cotización de equipo computacional con características al equipo usado en los entrenamientos.....	118
Tabla 29. Consumo en <i>watts</i> de la GPU y tarifa de energía eléctrica.....	119
Tabla 30. Comparación entre básculas y programa de pesaje con IA	120

ÍNDICE DE FIGURAS

Figura 1. Representación de los elementos de un sistema de un cuerpo en movimiento.	23
Figura 2. Modelo de comunicación	25
Figura 3. Tipos de aprendizaje automático	27
Figura 4. Procesamiento de datos en una red neuronal profunda	28
Figura 5. Tabla Nacional de Pesos y Dimensiones.....	34
Figura 6. Tabla nacional de pesos y dimensiones posibles combinaciones	35
Figura 7. Representación de fuerza y dirección del coeficiente de correlación.....	43
Figura 8. Diagrama de flujo del proceso de pesaje de vehículos.	45
Figura 9. Tiempos promedio en báscula por tipo de vehículos de carga	46
Figura 10. Medidas del espacio ocupado por báscula de referencia en metros (m).	52
Figura 11. Estructura de una red neuronal convolucional.....	59
Figura 12. Arquitectura de ResNet en comparación con <i>ResNext</i>	61
Figura 13. Arquitectura de <i>DenseNet</i>	62
Figura 14. Escalamiento del modelo <i>EfficientNet</i>	62
Figura 15. Representación del proceso de captura de imágenes por ángulo.....	71
Figura 16. Representación del proceso de aumento de datos al tomar más fotos de un mismo vehículo en un mismo ángulo.	72
Figura 17. Imágenes unidas por lado izquierdo y derecho para la parte delantera, trasera y lateral.	81
Figura 18. Imágenes unidas entre sus 4 esquinas y de manera lateral.	89
Figura 19. Arquitectura ResNet 50 y ResNet-68 (Adaptada)	96
Figura 20. Gráfico de pérdida por modelos entrenados.....	100
Figura 21. Gráfico histórico de la métrica MAE a lo largo de las épocas.....	101
Figura 22. Entrenamiento y consumo de recursos de computador	102

Figura 23. Rendimiento con métricas MAE por modelo y clasificación de vehículo	106
Figura 24. Rendimiento con métricas MSE por modelo y clasificación de vehículo.	106
Figura 25. Rendimiento con métrica MAE por modelo y lugar de captura de imágenes.....	107
Figura 26. Rendimiento con métricas MSE por modelo y lugar de captura de imágenes.....	108
Figura 27. Valor de pérdida por estadístico RMSE.....	110
Figura 28. Ajuste de curva de función exponencial negativa.....	112
Figura 29. Predicción de magnitud de pérdida a 40 épocas posteriores al entrenamiento.	113
Figura 30. Filtros aplicados a una imagen del <i>dataset</i> de prueba del modelo 3.	115
Figura 31. Filtros aplicados a una imagen del <i>dataset</i> de prueba del modelo 2.	116
Figura 32. Filtros aplicados con distinta capa a una imagen del <i>dataset</i> de prueba del modelo 2.	117
Figura 33. Diagrama de dispersión entre el peso real y peso predicho por el programa.	123
Figura 34. Región de rechazo para el estadístico t en prueba T de <i>Student</i>	126

ÍNDICE DE ANEXO

Anexo 1. Acta de la sustentación de Predefensa del TIC.....	143
Anexo 2. Certificado del abstract por parte de idiomas	144
Anexo 3. Fichas de observación.....	144
Anexo 4. Fichas documentales.....	148
Anexo 5. Cuestionario.....	150
Anexo 6. Solicitud para levantamiento de información en Transbolivariana C.A. ...	151
Anexo 7. Solicitud para levantamiento de información a SENAE.....	152
Anexo 8. Resolución Nro. INEN-INEN-2023-0007-RES	154
Anexo 9. Código de CNN ResNet -68.....	155

RESUMEN

La investigación tuvo como objetivo desarrollar un programa de pesaje para vehículos de carga pesada a partir de imágenes con Inteligencia Artificial (IA). La información utilizada consiste en imágenes de vehículos identificados con sus respectivos pesos para el entrenamiento de los modelos. Se analizaron las básculas empleadas para pesar vehículos, encontrando que son sistemas complejos de implementar, costosos, pero necesarios en el transporte terrestre. La propuesta se desarrolla con una arquitectura de redes neuronales ResNet, entrenada con imágenes tomadas a vehículos con y sin carga de 1390 registros fotográficos. Las imágenes se etiquetaron con los pesos registrados en básculas camioneras de varios centros de almacenamiento. Se generaron tres *datasets*: el primero incluyó las imágenes originales; el segundo imágenes unidas por pares correspondientes a los lados izquierdo y derecho de los vehículos; y en el tercero, las imágenes fueron tratadas para mostrar las partes más visibles de cada ángulo y se unificaron en una sola imagen como un mosaico de 4 imágenes unidas en pares en los ángulos laterales. El modelo 3, entrenado con el tercer *dataset* tuvo mejores resultados. Al usar estrategias de aumento de datos se obtuvo una base de 11207 imágenes para entrenamiento y 2891 imágenes para prueba con una resolución de 1400 x 1400 píxeles y en escala de grises. Con los datos de prueba se alcanzó una métrica general MAE de 5345.16 *kg*. Las imágenes unidas lateralmente presentaron un menor error MAE de 3534.38 *kg*. De esta forma, se pudo identificar que el modelo fue capaz de resaltar partes asociadas con el peso de un vehículo, como las llantas, espacio entre estas, chasis y forma del vehículo. Para mejorar este error, es necesario aumentar la cantidad de imágenes, tomar fotografías en entornos controlados y entrenar los modelos con una cantidad mayor a 60 épocas.

Palabras Claves: Programa de pesaje, Inteligencia Artificial, básculas, ResNet, Aprendizaje profundo.

ABSTRACT

The objective of this research was to develop a weighing program for heavy-duty cargo vehicles using Artificial Intelligence (AI) from image data. The information employed consisted of images of vehicles identified with their corresponding weights, which were used to train the models. An analysis of the weighing scales commonly employed for vehicles revealed that, although necessary in road transportation, they are complex to implement and costly. The proposed approach was developed using a ResNet neural network architecture, trained with 1,390 photographic records of vehicles with and without cargo. The images were labeled with the weights recorded by truck scales in various storage centers. Three datasets were generated: the first included the original images; the second consisted of paired images corresponding to the left and right sides of the vehicles; and the third was constructed by processing the images to highlight the most visible parts from each angle, which were then combined into a single image as a mosaic of four paired side-angle views. The third model, trained with this third dataset, yielded the best results. By applying data augmentation strategies, a total of 11,207 training images and 2,891 test images were obtained, with a resolution of 1400×1400 pixels in grayscale. On the test data, the model achieved an overall MAE metric of 5,345.16 *kg*. The laterally paired images resulted in a lower MAE error of 3,534.38 *kg*. In this way, it was possible to identify that the model was able to emphasize features associated with a vehicle's weight, such as the tires, spacing between them, chassis, and overall shape of the vehicle. To further reduce this error, it is necessary to increase the number of images, capture photographs under controlled conditions, and train the models for more than 60 epochs.

Keywords: Weighing Program, Artificial Intelligence (AI), Weighbridges, ResNet, Deep Learning.

INTRODUCCIÓN

El desarrollo del presente trabajo de investigación se basó en la propuesta de un programa basado en inteligencia artificial (IA), para estimar el peso de los vehículos de carga pesada, pues los sistemas que se usan actualmente son muy complejos y costosos de implementar, por lo tanto, la mayoría de las empresas prefieren no invertir en estos sistemas. Así mismo, se busca aportar con una investigación respecto al uso de redes neuronales convolucionales, y que tan efectivas pueden llegar a ser con respecto a problemas de regresión lineal. Ante esto, el empleo de tecnologías basadas en la inteligencia artificial se presenta como una opción eficaz e innovadora para optimizar estos procedimientos, representando más rapidez, exactitud y automatización. Este planteamiento saca partido del potencial de la visión artificial para calcular el peso de los camiones a partir de rasgos visuales, lo que supone una innovación considerable en el área del control logístico y del transporte terrestre. La investigación se organiza en cinco apartados. En el Capítulo 1, se aborda el problema, señala fallos y limitantes recurrentes del sistema de pesaje actual, su necesidad en el transporte y falta de implementación de inteligencia artificial en el sector, estableciendo el contexto, justificación y objetivos del estudio. El Capítulo 2, fundamentos teóricos, trata los conceptos fundamentales que respaldan el trabajo con referencia a los sistemas de pesaje y transporte de mercancías, las redes neuronales y visión por computador para resolución de problemas en el transporte. El Capítulo 3, metodología, describe el enfoque cuantitativo adoptado, así como el diseño experimental, la recopilación y procesamiento de los datos, y los instrumentos tecnológicos usados en la creación del modelo. El Capítulo 4, resultados y discusión, muestra los descubrimientos derivados del entrenamiento del modelo, la estimación de pesos con respecto a los datos de prueba y el análisis comparativo en contraste con los sistemas actuales. Por último, el Capítulo 5, conclusiones y recomendaciones, resume las contribuciones del estudio, valora el cumplimiento de los objetivos marcados y propone opciones de mejora a futuras investigaciones. Este proyecto aspira a contribuir al desarrollo de alternativas tecnológicas en el sector logístico, impulsando un sistema de pesaje más rápido, rentable y adecuado a las exigencias del transporte actual.

I. EL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

La tecnología avanza a grandes pasos en el mundo, cada vez se presenta más automatización y por tanto procesos más eficientes, lo que hace que la industria tenga que adaptarse a estos nuevos avances. Uno de estos es la inteligencia artificial (IA), que según, Alvarado (2015), "se define como la ciencia que tiene un enfoque importante en el desarrollo de programas o máquinas, que son capaces de razonar solo para la solución de un problema" (p. 27), consiguiendo optimizar tareas, al eliminar el error humano y mantenerse operativa sin descansos. Siendo crucial implementarla dentro de cadenas de suministro, tal como lo mencionan Gutiérrez y Polo (2023), "la IA puede optimizar la planificación y gestión de la cadena de suministros, mejorando la previsión de la demanda, la gestión de inventarios, la programación de la producción y la logística" (p. 12).

Uno de los elementos más importantes en la cadena de suministro es el transporte, debido al flujo de materiales que existe, esto debido a que, sin importar el sector económico, siempre se deben movilizar mercancías, de igual manera, "Si la globalización implica transportar cada vez más productos a mayores distancias, el manejo óptimo de todos los recursos implicados puede no solo significar mejores resultados financieros, sino la supervivencia de las empresas" (González, 2013, p.1). Y de los tipos de transporte, según lo mencionado por Martínez (2006), el transporte terrestre es el más importante, debido a que, sin este, el transporte marítimo o aéreo no podrían culminar sus entregas, y, además, este tiene la ventaja de prevalecer de manera independiente, siendo el marítimo y aéreo imprescindible.

El transporte terrestre por carretera es un modo que tiene muchas implicaciones a considerar, una de ellas es la seguridad, tal y como expone Malave (2018) el gestionar la seguridad se basa en protección de usuario, población y medio ambiente, donde para cumplir esto es importante tener inspecciones sobre los vehículos, frenos, batería, combustible, neumáticos y demás componentes mecánicos de los mismos. Y otra parte importante es la infraestructura, es decir redes viales, y estas tienen un gran inconveniente, y de aquí un problema que llegan a tratar Ayala y Campos (2013):

Todas las carreteras se diseñan bajo parámetros definidos, para cumplir con una determinada vida útil, tolerando cargas máximas establecidas; pero si estas cargas son superadas por los vehículos que se sirven de ellas, será inevitable el deterioro progresivo de la estructura del pavimento. (p. 51)

En relación con lo referido anteriormente el pesaje de los vehículos de carga pesada es de gran importancia, tanto para preservar los caminos, como para evitar los peligros que significa un exceso de carga, con base en lo mencionado por TYGOM (2023), que el exceso de carga afecta en la dinámica de los vehículos, genera un aumento de distancia al momento de frenar, los neumáticos y los sistemas de seguridad se pueden ver alterados con el sobrepeso y se da un desgaste prematuro de los vehículos, lo que conlleva a un inminente peligro en carretera. Es por estas razones que existen reglamentaciones que controlan estas irregularidades en los pesos de los vehículos, como la prohibición de la Reforma Reglamento a la Ley de Caminos de la República del Ecuador del Ministerio de Transporte y Obras Públicas (MTO) (2015) sobre, "Vehículos cuyos pesos y dimensiones excedan a los permitidos y que no tengan certificados de operación especial, acorde a lo prescrito en el presente Capítulo" (p. 2).

Para poder realizar esta medición del peso, es indispensable una báscula diseñada para este tipo de vehículos pesados, pero estas tienen sus limitaciones, que según Mettler-Toledo (2020), son las complicaciones físicas, puesto que con el tiempo y el uso se van deteriorando y su precisión se ve afectada, por lo que, en básculas en mal estado pueden perder mucha mercadería por una mala calibración, además son susceptibles a impactos de rayos y cambios en el ambiente, es por esta misma razón que su electrónica no es muy avanzada, pero aun así entran en gastos de mantenimientos y reparaciones. La inteligencia artificial ha generado grandes beneficios en la cadena de suministro, sin embargo, se evidencia que, en este proceso de especial relevancia para el transporte, a nivel tecnológico aún no ha producido cambios significativos.

La adaptabilidad de la IA es un tema de gran relevancia, puesto que, permite aplicarla en diferentes campos, según Ruano et al. (2019), dentro de la Zona 1 del Ecuador se puede esperar el incremento de productividad sobre bienes primarios, tal es el caso de la manufactura en tejidos o artesanías, en el sector agrícola, florícola, producción láctea y ganadera, estimando que Carchi, Imbabura, Esmeraldas y Sucumbíos sean afectados de manera positiva implementando IA en procesos

productivos. Un ejemplo de este tipo de herramientas propone Ormeño y Zapata (2024) es de implementar un programa capaz de valorar el peso de cuyes utilizando la cámara de un teléfono celular. Pero a pesar de esta gran capacidad de la IA, no se han implementado este tipo de herramientas en el Ecuador, para Barragán (2023), "Existe una brecha significativa al comparar al ecosistema de la IA en el Ecuador con los ecosistemas de Argentina, Brasil, Chile, Colombia, y Uruguay; mostrando un incipiente uso, adopción de esta tecnología" (p. 23).

Ahora, en el problema que se comprende de los sistemas de pesaje de los vehículos se da al tipo de tecnología que usan, como el ambiente afecta en sus operaciones y estas generan costos para la empresa, aun así, no se han cambiado estos sistemas porque hasta la actualidad no existen otros métodos para poder estimar el peso de los vehículos sin necesidad de un equipo físico. Por otro lado, la IA no ha sido utilizada a nivel nacional en diversas actividades que se pueden implementar, pero se sabe que para poder estimar el peso de un objeto se puede llegar a requerir solamente una fotografía, entonces, existe la posibilidad de adaptar esta tecnología a estos sistemas, sin embargo, aún no se ha usado en el sector.

1.2. FORMULACIÓN DEL PROBLEMA

¿Es posible optimizar los sistemas de pesaje de vehículos de carga pesada con el diseño de un programa de pesaje a partir de imágenes con inteligencia artificial?

1.3. JUSTIFICACIÓN

Los beneficios de implementar IA en cualquier proceso son significativos, especialmente cuando se busca optimizar operaciones, tal como lo menciona Marín et al. (2023), "Ante las limitaciones de los seres humanos, la IA es una herramienta excelente para detectar defectos en la industria, y realizar procesos más eficientes y con errores mínimos" (p. 3716). En el caso de un nuevo sistema de pesaje que no dependa de una báscula física, se eliminan muchos de los problemas comunes asociados con las básculas tradicionales, como los daños causados por el entorno donde se encuentran o el desgaste de sus componentes. Además, al no incluir los diversos mecanismos que utilizan las básculas convencionales, se reducen considerablemente los costos asociados al mantenimiento y las reparaciones, lo que resulta en una solución más eficiente y económica a largo plazo.

Otro aspecto favorable es su adaptabilidad, ya que no requiere una infraestructura especializada. Este sistema puede implementarse fácilmente en diferentes procesos, como el transporte, aduanas, puertos o estaciones de peaje, independientemente

de su ubicación. Su funcionamiento se basa en la captura de una fotografía del vehículo de carga pesada y posteriormente subirla al programa, lo que puede implicar cumplir con algunos requisitos mínimos, pero sin necesidad de hacer modificaciones físicas. Esto hace que el proceso sea versátil y rápido, dado que solo toma el tiempo necesario para capturar la imagen. Una vez que la fotografía se sube al sistema, el resultado se obtiene casi de inmediato, lo que ahorra tiempo y simplifica la operación.

Este sistema se planteó por su gran utilidad en varios casos prácticos, como seguridad vial, donde autoridades la incorporaren en el pesaje de vehículos de carga pesada en sus controles sin que estos procedimientos requieran mucho tiempo o sistemas costosos, lo que facilita la verificación inmediata del peso que un vehículo transporta en un momento dado, esto gracias a que el programa se realizó tomando en cuenta la eficiencia y accesibilidad. Por esta misma razón, lo convierte en una alternativa viable para organizaciones que, debido al costo elevado de las básculas tradicionales, no pueden integrarlas en sus procesos. Así, las autoridades aumentan sus opciones de pesaje de modo más flexible en diferentes lugares dentro de sus controles viales.

A su vez, el trabajo buscó constituirse en una opción para las empresas que, hasta ahora, no han implementado un proceso de pesaje debido a que les resulta más rentable no incluirlo. La posibilidad de integrar este sistema accesible permite a dichas organizaciones llevar un control más eficiente sobre la entrada y salida de mercancías, sin la necesidad de incurrir en grandes gastos. Esto no solo reduce los costos operativos para empresas que ya requieren pesaje, como las de transporte de carga pesada, sino que también ofrecería una opción viable para aquellas que no crean necesario un proceso de control de pesaje costoso, ahora lo puedan incluir debido a la accesibilidad del programa consiguiendo una ventaja competitiva. Los principales beneficiarios corresponden al sector del transporte terrestre, empresas que mueven grandes volúmenes de productos y peajes, todos con el objetivo de mejorar la seguridad vial y el control logístico.

Además, la ejecución de este proyecto abre nuevas oportunidades para explorar el uso de la inteligencia artificial, mostrando los alcances que puede lograr y cómo es capaz de automatizar procesos que tradicionalmente parecen requerir maquinaria específica o la supervisión directa de una persona. Este proyecto busca expandir la perspectiva empresarial hacia el desarrollo de programas más eficientes, que les

permite a las empresas volverse más competitivas y acercar estas herramientas tecnológicas a un mayor número de personas. El objetivo final es cumplir con uno de los principios clave de la logística, optimizar cualquier proceso dentro de la cadena de suministro, y la mejor manera de lograrlo es implementando soluciones que no sean excesivamente complejas y que, a su vez, cumplan con los requisitos para los cuales fueron diseñadas, este tipo de soluciones hoy en día se pueden satisfacer con la Inteligencia Artificial, solamente hay que saber aplicarla.

1.4. OBJETIVOS Y PREGUNTAS DE INVESTIGACIÓN

1.4.1. Objetivo General

Optimizar los sistemas de pesaje de vehículos de carga pesada con el diseño de un programa de pesaje a partir de imágenes con inteligencia artificial.

1.4.2. Objetivos Específicos

- Determinar los procesos que conlleva el pesaje de los vehículos de carga pesada.
- Determinar los requerimientos técnicos para el desarrollo del programa de pesaje con imágenes.
- Diseñar el programa de pesaje de vehículos de carga pesada a partir de imágenes con Inteligencia Artificial.
- Evaluar el rendimiento del programa de pesaje de vehículos de carga pesada a partir de imágenes.

1.4.3. Preguntas de Investigación

- ¿Cuáles son los procesos de pesaje de los vehículos de carga pesada?
- ¿Cuáles son los parámetros que se deben tomar en cuenta para saber que el programa logra optimizar los sistemas de pesaje?
- ¿Qué método de la Inteligencia Artificial será necesario para poder desarrollar el programa de pesaje?
- ¿Cuáles son los requerimientos técnicos para poder desarrollar el programa de pesaje?
- ¿Cuál es el procedimiento para poder desarrollar el programa de pesaje a partir de imágenes?
- ¿Cuál es el proceso para pesar los vehículos de carga pesada con el programa de pesaje?
- ¿Cuáles son los resultados de las métricas de rendimiento del programa para saber si es óptimo?

II. FUNDAMENTACIÓN TEÓRICA

2.1. ANTECEDENTES DE LA INVESTIGACIÓN

Entre los principales proyectos, el realizado por Chen et al. (2022), propone un modelo de regresión de redes neuronales convolucionales (CNN) para estimar los pesos de cadáveres de patos blancos de *Pekin*. Este proyecto cuenta con un enfoque cuantitativo, puesto que, trabaja con redes neuronales convolucionales que se basan en modelos matemáticos y estadísticos, y al ser una investigación aplicada busca resolver un problema de regresión para encontrar el peso de los cadáveres de patos, además, de aplicar técnicas estadísticas para medir su eficiencia. Para la recolección de datos, los investigadores utilizaron la observación para capturar imágenes de los cadáveres de los patos desde distintos ángulos, ajustando el entorno con el fin de mejorar la calidad de estas. Y así, se obtuvo como resultado una estimación del modelo precisa, con una desviación absoluta media (MAD) de 58.8 gramos y un error medio relativo (MRE) de 2.15% del conjunto de datos de prueba. Este proyecto aporta a la presente investigación mostrando un uso de las redes neuronales convolucionales para poder estimar el peso de un objeto, en este caso del cuerpo de un pato, mostrando el procedimiento para aplicar la inteligencia artificial, dando una guía sobre qué aspectos se deben tener en cuenta al momento de tomar las imágenes y una explicación sobre el funcionamiento de las CNN en la investigación.

En el artículo realizado por Ormeño y Zapata (2024), plantean estimar el peso de un cuy con base en una imagen de este, adaptando la red convolucional Mask R-CNN. Este proyecto adopta un enfoque cuantitativo, ya que emplea redes neuronales convolucionales (CNN) para analizar las características de las imágenes mediante algoritmos que encuentran patrones y predicen con precisión, buscando resolver el problema de estimar el peso de un cuy con algoritmos de regresión, siendo así una investigación aplicada. Mediante la observación se realizó la recolección de imágenes, capturando fotografías estandarizadas de los cuyes en una fosa y desde distintos ángulos, utilizando la cámara de un celular, de igual manera, se realizó un análisis documental para identificar modelos, arquitecturas y técnicas relevantes que aporten al desarrollo del modelo. Dando como resultado una herramienta útil y

accesible para el sector, con una correlación de 80.21% en el conjunto de validación. Este proyecto aporta a la investigación, puesto que, al ser una aplicación relativamente nueva en el sector, abre las posibilidades a ampliar el uso de la inteligencia artificial a otros proyectos, dando una guía sobre cómo realizar un sistema de pesaje, que, en este caso, a pesar de ser en cuyes, el procedimiento será similar pero aplicado a vehículos de carga, porque al final buscan estimar el peso de un objeto.

En la investigación de Maldonado (2021), desarrolla un modelo predictivo basado en aprendizaje profundo utilizando imágenes, para eliminar costos de utilizar una báscula y aumentar el número de pesajes de cerdos. Teniendo un enfoque cuantitativo, debido a que el aprendizaje profundo con base en imágenes requiere analizar las características a través de algoritmos de visión por computadora y al ser una investigación aplicada, busca resolver el problema de encontrar el peso de los cerdos a partir de regresión lineal y regresión basada en redes neuronales convolucionales (CNN). Donde para poder recolectar las imágenes usaron de la observación, para identificar los momentos en que se tomen las fotos y el peso, y así con una cámara y un PC se registraban las fotos con los pesos una vez que los cerdos pasaran por una zona específica donde colocan chips que envían señales en el momento específico. Al final resultó el mejor método de estimación de pesos el uso de CNN con un error absoluto medio (MAE) de 3.6 *kg* en 600 muestras, el 10% de los datos, correspondientes a la validación. Este proyecto aporta a la investigación, al ser un ejemplo que muestra el uso del aprendizaje profundo para estimar el peso de un animal, y que el mejor modelo haya sido las CNNs, brindando un buen estimado del peso real, generando una guía para saber cómo implementar la inteligencia artificial para el pesaje de algún objeto, puesto que el procedimiento no cambia, si no, las imágenes de entrada, dando así puntos fundamentales a valorar para la recolección de datos y modelos que dan mejores resultados.

Arbañil (2020), buscó mejorar el registro del peso de minerales transportados por camiones bitren de la minera Volcán en un área de carga, al desarrollar un sistema de pesaje por ejes. Con un enfoque cuantitativo, al analizar un sistema que mejore los indicadores, también corresponde a una investigación aplicada, al tener como objetivo solucionar el problema mediante la recopilación de información útil para su implementación, es una investigación de diseño experimental, donde identifica causas que afectan dentro del estudio y aquí el investigador manipula una variable

para controlar el resto de ellas, además, es una investigación descriptiva con un análisis de registros de mineral transportado. Para la recolección de datos empezó con registros históricos y análisis de documentos, buscando información en manuales, internet y página de Mettler Toledo, junto a una guía de preguntas para los operarios y conductores de los vehículos de carga. Llegando a concluir, sobre los pesos a circular por conjunto de eje y así elegir una plataforma metálica que se adapte a esas necesidades, siendo una balanza metálica multimodular, obteniendo un sistema de pesaje de camiones con capacidad de 120 toneladas con celdas de carga de 50 toneladas, y también llega a recomendar la adquisición de este sistema diseñado para que registre el material movilizado en el área de carga, para ser más precisos en la producción del mineral concentrado. Esta investigación aporta al proyecto, al mostrar la importancia de un sistema de pesaje para los camiones y que estos sepan cuánto producto manejan, pero para poder implementarlo es necesario saber qué tipo de sistema se ajusta mejor a los vehículos que se espera pesar y para esto se tiene que realizar una investigación sobre varios métodos y técnicas, el resultado es una báscula con ciertas características, entonces, mostrando que para poder implementar un sistema de pesaje, debe ser muy preciso y adaptable a la necesidad de la empresa.

Finalmente, el trabajo realizado por López (2020), se enfoca en mejorar los sistemas de pesaje, con el fin de reducir los tiempos de registro y permitir la escalabilidad en diferentes plataformas sin afectar el núcleo del modelo de negocio, implementando patrones de diseño en el sistema a desarrollar. Esta investigación tiene un enfoque cuantitativo, no experimental, de tipo descriptiva correlacional. Aquí usó las siguientes técnicas para poder recolectar información: encuestas dirigidas a las personas involucradas en el registro de pesaje, análisis documental para identificar investigaciones similares, entrevistas para obtener datos relacionados con el registro de pesaje, y también, la observación directa del sistema de pesaje para detectar necesidades. Los instrumentos utilizados fueron cuestionarios y fichas de observación. Llegando a obtener en la encuesta una fiabilidad de 0.989 superior a 0.7 con variables fiables para llevar a cabo el estudio, llegando a concluir que, si hay una relación entre las variables 1 y 2, al mejorar el tiempo de respuesta en el registro del sistema de pesaje y que el 70 % de usuarios perciben mejora de integridad del registro del pesaje. Esta investigación aporta al proyecto una perspectiva de cambio, demostrando que, si puede mejorar los sistemas de pesaje, pues al implementar un sistema de patrones

en el proceso de pesaje mejora y se vuelve una actividad más eficiente, sin la necesidad de alterar los mecanismos del sistema, por lo tanto, se espera que al automatizar estos procesos se optimice aún más los sistemas de pesaje.

2.2. MARCO TEÓRICO

2.2.1. Teoría General de Sistemas

La teoría General de Sistemas según Bertoglio (1993) trata de una herramienta utilizada para dar explicación a fenómenos de la realidad y a su vez de la predicción de su comportamiento futuro, a partir de un análisis de totalidades e interacciones internas y externas que tienen con su medio. Teniendo como una totalidad a la realidad o sistema, esta teoría aborda dicha totalidad a partir de una visión integral con mecanismos interdisciplinarios, contrario a un enfoque reduccionista, donde la realidad es dividida en subsistemas analizados como ciencias diferentes que no logran dar una explicación. Entendiendo a un sistema como, partes que conforman un conjunto que se coordinan e interactúan con el fin de conseguir cualquier objetivo, formando un todo, donde para desarrollarse deberá cumplir algunos pasos, como se aprecia en la Figura 1, el primero sobre corrientes de entrada se encarga de dar los recursos necesarios al sistema, posteriormente pasa por un proceso de conversión donde los recursos recolectados podrán ser utilizados para desarrollar las funciones determinadas y así pasar a la corriente de salida, que el sistema exporta al medio como un producto final, pero no termina ahí, debe pasar por una comunicación de retroalimentación dando así información sobre el funcionamiento del sistema en relación al objetivo buscado y enviar correcciones de ser necesario.

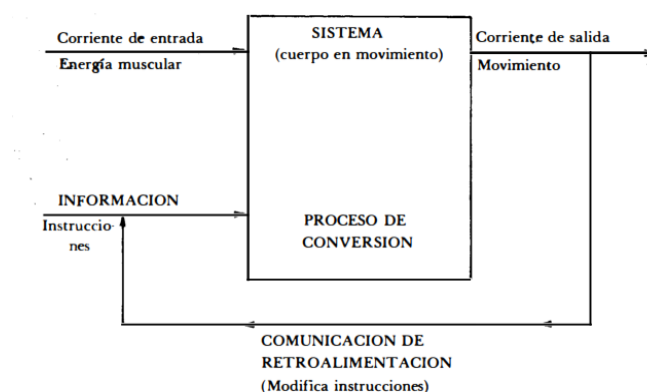


Figura 1. Representación de los elementos de un sistema de un cuerpo en movimiento.

Fuente: Introducción a la teoría general de sistemas, Bertoglio (1993)

Al hablar del diseño de un programa se menciona un sistema, pues todos los pasos que se requieren para su elaboración son subconjuntos que si no tienen una relación entre sí no lograrían nada, donde al tomar los pasos como un todo tienen el objetivo de lograr un programa capaz de pesar vehículos de carga pesada, logrando así entender el funcionamiento de dicho programa siempre que se entienda la relación que tiene cada una de las partes. A este proceso, se considera como un sistema, dado que cumple con los elementos para el desarrollo del mismo, en la corriente de entrada se ven las imágenes, que son los recursos en los cuáles las redes neuronales se basan para su entrenamiento, de ahí el proceso de conversión se da cuando las imágenes son tomadas como representaciones numéricas para poder ser utilizadas por un ordenador, así mismo darles un procesamiento adecuado dependiendo el algoritmo que se utilice, de ahí en la corriente de salida, será el resultado, de modo que el programa estime el peso de un vehículo de carga pesada al subirle una imagen, y para que esto sea lo más preciso posible las redes neuronales cumplen con el proceso de retroalimentación, debido a que, al no estimar bien una respuesta ajusta nuevamente los valores para que cada vez sea más precisa la respuesta.

2.2.2. Teoría de la Información

Para Correa (2008), la teoría de la información tiene su fundamento en la capacidad de información que pueda ser transmitida desde un receptor en el proceso de enviar mensajes, donde su mayor interés se da en la capacidad y fidelidad en el que a los sistemas de comunicación se les transmite la información, por lo que, el tipo de información no es de relevancia, mientras que cuánta información, si lo es. Dentro de los sistemas de comunicación se dan 3 análisis, el técnico, sobre los posibles problemas en relación con la fidelidad de información, el semántico, basado en el significado de mensaje e interpretación de este, y el pragmático, basado en la influencia del mensaje sobre una conducta, es por eso, que esta teoría surge para solucionar los problemas técnicos en el proceso de comunicación. Dichos problemas surgen sobre el modelo de transmisión de información, tal como está representado en la Figura 2, el cuál consta primero de una fuente de información, de un conjunto de mensajes posibles selecciona el deseado, luego pasa a un transmisor, donde la información es transformada en forma de señales que pueda ser enviada por el canal de comunicación, esta puede verse afectada por el ruido, que son distorsiones sobre la fuente de información, y al final las señales llegan al receptor, donde se decodifican tratando de obtener el mensaje original o una aproximación de este.

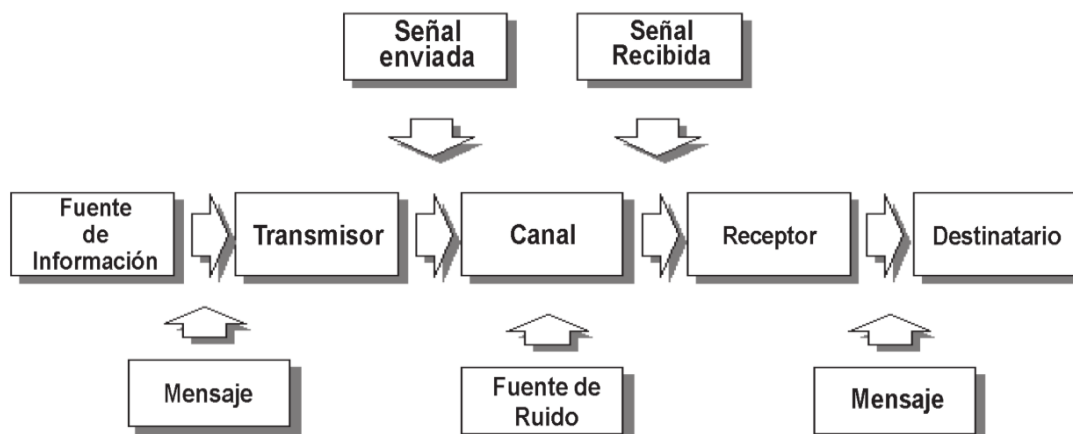


Figura 2. Modelo de comunicación

Fuente: Fundamentos de la teoría de la información, Correa (2008).

Dentro del programa de pesaje, las imágenes toman un rol muy importante, pues el entrenamiento del modelo se basa en estas como referencias porque cuentan con su peso real, con esta teoría se apoya la optimización del proceso de captura y transmisión de imagen, tratando de hacer una base de imágenes eficiente en cuanto a contenido visual, que la cantidad de imágenes sea la necesaria para obtener un buen rendimiento del programa y además sin distorsionar información al manipular o tomar las imágenes. En el proceso de desarrollo del programa se requerirá de, codificar las imágenes a números que puedan entender las redes neuronales y transmitirlos para su procesamiento en el entrenamiento, aquí también se debe analizar la capacidad de información que puede ser manejada por los modelos, para que el procesamiento de este sea más eficiente en recursos sin perder precisión en los resultados. Además, otro aspecto importante, es la fidelidad de información, en este caso el análisis del ruido que se pueda dar en el proceso de trabajar con imágenes, debido a que se puede tener mala calidad de imagen, zonas poco favorables de luz, entre otros aspectos que afecten al modelo cuando busque las características necesarias, siendo que estas no serán una buena referencia para el modelo, por lo que comprometerá en la precisión del mismo.

2.2.3. Inteligencia Artificial

Según Rouhiainen (2018) la Inteligencia Artificial (IA), trata del desarrollo de tareas que requieren razonamiento humano, mediante una máquina, esto mediante algoritmos que aprenden de datos y con estos toman decisiones, teniendo muchas ventajas en relación con las personas, debido a que son máquinas y no requieren de descanso, asimismo, el error en comparación con los humanos es muy bajo, también

se considera que la IA puede trabajar con gran cantidad de información. Teniendo aplicaciones en gran cantidad de áreas, dado que, ya puede realizar tareas pensadas como únicas de las personas, pudiendo llegar a tener actividades como reconocimiento de imágenes, clasificación, procesamiento de datos en pacientes, predicción, entre otras. Así llegando a tener un impacto social y económico. Además, es considerada una rama de las ciencias de la computación encargadas de desarrollar sistemas cada vez más complejos e inteligentes, buscando imitar la inteligencia humana, con el objetivo de superarla.

2.2.4. Aprendizaje automático

El aprendizaje automático (*machine learning*) para Fauland (2018) utiliza un algoritmo que, "se alimenta de información que debe analizar y reconocer para obtener un resultado específico" (p. 6), estos algoritmos trabajan en la detección de patrones sobre los datos que se suministran, para darles una clasificación a estos o realizar predicciones. De igual manera, el *machine learning* nace del interés por explorar si una computadora puede ir más allá de seguir instrucciones predefinidas y llegar a "razonar", en este enfoque, un elemento esencial es la capacidad de detectar patrones significativos, los cuales suelen encontrarse en los datos y sus relaciones con otros eventos. No obstante, su efectividad para abordar problemas concretos dependerá del contexto y las características particulares de cada situación. (Fauland, 2018). El aprendizaje automático puede ser utilizado con base en tres subconjuntos:

2.2.4.1. Aprendizaje supervisado

Según Rouhiainen (2018) "En el aprendizaje supervisado, los algoritmos usan datos que ya han sido etiquetados u organizados previamente para indicar cómo tendría que ser categorizada la nueva información" (p. 21). Por esta razón, en este método, es necesario que una persona capacitada intervenga en el etiquetado de datos.

2.2.4.2. Aprendizaje no supervisado

Según Rouhiainen (2018) "En el aprendizaje no supervisado, los algoritmos no usan ningún dato etiquetado u organizado previamente para indicar cómo tendría que ser categorizada la nueva información, sino que tienen que encontrar la manera de clasificarlas ellos mismos." (p. 21). Dado esto, aquí los modelos son los encargados de buscar la clasificación, siendo así que no requieran que el análisis humano intervenga.

2.2.4.3. Aprendizaje de refuerzo

Según Rouhiainen (2018) "El aprendizaje por refuerzo, los algoritmos aprenden de la experiencia. En otras palabras, tenemos que darles «un refuerzo positivo» cada vez que aciertan" (p. 21), al usar estos algoritmos las personas irán analizando el comportamiento en los resultados e irá calificando la respuesta desde su conocimiento.

En la Figura 3 se observan los 3 tipos de aprendizaje automático.



Figura 3. Tipos de aprendizaje automático

Fuente: "Inteligencia Artificial, 101 cosas que debes saber hoy sobre nuestro futuro", Rouhiainen (2018)

2.2.5. Aprendizaje Profundo

El aprendizaje profundo (*Deep learning*), tiene un enfoque en el aprendizaje supervisado, además, de considerarse un subconjunto del aprendizaje automático, este aprende características de los datos por medio de una representación de capas sucesivas, esta cantidad de capas define la profundidad del modelo, que pueden ir desde decenas hasta cientos de capas, y estas casi siempre se aprenden por medio de redes neuronales que trata de simular la neurobiología, por lo que, son modelos inspirados en el cerebro. Una red profunda se la puede tomar como el obtener información en varias etapas, en estas etapas la información pasa por filtros y así cada vez sale información más pura (Chollet, 2018).

Así el *Deep learning*, parte de entradas con sus etiquetas que por medio de redes neuronales profundas analizan la información al pasar por la secuencia de capas, donde su aprendizaje se da por pesos asignados a las capas, que al asignar valores correctamente en las capas de la red, se entiende por aprendizaje, y con estos valores dan un resultado esperado, este proceso puede llegar a contar con hasta

millones de parámetros tomados en cuenta en la red. Este proceso de aprendizaje debe pasar por algunos pasos, mismos que se ven en la Figura 4, primero un trabajo de función de pérdida, que mide que tan alejada esta la respuesta de lo esperado y así capturar el desempeño que ha tenido, luego pasa por una retroalimentación con la puntuación que obtiene en el desempeño, pasando a un ajuste de los valores en los pesos y así reducir la puntuación de pérdida, esto realizado por un optimizador, usando un algoritmo de retro propagación y es este algoritmo el centro de los modelos de aprendizaje profundo. Entendiendo esto, cuando se inicia el funcionamiento de la red, se designan valores aleatorios, por lo que, los pesos se van ajustando, tratando de minimizar la función de pérdida, esto con cada dato de entrada procesado, repitiendo este ciclo las veces necesarias, para tratar de tener una función de pérdida mínima, puesto que esto dará un resultado lo más cercano posible al real (Chollet, 2018).

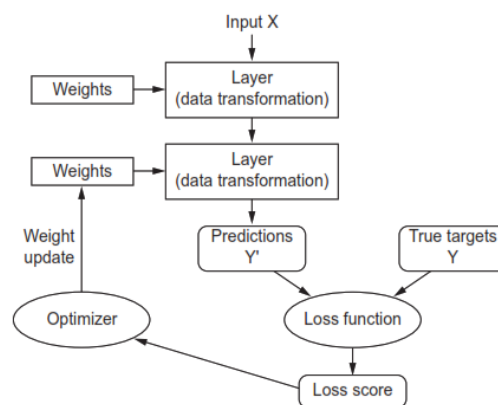


Figura 4. Procesamiento de datos en una red neuronal profunda
Fuente: Deep Learning with *Python*, Chollet (2018)

2.2.5.1. Redes Neuronales

Según Basogain (2008), las redes neuronales son el producto de los científicos de tratar de modelar el funcionamiento humano, es por eso que se dice que están inspiradas en las redes neuronales biológicas, estas cumplen con características como lo es, el aprendizaje, que lo hace con base en las entradas entregadas, el generalizar, que son las salidas las cuales cuentan con un cierto margen de errores, y el abstraer, al ser redes capaces de encontrar características en conjuntos no relacionados directamente. Estas redes neuronales artificiales se conforman de un elemento procesador, el cual cuenta con varias entradas y tiene una combinación que generalmente se da con sumas básicas, esta se modifica al pasar por una función de transparencia y este valor sale de este elemento procesador, de igual manera puede

partir hacia otras neuronas, siendo así que las salidas de unas neuronas sean las entradas de otras.

2.2.5.2. Entrenamiento de redes neuronales

Las redes neuronales tienen una gran capacidad de aprendizaje, pero, depende de cómo se realice el entrenamiento, para que se pueda obtener los mejores resultados, teniendo en cuenta lo que dice Basogain (2008) sobre el entrenamiento:

El objetivo del entrenamiento de una ANN es conseguir que una aplicación determinada, para un conjunto de entradas produzcan el conjunto de salidas deseadas o mínimamente consistentes. El proceso de entrenamiento consiste en la aplicación secuencial de diferentes conjuntos o vectores de entrada para que se ajusten los pesos de las interconexiones según un procedimiento predeterminado. Durante la sesión de entrenamiento los pesos convergen gradualmente hacia los valores que hacen que cada entrada produzca el vector de salida deseado. (p. 17)

Por lo tanto, para poder tener un buen entrenamiento es importante alimentar a las redes neuronales con una gran cantidad de datos, junto con su valor que se desee analizar y así tener un buen procesamiento de los datos en el entrenamiento.

2.2.5.3. Redes Neuronales Convolucionales

Este tipo de redes son un caso especial de las redes neuronales, puesto que son muy similares, al estar formadas por neuronas con pesos y sesgos que se pueden aprender, pero estas redes son capaces de trabajar con entradas iguales a una imagen, con las convoluciones se capturan la invariancia, al ser los filtros independientes de la ubicación, además, cuenta con una secuencia de capas, que transforma las activaciones o salidas a través de funciones diferenciables. Este tipo de redes trabaja con arquitecturas que en su mayoría contienen la capa de convolución, otra de agrupación y las conexiones entre capas, se caracterizan porque, las capas extraen características, son tipo capas de reducción, de dimensionalidad y de clasificación (Teuwen y Moriakov, 2020).

2.2.6. Métricas de rendimiento

Los modelos de IA, tras el entrenamiento requieren ser validados, tal como menciona Camarillo et al. (2013), "La validación de modelos es la etapa más importante tanto en el modelado como la estimación de parámetros de modelos de sistemas físicos" (p.111), estos permiten conocer el comportamiento de los modelos, tras analizar con

datos de prueba y estimaciones con base a la misma entrada, obteniendo errores de medición y permitiendo observar el rendimiento del modelo con respecto a un sistema real.

2.6.6.1. Error Absoluto Medio

Según Camarillo et al. (2013), "El error medio absoluto o MAE por sus siglas en inglés, mide la magnitud promedio del error entre los datos medidos y los datos estimados por el modelo" (p. 112), permitiendo facilidad de interpretación del error promedio, a partir de las mismas unidades de la variable objetivo, siendo una métrica clara para validar datos, esta se la obtiene a través de la fórmula:

$$MAE = \left(\frac{1}{n}\right) \sum_{i=1}^n |y_i - \hat{y}_i|, \text{ donde,}$$

y_i = valor medido y

\hat{y}_i = valor estimado

2.6.6.2. Error Cuadrático Medio

Para obtener métricas analicen mejor los errores más graves, se tiene al MSE, pues según Camarillo et al. (2013), "pondera las estimaciones que están más alejadas del valor medido [...] su resultado está en unidades al cuadrado de la variable medida" (p. 113), siendo así que penaliza aquellos errores más grandes, de manera útil al querer evitar este tipo de errores críticos, para poder ser sancionados eficientemente. Para obtener el valor de esta métrica se usa la fórmula:

$$MSE = \left(\frac{1}{n}\right) \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

2.6.6.3. Raíz de Error Cuadrático Medio

Esta métrica, con base en Camarillo et al. (2013), corresponde a medir el valor medio cuadrático de error, mayormente usado en la validación de modelos físicos, donde se obtiene como resultado las unidades de la variable objetivo, manteniendo la ponderación de predicciones alejadas del valor medio. Siendo esta una opción para mantener buena interpretación con las unidades originales de las variables y combina con la penalización de errores grandes, esta métrica se obtiene a través de la función:

$$RMSE = \sqrt{\left(\frac{1}{n}\right) \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

2.2.7. Lenguajes de programación

2.2.7.1. Python.

"Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML)." (Amazon Web Services, 2024), siendo un lenguaje adecuado para trabajar con diferentes modelos de IA, de manera apropiada.

2.2.7.2. Tensor Flow.

Para poder hacer uso de modelos de aprendizaje automático, es importante una herramienta que cuente con paquetes para poder utilizar estos modelos como lo es el *framework* de TensorFlow, según TensorFlow (2024), el *framework*, "facilita la creación de modelos de aprendizaje automático para computadoras de escritorio, dispositivos móviles, la web y la nube, sin importar si eres principiante o experto", siendo así posible, el uso de diversos recursos para el desarrollo de modelos de IA.

2.2.8. Pesaje en el transporte de mercancías

El saber el peso real de la carga es importante dentro de la logística, para Bascosta (2024), "Se refiere a la medición exacta del peso de la carga transportada, asegurando que los datos sean confiables y consistentes" (p. 1). Este proceso es importante para actividades como el diseño de rutas, cálculo de costos, seguridad vial y cumplir con regulaciones.

2.2.8.1. Tipos de básculas

2.2.8.1.1. Básculas de Plataforma.

Según Bascosta (2023), estas básculas son las más comunes, se pueden encontrar en carreteras, puertos y almacenes, diseñada con una plataforma plana y resistente, en esta, el camión se posiciona encima para pesarlo, están diseñadas para una gran capacidad de carga, resaltando su precisión en el pesaje, adaptables a cualquier tipo de vehículos y además están construidas con materiales resistentes por su alto tráfico.

2.2.8.1.2. Básculas de Ejes.

Bascota (2023), dice que estas básculas miden el peso de los vehículos en los ejes del camión, están hechas para garantizar las regulaciones de carga por eje, puesto que miden el peso individual de cada eje y, por lo tanto, son muy eficientes para cumplir con las normativas sobre pesos y dimensiones. Siendo muy útiles para optimizar la

carga dentro del vehículo, al distribuirla con mayor precisión por cada eje, mejorando en estabilidad y maniobrabilidad en el transporte.

2.2.8.1.3. Básculas portátiles

Este tipo de básculas tienen facilidad en el transporte por ser portátiles, al ser fabricadas de materiales livianos y resistentes, tienen la capacidad de ser instaladas de modo rápido y sencillo, al no necesitar mucha preparación cuando se las instala en el suelo, y complementando esas características cumplen con ser muy confiables y precisas en las medidas, brindando flexibilidad para su uso sobre diversos escenarios y un ahorro de espacio, puesto que se pueden guardar en zonas de poco espacio (Bascota, 2020).

2.2.9. Vehículos de transporte de carga

2.2.9.1. Definiciones de los vehículos

Tabla 1. Definiciones utilizadas sobre las características de los vehículos de carga pesada.

Definiciones consideradas en los vehículos de carga	
Camión	"Vehículo autopropulsado de carga, puede ser también utilizado para remolar. Su diseño puede incluir una carrocería o estructura portante"
Capacidad de carga	"Carga útil máxima permitida para el cual fue diseñado el vehículo."
Eje motriz	"Eje utilizado para transmitir la fuerza de tracción."
Eje no motriz	"Eje que no transmite fuerza de tracción."
Eje(s) direccional(es)	"Eje(s) a través del (de los) cual(es) se aplican controles de dirección al vehículo."
Eje(s) delantero(s)	"Eje(s) situado(s) en la parte anterior del chasis."
Eje(s) central(es)	"Eje(s) situado(s) en la parte central del chasis."
Eje(s) posterior(es)	"Eje(s) situado(s) en la parte posterior del chasis."
Eje simple	"Elemento constituido por un solo eje no articulado a otro, puede ser: motriz o no, Direccional o no, anterior, central o posterior."
Eje compuesto (Tándem)	"Elemento constituido por dos (2) ejes articulados al vehículo por dispositivo(s) común(es), separados 1,20 m y 1,60 m entre líneas de rotación extremas (centro de ejes extremos), pudiendo ser motriz, portante o combinado."
Eje compuesto (Trídem)	"Elemento constituido por tres (3) ejes articulados al vehículo por dispositivo(s) común(es), separados 2,0 m y 3,2 m entre líneas de rotación extremas (centro de ejes extremos), pudiendo ser motriz, portante o combinado."
Eje retráctil	"Eje que puede transmitir parte de la carga del vehículo a la superficie de la vía o aislarse de ésta mediante dispositivos mecánicos, hidráulicos o neumáticos."
Nodriz	"Parte de la carrocería, remolque o semi remolque diseñado exclusivamente para el transporte de vehículos ensamblados."

Definiciones consideradas en los vehículos de carga	
Peso bruto vehicular máximo	"Tara del vehículo más el peso de la carga máxima permitida."
Peso bruto vehicular	"Tara del vehículo más el peso de la carga que transporta."
Remolque	"Vehículo no autopropulsado con eje(s) delantero(s) y posterior(es) cuyo peso total, incluyendo la carga, descansa sobre sus propios ejes, y es remolcado por un camión o tractocamión."
Remolque balanceado	"Vehículo no autopropulsado en el cual el (los) eje(s) que soporta(n) la carga está(n) ubicado(s) aproximadamente en el centro de la carrocería portante, y es remolcado por un camión o tracto-camión"
Semirremolque (furgón, plataforma, tolva, tanque fijo)	"Vehículo no autopropulsado con eje(s) posterior(es), cuyo peso y carga se apoyan (transmiten parcialmente) en el tracto-camión que lo remolca."
Tara del vehículo	"Peso neto del vehículo con tripulación, provisto de combustible y equipo auxiliar habitual, en orden de marcha, excluyendo la carga."
Tractocamión (chuto-cabezal)	"Vehículo autopropulsado, diseñado para remolcar y soportar la carga que le transmite un semi-remolque a través de un acople adecuado para tal fin."
Unidad de Carga	"El remolque o semirremolque (furgón, plataforma, tolva, tanque fijo) registrado ante los organismos nacionales de transporte y aduana."
Vehículo de Carga	"Vehículo autopropulsado destinado al transporte de mercancías por carretera. Puede contar con equipos adicionales para la prestación de servicios especializados."

Nota. Información obtenida de la Decisión 491, Comunidad Andina (2001)

2.2.9.2. Pesos Brutos Máximos Permisibles

Los límites de peso y dimensiones para vehículos motorizados, remolques y semirremolques se especifican según el tipo de vehículo. Esto incluye la nomenclatura vehicular, la distribución máxima de carga por eje en relación con el peso máximo permitido por eje simple o conjunto de ejes, autorizado para la circulación en la Red Vial del país. Además, se describen la configuración según la disposición y número de ejes, el peso bruto máximo permitido según el tipo de vehículo, así como las dimensiones máximas de longitud, ancho y altura permitidas para transitar por la red vial del país (Ministerio de transporte y obras públicas [MTO], 2016), estas se presentan en la Figura 5 y Figura 6.

TIPO	DISTRIBUCIÓN MÁXIMA DE CARGA POR EJE	DESCRIPCIÓN	PESO MÁXIMO PERMITIDO (Ton.)	LONGITUDES MÁXIMAS PERMITIDAS (metros)		
				Largo	Ancho	Alto
2 D			7	5,00	2,60	3,00
2DA			10	7,50	2,60	3,50
2DB			18	12,20	2,60	4,10
3-A			27	12,20	2,60	4,10
4-C			31	12,20	2,60	4,10
4-0 OCTOPUS			32	12,20	2,60	4,10
V2DB			18	12,20	2,60	4,10
V3A			27	12,20	2,60	4,10
VZS			27	12,20	2,60	4,10
T2			18	8,50	2,60	4,10
T3			27	8,50	2,60	4,10
S3			24	13,00	2,60	4,10
S2			20	13,00	2,60	4,10
S1			11	13,00	2,60	4,10
R2			22	10,00	2,60	4,10
R3			31	10,00	2,60	4,10
B1			11	10,00	2,60	4,10
B2			20	10,00	2,60	4,10
B3			24	10,00	2,60	4,10

Figura 5. Tabla Nacional de Pesos y Dimensiones
Fuente: MTOP (2016)

TIPO	DISTRIBUCIÓN MÁXIMA DE CARGA POR EJE	DESCRIPCIÓN	PESO BRUTO VEHICULAR MÁXIMO PERMITIDO (toneladas)	LONGITUDES MÁXIMAS PERMITIDAS (metros)		
				Largo	Ancho	Alto
2S1			29	20,50	2,60	4,30
2S2			38	20,50	2,60	4,30
2S3			42	20,50	2,60	4,30
3S1			38	20,50	2,60	4,30
3S2			47	20,50	2,60	4,30
3S3			48	20,50	2,60	4,30
2R2			40	20,50	2,60	4,30
2R3			48	20,50	2,60	4,30
3R2			48	20,50	2,60	4,30
3R3			48	20,50	2,60	4,30
2B1			29	20,50	2,60	4,30
2B2			38	20,50	2,60	4,30
2B3			42	20,50	2,60	4,30
3B1			38	20,50	2,60	4,30
3B2			47	20,50	2,60	4,30
3B3			48	20,50	2,60	4,30

Figura 6. Tabla nacional de pesos y dimensiones posibles combinaciones
Fuente: MTOP (2016)

2.2.9.3. Distribución de carga

La distribución de la carga debe estar uniforme a lo largo y ancho de la superficie útil del vehículo, y en caso de no ser posible, no colocar con sobrecarga el centro de gravedad entre ejes (Comunidad Andina, 2001).

III. METODOLOGÍA

3.1. ENFOQUE METODOLÓGICO

3.1.1. Enfoque

El desarrollo de esta investigación se realizó bajo un enfoque cuantitativo, ello se debe al análisis que se les da a los datos, que son imágenes representadas numéricamente y subidas a un modelo de inteligencia artificial para que aprenda a reconocer características de vehículos de carga pesada y relacionarlas con el peso real de estos vehículos, de igual manera, Hernández et al. (2014), explica que , “se analizan las mediciones obtenidas utilizando métodos estadísticos, y se extrae una serie de conclusiones respecto a la o las hipótesis” (p. 4), debido a esto, los resultados fueron evaluados mediante métricas estadísticas para asegurar la precisión y confiabilidad, permitiendo evaluar el desempeño del programa y así llegar a una conclusión sobre la hipótesis planteada.

3.1.2. Tipo de Investigación

En el trabajo se emplean los tipos de investigación documental, descriptiva, explicativa y de campo. Con relación al primer tipo de investigación planteada, según Bernal (2010), se trata de un, “análisis de la información escrita sobre un determinado tema, con el propósito de establecer relaciones, diferencias, etapas, posturas o estado actual del conocimiento respecto al tema objeto de estudio” (p. 111), debido a esto, se logró identificar literatura y proyectos relacionados con la inteligencia artificial y visión por computadora, lo cual permitió recopilar la información necesaria que sirvió de base para el desarrollo de este proyecto. Además, se adquirieron conocimientos que facilitaron la estructuración del marco teórico y aportaron información relevante sobre el uso de básculas para vehículos de carga pesada dentro del sector del transporte.

El tipo de investigación descriptiva tiene “la capacidad para seleccionar las características fundamentales del objeto de estudio y su descripción detallada de las partes, categorías o clases de ese objeto” (Bernal, 2010, p.113). Por consiguiente, se buscó detallar el proceso sobre el diseño del programa, su funcionamiento, los parámetros tomados en cuenta en la recolección de datos y las técnicas utilizadas

dentro del entrenamiento, entre otros aspectos. También, se realizó una descripción sobre cómo se lleva a cabo el pesaje con básculas convencionales y como se la realiza con la implementación del programa.

En la investigación explicativa, según Hernández et al. (2014) su alcance, “está dirigido a responder por las causas de los eventos y fenómenos físicos o sociales. Se enfoca en explicar por qué ocurre un fenómeno y en qué condiciones se manifiestan, o por qué se relacionan dos o más variables” (p. 98). De este modo, se establecieron las relaciones causales sobre cómo la inteligencia artificial logra estimar el peso de un vehículo, con base en el conjunto de imágenes previamente entregado, y así mismo, establecer cuáles son los efectos que tiene el programa al implementarlo en los sistemas de pesaje de los vehículos de carga pesada.

Y, por último, la investigación de campo, “es aquella que se realiza en el mismo lugar y en el tiempo donde ocurre el fenómeno (...) su objetivo es levantar la información de forma ordenada y relacionada con el tema de interés” (Arias y Covinos, 2021, p.67). Por esta razón, se buscó el lugar del sistema de pesaje de los vehículos, para poder obtener las fotos en el momento en que realiza el pesaje y así etiquetar las imágenes con el peso real, así organizando los datos con los valores necesarios para el entrenamiento del programa, además, de analizar el entorno donde se realiza el proceso y en el momento que se lleva a cabo.

3.2. HIPÓTESIS

H_0 : Un programa de pesaje basado en imágenes no permite optimizar el proceso de pesaje de vehículos de carga pesada.

H_1 : Un programa de pesaje basado en imágenes permite optimizar el proceso de pesaje de vehículos de carga pesada.

3.3. DEFINICIÓN Y OPERACIONALIZACIÓN DE LAS VARIABLES

3.3.1. Variables

Variable Independiente: Programa con inteligencia artificial

Variable Dependiente: Sistemas de pesaje de vehículos de transporte de carga pesada.

3.3.2.1. Variable Independiente

Un programa con inteligencia artificial, para Rouhiainen (2018), se refiere a un sistema que permite a las máquinas aprender con datos y posteriormente tomar decisiones, utilizando algoritmos.

3.3.2.2. Variable Dependiente

Los sistemas de pesaje de vehículos de transporte de carga pesada, según Salguero (2003), son sistemas que utilizan básculas, para el control de pesos de los vehículos de transporte, usados en el control de despacho, tanto en la entrada como en la salida de mercancías.

3.3.2. Operacionalización de Variables

Tabla 2. Operacionalización de las variables para el tema "Diseño de un programa para el pesaje de vehículos de carga pesada a partir de imágenes con Inteligencia Artificial"

Variable	Dimensión	Indicadores	Técnica	Instrumento
Independiente: Programa con inteligencia artificial.	Entrenamiento del modelo	Arquitectura de red neural seleccionada	Análisis documental	Ficha de documental
		Conjunto de datos	Observación	Ficha de observación
		Precisión del modelo	Análisis documental	Ficha documental
	Rendimiento del modelo	Tiempo de respuesta del programa	Identificación de características relevantes de las imágenes	
Adaptabilidad		Exactitud del modelo		
		Capacidad de estimar el peso en diferentes condiciones	Facilidad de implementar en entornos reales	Análisis documental

Variable	Dimensión	Indicadores	Técnica	Instrumento
		Compatibilidad con las normativas		
Dependiente: Sistemas de pesaje de vehículos de transporte pesado	Desempeño de los sistemas	Tiempo de operación de pesaje	Observación	Ficha de observación
		Capacidades máximas de las básculas	Análisis documental	Ficha documental
	Implementación de básculas	Exactitud de las básculas		
		Infraestructura necesaria para la instalación		
		Impacto del uso de básculas en el sector del transporte.	Entrevista	Guía de preguntas
	Costos	Precios de básculas Frecuencias de mantenimiento Costos operativos	Análisis Documental	Ficha documental

3.4. MÉTODOS UTILIZADOS

3.4.1. Método hipotético-deductivo

Según Bernal (2010) el método hipotético-deductivo "Consiste en un procedimiento que parte de unas aseveraciones en calidad de hipótesis y busca refutar o falsear tales hipótesis, deduciendo de ellas conclusiones que deben confrontarse con los hechos."(p. 60), por esta razón, permite refutar la hipótesis sobre el programa de pesaje, al analizar los datos del programa y su funcionamiento, permitiendo validar el diseño del programa y probar su rendimiento en comparación con los sistemas de pesaje tradicionales.

3.4.2. Método analítico-sintético

El método analítico-sintético "estudia los hechos, partiendo de la descomposición del objeto de estudio en cada una de sus partes para estudiarlas en forma individual (análisis), y luego se integran esas partes para estudiarlas de manera holística e integral (síntesis)" (Bernal, 2010, p. 60), esto ayuda en la investigación, debido a que se divide en varias partes, como lo es el entramado del programa y observar su desempeño, también, analizar las características de las básculas y sus costos

operativos, entre otros. De ahí, pasando a integrar estos aspectos para tener una visión general sobre el impacto del programa con respecto a los sistemas de pesaje tradicionales. Ayudando así, al análisis de las interacciones de los elementos y de la consolidación de los hallazgos.

3.4.3. Técnicas de Investigación

En la investigación para poder recolectar información se usó tres técnicas o herramientas que permiten recolectar datos al investigador, las cuales son: observación, análisis documental y entrevista.

3.4.3.1. Observación

“Este método de recolección de datos consiste en el registro sistemático, válido y confiable de comportamientos y situaciones observables, a través de un conjunto de categorías y subcategorías” (Hernández, 2014, p. 252), en la investigación, se emplea para evaluar directamente el desempeño del programa basado en inteligencia artificial y los sistemas tradicionales de pesaje de vehículos de transporte pesado. Este método permite registrar aspectos como el tiempo de operación, las condiciones en las que se lleva a cabo el pesaje con la báscula y el programa. La observación proporciona una base confiable para comparar y validar las métricas de rendimiento y adaptabilidad del programa desarrollado con respecto a los sistemas de pesaje tradicionales.

3.4.3.2. Análisis documental

Para Hernández (2014), este método “implica la revisión de documentos, registros públicos y archivos físicos o electrónicos” (p. 252), el análisis documental se utiliza para recopilar información relevante sobre normativas, estándares y procedimientos relacionados con el pesaje de vehículos de transporte pesado, también, ayudando en la recolección de datos sobre los resultados del entrenamiento de modelo, para poder realizar el cálculo de las métricas de rendimiento. Además, permite examinar estudios previos, estadísticas y datos técnicos sobre sistemas de pesaje tradicionales y sus costos operativos.

3.4.3.3. Entrevista

Para Hernández (2014), las entrevistas “implican que una persona calificada (entrevistador) aplica el cuestionario a los participantes; el primero hace las preguntas a cada entrevistado y anota las respuestas” (p. 233), esta técnica resulta

esencial en la investigación, lo que permite recopilar información directa y detallada de las percepciones, experiencias y conocimientos de los representantes del sector transporte en relación con el uso de básculas tradicionales y su impacto en la logística y el cumplimiento normativo, proporcionando una visión integral de los retos y beneficios asociados al pesaje vehicular. Asimismo, permite explorar posibles barreras para la implementación de soluciones basadas en inteligencia artificial, contribuyendo a diseñar estrategias más efectivas y alineadas con las necesidades del sector.

3.5. ANÁLISIS ESTADÍSTICO

El análisis estadístico se desarrolló a partir de una prueba T de *Student* apareada, la cual permite saber si existe o no diferencia significativa entre dos muestras de datos, para el caso de estudio, el primer valor serán los pesos estimados por el programa y el otro valor serán los valores reales, esto permite saber si el modelo da buenas predicciones, el desarrollo de la prueba de hipótesis se da en *Python*, tras entrenar el modelo y hacer predicciones del *dataset* de prueba, relacionado con el modelo 3, mismo que contiene 2851 datos.

Esto representa la primera parte, porque si hay diferencia significativa, resulta que el modelo no es eficiente, y, por lo tanto, se rechaza la hipótesis nula sobre las distancias significativas y así se aceptará la hipótesis nula general, que es sobre la optimización del modelo, y si el caso es que no hay diferencia significativa, se buscará hacer otra prueba de hipótesis que permita saber si el programa mejora los sistemas de pesaje. Para así rechazar la hipótesis nula general y decir que si se puede optimizar el proceso de pesaje.

3.5.1. Prueba T de *Student*

La distribución T de *Student* es usada para la resolución de problemas sobre la media de la población, u problemas de comparación, que permite determinar una gran diferencia entre dos muestras (Walpole et al, 2012). Siendo precisamente lo que se busca analizar, una comparación entre dos muestras del peso de los vehículos, para conocer si cuenta con diferencia.

3.5.1.1. Prueba T de *Student* Apareada

También conocida como prueba de hipótesis de dos muestras dependientes o relacionadas, Lind et al. (2012) menciona que la hipótesis tiene su interés en la distribución de la diferencia de los valores de dos observaciones, quedando una sola

muestra, en el que investiga si este valor es cero, siendo que si en ambas observaciones registran datos similares la media de la distribución de las diferencias será cero, pero caso contrario que una observación difiera en la media de las diferencias no será cero, para esto se usa μ_d como símbolo que indica la media poblacional de la distribución de las diferencias y se toma en cuenta que esta diferencia sigue una distribución normal y el estadístico de prueba sigue la distribución t , con la fórmula:

$$t = \frac{\bar{d}}{\frac{s_d}{\sqrt{n}}}, \text{ donde,}$$

existe $(n - 1)$ grados de libertad,

\bar{d} media de la diferencia entre las observaciones apareadas,

n número de observaciones.

Y la desviación estándar de las diferencias se obtiene a través de la fórmula de la desviación estándar, sustituyendo X por d (la diferencia entre las observaciones):

$$s_d = \sqrt{\frac{\sum(d - \bar{d})^2}{n - 1}}$$

3.5.1.2. Prueba de hipótesis con valor p

El valor p , según Lind et al. (2012), lo define como "Probabilidad de observar un valor muestral tan extremo o más que el valor observado, si la hipótesis nula es verdadera" (p. 346). En otras palabras, compara la probabilidad, valor p , respecto al nivel de significancia, así mismo, Lind et al. (2012), menciona que "Si el valor p es menor que el nivel de significancia, H_0 se rechaza. Si es mayor que el nivel de significancia, H_0 , no se rechaza" (p.346), permitiéndonos así, con este valor tomar la decisión sobre la H_0 , siendo una alternativa en la prueba de hipótesis.

3.5.1.3. Teorema central del límite

Para poder usar la prueba *T Student*, es importante tener datos que sigan una distribución normal, en esta investigación se tiene una cantidad de muestra de 2581 pesos, con base a lo mencionado por Lind et al. (2012), "El teorema central del límite hace hincapié en que, en el caso de muestras aleatorias grandes la forma de la distribución muestral de la media se aproxima a la distribución de probabilidad normal" (p. 279), muestra que si se puede hacer uso de esta prueba de hipótesis, al tener una gran cantidad de datos. Esto también se considera pues Lind et al. (2012), menciona que es posible requerir una muestra de 30 o más datos en los que se puede

observar normalidad. Al tener en cuenta esto, la cantidad manejada en este estudio es mucho mayor.

3.5.1.4. Análisis de correlación

Según Lind et al. (2012), este análisis representa la medición de la relación entre dos variables, esta asociación primero empieza observándose por un diagrama de dispersión, luego se calcula la fuerza de la relación en una medida cuantitativa, con el coeficiente de correlación, el cual describe esta fuerza en escala de intervalo o razón de dos conjuntos, esta se encuentra designada con la letra r , y puede tomar un valor entre -1 y $+1$, la fórmula para encontrar el coeficiente de correlación corresponde al siguiente:

$$r = \frac{\sum(X - \bar{X})(Y - \bar{Y})}{(n - 1)S_x S_y}$$

La representación de la fuerza y dirección del coeficiente de correlación se muestra en la Figura 7.

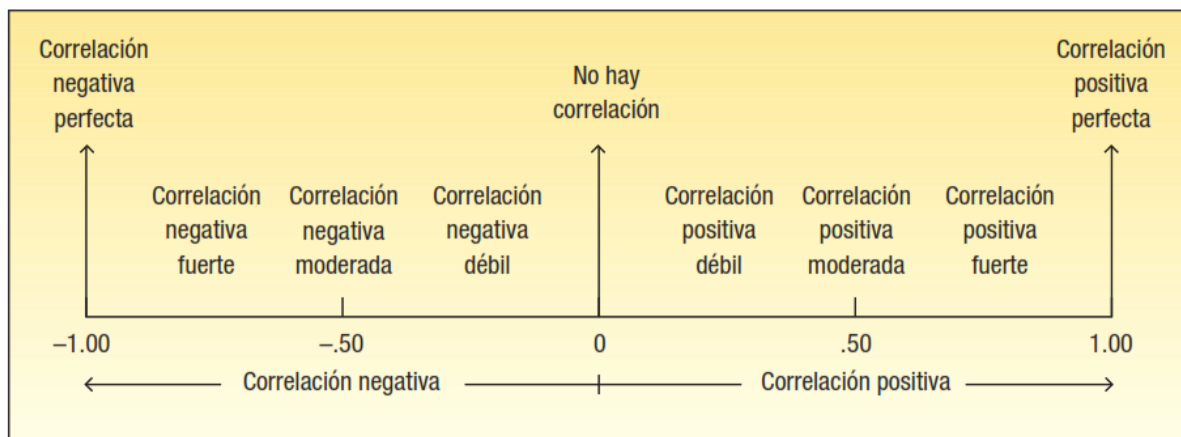


Figura 7. Representación de fuerza y dirección del coeficiente de correlación.

Fuente: Estadística aplicada a los negocios y la economía, Lind et al. (2012)

Con esto, se puede encontrar el coeficiente de correlación entre el peso de los vehículos reales y los estimados por el programa, para así ver la fuerza de relación que tienen entre sí, esperando una relación fuerte, esto porque las estimaciones buscan dar con los valores reales, en el caso que esta relación no lo sea, el programa simplemente no estaría analizando el peso correctamente.

3.5.1.5. Coeficiente de determinación

Con este estadístico, representado por R^2 , se busca analizar la variación entre las dos variables del peso, esto en base a lo mencionado por Walpole et al (2012), "se denomina coeficiente de determinación y es una medida de la proporción de la

variabilidad explicada por el modelo ajustado" (p. 407), llegando así a comprender la exactitud del modelo, y para encontrar este valor se usa la fórmula:

$$R^2 = 1 - \frac{SCE}{STCC} \text{ donde,}$$

$$SCE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \text{ y } SCE = \sum_{i=1}^n (y_i - \bar{y}_i)^2$$

IV. RESULTADOS Y DISCUSIÓN

4.1. RESULTADOS

4.1.1. Proceso de pesaje de los vehículos de carga pesada

El uso de básculas tradicionales en el pesaje de vehículos de carga pesada sigue una estructura simple como se muestra en la Figura 8, la cual se realiza en una estructura física, que consta de dos partes, la primera, una plataforma de hormigón y vigas de acero que va montada sobre celdas de carga, y la segunda, un *display* que se conecta con las celdas y muestra los pesos.

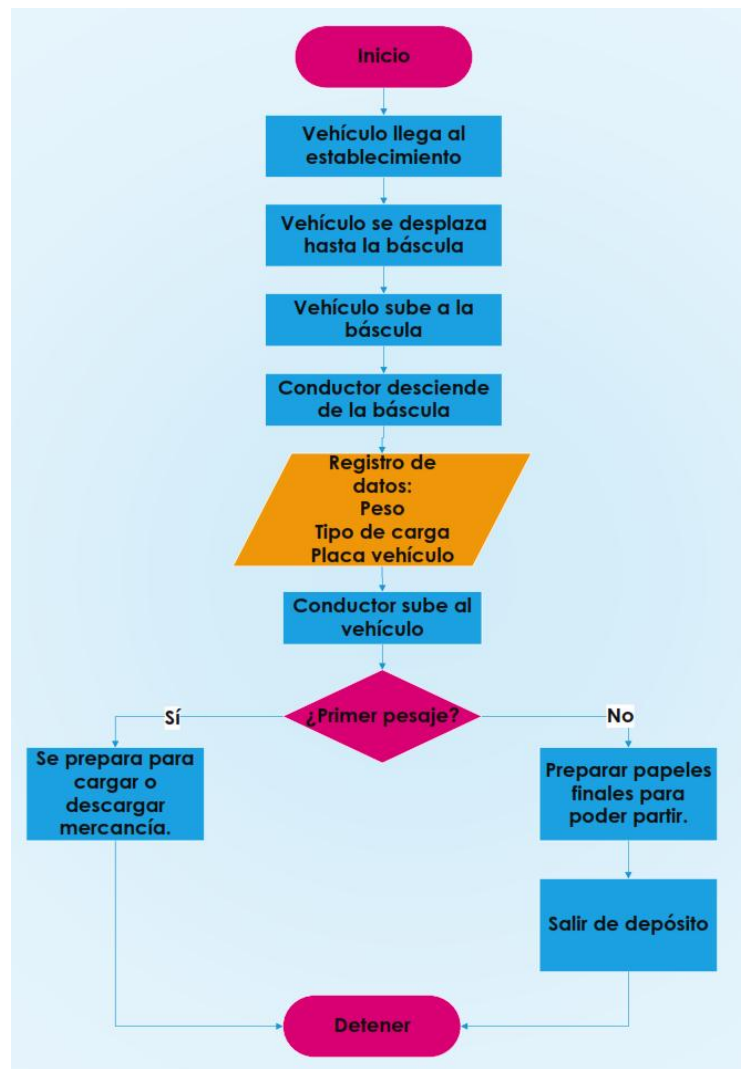


Figura 8. Diagrama de flujo del proceso de pesaje de vehículos.

Así, se ve en la Figura 8, que, para tomar el peso de un vehículo, es necesario que este se ponga encima de la plataforma, de ahí, el tiempo que se demora en hacer

este procedimiento puede variar, puesto que, las básculas pueden estar situadas en diferentes ubicaciones de las instalaciones.

El análisis sobre el comportamiento de los sistemas de pesaje se observó en 4 depósitos temporales de la ciudad de Tulcán, el primero con el apoyo de TRANSBOLIVARIANA C.A., y mediante el acuerdo entre la SENA – UPEC, con BODEGAS PRIVADAS TERAN CIA. LTDA., ADUANOR CIA. LTDA. y MIDECAR CIA. LTDA., las solicitudes que permitieron el ingreso a estas instalaciones se muestran en el Anexo 6 y 7, respectivamente.

La ubicación de las básculas en los depósitos no sigue patrones específicos, en algunos se encuentran en la entrada y en otras tienen que movilizarse hacia alguna zona dentro de las instalaciones. Por esta razón, no se puede tener certeza en un valor de tiempo para generalizar el tiempo de pesaje desde la llegada de un vehículo, más, sin embargo, si se puede ver los tiempos que se demoran los vehículos en la plataforma, desde su ingreso a la misma, donde el conductor desciende y luego de tomarse el peso vuelve a subir, y sale hacia la zona que se le designe, estos tiempos se tomó una muestra de 10 vehículo por cada tipo de vehículo, lo que permitió buscar el promedio que se demora cada tipo de vehículo sobre las básculas, tal como se ve en la Figura 9.

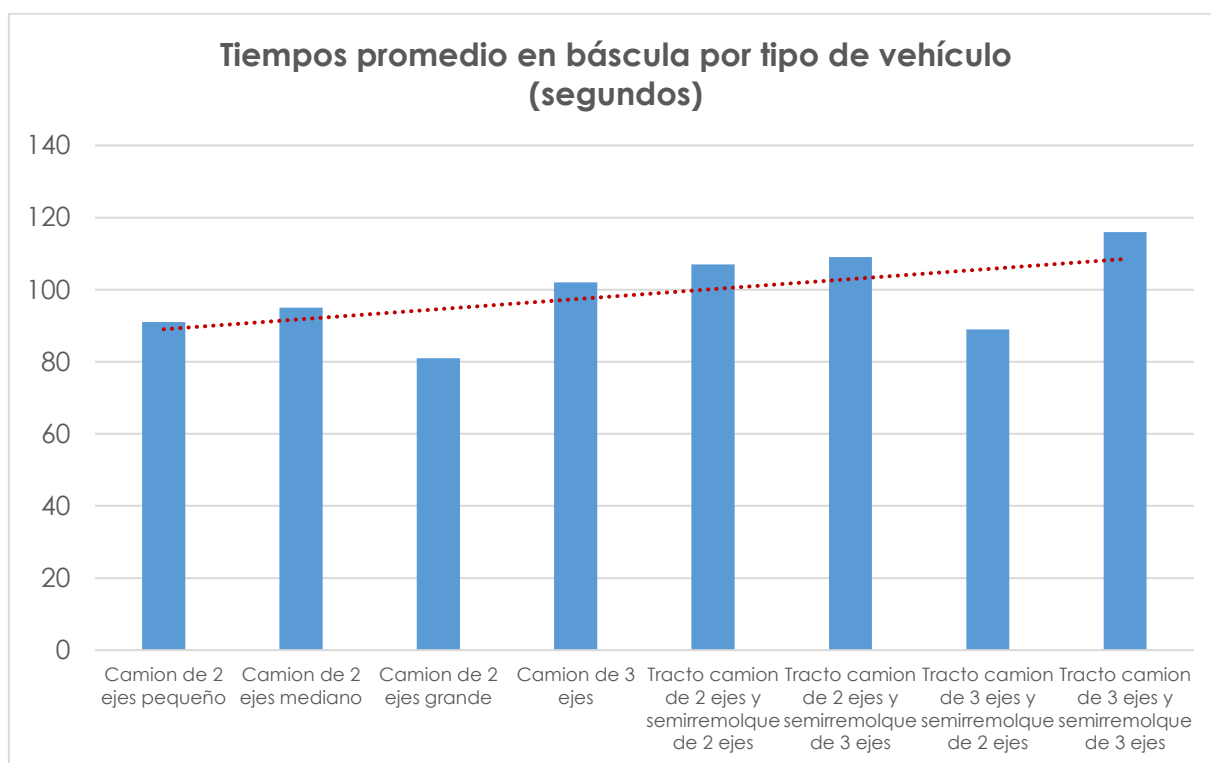


Figura 9. Tiempos promedio en báscula por tipo de vehículos de carga

La tendencia que muestra la Figura 9, representa un comportamiento en el que mientras más grande es el vehículo, más tiempo en promedio va a permanecer en las básculas. El gráfico se ordena en manera que los vehículos más pequeños se muestran primero y luego se muestra vehículos con dimensiones más grandes, existiendo un pequeño cambio en los promedios de tiempo, pues el camión de 2 ejes pequeño, tiene una media de 91 segundos sobre la báscula, mientras el tractocamión de 3 ejes y semirremolque de 3 ejes de 116 segundos, siendo así, un cambio de 25 segundos entre estos, que puede ser explicado por la dificultad que existen al incrementar las dimensiones de los vehículos para poder ser maniobrados, sin embargo, influyen más factores, como el tiempo que el operador se demore en tomar el peso, o si este debe tomar más datos del vehículo, actividades que no se centran en el pesaje, pero influye en su tiempo, y estos procesos son manejados de diferente manera por cada organización, dependiendo sus actividades.

Por lo tanto, no es un dato que permita un análisis más profundo, pues tal como muestra el tractocamión de 3 ejes y semirremolque de 2 ejes, tiene un valor de 89 segundos, siendo inferior inclusive al camión de 2 ejes pequeño, así mismo, se toma en cuenta que algunos depósitos permiten que los vehículos permanezcan un tiempo prolongados de más de 300 segundos, mientras realizan actividades relacionadas con el transporte, sin embargo, en otros, prefieren tomar únicamente el peso para evitar que las vibraciones puedan llegar a descalibrar las básculas.

4.1.1.1. Estructura de básculas tradicionales

Las básculas más comunes, son las de plataforma, por su gran capacidad de carga y precisión, mismas que se encontraron en los depósitos visitados, donde cada una correspondía a diferentes marcas, estructuras, diseño, etc., pero con el mismo funcionamiento, una plataforma de hormigón, sobre celdas de carga, que envían la información del peso a un *display* en una caseta.

Es por esta razón, que se toma como referencia a una sola báscula, que corresponde a la utilizada por el depósito temporal Transbolivariana C.A., esto permite identificar aquellas especificaciones importantes de estos sistemas sobre su implementación, mismas que se presentan en la Tabla 3.

Tabla 3. Especificaciones de Básculas tradicionales

	Obra Civil	Indicador Digital
Especificaciones	<ul style="list-style-type: none"> - Excavación y preparación de terreno. - Construcción de plintos y cadenas (Base a planos – 8 plintos). - Muros de aproximación - Vigas paragolpes longitudinales - Vigas paragolpes transversales - Instalación de sistema Toma Tierra - Tubería interna para cableado - Canales para evacuación de aguas - Ductos de cableados entra báscula y terminal - Ángulos metálicos en filos de rampa - Losa de asentamiento - Rampas de entrada - Losa de rodadura - Muros guías - Caseta de control - Celdas de carga de compresión tipo botella (6 unidades) 	<ul style="list-style-type: none"> - Marca: Berman Weighing Technology - Escala de medición de 10 kilos - ISO 9001 – 2000, NTEP, - OIML - Carcaza de acero inoxidable - Voltaje de operación 110 voltios - Control de: Fecha, hora, gross, neto, tara, observaciones, nombre de chofer, peso de vehículo entrando, peso de vehículo saliendo. - Temperatura de operación: -10 C to 40 C. - Valores en Kg, Lb, g, T - Capacidad: 60 000 <i>kg</i> - Peso de equipo: 10 <i>kg</i>

Las especificaciones mencionadas en la Tabla 3, corresponden específicamente a la báscula de Transbolivariana C.A., pero muestra todas las especificaciones fundamentales para su funcionamiento, con esto en cuenta, se mira que la infraestructura de estos sistemas en obra civil es muy robusta, y esto a su vez hace que sea muy costosa y difícil de implementar, pero estas especificaciones son necesarias en una báscula, pues deben soportar vehículos de gran capacidad, siendo aquellos los de tipos tractocamión de 3 ejes con semirremolque de 3 ejes, en este caso, tanto la plataforma y el *display*, pueden soportar una capacidad máxima de 80 000 kilos, aunque su rango útil es de 60 000 kilos, así mismo, su implementación al adquirir una báscula, puede llevar un aproximado de 30 días. Y también, según Benitez (2025), subgerente de Transbolivariana C.A., el costo aproximado de estos sistemas ronda entre los 30 000 y 35 000 dólares, en especificaciones similares.

4.1.1.2. Calibración de básculas

El Servicio Ecuatoriano de Normalización (INEN), es aquel organismo encargado de elaborar y adaptar normativas con respecto a la metrología, lo cual, en el ámbito de control de balanzas, el INEN (2023), realiza este proceso con base en la Resolución Nro. INEN-INEN-2023-0007-RES (Anexo 8), que se enfoca en el control de instrumentos digitales y análogos que tiene una función comercial.

De ahí que, tomando la misma báscula de referencia que se usó para observar sus componentes, también se analizó el proceso que conlleva la calibración periódica de estos sistemas. En este caso, realizaron una calibración en noviembre de 2024, por el laboratorio de Certificación Metrológica, CERTMETROL, en las instalaciones donde se encuentra la báscula, donde toman en cuenta las condiciones ambientales, en la que tuvo una temperatura de inicio en 22.2 °C y final de 19.6 °C, así mismo, el porcentaje de humedad relativa (%HR), la inicial fue de 31.2% y la final 59.0%.

Primero consta con una inspección visual, en este aspecto, el sistema no dispone de instrumentos para control de temperatura y humedad, pero si para control de voltaje, también observan que existen avisos preventivos, rompe velocidades al ingreso y salida de la balanza, cuentan con la plataforma y el instrumento indicador de lectura limpio, celdas de carga protegidas, el conjunto pintado y un drenaje alrededor de la balanza adecuado, sin embargo, no cuentan con un sistema de orden para entrada y salida de vehículos, además de no tener el área iluminada.

De ahí, para los ensayos de verificación utilizan pesas, principalmente de 500 kg, que permiten saber que las básculas se encuentran en parámetros correctos, los ensayos que realizaron se muestran en la Tabla 4.

Tabla 4. Ensayo de excentricidad y repetibilidad en báscula referencial.

Ensayos						
Excentricidad	Sentido	Lectura (kg)			Excentricidad máxima (kg)	Error máximo permitido (kg)
		Izquierda	Centro	Derecha		
		Entrada	26400	26400		
Retorno	26410	26410	26400			
Repetibilidad	Carga (kg)	Lectura kg			Error de Repetibilidad y lectura de cero (kg)	Error máximo permitido (kg)
		Primera	Segunda	Tercera		
		49 110	49110	49110		
Lectura de cero:	0	0	0	0	+- 10	

Los ensayos resultados en la calibración que se observan en la Tabla 4, determinan la eficiencia en las que operan estos sistemas, con respecto a la excentricidad, donde se refiere a que la báscula debe dar el mismo resultado sin importar en que parte de la báscula se sitúe el peso, el resultado o error máximo que tuvo fue de 10 kg, en el que el laboratorio de certificación metrológica, permite como máximo un error de más o menos 30 kg con respecto al valor real, siendo este el rango en el que pueden operar este tipo de básculas, con respecto a la repetibilidad, refiere a realizar un

pesaje 3 veces y en todas se debe obtener el mismo resultado, caso contrario la diferencia consta como un error, aquí se observa que para este caso las mediciones son las mismas, y una vez se quita el peso de la báscula el dispositivo muestra cero en cada caso, siendo así, en este ensayo el error fue de cero, pero al igual que el anterior el máximo con el que puede operar corresponde a más o menos 30 *kg*, y en el caso que en la báscula quede sin peso, el indicador puede mostrar un error de más o menos 10 *kg* respecto a cero. Luego de esto, el tercer ensayo corresponde a aumentar pesas en la báscula, y esta debe dar los resultados más precisos, estos resultados se muestran en la Tabla 5.

Tabla 5. Ensayo de carga de báscula referencia.

Carga (kg)	Lectura (kg)	Error (kg)	Error máximo permitido (kg)
0	0	0	+10
1000	1000	0	+10
2000	2000	0	+10
3000	3000	0	+10
4000	4000	0	+10
5000	5000	0	+10
6000	6000	0	+20
7000	7000	0	+20
8000	8000	0	+20
9000	9000	0	+20
10000	10000	0	+20
11000	11000	0	+20
12000	12000	0	+20
12780	12780	0	+20
13780	13780	0	+20
14780	14780	0	+20
15780	15780	0	+20
16780	16780	0	+20
17780	17780	0	+20
18780	18780	0	+20
19780	19780	0	+20
20780	20780	0	+30
21780	21780	0	+30
22780	22780	0	+30
23700	23700	0	+30
24700	24700	0	+30
25700	25700	0	+30
26700	26700	0	+30
27700	27700	0	+30
28700	28700	0	+30
29700	29700	0	+30
30700	30700	0	+30
31700	31700	0	+30
32700	32700	0	+30
33700	33700	0	+30
34700	34700	0	+30
49110	49110	0	+30

Con respecto al error que tienen las básculas, este tenderá a ser mayor mientras mayor sea el peso sobre la misma, como se ve en la Tabla 5, en la cual, muestra que

en este ensayo para esta báscula no tiene errores en ningún momento al ir aumentando el peso, por lo que, queda claro que este tipo de básculas, con buen mantenimiento y las calibraciones correspondientes son muy fiables, esto tomando en cuenta, que en el caso de tener errores, el mayor permitido es de más o menos 30 *kg* en pesos superiores a 20780 *kg*, de más o menos 20 *kg* en pesos inferiores a este valor pero superiores a 6000 *kg*, y de más o menos 10 *kg* en menores pesos. Con esto se comprende un nivel en el que se puede comparar a cualquier sistema de pesaje para vehículos para que pueda operar en los mismos parámetros de precisión.

Otro aspecto importante es tener en cuenta el costo que genera este proceso, donde el laboratorio de certificación metrológica, al realizar esta certificación, cotizó en USD 820 la calibración y certificación de balanzas de gran capacidad con un rango mayor a 60000 *kg*, y USD 55, por calibración y certificación de balanzas de mediana capacidad en rangos de 50 *kg* y 500 *kg*, a lo que suma un costo de USD 875, y a esto se incluye el IVA al 15%, que corresponde a USD 135.25, por tanto, se toma en cuenta que para realizar la calibración tiene un costo total de USD 1006.25. Y con esto, estas calibraciones por la misma organización, establece que el usuario debe de manera obligatoria recalibrar la balanza en intervalos apropiados, estos tiempos dependen mucho del uso que se le dé, pues a más uso, se deberá recalibrar más rápido, caso contrario sin tanto uso, el tiempo se puede aplazar, sin embargo, la sugerencia que hace el laboratorio es hacerlo semestralmente, y para el caso analizado, recomienda la fecha en mayo de 2025 y tomando en cuenta que la calibración fue realizada en noviembre de 2024, da un intervalo de 6 meses.

4.1.1.3. Costos asociados al sistema de pesaje

Con lo mencionado hasta el momento, los principales costos corresponden a la implementación de una báscula camionera y las calibraciones periódicas, de ahí, para su uso, una persona debe controlar el flujo de los vehículos, conforme llegan a las instalaciones que cuentan con estos sistemas, así mismo, son quienes toman los principales datos de los vehículos y con estos anotan el peso de los vehículos, es por esta razón, que se toma en cuenta un operador, el cual cuenta con un salario básico, y también, el consumo de energía del sistema, pues todos sus componentes son electrónicos, por tanto, se toma en cuenta un aproximado del consumo de un mes, teniendo en cuenta estos 4 valores principales, se muestran en la Tabla 6 los datos.

Tabla 6. Costos asociados con el uso de báscula.

Costos asociados	Monto (USD)	Frecuencia
Costo Inicial de báscula	30 000 - 35 000	Valor inicial
Calibración	1006.25	Semestral
Operario	470	Mensual
Energía eléctrica	8.15	Mensual

Los datos presentados en la Tabla 6, se obtuvieron a través del análisis documental (Anexo 4), así como la información de referencia sobre estructura y calibración de básculas. Estos costos son elevados, esta es una de las razones por la que, la mayoría de empresas optan por pagar por este servicio, y no por comprar una, esto por su precio inicial y costos de calibración que son los más altos, prefiriendo pagar el servicio, que puede variar entre 8 a 10 dólares, dependiendo el tipo de vehículo y la tarifa de cada empresa con estos sistemas, caso contrario pasa con los depósitos temporales, estos si deben contar con estas básculas para poder ofrecer el servicio y cumplir con las normativas de transporte de carga, siendo así, que pueden optar por prestar este servicio a empresas que lo requieran.

4.1.1.4. Espacio ocupado por báscula

La infraestructura de este sistema debe abastecer a vehículos de carga pesada con gran volumen de carga, que en el tiempo que se recolectó información, el tipo de vehículo con dimensiones más grandes, fueron los tractocamiones de tres ejes con semirremolques de 3 ejes. Nuevamente tomando la báscula de referencia, se hace un análisis del espacio que pueden ocupar estos sistemas, siendo así, se tomaron las medidas tanto de la base de concreto o fosa, la plataforma encargada de recibir los vehículos, la caseta de control y una zona por la cual las personas pueden movilizarse entre la báscula y la caseta caminando, estas medidas se representan en la Figura 10.

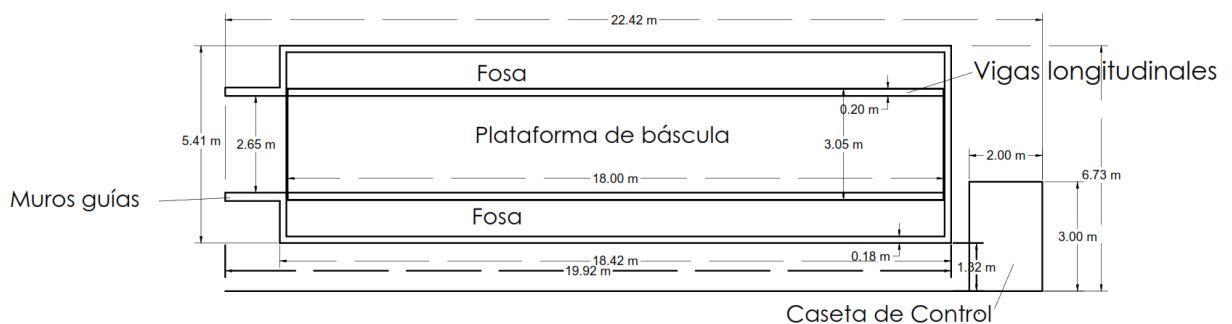


Figura 10. Medidas del espacio ocupado por báscula de referencia en metros (m).

De la Figura 10, las medidas que más representan son la de toda la báscula, pues desde los muros guías hasta la caseta de control, termina midiendo 22.42 *m* de largo y a lo ancho, incluyendo la caseta y el camino para movilizarse que mide 1.3 *m*, termina midiendo en total 6.73 *m*, esto generando un área de uso de 150.83 *m*², siendo este un referente de espacio que se requiere para poder implementar una báscula, otra parte importante a considerar es la fosa, pues es la cimentación sobre la que va a ir la plataforma, y por lo tanto, su superficie es la que debe estar preparada para soportar grandes pesos, esta corresponde a 18.92 *m* de largo y 5.41 *m* de ancho, dando un área 102.35 *m*², referente a tener en cuenta para el terreno que debe ser excavado y preparado con plintos, cadenas y los muros que lo rodean, también la plataforma de la báscula, pues esta va encima de las celdas de carga, mismas que se sitúan encima de la fosa, de las dimensiones de esta dependerá que tipos de vehículos puedan ingresar en la báscula, en este caso las medidas corresponden a 18 *m* de largo y 3.05 *m* de ancho, siendo las medidas suficientes para poder pesar vehículos de tres ejes con semirremolques de 3 ejes, lo cuál termina siendo un área total de 54.9 *m*², aunque en el ancho se resta las medidas de las vigas paracolpes longitudinales, que sirven como guía para ingreso del vehículo, estas al medir 0.20 *m* en cada lado, el espacio de ancho con el que cuenta la báscula para el ingreso de los vehículos es de 2.65 *m*, siendo suficiente para la mayoría de vehículos de carga pesada.

4.1.1.5. Normativa aplicable a las básculas

En el artículo 38 de la ley del sistema ecuatoriano de la calidad (2021) menciona que, el INEN, es el ente que formula regulaciones para los aspectos de uso, control y mantenimiento sobre, unidades de peso y medida, de aquellos dispositivos usados en el pesaje o medición, con el propósito de asegurar su exactitud. Por tanto, tomando esto en cuenta, las básculas de pesaje de vehículos son necesarias para cumplir con estas regulaciones. Así mismo, las actividades destinadas para estos prototipos, según el artículo 38 de la ley del sistema ecuatoriano de la calidad (2021) son:

- a) Realizar transacciones comerciales o para determinar el precio de un producto o servicio;
- b) Realizar las actividades que tengan relación con los propósitos de la Ley y de manera especial los relacionados con el servicio de salud; y,
- c) Realizar actos de naturaleza pericial, judicial o administrativa.

Para el caso de los sistemas tradicionales de pesaje de vehículos, tienen que determinar el peso para así poder brindar el servicio de transporte de los productos, y que las empresas sepan con certeza que cantidad de producto están adquiriendo, evitando conflictos entre las partes.

4.1.1.5.1. Infracciones

Según el artículo 53 de la ley del sistema ecuatoriano de la calidad (2021), sanciona por, "La comercialización, ofrecimiento, y/o exposición de mercadería, que sea expresada en unidades diferentes a las del Sistema Internacional de Unidades (SI) o que no cumplan con las tolerancias de peso o medida establecidas en los reglamentos técnicos pertinentes" (p. 18), con una multa pecuniaria, que puede llegar a los 15 salarios básicos unificados.

Además, el artículo 53 de la ley del sistema ecuatoriano de la calidad (2021), sanciona por, "el uso de indicaciones o etiquetas tendientes a desorientar o engañar a los compradores de mercaderías, en lo referente a peso, medida o expresión de número de unidades inexactas" (p. 19).

Con esto, los sistemas de pesaje se vuelven obligatorios para la comercialización de productos, considerando que su función es dar a conocer la cantidad de producto que se está negociando, y por esta misma razón es de suma importancia tener en cuenta las calibraciones periódicas para evitar este tipo de sanciones, pues al tener fallas en los pesos, pueden ser consideradas como mediciones que engañan a los compradores, así, al realizar los mantenimientos y mantenerse al margen de las regulaciones se evita gastos innecesarios que pueden afectar a la operatividad de las empresas.

4.1.1.5.2. Reglamentación de la CAN, límites de pesos y dimensiones

Según la Comunidad Andina (2001), es importante regular los pesos y dimensiones de los vehículos en el transporte internacional, "a fin de dar máxima seguridad y eficiencia en la prestación del servicio y preservar el patrimonio vial de los Países Miembros" (p.1). Siendo así de suma importancia los tipos de vehículos y cuáles son los pesos que estos tiene permitido manejar. Para esto, algunas configuraciones de vehículos de carga según la disposición y número de ejes son presentadas por el MTOP (Ministerio de Transporte y Obras Públicas), en la Figura 5, muestra los pesos máximos permitidos en toneladas para los diferentes tipos de vehículos de carga, mismos que se clasifican como camiones, tractocamiones, semirremolques y

remolques, junto al número de ejes con el que cuenten, así mismo en la Figura 6 se mira las diferentes configuraciones entre los mencionados, tractocamiones y semirremolques, y camiones remolcadores y remolques

Las básculas permiten pesar cualquier combinación posible de estas, siempre y cuando se encuentre en la medida de ancho y longitud de la báscula a usar, por ende, si la configuración del vehículo se vuelve muy larga, esta deberá usar básculas con dimensiones más grandes, que permitan que todo el vehículo ingrese en la plataforma, en la investigación no se trabajó con los camiones remolcadores y remolques, además, el tipo de vehículo más grande que se observó fue el tractocamión de tres ejes y semirremolque de 3 ejes, mismo que le permite tener un largo máximo de 20.50 m, ancho de 2.60 m y alto 4.30 m, junto a un peso bruto máximo permitido de 48 toneladas, en este aspecto la mayoría de básculas, en el caso de la analizada si permite el pesaje, en el caso del largo, al ser el largo de los vehículos menor a 18 m con respecto a los ejes,

4.1.1.6. Impacto de las básculas en el transporte de carga pesada.

Las básculas deben permitir conocer el peso de los vehículos con precisión y eficiencia, esto porque la industria requiere de ser optima en sus servicios, y al obtener mediciones lo más exactas posibles, se logra una buena logística, cumplimiento de regulaciones y seguridad (BASCOSTA, 2024).

4.1.1.6.1. Sobrecargas:

El uso de básculas previene el sobrecargar los vehículos con carga que sobrepasan sus especificaciones, evitando así, el desgaste prematuro o daño de la infraestructura vial, tal como lo menciona Moreno (2004), "el daño del pavimento, sin embargo, tiene tres peculiaridades: 1) es atribuible exclusivamente a los vehículos de carga; 2) crece muy bruscamente al aumentar los pesos que llevan, y 3) no se percibe de inmediato" (p. 3), ocasionando un gasto mayor para los encargados de dar mantenimiento a las vías, y mientras esto no se haga en zonas afectadas, afectará a otros conductores que deban circular por las mismas vías.

A su vez, el sobrecargar un vehículo, tiene un efecto sobre la seguridad vial, puesto que, como menciona Pinto (2021)," El sobrepeso en el vehículo involucra mayor fuerza para una misma aceleración, haciendo que la fuerza de impacto sea mayor en un supuesto de colisión frente a otros vehículos" (p. 169) , y también se vuelven un riesgo en curvas, así lo menciona Pinto (2021), "a mayor carga en una curva se

genera un dificultoso frenado lo que puede provocar en un volcamiento de vías, para esto es recomendable mantener una velocidad prudente" (p. 169), esto siendo no solamente un riesgo para la persona que conduce el vehículo, si no, se convierte en un peligro para otros conductores que circulen por la misma zona que este.

4.1.1.6.2. Optimización de carga:

Al conocer las medidas exactas de las cargas, es posible manipularlas para maximizar el rendimiento del espacio para toda una flota de vehículos, optimizando así los viajes, igualmente, esta información ayuda a gestionar no solo los vehículos, si no también almacenes, logrando tener una gestión de inventarios optima y así tener una buena planificación logística, logrando reducir costes de operación. Según Rosas (2025), operador logístico de Cetlog S.A., en la ciudad de Tulcán, tras una entrevista (Anexo 5), menciona que existen básculas fijas de plataforma, porque son las más eficientes, a pesar de eso, no representan optimización en los procesos, debido a que siguen siendo sistemas que llegan a fallar o por falta de energía estos no se pueden usar, pero son muy importantes, puesto que al hacer buen uso de estos sistemas se genera cargas homogéneas en los vehículos, evitando sobrepeso en los mismos y así los problemas anteriormente mencionados, otro aspecto importante que toman en cuenta es la normativa de la CAN, donde menciona que la mayoría de transportistas no se encuentran informados, y por tanto, basan la carga del vehículo en su propia experiencia, sobre cuánto puede cargar un vehículo, por ende, estas normativas son tomadas como un desafío, pues se deben cambiar desde el punto de vista cultural, donde la creencia sigue siendo que los vehículos pueden ser sobrecargados para poder sacar un margen de ganancia mejor, tanto para empresas de transporte que pueden ahorrarse contratar un vehículo más, o en los transportistas, que por un pago un poco más alto, acceden a llevar más carga, por ende, es importante usar las básculas, para optimizar recursos.

En este aspecto, en la región, aún no se logra optimizar las flotas vehiculares con respecto al uso correcto de las básculas, pues al no haber un control exigente, los transportistas y empresas de transporte, al querer mejorar el margen de ganancia llegan a sobrecargar los vehículos, sin tomar en cuenta que esto puede resultar en algo contrario, dado que los vehículos, exceden su uso y por lo tanto empiezan a fallar antes, llegando a sufrir daños estructurales de los vehículos, o de sus mecanismos, por tanto, se generan gastos altos, pero no de manera inmediata.

4.1.1.6.3. Impacto ambiental:

En este aspecto, BASCOSTA (2023), menciona que el uso de básculas evita la sobrecarga de vehículos, evitando así un consumo de combustible excesivo, esto debido a una correcta distribución de las cargas en los vehículos logrando una reducción de emisiones. Esto, también expuesto por Pinto (2021), "un vehículo cargado consume mayor cantidad de combustible, pero también ocasiona en las llantas inestabilidad exponiéndolas a una explosión y accidentes irreparables en las vías" (p. 169), con esto, queda claro que manejar un vehículo con sobrepeso es una muy mala decisión, tanto por el peligro al que se está expuesto, como por la mayor cantidad de emisiones que producen los vehículos. Así mismo, Rosas (2025), operador logístico de Cetlog S.A., menciona que es un problema más que económico, pues, al forzar el vehículo falla antes el motor, transmisiones, llantas e inclusive carrocerías, lo que hace que se deban botar estas partes dañadas y cambiar por nuevas, no optimizando los recursos y generando contaminación por componentes que no se pueden volver a usar.

Por estas razones, el uso de una báscula vuelve más eficientes los procesos logísticos, además, que son fundamentales para evitar sobrecargas y a su vez, tener seguridad en el transporte al mismo tiempo que se vuelve una práctica sostenible con el medio ambiente.

4.1.1.7. Problemas comunes en las básculas

Según Qichuang (2024), explica los errores comunes asociados a las básculas, que, al ser un equipo de trabajo, experimentan problemas en el tiempo, por lo cual, es crucial tener mantenimientos preventivos y realizar reparaciones de forma oportuna, para así mantener precisión y tener eficiencia, estos errores se observan más detalladamente en la tabla 7.

Tabla 7. Errores comunes en las básculas tradicionales

Error	Problema
<i>Inexactitud</i>	Las básculas para camiones pueden presentar inexactitud en el pesaje por problemas como la acumulación de residuos, una distribución desigual de la carga o errores en la calibración. Para su correcto uso es recomendable mantener la báscula limpia, también darle una calibración periódicamente siguiendo las indicaciones del fabricante, recurriendo a una recalibración profesional si se detectan fallos recurrentes. Asimismo, es fundamental que el camión esté correctamente centrado sobre la plataforma y que la carga se distribuya de manera equitativa.
<i>Celdas con mal funcionamiento</i>	Las celdas de carga son elementos esenciales, ya que se encargan de medir el peso con precisión. Si alguna de estas celdas presenta fallas, se pueden obtener mediciones incorrectas o, en casos extremos, un mal funcionamiento total del equipo. Su prevención, consta en una inspección frecuente de las celdas,

Error	Problema
	verificando que no presenten daños, corrosión o desgaste, y al detectar una celda defectuosa, se debe reemplazar de inmediato. Adicionalmente, es recomendable utilizar protecciones o cubiertas para resguardar estos componentes de factores como la humedad, productos químicos o acumulación de residuos.
<i>Problemas eléctricos</i>	Los problemas eléctricos son otra posible causa de fallos en las básculas de camiones, ya que estas dependen de componentes electrónicos para su funcionamiento. Entre los principales inconvenientes se encuentran el cableado en mal estado, las sobrecargas eléctricas y la filtración de agua. Para minimizar estos riesgos, se debe revisar periódicamente el cableado, asegurando que no haya conexiones flojas, desgastadas o corroídas, y reemplazarlas si es necesario. También es útil instalar protectores contra sobretensiones para evitar daños por picos de corriente y garantizar que los componentes electrónicos estén sellados correctamente para impedir la entrada de agua, lo que puede provocar cortocircuitos o corrosión.
<i>Base en mal estado</i>	La base de la báscula tiene un gran impacto en la precisión de las mediciones, si presenta grietas, hundimientos o erosión, puede generar lecturas incorrectas, y para evitar estas fallas, es importante inspeccionar con regularidad la estructura de la base en busca de daños y efectuar las reparaciones necesarias de inmediato. Además, se debe garantizar un adecuado sistema de drenaje alrededor de la báscula, esto por la acumulación de agua, que puede ocasionar erosión o hundimientos que comprometan su estabilidad y funcionamiento.
<i>Dispositivo indicador con mal funcionamiento</i>	El dispositivo que indica el peso muestra las mediciones realizadas por la báscula, por lo que, cualquier fallo en su funcionamiento puede generar lecturas incorrectas, para asegurar su óptimo desempeño, es recomendable revisar las conexiones entre el indicador y la báscula, verificando que estén firmes y libres de corrosión. También es importante comprobar la fuente de alimentación, resolviendo cualquier problema eléctrico que pueda estar afectando su operatividad.
<i>Desgaste de estructura</i>	El uso continuo de una báscula para camiones puede provocar el desgaste de sus componentes mecánicos, como la plataforma, rodamientos y juntas, lo que afecta su rendimiento y precisión. Siendo necesario, aplicar lubricación regularmente en las partes móviles, siguiendo las instrucciones del fabricante. Asimismo, es importante sustituir piezas desgastadas, como cojinetes o articulaciones, antes de que fallen por completo, evitando problemas mayores en el sistema de pesaje.

Nota. Información tomada de Qichuang (2024).

En la Tabla 7, se observa como cada parte del sistema de pesaje es fundamental para poder brindar una correcta medición del peso, y para que este sea siempre el más preciso, es necesario dar mantenimiento o reparación lo más pronto posible a cada parte, una vez se encuentren fallas en el pesaje, esto debido a que cualquier parte del sistema puede provocar errores de medición o inclusive provocar el deterioro de todo el conjunto.

4.1.2. Métodos de IA

Según Ramírez y Chacón (2011), menciona que las redes neuronales artificiales son las más usadas en segmentación de formas, colores, para encontrar texturas, movimientos y bordes de las imágenes, debido a sus formas irregulares, no tienen

patrones tan repetibles, por lo que, no se pueden extraer esas características con métodos clásicos para procesamiento de imágenes. Para estimar el peso se requiere que el modelo sea capaz de encontrar las características que estén relacionadas, por ende, las CNNs son la opción más adecuada para este trabajo, teniendo en cuenta su usabilidad en clasificación de objetos, donde muestra que es capaz de encontrar características únicas de los objetos y a partir de estos realiza la predicción con un alto porcentaje de acierto, este proceso se lo puede visualizar en la Figura 11.

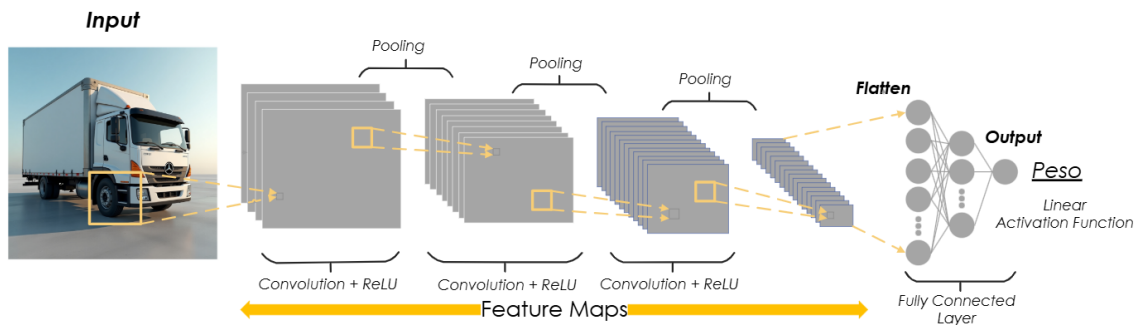


Figura 11. Estructura de una red neuronal convolucional

Siendo así, para el caso de estudio la CNN, sigue la estructura mostrada en la Figura 11, donde buscará patrones únicos a partir de convoluciones y mediante una red neuronal buscará la solución más pertinente, cambiando la estructura final a regresión lineal, pues buscará el valor lo más cercano posible al correcto, el cuál ha sido obtenido directamente de una báscula, asegurando la calidad de los datos.

4.1.2.1. Análisis de arquitecturas CNN para implementar

Las CNNs por si solas tienen una gran eficiencia, pero son propensas a cometer errores o sufrir sobreajuste, donde, He, et al. (2016), menciona mayor dificultad de entrenamiento en redes neuronales más profundas, a lo cual, presentaron redes profundas mediante un marco de aprendizaje residual, agregando capas como funciones residuales de aprendizaje referenciando a las capas de entrada, facilitando la optimización de los modelos y ganando precisión con profundidades mayores, como lo es la red residual profunda de 152 capas, que cuenta con menor complejidad que una VGG a pesar de ser 8 veces más profunda. Mostrando mejores resultados en el reto *ILSVRC*, comparada con métodos de vanguardia del momento, como *VGG*, *PreLU-net*, *GoogLeNet* y *BN-inception*, datos que se muestran en la Tabla 8.

Tabla 8. Tasas de error (%) de los resultados de single-model en el conjunto de validación de ImageNet

method	Top-1 err.	Top-5 err.
VGG [40] (ILSVRC'14)	-	8.43
GoogLeNet [43] (ILSVRC'14)	-	7.89
VGG [40] (v5)	24.4	7.1
PReLU-net [12]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	18.83	4.49

Nota. Datos tomados de He, et al. (2016)

El modelo de *ResNet* con 152 capas mostró el menor error de top 1, lo que quiere decir, que tiene un error de 18.83 % en que el modelo no tuvo como mayor probabilidad a la clasificación correcta, y en el caso del top 5, tuvo un error de 4.49 % en el que en el resultado del modelo no se encontró en las 5 clases con mayor probabilidad del modelo, por lo tanto, entre los otros modelos PReLU-net obtuvo menor error en el top-1 con 21.59 %, con esto en cuenta, sin embargo ResNet-34 C reduce este error a 21.53 %, esto basado en el *dataset* de test de *ImageNet*, con esto los desarrolladores de este modelo lograron mostrar que la Arquitectura *ResNet* tiene un gran rendimiento en comparación con otros modelos, por lo tanto, será tomada como la opción más viable para implementar en el entrenamiento del modelo para pesaje.

Al tomar *ResNet*, también se tiene en cuenta que han ido surgiendo nuevos modelos, y, por lo tanto, se realizó un análisis comparativo con Redes como *ResNext*, *DenseNet* y *EfficientNet*, que han demostrado mejores resultados, pero en *datasets* de clasificación, analizando así la mejor opción respecto a las necesidades de este proyecto, que son el tiempo para desarrollarlo, la capacidad computacional y adaptabilidad.

4.1.2.1.1. ResNext

Con el avance investigativo en esta rama de visión por computadora, han ido surgiendo más arquitecturas, como menciona Hitawala (2018), entre las más recientes, se encuentra el modelo *ResNeXt*, este propuesto por *Facebook AI Research* en el año 2017, basado en las arquitecturas *ResNet* y *VGG*, con un gran rendimiento

en la clasificación de imágenes, al ser el ganador del Desafío de Reconocimiento Visual de *ImageNet* en el 2017, su estructura de profundidad y capas, se las puede observar en la Figura 12.

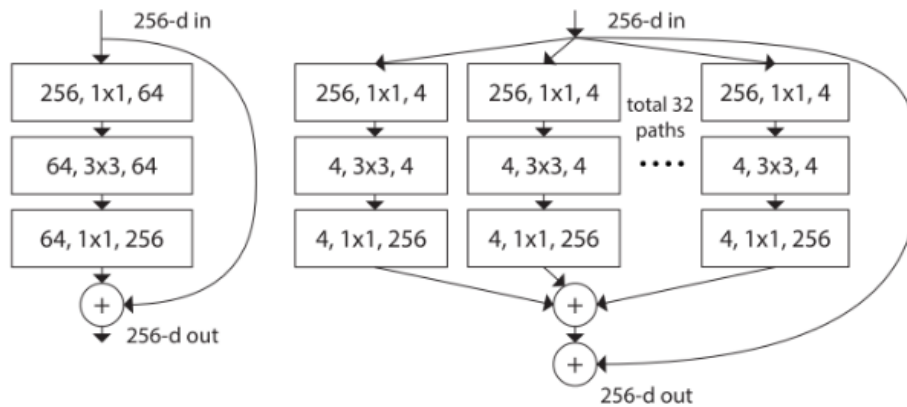


Figura 12. Arquitectura de ResNet en comparación con ResNext
Fuente: Hitawala (2018)

En la Figura 12, se analizar al lado derecho, el bloque de construcción de la arquitectura de *ResNext*, siguiendo un paradigma de división, transformación y fusión, y en este caso, las salidas de las diferentes rutas se fusionan al sumarse.

4.1.2.1.2. DenseNet

En el artículo de Huang et al. (2018), muestran que al tener conexiones más cortas entre las capas cercanas a la entrada y salida, los resultados son más precisos y eficientes, desarrollando la arquitectura *DenseNet*, con base en este comportamiento, por lo tanto, no tiene una misma cantidad de conexiones como capas, si no, la red cuenta con conexiones directas, logrando presentar mejoras de gradiente de desaparición, refuerzan la propagación de características, fomentan la reutilización de características y reducen el número de parámetros, obteniendo mejores resultados que otras arquitecturas, con la estructura como se muestra en la Figura 13.

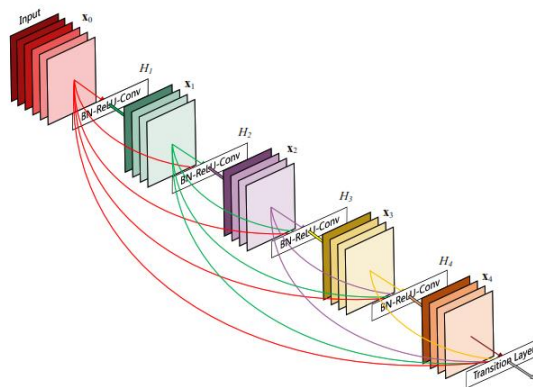


Figura 13. Arquitectura de *DenseNet*
Fuente: Huang (2018)

En la Figura 13, se puede ver cómo se hace una arquitectura sin la estructura convencional, con la idea de un patrón de conectividad simple, basado en garantizar el flujo de información máxima, al conectar todas las capas directamente entre ellas, y a diferencia de *ResNet*, no se combinan las características sumándolas y transmitiéndolas a otra capa, si no, concatenándolas.

4.1.2.1.3. EfficientNet

De acuerdo con lo mencionado por Tan y Le (2020), se llevan a cabo redes neuronales convolucionales con presupuestos fijos para posteriormente escalarlos a más recursos en el caso de ser posible, con el fin de obtener mejores resultados, por eso, identifican mediante un escalado de modelos, el equilibrio entre profundidad, ancho y resolución de la red para mejorar el rendimiento, a lo cual, proponen un método que escala uniformemente todas las 3 dimensiones utilizando un compuesto simple pero efectivo, obteniendo eficiencia para el escalado de redes móviles y *resnets*. En la Figura 14 se observa su función en comparación con otros modelos.

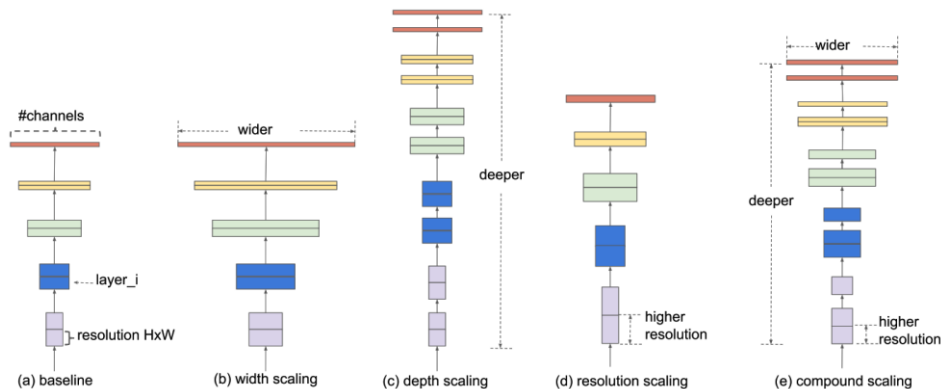


Figura 14. Escalamiento del modelo *EfficientNet*.
Fuente: Tan y Le (2020)

En la Figura 14, se ilustra la diferencia del método de escalamiento y métodos convencionales, en el caso de (a) se tiene una red base, el (b), (c) y (d) muestran escalamientos convencionales con aumento en una dimensión, ancho, profundidad o resolución, respectivamente, mientras en el (e) muestra el escalamiento compuesto, al escalar de manera uniforme entre las tres dimensiones.

En la Tabla 9, se presentan cuáles son las ventajas y desventajas de las diferentes arquitecturas tomadas en cuenta, para poder desarrollar el programa, de acuerdo con las necesidades requeridas por el proyecto.

Tabla 9. Ventajas y limitaciones de ResNet, ResNext, DenseNet y EfficientNet.

Arquitectura	Ventajas	Limitaciones	Rendimiento ImageNet
ResNet. He, et al. (2016)	<ul style="list-style-type: none"> Logra entrenamiento con redes profundas, evitando degradación por bloques residuales. Alta estabilidad y buena generalización. Amplio soporte en frameworks (Tensorflow), fácil transferencia de learning. 	<ul style="list-style-type: none"> Requiere ajuste de hiperparámetros. 	ResNet-50 Top 1 Acc.: 76.0% Top 5 Acc.: 93.0% Parámetros: 26 Millones
ResNext. Hitawala (2018)	<ul style="list-style-type: none"> Implementa "cardinalidad" (ramas paralelas), mejorando la expresión sin aumentar profundidad. Mejora precisión de ResNet. 	<ul style="list-style-type: none"> Mayor complejidad de implementar por la estructura multi-rama. Difícil implementar manualmente, requiere cambiar número de canales, aplicar regularización o adaptar para tareas distintas a la clasificación. Más complejo el ajuste de hiperparámetros: cardinalidad, profundidad y ancho de ramas. Por sus ramas paralelas, es difícil interpretar lo que cada parte del bloque está aprendiendo. 	ResNext-101 Top 1 Acc.: 80.9% Top 5 Acc.: 95.6% Parámetros: 84 Millones
DenseNet. Huang, et al. (2018)	<ul style="list-style-type: none"> Mejora la propagación de gradiente y reutilización de características mediante conexiones densas. Precisión similar a ResNet con menor número de parámetros. 	<ul style="list-style-type: none"> Las conexiones entre cada capa dificultan adaptación e interpretabilidad en su estructura. Más costosa computacionalmente por su cantidad de conexiones más grande. En el aumento de conexiones, los tensores que pasan entre capas crecen cuadráticamente, requiere más memoria GPU, a pesar de tener menos parámetros finales. Difícil manipular las capas internas, al ser interdependientes, modificar 	DenseNet-169 Top 1 Acc.: 76.2% Top 5 Acc.: 93.2% Parámetros: 14 Millones

Arquitectura	Ventajas	Limitaciones	Rendimiento <i>ImageNet</i>
EfficientNet Tan y Le (2020)	<ul style="list-style-type: none"> Optimización simultánea de profundidad, ancho y resolución. Excelente precisión, eficiencia parámetros y velocidad de inferencia. 	<p>una capa intermedia impacta en las siguientes.</p> <ul style="list-style-type: none"> Así mismo, al estar mezcladas no se puede saber que está aprendiendo cada capa o parte de la red. Estructura más compleja (MBCConv + SE blocks). Difícil adaptar (reemplazo y ajuste) para regresión debido a escalado y resoluciones fijas. Requiere redimensionar las imágenes para cada variante, perdiendo información de estas. Entrenar el modelo es más difícil debido a que se necesita un balance cuidadoso entre las 3 dimensiones o pierde rendimiento rápidamente. La red fue diseñada para aprovechar recursos de los modelos pre entrenados, al entrenar desde cero requiere muchos recursos. 	EfficientNet-B7 Top 1 Acc.: 84.3% Top 5 Acc.: 97.0% Parámetros: 5.3 Millones

Para el desarrollo del modelo, se requiere una arquitectura, que muestre un buen desempeño, que sea eficiente a nivel computacional y sea adaptable al problema de regresión, pues estas arquitecturas son más usadas para clasificación, con esto en cuenta, *ResNet* cumple con estas condiciones, a pesar de no ser la que mayor precisión tiene con respecto a otras redes, mantiene un uso de recursos computacionales normal, tiene una adaptabilidad al problema de regresión, al cambiar en la última capa "fc" de "softmax" usada para clasificar, por regresión lineal con "linear", por ser un problema para encontrar un solo valor numérico, también, permite ver lo que están aprendiendo las capas de la red. En el caso de *EfficientNet* a pesar de tener la mejor precisión, no es una buena opción para el problema en específico, pues tiene una estructura muy compleja, siendo menos adaptable y su dificultad de entrenamiento aumenta para entrenarla desde cero, en el caso de *ResNet*, sus capas en paralelo vuelven difícil su manipulación para cambiar a un problema de regresión, y *DenseNet*, también dificulta la adaptabilidad del modelo, al querer modificar las capas, teniendo en cuenta que su interconexión afecta toda la red.

Por esta razón, la estructura de *ResNet*, fue la seleccionada para el desarrollo del modelo, en específico *ResNet-50*, al tener un equilibrio entre precisión y costo computacional, además, permite cambiar la profundidad y su estructura para adaptarla a las imágenes de entrada. Estas mismas razones, permiten saber si al entrenar el modelo se obtendrán valores óptimos, es decir, con errores bajos, en el

caso que los resultados no sean precisos, evita usar redes más complejas que conlleven más tiempo en adaptarlas, en entrenarlas y que requieran de más recursos de cómputo, aportando información importante sobre los posibles errores en la estructura o en el *dataset*, caso contrario, se tendrá una buena base que muestre resultado aceptables, y en este caso se optaría por mejorar el modelo, con redes más complejas que puedan mejorar el resultado y hacerlo más preciso.

4.1.2.2. Requerimientos de software

Para el desarrollo del programa se parte de *Python*, el cuál es un lenguaje de programación ampliamente usado para el desarrollo de modelos de IA, con la implementación de librerías que permitirán el procesamiento de los datos y la generación de los modelos, en la Tabla 10, se muestra aquellas librerías usadas y su función en todo el proceso.

Tabla 10. Librerías usadas para el desarrollo del programa

Librería	Función
Tensorflow	<p>Preparar datasets ".tfrecord", formato optimizado para almacenar gran cantidad de datos en Tensorflow. <u>Funciones:</u> - tf.train.feature(): Define características (features) para cada ejemplo. - tf.train.BytesListy(), tf.train.Int64LIST(): Define listas de bytes y enteros en features, respectivamente. - tf.train.Example(): Crea un objeto de ejemplo serializable (objeto para ser convertido en una secuencia de bytes para su almacenamiento o transmisión). - tf.io.TFRecordWriter(): Escribe en un archivo de formato ".tfrecord" los datos para almacenar.</p> <p>Cargar datasets Carga y decodificar archivos de formato ".tfrecord", preprocesa imágenes, extrae etiquetas y prepara los <i>datasets</i> para los entrenamientos de los modelos. <u>Funciones:</u> - tf.io.FixedLenFeature(): Define el tipo y forma de los datos almacenados en el <i>TFRecord</i>. - tf.io.parse_single_example(): Decodifica un solo ejemplo del archivo <i>TFRecord</i> según el esquema de features. - tf.io.decode_jpeg(): Decodifica una imagen JPEG codificada en bytes. - tf.image.rgb_to_grayscale(): Convierte la imagen de formato RGB a escala de grises. - tf.image.resize(): Redimensiona las imágenes a un tamaño específico de píxeles. - tf.cast(---, tf.float32): Convierte la imagen a valor de flotantes.</p>

Librería	Función
	<ul style="list-style-type: none"> - tf.data.TFRecordDataset(): Crea un <i>dataset</i> de un archivo ".tfrecord". - tf.data.Dataset.map(): Aplica una función que decodifica un único ejemplo del archivo ".tfrecord" para obtener los datos. - tf.data.Dataset.shuffle(): Mezcla los datos de modo aleatorio. - tf.data.Dataset.batch(): Agrupa los datos en lotes. - tf.data.Dataset.prefetch(): Precarga los datos para mejorar el rendimiento de los entrenamientos. <p>Modelo con Keras: Permite personalizar el modelo de <i>ResNet</i> con <i>Keras</i>, compilar y entrenar los <i>datasets</i>. Para la implementación del modelo el uso de las funciones, <i>Input()</i>, <i>Conv2D()</i>, <i>BatchNormalization()</i>, <i>Activation()</i>, <i>Add()</i>, <i>ZeroPadding2D()</i>, <i>MaxPooling2D()</i>, <i>AveragePooling2D()</i>, <i>flatten()</i>, <i>Dense()</i>, <i>Model()</i>, posterior a esto se configura el modelo para entrenarlo con la función <i>model.compile()</i> y se ejecuta el mismo con <i>model.fit()</i></p>
pandas	<p>Preparar datasets Permite leer archivos Excel que contengan las etiquetas e información asociadas a las imágenes.</p> <p><u>Funciones:</u></p> <ul style="list-style-type: none"> - pd.read_excel(): Permite leer el archivo en formato Excel. - df.iloc[:,0], df.iloc[:,1]: Accede a la columna especificada por el número, teniendo en cuenta la primera columna como 0. - df["-----"]: Accede a la columna especificada por el nombre. - pd.DataFrame(): Crea un DataFrame de Pandas y con la función to_excel(), exporta los datos a un archivo en formato Excel. - pd.Series(): Crea una serie de pandas (columna)
os	<p>Preparar datasets: Permite operar con rutas de archivos, crear, modificar y eliminar carpetas.</p> <p><u>Funciones:</u></p> <ul style="list-style-type: none"> - os.makedirs(): Crea la carpeta de salida si no existe. - os.path.join(): Genera rutas de acceso válidas para el sistema operativo - os.listdir("ruta_directorio"): Permite listar los nombres de archivos en un directorio especificado.
cv2 (OpenCV)	<p>Preparar datasets: Permite leer y preprocesar las imágenes (lectura, conversión de color, cambio de tamaño y codificación).</p> <p><u>Funciones:</u></p> <ul style="list-style-type: none"> - cv2.imread(): Leer una imagen desde un archivo de una ruta especificada, también permite leer las imágenes en un solo canal, blanco y negro. - cv2.cvtColor(): Convierte imágenes a RGB, blanco y negro, desde el formato por defecto de cv2 (BGR). - cv2.resize: Redimensiona la imagen a un tamaño fijo. - cv2.imencode(): Codifica la imagen a un formato como ".jpg" en bytes
Matplotlib.pyplot	Graficar en python

Librería	Función
	Permite visualizar las imágenes y las etiquetas de los datasets, <u>Funciones:</u> - plt.figure(): Crea una nueva figura para agrupar subgráficas. - plt.subplot(): Divide la figura en cuadrícula y selecciona una celda para graficar -plt.xticks([]) y plt.yticks([]): Elimina ejes X y Y para limpiar la vista. - plt.grid(False): Oculta cuadrícula - plt.imshow(): Muestra la imagen como matriz de píxeles. - plt.xlabel(): Ubica la etiqueta debajo de la imagen. - plt.show(): Muestra la figura completa con el conjunto de subgráficas.
Scikit-learn	Preparar Datasets Con train_test_split() permite dividir los datos en conjuntos de entrenamiento y prueba

En la Tabla 10, se encuentran las librerías y funciones más representativas que permiten el manejo y procesamiento de los datos, conjuntamente, para aquellas que permiten modelar la CNN, por lo tanto, para ciertas acciones que se requieren pueden darse más funciones para complementar, pero las presentadas son las principales, lo que permite entender los procesos que se llevan a cabo antes, durante y después del entrenamiento de modelos de redes neuronales convolucionales.

4.1.2.3. Hardware utilizado

El proceso de entrenamiento de redes neuronales convolucionales requiere de infraestructura computacional con gran capacidad para procesar abundante cantidad de información, garantizando tiempos de procesamiento razonables, por esta razón se analiza el hardware que se utilizó para entrenar los modelos de IA, con base en sus principales características y capacidades que influyen directamente al rendimiento del entrenamiento, las especificaciones se presentan en la Tabla 11.

Tabla 11. Especificaciones del computador utilizado en los entrenamientos de IA.

Componente	Especificaciones	Observaciones
CPU	AMD Ryzen 9 5950 X 16-Core Processor	El procesador de 16 núcleos físicos y 32 hilos, basado en la arquitectura Zen 3, ofrece alto rendimiento en tareas de cálculo paralelo, por lo tanto, es útil para preparar los datos. Esto se evidencia, al tener imágenes con resolución de 1400 x 1400 a RGB, el procesador ocupa un aproximado del 20% de su capacidad y permite que la GPU trabaje maximizando sus recursos, como se mira en el Figura 22.
GPU	NVIDIA GeForce RTX 3090 y memoria de la GPU dedicada 24 GB.	Componente central para el entrenamiento de los modelos. Sus 24 GB de VRAM es la capacidad que soporta para entrenar el modelo, esto permitió realizar los entrenamientos en lotes de 8 imágenes, a una resolución máxima de 1400 x 1400 píxeles y a escala de grises.

Componente	Especificaciones	Observaciones
Memoria	63,9 GB, Velocidad, 2666 MT/s, Ranuras Usadas: 4 de 4	La capacidad de memoria es fundamental para trabajar con grandes volúmenes de datos, pues permite guardar los datos en Cache y así los datos ya se encuentren cargados y listos para ser procesados. Aunque para este proyecto, la capacidad de la memoria no fue suficiente en la resolución de imágenes requerida, lo cual fue cubierto por la capacidad del procesador al preprocesar los datos directamente desde la memoria, así mismo, para evitar que este no llene su capacidad, las imágenes fueron preprocesadas en un archivo en formato <i>TFRRecord</i> a un tamaño de 1400 x 1400 en formato RGB, esto reduce drásticamente el tamaño de las mismas y permite que el procesador no use recursos en leer imágenes pesadas y redimensionarlas.
Disco 0 (D:)	Samsung SSD 870 EVO 4TB, capacidad de 3,6 TB	Esta unidad de estado sólido permitió almacenar los <i>datasets</i> de gran tamaño, así mismo ofrece velocidad para lectura y estructura secuencia suficiente para precargar los datos.
Disco 1 (C:)	KINGSTONE SNV2S1000G, capacidad 932 GB (Disco del sistema)	Esta unidad de estado sólido alberga el sistema operativo, entornos de desarrollo y bibliotecas necesarias, su velocidad es superior a la del Disco 0 (D:), mejorando en los tiempos de carga y respuesta del sistema.
Wi-Fi	Intel(R) Wireless-AC 9260 160 MHz	Este componente permite conectarse a redes Wi-Fi de alta velocidad, permitiendo una conexión estable con el entorno de ejecución de Google Colab.

Con la información de la Tabla 11, se mira una infraestructura óptima para desarrollar modelos de IA, además, con estas características puede dar información importante para saber si esta capacidad es adecuada para el programa en específico o si requiere de una mayor robustez en infraestructura computacional. Además, es importante mencionar que, este computador es un equipo de la Universidad Politécnica Estatal del Carchi, más específicamente se encuentra en funcionamiento en el laboratorio de Realidad Virtual y Simulación de la carrera de Logística y Transporte, mismo lugar en el que se llevaron a cabo las pruebas y entrenamientos de los modelos.

La captura de imágenes, se la realizó con un teléfono celular personal, de la marca Samsung, en específico el modelo Galaxy A51, el cual permitió tomar imágenes a una resolución de 4000 x 3000 píxeles en formato RGB, donde para aprovechar la capacidad del dispositivo, se aseguró la calidad con respecto al entorno, como fotografías con mala iluminación u obstrucciones frente a los vehículos, este dispositivo en el mercado cuenta con una cotización aproximada en USD 200, de la misma manera, se hizo uso de un trípode, para evitar que la cámara del celular tomara fotos borrosas por el movimiento.

4.1.2.4. Datasets

4.1.2.4.1. Captura de imágenes.

La cantidad de imágenes en modelos de IA en visión por computadora no cuenta con una cantidad predeterminada que asegure un funcionamiento óptimo, pero para obtener mejores resultados, es mejor, una gran cantidad de datos, para que pueda lograr generalizar los diferentes casos de mejor manera.

El proceso de recolección de información consta de la captura de imágenes de vehículos de carga pesada y anotar el peso que corresponde a ese momento de la fotografía, ya sea, estando descargado, cargado total o parcialmente, esto se logró con el apoyo de varios centros de almacenamiento en la ciudad de Tulcán, el primero TRANSBOLIVARIANA C.A, BODEGAS PRIVADAS TERAN CIA. LTDA., ADUANOR CIA. LTDA. y MIDECAR CIA. LTDA., asegurando la fidelidad de los datos al anotar el valor dado por las básculas, este proceso de recolección de imágenes y pesos llevó aproximadamente 4 meses.

El capturar imágenes en estas 4 bodegas, permitió, tener una gran base de imágenes para entrenar el modelo, con variedad, tanto por el lugar en donde se toma las imágenes y los tipos de vehículos, con el fin que la red no aprenda datos por un lugar o un tipo de vehículo específico, por lo tanto, estas bodegas lo permiten, pues manejan diversidad de productos que se almacenan para su transporte a nivel nacional e internacional, por esta misma razón, llegan diferentes tipos de vehículos, los cuales cargan o descargan mercancía, obteniendo así imágenes en las condiciones de diferentes pesos, de igual manera, se asegura la calidad en las etiquetas de los datos, pues a cada vehículo que ingresara o saliera se le tomaba la foto y se registraba el peso que daba como resultado la báscula, logrando así asociar imagen y etiqueta del peso real.

Algunas condiciones que se tomaron en cuenta para asegurar la calidad del *dataset*, fue, tomar algunas fotografías con el conductor en el vehículo, asegurando que el peso registrado correspondiese al del vehículo más el conductor, y otras fotografías sin el conductor, asegurando que el peso registrado sea solamente del vehículo, a su vez, las imágenes se tomaron a lo largo del día, manteniendo variedad con respecto al fondo de imagen y cantidad de luz. Así mismo, la cantidad de datos no es muy elevada, en total se llevó a cabo el proceso de tomar fotografías 1390 veces, siendo este el registro fotográfico, en la Tabla 12 se analiza algunas formas en las que se tomaron las fotos.

Tabla 12. Condiciones en las que fueron tomadas las fotografías,

Condición	Especificación	Observación
<i>Mismos vehículos</i>	Los vehículos que realizan transporte de carga no usan los servicios de bodega una sola vez.	El repetir vehículos, permite tener en la base un vehículo con diferentes pesos y al tomar la foto en diferentes momentos o días, el fondo también puede cambiar, por lo tanto, el modelo deberá aprender a identificar características más asociadas con el peso que con el entorno.
<i>Por proceso de fotografías</i>	Los vehículos tienen el doble de puntos de apoyo por cada eje, por lo tanto, las alteraciones por el peso se verán afectadas en cada punto de apoyo.	Una mala distribución en el peso puede alterar un lado del vehículo, más que el otro, y si solo se toma fotografías de uno, el modelo tendrá errores, pues no analizaría el cambio en todos los puntos donde podrían sufrir cambios por el peso, por lo tanto, se toma fotografía por cada esquina del vehículo, con denominación de parte delantera izquierda, delantera derecha, trasera izquierda y trasera derecha, además, en vehículos pequeños y medianos la cámara del teléfono celular permite capturar las imágenes por la parte lateral izquierda y derecha.
<i>Aumento de datos</i>	Las imágenes son representaciones numéricas de los píxeles, por lo tanto, la lectura de las imágenes sobre una posición diferente del vehículo en la cámara será tomada como una imagen diferente.	Teniendo en cuenta que la cantidad de imágenes para un modelo de IA debe ser elevado, y que la recolección de este <i>dataset</i> dio una cantidad baja, se opta por aprovechar esta lógica de tomar fotografías de un mismo vehículo, entre una, dos y tres veces por ángulo, permitiendo al modelo encontrar características sobre el peso sin importar la ubicación del vehículo en la posición de la imagen, esto, permitió aumentar el <i>dataset</i> a 14249 imágenes.

En la Tabla 12, se puede analizar de mejor manera como se consiguieron las imágenes que permiten entrenar el modelo de IA para estimar el peso de los vehículos, que, a pesar de haber 1390 registros fotográficos de los vehículos, la cantidad de datos final fue de 14249, por haber tomado imágenes en diferentes ángulos de los vehículos y que de cada ángulo se tomaran más de una, pero con diferentes posiciones del vehículo en la imagen, el proceso de tomar las imágenes por ángulo se lo presenta en la Figura 15.



Figura 15. Representación del proceso de captura de imágenes por ángulo.

En la figura 15, se muestra visualmente mediante una referencia la captura de imágenes de un vehículo, desde cada uno de sus ángulos, en la imagen se presenta un camión, mismos que además de las 4 esquinas también permitieron tomar fotos de manera lateral, pues la cámara del celular con el que se trabajó, no tiene la amplitud angular para tomar imágenes laterales en tractocamiones sin tener que alejarse del vehículo una distancia muy extensa, por lo tanto, para el caso de tractocamiones con semirremolques solo se tomaron imágenes desde los ángulos delantero izquierdo y derecho y Trasero izquierdo y derecho. De ahí, con respecto al aumento de datos tomando más fotos del mismo vehículo en un mismo ángulo, se observa en la Figura 16 para una mejor comprensión.

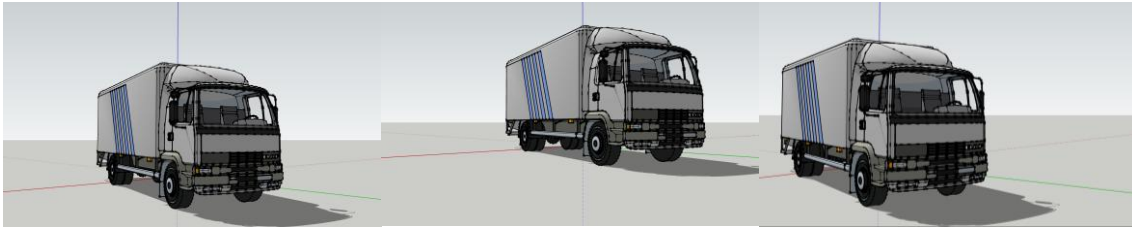


Figura 16. Representación del proceso de aumento de datos al tomar más fotos de un mismo vehículo en un mismo ángulo.

El aumento de datos, como se mencionó antes, se debe a que las imágenes son representaciones numéricas en los píxeles, por esta razón, en la Figura 16, se observa un mismo vehículo en el mismo ángulo delantero izquierdo, la diferencia se da en la posición en la imagen, hace que cada imagen ante una computadora sea diferente, pues en el lado izquierdo la imagen del vehículo se encuentra más ubicada en la parte inferior derecha, mientras la imagen central muestra el vehículo en la parte superior derecha y la imagen de la derecha muestra el vehículo centrado al lado izquierdo de la imagen, con esto en cuenta, se puede aumentar por cada vehículo en muchas formas, por lo que, para cada vehículo, se le tomó entre dos o tres imágenes por ángulo, siendo así que el *dataset* sea más grande y al mismo tiempo, haciendo que la red no se aprenda una posición del vehículo, sino se la fuerce a buscar las características que determinan el peso del vehículo en la imagen.

Un aspecto importante que se toma en cuenta es el lugar donde se tomaron las imágenes, pues la infraestructura de cada uno permitía o limitaba los ángulos en los que se toma una imagen, en la Tabla 13, las distancias a las que se tomaron las imágenes en cada bodega.

Tabla 13. Distancia para la captura de imágenes entre el un ángulo del vehículo y la cámara, y por bodegas.

Depósito	Distancia de cámara y el ángulo del vehículo	
TRANSBOLIVARIANA C.A	Camión: Ancho ≈ 5.41 m Largo ≈ 18 m Tractocamión: Ancho ≈ 4.05 m Largo ≈ 18 m	Observación: En esta bodega se usó el área de la báscula, no una distancia entre la cámara y el ángulo del vehículo.
BODEGAS PRIVADAS TERAN CIA. LTDA.	Alto ≈ 1.45 m Camión: A lo largo se aleja 1 m y a lo ancho 3 m. Distancia $\approx 3,16$ m. Tractocamión: A lo largo se aleja 2 m y a lo ancho 4 m. Distancia $\approx 4,47$ m.	Área aproximada Camión: Largo ≈ 15 m Ancho ≈ 6 m Tractocamión: Largo ≈ 22 m Ancho ≈ 8 m

Depósito	Distancia de cámara y el ángulo del vehículo	
ADUANOR CIA. LTDA.	<p>Alto \approx 1.45 m</p> <p>Camión: A lo largo se aleja 1 m y a lo ancho 3 m. Distancia \approx 3,16 m. Lateralmente \approx 5 m</p> <p>Tractocamión: A lo largo se aleja 2 m y a lo ancho 4 m. Distancia \approx 4,47 m.</p>	<p>Área aproximada</p> <p>Camión: Largo \approx 15 m Ancho \approx 6 m Lateral \approx 17 m</p> <p>Tractocamión: Largo \approx 22 m Ancho \approx 8 m</p>
MIDECAR CIA. LTDA.,	<p>Alto \approx 1.45 m</p> <p>Camión: A lo largo se aleja 1 m y a lo ancho 3 m. Distancia \approx 3,16 m. Lateralmente \approx 5 m</p> <p>Tractocamión: A lo largo se aleja 2 m y a lo ancho 4 m. Distancia \approx 4,47 m.</p>	<p>Área aproximada</p> <p>Camión: Largo \approx 15 m Ancho \approx 6 m Lateral \approx 17 m</p> <p>Tractocamión: Largo \approx 22 m Ancho \approx 8 m</p>
	Alto \approx 1.45 m	

En la Tabla 13, se encuentran las distancias a las que se aleja la cámara de las esquinas de los vehículo, para poder tomar la foto y que en la imagen salga todo el vehículo, sin embargo, en Transbolivariana no existe esta distancia, pues las imágenes se tuvieron que tomar mientras los vehículos se encontraban en la báscula, debido a que fuera de esta, las condiciones del piso no era propicias para el programa, dado esto, se tomó como referencia en vehículos de dos ejes las esquinas de los muros de la báscula, mientras para vehículos más grandes, por motivo que las vigas sobresalían de los laterales de la plataforma, las imágenes a lo ancho se tomaron más pegadas al vehículo, por ende, el área referencial fue más pequeña. De ahí, en bodegas privadas Terán, las imágenes fueron tomadas antes de ingresar a báscula y por eso permitió estandarizar una distancia entre las esquinas de los vehículos y la cámara, estas medidas se dieron en las demás bodegas, sin embargo, en este lugar en específico su ancho en el ingreso no permitió tomar imágenes de los camiones lateralmente, mientras en los demás lugares si fue posible, siendo esta una distancia de 5 metros frente al eje trasero.

Además, hubo complicaciones al tomar las imágenes que se dieron porque no se podía intervenir en el flujo de los vehículos en las bodegas, por lo tanto, no fue posible tomar imágenes en una zona controlada, más bien, se pedía a los conductores que paren el vehículo un momento y con ayuda de un trípode a una altura de 1.45 m, la altura en todos los casos, se posicionaba el celular estimando las distancias

mencionadas anteriormente alrededor del vehículo, evitando intervenir en las actividades de las bodegas y facilitando el trabajo de recolección de imágenes, esta misma razón, hizo tomar la decisión de no tomar la presión de las llantas de los vehículos, pues el proceso para hacerlo es muy tardado, teniendo en cuenta un vehículo de 2 ejes que tiene 6 llantas puede ser accesible, aunque demoroso, pero en el caso de tractocamiones de 3 ejes y semirremolques de 3 ejes, representa una cantidad de 22 llantas, siendo no posible no intervenir en el flujo de las actividades de los vehículos, igualmente no se puede tomar una muestra y generalizar para todas las llantas, considerando que la IA trabaja con datos lo más reales posibles para encontrar los patrones específicos que afecten al peso, por lo tanto, se comprenden como datos que no proporcionan eficiencia en el modelo.

Por otro lado, es necesario comprender todas las dificultades que se presentaron al recolectar esta información, pues, primero al no tener un ambiente controlado hace necesario adaptarse a las actividades de las bodegas y evitar alterar su flujo de trabajo, además, la captura de imágenes tenía que ser realizada personalmente, esto por la falta de recursos que implica adaptar un sistema automatizado, con esto surge el problema de los horarios, dado que existen actividades del investigador, como clases de la universidad, que intervienen en la toma de imágenes con respecto a los días que se toman las fotografías, que son lunes, martes, miércoles, jueves y viernes, días que laboran normalmente los depósitos, también se incluyen las dificultades presentadas al tomar las imágenes, como climas poco favorables, que son días lluviosos, días en los que cierran las vías por protestas y días no laborables, afectando directamente al tamaño del *dataset*.

4.1.2.4.2. Base de información general

Luego de haber obtenido la mayor cantidad de datos posibles es necesario tratar esta información antes de entrenar el modelo, esto al etiquetar las imágenes con los pesos, el número de vehículo en el registro fotográfico, el ángulo del vehículo capturado, el tipo de vehículo y el lugar donde se tomaron las fotos, esto en una base de Excel, organizando los datos como se muestra en la Tabla 14.

Tabla 14. Resumen de base de Excel con la información de las imágenes.

Imagen	Registro vehicular	Peso	Ángulo	Lugar	Tipo de vehículo
14249	1390	Referencial al número del registro del vehículo	6 formas: - Esquina delantera izquierda	- Transbolivariana - Bodegas privadas Terán - Aduanor	Camión de 2 ejes grande Camión de 2 ejes mediano

Imagen	Registro vehicular	Peso	Ángulo	Lugar	Tipo de vehículo
			- Esquina delantera derecha	- Midecar	Camión de 2 ejes pequeño
			- Esquina trasera izquierda		Camión de 3 ejes
			- Esquina trasera derecha		Tractocamión de 2 ejes y semirremolque de 2 ejes
			- Lateral izquierda		Tractocamión de 2 ejes y semirremolque de 3 ejes
			- Lateral Derecha		Tractocamión de 3 ejes y semirremolque de 2 ejes
					Tractocamión de 3 ejes y semirremolque de 3 ejes
					Tractocamión de 3 ejes
					Tractocamión de 2 ejes

En la Tabla 14, destaca el total de imágenes recolectadas, que es de 14 249, en un registro de 1390 vehículos, asignando un número específico a cada registro, por lo tanto, cada número de registro cuenta con varias imágenes, que se encuentran etiquetadas con el ángulo en que fue tomada la imagen, el peso se encuentra en *kg* y también el número de registro se encuentra relacionado con un lugar en el que se tomaron las imágenes, el tipo de vehículo, y la primera columna correspondiente al nombre o identificación de cada imagen, las cuales se encuentran en una carpeta del computador almacenadas con dichos nombres. En la Tabla 15, se observa la forma que tiene la base del Excel.

Tabla 15. Forma de la base de Excel con la información de las imágenes.

Id_Imagen	N_vehiculo	Peso	Angulo	Lugar	Tipo_vehiculo
1	1	13580	Lateral Izquierda	TransBolivariana	Camion de 2 ejes pequeño
			Esquina Delantera		
2	1	13580	Izquierda	TransBolivariana	Camion de 2 ejes pequeño
			Esquina Delantera		
3	1	13580	Derecha	TransBolivariana	Camion de 2 ejes pequeño
			Esquina Delantera		
4	2	3560	Derecha	TransBolivariana	Camion de 2 ejes pequeño
			Esquina Trasera		
5	2	3560	Izquierda	TransBolivariana	Camion de 2 ejes pequeño
			Esquina Trasera		
6	2	3560	Izquierda	TransBolivariana	Camion de 2 ejes pequeño
			Esquina Trasera		
7	2	3560	Derecha	TransBolivariana	Camion de 2 ejes pequeño
			Esquina Delantera		
8	2	3560	Izquierda	TransBolivariana	Camion de 2 ejes pequeño

Id_Imagen	N_vehiculo	Peso	Angulo	Lugar	Tipo_vehiculo
9	2	3560	Esquina Delantera Izquierda	TransBolivariana	Camion de 2 ejes pequeño
10	3	4180	Esquina Trasera Izquierda	TransBolivariana	Camion de 2 ejes mediano
⋮	⋮	⋮	⋮	⋮	⋮
14249	1389	7520	Esquina Trasera Izquierda	Midecar	Camion de 2 ejes grande

En la Tabla 15, muestra como cada imagen se relaciona a los datos con el nombre de la imagen, obteniendo el registro al que pertenece, el peso que tiene en el momento de ser tomada la imagen, el ángulo, el lugar donde fue tomada y el tipo de vehículo. Con esta información, es posible entrenar el modelo, al subir las imágenes y relacionarlas con el peso, para luego realizar predicciones con un *dataset* de prueba y con las etiquetas sobre la demás información mencionada, para segmentar los datos y observar si existe mejora con alguna clase en específico. Además, también se toma en cuenta el número de vehículos que hay, por cada tipo de vehículo, estos datos se muestran en la Tabla 16.

Tabla 16. Número de vehículos por cada tipo de vehículo,

Tipo de vehículo	Cantidad
Camión de 2 ejes grande	477
Camión de 2 ejes mediano	151
Camión de 2 ejes pequeño	135
Camión de 3 ejes	115
Tractocamión de 2 ejes y semirremolque de 2 ejes	49
Tractocamión de 2 ejes y semirremolque de 3 ejes	21
Tractocamión de 3 ejes y semirremolque de 2 ejes	57
Tractocamión de 3 ejes y semirremolque de 3 ejes	380
Tractocamión de 3 ejes	4
Tractocamión de 2 ejes	1
TOTAL	1390

En la Tabla 16, se muestra aquellos vehículos, que realizaron actividades en las bodegas y por ende, se tomaron en cuenta en la toma de fotografías, el tipo de vehículo fue clasificado mediante la tabla de pesas y medidas de la ANT, de la figura 6, mirando todas aquellas combinaciones que constan en el *dataset* de imágenes, en donde algunas combinaciones, el registro de vehículos es muy pequeña, tal es el caso de vehículos tractocamión de 2 ejes y semirremolque de 3 ejes que solo cuenta con 21 registros, en el caso de los camiones el de 3 ejes es el que menor cantidad tiene, con 115 registros, por lo tanto, estos datos no se encuentran en cantidades iguales, lo ideal para entrenar el modelo, es eliminar algunos registros de ciertos

vehículos para igualar las cantidades, pero no es mejor opción, ya que se pierde gran cantidad de información, por ende, un aspecto importante que se tomó en cuenta, es la forma en la que se tomaron las imágenes, pues la distancia necesaria para capturar una imagen de un camión es similar y así mismo la distancia para captura de imágenes de tractocamiones con semirremolques, es por esta razón, que se toma dos categorías, las cuales consisten en camión y tractocamión, esto por la similitud en el proceso que se tomaron las imágenes. Por eso en la Tabla 17 se muestra el número de vehículos con esta nueva clasificación.

Tabla 17. Número de vehículos por clasificación de camión y tractocamión

Clasificación de vehículos	Cantidad de Registros
Camiones	878
Tractocamiones	512
TOTAL	1390

Con esta nueva clasificación, se logra tener una cantidad de registros de los vehículos más grande para cada tipo, por ende, permite medir de mejor manera la eficiencia del modelo, a través de estas dos clases.

4.1.2.4.3. Bases para diferentes modelos

Como se mencionó anteriormente, las imágenes fueron tomadas por cada vehículo en diferentes ángulos, para evitar problemas sobre predicciones inexactas al haber vehículos con mayor carga en un lado del vehículo, por lo tanto, al tener diferentes ángulos, se genera 2 *dataset* más para diferentes modelos, en el primero se cargará las imágenes normales, sin modificación alguna y sin tener en cuenta que cada imagen solo cuenta con información de un solo lado, el segundo *dataset*, tomará en cuenta las imágenes por lados, así se relacionan en ángulos, delanteros, traseros y laterales, mismas que muestran un problema observado al tomar las imágenes, pues en el caso de tomar una imagen desde una esquina, la esquina posterior, a pesar que si sale en la fotografía, pierde mucha información debido a la lejanía a la que se encuentra la cámara, por eso, cada imagen relacionada con una esquina tendrá mejor calidad con dicho ángulo, por lo tanto, se relacionan estos ángulos para que la red tome en cuenta cada parte con mayor información y así pueda analizarla para obtener mejor precisión. Este proceso se detalla de mejor manera a continuación:

4.1.2.4.3.1. Creación de la segunda base relacionando por lados

Para ambos casos de la creación de datos, se toma en cuenta que al relacionar las imágenes, se refiere a unificarlas, pues las CNNs reciben imágenes de entrada, y esta

es la forma más fácil de hacer que ambos lados sean analizados al mismo tiempo, por ende, el proceso se basa en seleccionar un número de vehículo por vez, el cual, tendrá varias imágenes relacionadas, cada una identificada por un ángulo en específico, con esto en cuenta, se consideración que las imágenes tienen una resolución de 4000 x 3000 píxeles, por lo que, a la altura de cada una, se le recorta 1000 píxeles, perdiendo información de bajo valor, como parte del techo de los vehículos, sin embargo, estos no sufren deformaciones por el peso porque su función es cubrir la mercancía, de ahí, se unifica las imágenes y se guarda en una nueva carpeta con un nuevo nombre para la imagen. Este proceso se llevó a cabo mediante el código, mostrado en la Tabla 18.

Tabla 18. Código para unificar las imágenes por lados.

Función	Código
Crear nuevo dataset de la base original	<pre> # Obtener base original df = pd.read_excel(r"ruta_archivo.xlsx", sheet_name="Base") # Función para ver si tiene o no más de 3 fotografías def check_images(lista): if len(lista) <=3: return lista else: return random.sample(lista, 3) def check_pares(lista): lis = lista posibilidades = [] for i in range(3): if all(v in lis for v in ["di", "dd"]): posibilidades.append(1) lista.remove("di") lista.remove("dd") elif all(v in lis for v in ["ti", "td"]): posibilidades.append(2) lista.remove("ti") lista.remove("td") elif all(v in lis for v in ["li", "ld"]): posibilidades.append(3) lista.remove("li") lista.remove("ld") return posibilidades # Generar relaciones de las imágenes relaciones = [] vehiculo = [] etiqueta = [] Lugar = [] angulo = [] n_veh = list(df["N_vehiculo"].unique()) for i in n_veh: data = df[df["N_vehiculo"] == i] posibilidades = check_pares(list(data["Angulo"].unique())) for j in posibilidades: if j == 1: di = check_images(list(data[data["Angulo"]=="di"]["ld"])) dd = check_images(list(data[data["Angulo"]=="dd"]["ld"])) rel_d = [(x, y) for x in di for y in dd] relaciones = relaciones + rel_d vehi = [i]*len(rel_d) vehiculo = vehiculo + vehi </pre>

Función	Código
	<pre> peso = [data["Pesounico"].unique()[0]]*len(rel_d) etiqueta = etiqueta + peso lug = [data["Lugar"].unique()[0]]*len(rel_d) Lugar = Lugar + lug ang = ["delantera"]*len(rel_d) angulo = angulo + ang elif j == 2: ti = check_images(list(data[data["Angulo"]=="ti"]["Id"])) td = check_images(list(data[data["Angulo"]=="td"]["Id"])) rel_t = [(x, y) for x in ti for y in td] relaciones = relaciones + rel_t vehi = [i]*len(rel_t) vehiculo = vehiculo + vehi peso = [data["Pesounico"].unique()[0]]*len(rel_t) etiqueta = etiqueta + peso lug = [data["Lugar"].unique()[0]]*len(rel_t) Lugar = Lugar + lug ang = ["trasera"]*len(rel_t) angulo = angulo + ang elif j == 3: li = check_images(list(data[data["Angulo"]=="li"]["Id"])) ld = check_images(list(data[data["Angulo"]=="ld"]["Id"])) rel_l = [(x, y) for x in li for y in ld] relaciones = relaciones + rel_l vehi = [i]*len(rel_l) vehiculo = vehiculo + vehi peso = [data["Pesounico"].unique()[0]]*len(rel_l) etiqueta = etiqueta + peso lug = [data["Lugar"].unique()[0]]*len(rel_l) Lugar = Lugar + lug ang = ["laterales"]*len(rel_l) angulo = angulo + ang data = pd.DataFrame({"Uniones": relaciones, "Vehiculo": vehiculo,"Peso": etiqueta, "Lugar": Lugar, "Angulo": angulo}) data.to_excel(r"D:\Programa de Pesaje\2. Pesos\datos_pares.xlsx") </pre>
<p>Explicación de la creación de nuevo dataset en Excel</p>	<p>Este código guarda en la variable df, el archivo leído del Excel original. Luego, mediante una función <i>check_images()</i>, se analiza que para cada ángulo de un vehículo no tenga más de 3 imágenes, si tiene más, selecciona aleatoriamente 3, de ahí, en la función <i>check_pares()</i>, se analiza que por cada vehículo existan imágenes con ambos lados en la parte delantera e izquierda para poder relacionarlos. Posteriormente se crean variables para guardar los datos de las relaciones, el número del vehículo con el que se trabaja, el peso, el lugar y el ángulo, que en este caso es delantero, trasero y lateral, donde primero selecciona un vehículo, de ahí con la función <i>check_pares()</i> identifica los ángulos que tiene y si se pueden unir, para que posteriormente, identifique el ángulo al que corresponden, y mediante <i>check_images()</i> seleccione 3 elementos por ángulo y luego se relacionen los datos con las imágenes. Finalmente, con estas uniones se obtiene 16 362 imágenes, se guarda como un <i>dataframe</i> y lo guarda en un archivo Excel como "datos_pares.xlsx".</p>
<p>Crear imágenes con base en las imágenes originales</p>	<pre> # Carga de datos df = pd.read_excel(r"ruta_archivo\datos_pares.xlsx") salida = r"ruta_carpeta\Programa de Pesaje\Pares" # Función para recortar las imágenes def recortar_imagen(imagen): return imagen[500: , :] # Generar las imágenes image_ids_for_vehicle = [] # Recolectar los ids de las imágenes de la columna de uniones for union_list in data["Uniones"].apply(ast.literal_eval): image_ids_for_vehicle.extend(union_list) id_img = list(set(image_ids_for_vehicle)) </pre>

Función	Código
Explicación del código para crear imágenes con base en las imágenes originales	<pre> # Cargar las imágenes image_dict = {} for img_id in id_img: img_path = os.path.join(image_dir, f'{str(img_id)}.jpg') imag = cv2.imread(img_path) imag = cv2.resize(imag, (2000, 1500)) image_dict[img_id] = imag n = 0 nombres = data["Uniones"].index for lista in list(data["Uniones"].apply(ast.literal_eval)): name = nombres[n] n+=1 for i, n_img in enumerate(lista): if i == 0: img1 = image_dict[n_img] elif i == 1: img2 = image_dict[n_img] rec1 = recortar_imagen(img1) rec2 = recortar_imagen(img2) combinada = cv2.vconcat([rec1, rec2]) nombre_salida = f'{name}.jpg' path_salida = os.path.join(salida, nombre_salida) cv2.imwrite(path_salida, combinada) </pre>
	<p>El código empieza guardando la dirección de la carpeta donde se encuentran las imágenes y también de la carpeta donde se van a guardar las nuevas imágenes. De ahí, lee el archivo Excel con las relaciones en pares y lo guarda en una variable denominada <i>df</i>.</p> <p>De ahí, se define una función, <i>recortar_imagen()</i>, que tiene el fin de recortar cada imagen 500 píxeles desde la parte superior, esto debido a que más adelante la imagen se redimensiona a 2000 x 1500 píxeles, para acelerar el proceso de generar las imágenes.</p> <p>En este punto, se genera una variable, <i>imag_ids_for_vehicle</i>, que guardará las uniones las imágenes que se deberán cargar para una cierta cantidad de imágenes a generar, luego con esta lista, se cargan todas las imágenes y se guardan en un diccionario llamado <i>image_dict</i>, también se redimensionan a 2000 x 1500 píxeles y se las guarda con su id original.</p> <p>Con esto, se genera una variable de nombres, que guarda las imágenes que se relacionan por cada nueva imagen, luego se carga cada imagen, para poder ser recortadas con la función <i>recortar_imagen()</i>, y ser almacenadas en dos variables diferentes, las cuales con la función de <i>cv2.vconcat()</i>, se unifican la primera imagen encima de otra, para posteriormente guardar la imagen en la carpeta que se introdujo al principio agregado el nombre de la imagen.</p>

Con esto, en la base al código de la Tabla 18 se ha podido generar nuevas imágenes a través de las originales, pero la cantidad supera a la cantidad original, pues de 14249 imágenes, se obtuvo 16362, esto debido a la manera en que se relacionan los datos, siguiendo la lógica anterior de observar las imágenes parecidas como diferentes para una computadora, al seleccionar 3 imágenes de cada ángulo por lado, se logra realizar 9 combinaciones posibles, que ayuda a aumentar la cantidad de datos para los entrenamientos, pero se toma en cuenta que las imágenes por ángulo no en todos los casos eran 3, por lo tanto, no se generaron muchas más imágenes, y en algunos vehículos por cada ángulo eran de 4, en este caso para evitar esto, se seleccionó solamente 3, aumentando el tamaño del *dataset*, sin llegar a un

punto en que este crecimiento afecte con datos demasiado parecidos. El resultado de la unión de estas imágenes se representa en la Figura 17.

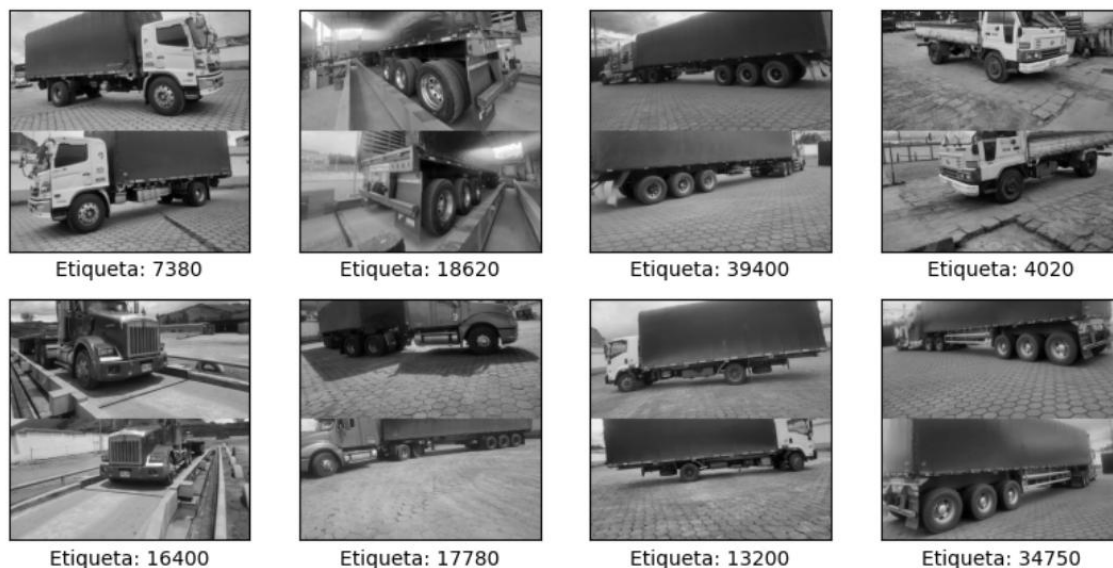


Figura 17. Imágenes unidas por lado izquierdo y derecho para la parte delantera, trasera y lateral.

En la Figura 17, se muestra el resultado de la unión de las imágenes por lados, al recortar las imágenes en la parte superior, se recorta en muchos casos la parte superior de los vehículos, pero las partes más importantes de los mismos siguen visibles, como lo son las llantas y la distancia que tiene carrocería con el piso, esto debido a que, la parte superior del vehículo cubre las mercancías, pero no sufren ningún cambio con respecto a un peso ejercido.

4.1.2.4.3.2. Creación de la tercera base relacionando por los 4 ángulos

Para este caso en específico, sigue una lógica parecida, pues selecciona de un vehículo 4 imágenes correspondientes a cada ángulo, de ahí, se implementa el modelo YOLO en su versión "yolov8x.pt", el cual permite identificar el vehículo, obtener su ubicación en la imagen y recortar la sección, además identifica el ángulo de la imagen y selecciona solamente la esquina con la que se trabaja, evitando aquello que no corresponde al vehículo y su mitad que no corresponde al ángulo, esto se lo hace con la imagen de cada esquina, obteniendo imágenes más específicas por cada ángulo, las cuales guardan mayor calidad y al unificarlas se obtiene la información más importante de cada ángulo en una sola imagen. El código que permitió llevar a cabo este proceso, se lo presenta en la Tabla 19.

Tabla 19. Código para unificar las imágenes por sus 4 ángulos.

Función	Código
Crear base de Excel de la base original	<pre data-bbox="395 226 1430 342"># Guardar la dirección de la carpeta donde se encuentran las imágenes image_dir = r"ruta_carpeta/Imágenes" # Cargar la base original df = pd.read_excel(r"ruta_archivo_Imágenes.xlsx")</pre> <p data-bbox="395 371 1430 427">Explicación: Estas dos primeras líneas, se carga la dirección de la carpeta donde se encuentran las imágenes y guardar el archivo de Excel en la variable df.</p> <pre data-bbox="395 461 1011 633"># Función para ver si tiene o no más de 3 fotografías def check_images(lista): if len(lista) <=3: return lista else: return random.sample(lista, 3)</pre> <p data-bbox="395 667 1430 728">Explicación: Se crea la función <code>check_images()</code>, para asegurar que no se repitan imágenes por cada ángulo más de 3 veces.</p>
	<pre data-bbox="395 763 858 1196">def check_union(lista): lis = lista posibilidades = [] for i in range(2): if all(v in lis for v in ["di", "dd", "ti", "td"]): posibilidades.append(1) lista.remove("di") lista.remove("dd") lista.remove("ti") lista.remove("td") elif all(v in lis for v in ["li", "ld"]): posibilidades.append(2) lista.remove("li") lista.remove("ld") return posibilidades</pre> <p data-bbox="395 1229 1430 1391">Explicación: En esta función, se busca que los vehículos tengan imágenes de los 4 ángulos para poder unirlas, caso contrario ese vehículo no es tomado en cuenta, así mismo, las imágenes laterales son tomadas como pares, pues al no estar en una esquina específica mantiene la misma calidad de imagen tanto en la parte delantera como trasera.</p>
	<pre data-bbox="395 1424 1107 2031">relaciones = [] vehiculo = [] etiqueta = [] Lugar = [] n_vehi = list(df["N_vehiculo"].unique()) for i in n_vehi: data = df[df["N_vehiculo"] == i] posibilidades = check_union(list(data["Angulo"].unique())) for j in posibilidades: if j == 1: di = check_images(list(data[data["Angulo"]=="di"]["Id"])) dd = check_images(list(data[data["Angulo"]=="dd"]["Id"])) ti = check_images(list(data[data["Angulo"]=="ti"]["Id"])) td = check_images(list(data[data["Angulo"]=="td"]["Id"])) relx4 = [(v, x, y, z) for v in di for x in dd for y in ti for z in td] if len(relx4) > 10: relx4 = random.sample(relx4, 10) relaciones = relaciones + relx4 vehi = [i]*len(relx4) vehiculo = vehiculo + vehi peso = [data["Pesounico"].unique()[0]]*len(relx4)</pre>

Función	Código
	<pre> etiqueta = etiqueta + peso lug = [data["Lugar"].unique()[0]]*len(relx4) Lugar = Lugar + lug elif j == 2: li = check_imagenes(list(data[data["Angulo"]=="li"]["Id"])) ld = check_imagenes(list(data[data["Angulo"]=="ld"]["Id"])) rel_l = [(x, y) for x in li for y in ld] if len(rel_l) > 8: rel_l = random.sample(rel_l, 8) relaciones = relaciones + rel_l vehi = [i]*len(rel_l) vehiculo = vehiculo + vehi peso = [data["Pesounico"].unique()[0]]*len(rel_l) etiqueta = etiqueta + peso lug = [data["Lugar"].unique()[0]]*len(rel_l) Lugar = Lugar + lug </pre>
	<p><i>Explicación:</i> En esta parte, se crean 5 variables, que almacenan las relaciones posibles por cada vehículo, luego el número de vehículo seleccionado, la etiqueta que representa el peso y el lugar.</p> <p>Este código avanza de registro de vehículo en vehículo, a lo cual selecciona los ángulos que tiene con la función <i>check_union()</i>, sabiendo así las posibilidades, que son unir las imágenes por las 4 esquinas o unir las en pares a las laterales.</p> <p>Con esto en cuenta, para cada posibilidad con la función <i>check_imagenes()</i>, busca seleccionar solo 3 imágenes como máximo por ángulo, luego las relaciona entre sí, a lo que tiene la capacidad de obtener en 3 imágenes por cada ángulo, 27 posibilidades, por lo que, genera demasiadas imágenes muy similares, por ende se limita el número y se selecciona 10 posibilidades aleatoriamente, así mismo en el caso de las imágenes unidas lateralmente, con la diferencia que se limita a escoger 8 uniones, de ahí, en cada proceso, se añade las combinaciones de imágenes, el número de vehículo, el peso y el lugar.</p> <p>De ahí, con el código <i>print(len(relaciones))</i>, se mira que se generaron 14150 imágenes, tomando que al unir las imágenes se pierde un 75% de la cantidad del <i>dataset</i>, con las uniones similares (no idénticas), se logra mantener un tamaño parecido de los datos.</p>
	<pre> #Crear un dataframe data = pd.DataFrame({"Uniones": relaciones, "Vehiculo": vehiculo,"Peso": etiqueta, "Lugar": Lugar, "Angulo": angulo}) </pre>
	<p><i>Explicación:</i> Con este código se logra guardar los datos en un <i>dataframe</i> para que estén relacionados entre sí, luego, con el código <i>len(data["Vehiculo"].unique())</i>, se mira que el número de vehículo bajó a 1372, siendo que en 18 vehículos no cumplen con tener imágenes para cada uno de los ángulos.</p>
	<pre> #Guardar dataframe en un excel data.to_excel(r"D:\Programa de Pesaje\2. Pesos\Base de Img Unidas.xlsx") </pre>
	<p><i>Explicación:</i> Con este código, se guarda el archivo con las relaciones de las imágenes originales para unificar todas las imágenes por sus 4 ángulos en un archivo de Excel llamado "Base de Img Unidas.xlsx".</p>
<p>Crear imágenes con base en las</p>	<pre> # Guardar la dirección de la carpeta de las imágenes image_dir = r"ruta_carpeta/Imagenes" # Leer archivo de excel con las relaciones de las imágenes df = pd.read_excel(r"ruta_archivo/Base de Img Unidas.xlsx") </pre>

Función	Código
imágenes originales	<p data-bbox="395 219 1428 324"><i>Explicación:</i> En estos dos códigos, se carga primero la dirección de la carpeta donde se encuentran las imágenes, luego se lee el archivo Excel y se guarda en la variable df.</p> <pre data-bbox="395 353 718 414"># Cargar modelo YOLO model = YOLO("yolov8x.pt")</pre> <p data-bbox="395 443 1428 515"><i>Explicación:</i> El modelo de YOLO permite identificar en cada imagen los vehículos para poder procesar las imágenes.</p> <pre data-bbox="395 544 869 2040">def redim(coordenadas, angulo): x = coordenadas[2] - coordenadas[0] y = coordenadas[3] - coordenadas[1] new = [] if angulo == 0: for j, i in enumerate(coordenadas): if j == 0: new.append(i + int(x / 4)) elif j == 1: val = i - int(y * 0.05) new.append(max(0, int(val))) elif j == 2: val = i + int(x * 0.05) new.append(min(4000, int(val))) elif j == 3: val = i + int(y * 0.05) new.append(min(3000, int(val))) elif angulo == 1: for j, i in enumerate(coordenadas): if j == 0: val = i - int(x * 0.05) new.append(max(0, int(val))) elif j == 1: val = i - int(y * 0.05) new.append(max(0, int(val))) elif j == 2: new.append(i - int(x / 4)) elif j == 3: val = i + int(y * 0.05) new.append(min(3000, int(val))) elif angulo == 2: for j, i in enumerate(coordenadas): if j == 0: new.append(i + int(x / 4)) elif j == 1: val = i - int(y * 0.05) new.append(max(0, int(val))) elif j == 2: val = i + int(x * 0.05) new.append(min(4000, int(val))) elif j == 3: val = i + int(y * 0.05) new.append(min(3000, int(val))) elif angulo == 3: for j, i in enumerate(coordenadas): if j == 0: val = i - int(x * 0.05) new.append(max(0, int(val))) elif j == 1: val = i - int(y * 0.05) new.append(max(0, int(val)))</pre>

Función	Código
	<pre> elif j == 2: new.append(i - int(x / 4)) elif j == 3: val = i + int(y * 0.05) new.append(min(3000, int(val))) return new </pre>
	<p><i>Explicación:</i> Esta función, obtiene las ubicaciones del vehículo en la imagen, luego de haber sido analizado por YOLO, por lo tanto, analiza primero el ángulo de la imagen y dependiendo eso, recorta el vehículo por la mitad, dejando solo la parte importante, en el caso de la parte trasera izquierda de un camión, toma la mitad del vehículo hacia atrás, y en las demás direcciones aumenta la distancia en un 5% para evitar que el recorte tome características necesarias, pero tomando en cuenta los límites de la imagen.</p>
	<pre> def redim_(coordenadas): x = coordenadas[2] - coordenadas[0] y = coordenadas[3] - coordenadas[1] new = [] for j, i in enumerate(coordenadas): if j == 0: val = i - int(x * 0.05) new.append(max(0, int(val))) elif j == 1: val = i - int(y * 0.05) new.append(max(0, int(val))) elif j == 2: val = i + int(x * 0.05) new.append(min(4000, int(val))) elif j == 3: val = i + int(y * 0.05) new.append(min(3000, int(val))) return new </pre>
	<p><i>Explicación:</i> Al igual que la función <i>redim()</i>, esta toma las coordenadas, y no las recorta, pues son las imágenes laterales, por ende, solo aumenta su tamaño en un 5%, para evitar perder información al recortar la imagen.</p>
	<pre> def recort(imagen, angulo): # Ejecutar detección results = model(imagen) # Extraer resultados boxes = results[0].boxes if boxes is None or boxes.cls is None or len(boxes) == 0: print("No se detectó nada.") return None clases = boxes.cls.cpu().numpy() # Clases detectadas coordenadas = boxes.xyxy.cpu().numpy() # Bounding boxes [x1, y1, x2, y2] # Crear una copia de la imagen original para dibujar sobre ella img_with_boxes = imagen.copy() # Dibujar solo camiones (clase 7 en COCO) max_area = 0 mejor_box = None for i, class_id in enumerate(clases): if int(class_id) == 7: </pre>

Función	Código
	<pre> x1, y1, x2, y2 = map(int, coordenadas[i][:4]) area = (y2 - y1) if area > max_area: max_area = area mejor_box = (x1, y1, x2, y2) if mejor_box: x1, y1, x2, y2 = redim(mejor_box, angulo) imagen = img_with_boxes[y1:y2, x1:x2] return imagen else: print("No se encontró nada") return None </pre>

Explicación: Esta función, toma una imagen y aplica el modelo de YOLO, buscando así la ubicación del objetos en la imagen, hay la posibilidad que no encuentre el nada en la imagen, en este caso mostrará "no se detectó nada", de ahí identifica todas las clases que identificó y las coordenadas, luego crea una copia de la imagen para modificarla, de ahí, identifica que en las clases esté un vehículo, si encuentra más de un vehículo, buscará las coordenadas que den una mayor área, pues las fotos se tomaron enfocadas en los vehículos, y por eso se miran más grandes. Posterior a eso, se toman las coordenadas, y se las redimensionan con la función *redim()*, específicamente para las imágenes de las esquinas, obteniendo así las coordenadas que recortaran las imágenes en las zonas requeridas.

Para el caso que las imágenes que se quieren redimensionar son las laterales, se utiliza el mismo código, pero la función para redimensionar es la especial para estas imágenes, denominada *redim_l()*.

```

def more_tall(formas):
    # Initialize alto and ancho with small values
    alto = 0
    ancho = 0
    # Iterate through each shape in the list
    for shape in formas:
        # Update alto if the current shape's height is greater
        if shape[0] > alto:
            alto = shape[0]
        # Update ancho if the current shape's width is greater
        if shape[1] > ancho:
            ancho = shape[1]
    return alto, ancho # Return the true maximum alto and ancho

```

Explicación: En esta función toma todas las resoluciones de las imágenes recortadas, para evitar perder información al redimensionarlas o alterar su estructura, se opta por mantener los recortes en su misma resolución, escoge cual imagen es la más grande y guarda esta resolución.

```

def _resize(imagen, alto, ancho):
    nuevo_ancho = ancho
    nuevo_alto = alto
    alto = imagen.shape[0]
    ancho = imagen.shape[1]

    # Calcular cuántos píxeles faltan
    top = (nuevo_alto - alto) // 2
    bottom = nuevo_alto - alto - top
    left = (nuevo_ancho - ancho) // 2
    right = nuevo_ancho - ancho - left

```

Función	Código
	<pre># Agregar bordes (relleno negro) imagen = cv2.copyMakeBorder(imagen, top, bottom, left, right, cv2.BORDER_CONSTANT, value=(0, 0, 0)) return imagen</pre>
	<p><i>Explicación:</i> Con esta función permite obtener las imágenes ya recortadas, y también las dimensiones más grandes por cada grupo de imágenes, por eso, en el caso que la imagen seleccionada sea más pequeña en alguna dirección, se aumenta un relleno negro, evitando así redimensionar imágenes a resoluciones que afecten su forma.</p>
	<p>Finalmente, con el código:</p> <pre>image_dir = r"ruta_carpeta/Imagenes" salida = r"ruta_carpeta/Unidasx4"</pre>
	<pre>for i in vehiculos: data = df[df["Vehiculo"] == i] image_ids_for_vehicle = []</pre>
	<pre># Recolectar todos los IDs de imagen del vehículo actual for union_list in data["Uniones"].apply(ast.literal_eval): image_ids_for_vehicle.extend(union_list) id_img = list(set(image_ids_for_vehicle))</pre>
	<pre># Cargar imágenes en un diccionario image_dict = {} for img_id in id_img: img_path = os.path.join(image_dir, f"{str(img_id)}.jpg") imag = cv2.imread(img_path) if imag is None: print(f"Imagen no encontrada o inválida: {img_path}") continue try: imag = cv2.resize(imag, (4000, 3000)) image_dict[img_id] = imag except Exception as e: print(f" Error al redimensionar {img_path}: {e}")</pre>
	<pre>nombres = data["Uniones"].index</pre>
	<pre>for n, lista in enumerate(data["Uniones"].apply(ast.literal_eval)): name = nombres[n] error = False imagenes = [] forma = []</pre>
	<pre>for z, img_id in enumerate(lista): if img_id not in image_dict: print(f"ID {img_id} no encontrado en image_dict") error = True break</pre>
	<pre>original_img = image_dict[img_id] if len(lista) == 2: procesada = recort_(original_img) else: procesada = recort(original_img, z)</pre>
	<pre>if procesada is None: print(f" recorte fallido para ID: {img_id}, índice: {z}") error = True</pre>

Función	Código
	<pre> break imagenes.append(procesada) forma.append(procesada.shape) if error or len(imagenes) != len(lista): continue alto, ancho = more_tall(forma) try: imagenes = [_resize(img, alto, ancho) for img in imagenes] except Exception as e: print(f" Error en redimensionamiento de imágenes: {e}") continue try: if len(imagenes) == 2: combinada = cv2.vconcat(imagenes) elif len(imagenes) == 4: fila_sup = cv2.hconcat([imagenes[1], imagenes[0]]) fila_inf = cv2.hconcat([imagenes[2], imagenes[3]]) combinada = cv2.vconcat([fila_sup, fila_inf]) else: print(f" Número de imágenes inesperado: {len(imagenes)}") continue if combinada is None: print(" Error al combinar imágenes") continue combinada = cv2.resize(combinada, (1500, 1500)) path_salida = os.path.join(salida, f"{name}.jpg") cv2.imwrite(path_salida, combinada) except Exception as e: print(f" Error al combinar o guardar imagen {name}: {e}") </pre>

Explicación: Por medio de este código se llaman a todas las funciones y hace que las imágenes sean procesadas. Primero guarda las direcciones de las carpetas de trabajo, donde toma las imágenes originales y donde guarda las imágenes generadas.

Luego por cada vehículo, selecciona las imágenes que van a ser procesadas y unidas, para cargarlas en el diccionario *image_dict*, con la resolución original de 4000 x 3000 píxeles, así mismo con el nombre de la imagen a generar.

Luego selecciona las imágenes a unir por cada nueva imagen, de ahí, analiza si las relaciones son en pares o no, para procesar las imágenes con la función *recort_1()*, con las pares y la función *recort()* para el caso de los diferentes ángulos, especificando el mismo en la función, de ahí, las imágenes se guardan en una variable *imagenes* y también la forma de estas, pues ya fueron recortadas. Luego se obtiene el valor de alto y ancho más grandes con la función *more_tall()*, para así pasar las imágenes por la función *_resize()*, que pone marcos negros para igualar la forma de la imagen más grande del conjunto.

Posteriormente se toman las imágenes ya procesadas con las mismas dimensiones y se las une con las funciones de *cv2.vconcat()* para imágenes pares, poniendo una encima de otra, y para las imágenes de los 4 ángulos, con *cv2.hconcat()* une las imágenes a un lado de la otra, esto por la parte delantera y trasera para posteriormente unir estas dos una encima de la otra.

Función	Código
	Por último, toma la imagen combinada, la redimensiona a 1500 x 1500 píxeles, para hacer que el <i>dataset</i> generado no sea pesado con el <i>dataset</i> original y finalmente genera la imagen en la carpeta especificada al inicio, con el nombre correspondiente.

En la Tabla 19, se tiene un código extenso, que permite tomar las imágenes de un vehículo, buscar la parte más importante de este con base en su ángulo de captura de imagen, para unir en una sola imagen las 4 partes del mismo vehículo con el mismo peso, y así generar un *dataset* que trate de forzar a la red a buscar las características que afecten por el peso, en la parte de las llantas de los vehículos. Resultando en imágenes como las que se puede apreciar en la Figura 18.



Figura 18. Imágenes unidas entre sus 4 esquinas y de manera lateral.

En esta figura 18, se muestra imágenes tipo mosaico, en las que cada una se compone específicamente de 4 sub imágenes, mostrando cada una, únicamente una parte específica del vehículo, eliminando información irrelevante para el entrenamiento, además, también se mira las imágenes tomadas de manera lateral, la decisión radica en que estas imágenes no muestran una parte del vehículo con más detalle, si no, muestra de forma similar todo un lado en la imagen, por lo tanto, se optó por unir las de la misma manera, en este caso, a diferencia del anterior *dataset*, que se recortaba la parte superior de las imágenes, se pudo recortar cada imagen alrededor del vehículo, gracias al modelo de YOLO, que permitió identificar la posición de los vehículos en las imágenes.

4.1.2.4.4. Generar *datasets* de entrenamiento y prueba

Para este caso en específico, el problema de tener un *dataset* con una cantidad de datos baja, no permite dividirlo en datos de entrenamiento, validación y prueba, los datos de validación sirven para examinar el comportamiento del modelo que va teniendo a lo largo de las épocas, sin embargo, al tener pocos datos se optó por generar solo dos *datasets*, uno para el entrenamiento y otro para prueba.

El generar los *datasets* con anterioridad, permite que no se consuman recursos computacionales al momento de entrenar el modelo, pues los datos ya se encontrarán mezclados y redimensionados, evitando así cuellos de botella al momento de generar los modelos.

4.1.2.4.4.1. Procesamiento de bases de Excel

Primero, mediante los Excel de cada uno de los *datasets*, se separan en datos de entrenamiento con el 80% de datos y de prueba con el 20%, pero esto no se lo hace con base en todas las imágenes, sino, al registro de vehículos, así se asegura que los datos de prueba sean totalmente diferentes a los datos de entrenamiento.

El proceso que se siguen mediante código de *Python* es el siguiente:

```
# Cargar nombres de imágenes
image_dir = r"dirección de carpeta donde se encuentren las imágenes"
image_files = os.listdir(image_dir)
image_files = pd.Series(image_files).replace(to_replace='.jpg',
                                             value="", regex=True).astype(int).sort_values().astype(str)
# df = pd.read_excel(r"dirección de archivo excel.xlsx")
```

En estas líneas de código, se obtiene los nombres de las imágenes que se encuentran en la carpeta, luego se lee un archivo de Excel, correspondiente a la carpeta de imágenes.

```
#Filtrar datos
image_files = list(image_files.astype(int))
df_filtrado = df[df['Id'].isin(image_files)]
```

Se transforma los nombres de los archivos a enteros. Luego, se filtra los nombres de las imágenes con los nombres almacenados en el dataframe, para evitar posibles errores.

```
data = df_filtrado[["Vehiculo", "Peso", "Lugar"]]
data = data.drop_duplicates().reset_index(drop=True)
```

Se selecciona las columnas del registro de vehículo, el peso y el lugar del *dataframe*, para luego eliminar los duplicados, así, se obtiene solamente los datos por número de vehículo y el peso asociado.

```
X = data.drop(columns=['Peso'])
```

```
y = data['Peso']
```

```
# Dividir en entrenamiento y test
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Divide el *dataframe* en variables X para seleccionar solo el vehículo y lugar, mientras la variable y seleccionar la variable del peso, de ahí, con la función de sklearn, `train_test_split`, se divide el 80% de los datos para entrenamiento y el 20% de los datos para prueba

```
# Obtener datos de entrenamiento
```

```
vehi = list(X_train['Vehiculo'])
```

```
df_train = df_filtrado[df_filtrado['Vehiculo'].isin(vehi)]
```

```
# Obtener datos de validacion
```

```
vehi = list(X_test['Vehiculo'])
```

```
df_test = df_filtrado[df_filtrado['Vehiculo'].isin(vehi)]
```

Ahora, ya separado los datos por el tipo de vehículo, se toma los datos del *dataframe* inicial para aumentar al *dataset* las imágenes correspondientes por cada registro de vehículo.

```
df_train.sample(frac=1, random_state=42).reset_index(drop=True)
```

```
df_test.sample(frac=1, random_state=42).reset_index(drop=True)
```

Con la anterior línea de código, los datos se ordenaban con base en el *dataframe*, original, por ende, se usa `df.sample()`, para mezclar todos los datos.

```
df_train.to_excel(r"ruta de salida para excel.xlsx", index=False)
```

```
df_test.to_excel(r"ruta de salida para excel.xlsx", index=False)
```

Finalmente, se guardan los datos en formato Excel, obteniendo así, de un *dataset*, los datos ya mezclados y listos para ser cargados en un archivo tipo *TFRRecord*.

Este proceso se lo realiza con los 3 *dataframes*, cambiando la dirección de la carpeta de imágenes, y la dirección del archivo Excel correspondiente, obteniendo así *datasets* únicos para cada *dataframe*, pues cada uno divide los datos de forma independiente, y con eso obtiene bases con las características mostradas en la Tabla 20.

Tabla 20. Cantidad de imágenes por *dataset*

Nombre <i>dataset</i>	Cantidad de imágenes	Registro Vehicular
<i>train_normales</i>	11349	1112
<i>test_normales</i>	2895	278
<i>train_unidasx2</i>	13013	1111
<i>test_unidasx2</i>	3349	278
<i>train_unidasx4</i>	11207	1094
<i>test_unidasx4</i>	2851	274

En la Tabla 20, se tiene el conjunto de imágenes de cada *dataset*. La cantidad del registro de vehículos con el que se formó cada uno, teniendo en cuenta que, en cada registro, se tomaron varias imágenes por vehículo, permitiendo mostrarlo desde cada ángulo, con respecto al número de datos son diferentes entre sí, a pesar de basarse en la misma base de imágenes, debido al procesamiento que tuvieron, lo que permitió tener 3 *datasets* diferenciados por la forma de la imagen. Los datos con el nombre “normales”, son las imágenes que no sufrieron alteraciones, las que tienen el nombre de “unidasx2”, se unieron las imágenes en pares, tomando los lados derecho e izquierdo, y las que hacer referencia a “unionesx4”, son aquellas en las que las imágenes se unen los 4 ángulos de un vehículo en una sola, así mismo, la cantidad de vehículos se ve afectada, pues en las unionesx2, un vehículo que no cumple con tener una imagen del par de adelante, atrás y lateral, este registro no es tomado en cuenta, en las unidasx4, sufren más pérdidas, a razón que al hacer falta de un ángulo el vehículo no se toma en cuenta todo el registro. Además, se toma en cuenta que los *datasets* con la denominación *train*, refiere a los datos usados en los entrenamientos y *test*, a los datos para prueba.

4.1.2.4.4.2. Generación de *datasets* en formato *TFRecord*

Ya teniendo una base en Excel con los datos mezclados y separados por entrenamiento y prueba, es más fácil generar los *datasets*, pues solo debe ir cargando las imágenes y almacenarlas secuencialmente, ya que no requiere de mezclar los datos o manipularlos. Con esto, se almacena las imágenes en el formato “.*tfrecord*”, específico para almacenar datos de Tensorflow, con el fin de evitar todo

este proceso al momento de realizar los entrenamientos y simplemente subir los datos de manera óptima.

Para convertir los datos a formato *TFRecord*, se usa el siguiente código:

```
def _bytes_feature(value):
    """Convierte un valor en una característica en formato byte para TFRecords"""
    return tf.train.Feature(bytes_list=tf.train.BytesList(value=[value]))

def _int64_feature(value):
    """Convierte un valor en una característica en formato int64 para TFRecords"""
    return tf.train.Feature(int64_list=tf.train.Int64List(value=[value]))

def serialize_example(name,image,label):
    """Convierte una imagen y su etiqueta en un formato de ejemplo TFRecords"""
    feature = {
        "name" : _int64_feature(name),
        "imagen" : _bytes_feature(image),
        "label" : _int64_feature(label)
    }
    example_proto = tf.train.Example(features=tf.train.Features(feature=feature))
    return example_proto.SerializeToString()

def convert_images_to_tfrecord(image_folder,
label_file,output_folder,output_filename):
    #Cargar etiquetas
    df = pd.read_excel(label_file)
    labels_dict = dict(zip(df.iloc[:,0], df.iloc[:,1]))
    # Crear la carpeta de salida si no existe
    os.makedirs(output_folder, exist_ok = True)
    output_path = os.path.join(output_folder, output_filename)
    writer = tf.io.TFRecordWriter(output_path)
    for file in list(df["Id"]):
        img_path = os.path.join(image_folder,f"{str(file)}.jpg")
        img = cv2.imread(img_path)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        img = cv2.resize(img, (1500, 1500))
        _, img_encoded = cv2.imencode(".jpg", img) #convertir imagen a bytes
```

```

label = int(file)
label = labels_dict.get(label, 0) # Obtener la etiqueta o asignar 0
example = serialize_example(int(file),img_encoded.tobytes(), label)
writer.write(example)
writer.close()
print(f"TFRecords guardado en {output_path}")
# Rutas
image_folder = r"ruta específica para la carpeta de las imágenes"
label_file = r"ruta específica para el archivo excel relacionado.xlsx "
output_folder = r"Carpeta de salida para almacenar el modelo "
output_filename = r"nombre_dataset.tfrecord "
convert_images_to_tfrecord(image_folder, label_file, output_folder,output_filename)

```

Explicación: En resumen, este código lee un archivo Excel el cual contiene los Id (Identificaciones) de las imágenes y las etiquetas, luego busca cada imagen en la carpeta especificada en *image_folder*, para convertirla en RGB y redimensionar las imágenes a 1500 x 1500 píxeles, posteriormente codifica la imagen como JPEG con el fin de obtener su representación en bytes. De ahí, crea un diccionario con el nombre del archivo (ID), la imagen codificada, y su etiqueta. Para pasar a serializar los datos en formato ". *tfrecord*" para poder hacer una lectura eficiente por parte de Tensorflow y posterior a eso guarda el archivo en la carpeta indicada en la variable *output_folder* con el nombre insertado en *output_filename*.

El tener estos archivos *TFRecord*, permite subir el *dataset* directamente en el entrenamiento del modelo, sin consumir recursos computacionales para mezclar datos, o redimensionar las imágenes, logrando tener un archivo con bajo peso, al haber sido redimensionadas con anterioridad, evitando que se generen cuellos de botella en el CPU mientras carga los datos, no aprovechando la capacidad de la GPU. De ahí, se observa en la Tabla 21, los *datasets* listos para los entrenamientos de los modelos y con su respectivo tamaño de almacenamiento.

Tabla 21. *Datasets* en formato *TFRecord* y su tamaño de almacenamiento.

Nombre <i>dataset</i>	Peso en memoria
<i>df_train_normales</i>	7.13 GB
<i>df_test_normales</i>	1.81 GB
<i>dt_train_unidasx2</i>	9.67 GB
<i>df_test_unidasx2</i>	2.46 GB
<i>df_train_unidasx4</i>	7.74 GB
<i>df_test_unidasx4</i>	1.97 GB

4.1.3. Diseño del sistema de pesaje con Inteligencia Artificial.

La arquitectura de CNNs que se escogió es ResNet-50, por su adaptabilidad y buen desempeño en comparación con otras redes, el único inconveniente es que el modelo recibe imágenes en una resolución de 224 x 224 píxeles, pero las imágenes de los vehículos se busca una resolución alta para que la información que se encuentra en partes específicas, como las llantas, no se llegue a perder significativamente, por lo que, la red debe ser capaz de recibir imágenes superiores a una resolución de 1000 x 1000 píxeles. Por lo tanto, tomando en cuenta la estructura original con 5 bloques de conjuntos de convoluciones, en el quinto bloque los mapas de activación tienen una resolución de 7 x 7 píxeles, esto muestra que este tamaño es óptimo al finalizar la red y evitar que el computador sature la memoria, dividiendo el bloque 4, el cuál es el que tiene 6 capas, se divide en 3 bloques y a cada uno se le aumenta 2 conjuntos de convoluciones, obteniendo así una versión de ResNet-68, esta se implementará en cada uno de los *datasets*, la configuración visual de las *ResNet* se observan en la Figura 19.

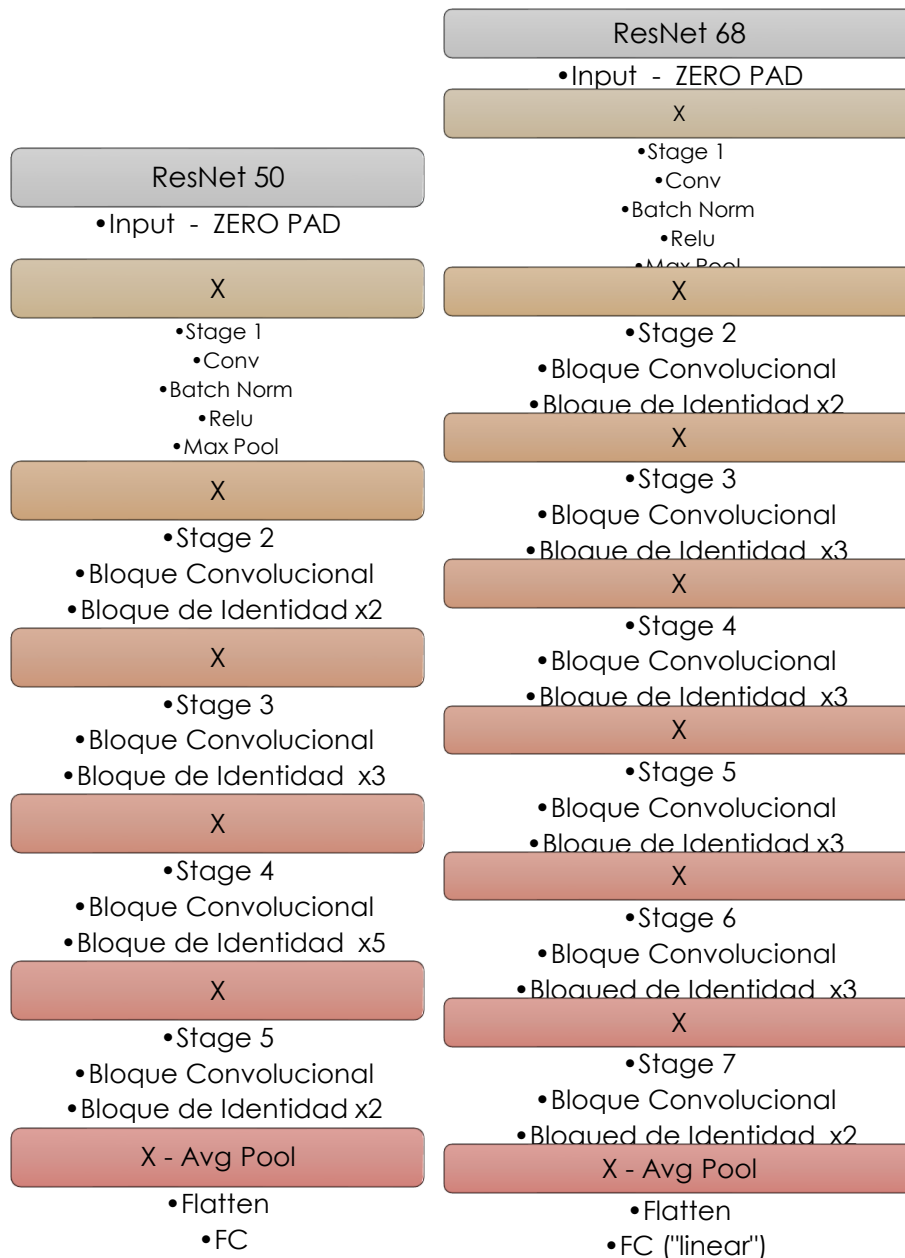


Figura 19. Arquitectura ResNet 50 y ResNet-68 (Adaptada)

Como se aprecia en la Figura 19, a la red se le han aumentado 2 bloques más, estos tienen la misma configuración que los demás, haciendo que, en el bloque convolucional, la imagen sea reducida a la mitad, pues su función es recibir la imagen y aplicar capas de convolución, al final reduce los mapas de activación a la mitad, de ahí pasan por los bloques de identidad, que pasan las entradas por más convoluciones. Esta red adaptada, aumenta la profundidad, no siendo un problema, debido a que el modelo de *ResNet* funciona mejor añadiendo más profundidad a la red, sin embargo, aumentará el tiempo de entrenamiento, otra diferencia es que al ser un problema de regresión, al final de la capa *fullyconnect(fc)*, se cambia "softmax"

por “linear” permitiendo buscar un solo valor por medio de regresión lineal, el código utilizado para ResNet-68 se encuentra en el Anexo 9.

Una vez teniendo listo la red es necesario llamar a los *datasets* para poder utilizarlos en la red. Para esto, se utiliza el código que permite leer archivos en formato *TfRecord*, siendo el detallado aquí:

```
def parse_tfrecord(example_proto):  
    """Decodifica un TfRecord para recuperar la imagen y la etiqueta"""  
    features = {  
        'name': tf.io.FixedLenFeature([], tf.int64),  
        'imagen': tf.io.FixedLenFeature([], tf.string),  
        'label': tf.io.FixedLenFeature([], tf.int64)  
    }  
    parsed_features = tf.io.parse_single_example(example_proto, features)
```

Esta parte del código permite obtener los datos de los *datasets* ya precargados, el nombre lo carga en formato “*int64*” así como el *label* (peso), y la imagen la recupera como formato *string* (en formato JPEG).

```
    # Decodificar la imagen  
    image = tf.io.decode_jpeg(parsed_features['imagen'])  
    image = tf.image.rgb_to_grayscale(image)  
    image = tf.image.resize(image, (alto, ancho))  
    image = tf.cast(image, tf.float32)/255.0  
  
    name = parsed_features['name']  
  
    label = parsed_features['label']  
    return image, label o name
```

En esta parte del código, toma las imágenes de los datos recuperados anteriormente, la convierte a escala de Grises y redimensiona a una resolución seleccionada para cada *dataset*, luego normaliza las imágenes (entre 0 y 1), para devolver las imágenes y sus respectivas etiquetas del peso.

```
def load_tfrecord(tfrecord_path, batch_size):
```

```

"""Carga un archivo TFRecord y lo devuelve como un dataset de TensorFlow"""
    raw_dataset = tf.data.TFRecordDataset(tfrecord_path,
num_parallel_reads=tf.data.AUTOTUNE)
    raw_dataset = raw_dataset.batch(batch_size)
    dataset = tf.data.TFRecordDataset(tfrecord_path)
    dataset = dataset.map(parse_tfrecord, num_parallel_calls=tf.data.AUTOTUNE)
    dataset = dataset.shuffle(500, seed=42)
    return dataset.batch(batch_size).prefetch(tf.data.AUTOTUNE), raw_dataset

```

`tfrecord_path = r"dirección del archivo .tfrecord"`

`test_dataset, raw_test_dataset = load_tfrecord(tfrecord_path_test, batch_size = 16)`

Mediante este código se carga el archivo en *format.tfrecord*, desde una ruta especificada en la variable *tfrecord_path*, se generan dos *datasets*, el uno como *raw_dataset*, que solo agrupa por lotes, permite saber la cantidad exacta de lotes con el que se va a trabajar, y el *dataset*, el cual es transformado con la función *parse_tfrecord*, para posteriormente ser aleatorizado, pero de a 500 imágenes, y con *tf.data.AUTUNE*, permite optimizar la carga de las imágenes y las organiza en lotes.

Al final solo se llama a la función *load_tfrecord*, que realiza todos los pasos anteriores, en esta función se ingresa la ruta del archivo y el número de imágenes por lote (*batch_size*).

Es importante aclarar los siguientes puntos:

Al cargar el *dataset* y no especificar que se guarde en cache, se hace que la computadora cargue las imágenes directamente desde el disco donde se encuentran almacenadas, las imágenes se van subiendo de acuerdo a la cantidad especificada en el lote, lo cual, la capacidad del CPU del computador usado permite realizarlo de un modo adecuado, sin generar cuellos de botella en la GPU, también el código optimiza la carga de lotes, al precargar lotes en la memoria cache para posteriormente pasar a la tarjeta gráfica.

Este código se utiliza para todos los *datasets*, con pequeños cambios para cada uno en especial, de los datos de entrenamiento se obtiene la imagen y el *label*, pero en los datos de prueba se obtiene la imagen y el nombre, para que su posterior análisis sea más fácil de interpretar, así mismo dependiendo de los 3 tipos de *datasets*, se selecciona una resolución diferente y de igual manera un número de imágenes por lote que permita cargar las imágenes sin saturar la VRAM de la GPU, tomando en

cuenta el no tomar menos de 8 imágenes por lote, para asegurar un entrenamiento eficiente. Con esto en cuenta, se cargan 3 modelos diferentes, y las configuraciones de cada una se muestran en la Tabla 22.

Tabla 22. Configuración por modelo a generar

Modelo	Dataset de entrenamiento usado	Resolución (ancho, alto, canal)	Tamaño de lote	Parámetros (alto x ancho x lote)
Modelo 1	<i>df_train_normales</i>	750 x 1000 x 1 (escala de grises)	16 imágenes por lote	12 000 000
Modelo 2	<i>dt_train_unidasx2</i>	1200 x 1200 x 1 (escala de grises)	12 imágenes por lote	17 280 000
Modelo 3	<i>df_train_unidasx4</i>	1400 x 1400 x 1 (escala de grises)	8 imágenes por lote	15 680 000

La resolución de la imagen fue aumentada con respecto a las uniones de imágenes que se realizaron, como se mira en la Tabla 22, pues al unir 2 imágenes su altura incrementa al doble, por lo tanto, no se puede redimensionar a la misma resolución de las imágenes normales, lo que genera tendencia a perder más información, por ende se trata de aumentar la resolución de estas para evitar eso, y se lo logra reduciendo el número de imágenes por lote, en el caso del modelo 3, tiene la misma lógica, uniendo 4 imágenes en una, por lo tanto, se logró incrementar la resolución hasta 1400 x 1400 píxeles, sin hacer que el lote sea menor a 8 imágenes.

Las resoluciones, se escogieron tras analizar cuál era la cantidad de parámetros permitidos por la VRAM de la GPU, con estos datos se obtuvo que, esta computadora permite entrenar modelos de hasta 17 280 000 parámetros, representando la mayor cantidad de los 3 modelos que tuvo un alto rendimiento y no saturó la VRAM, este comportamiento se lo observa en la Figura 22.

4.1.3.1. Entrenamiento

El entrenamiento de los modelos se llevó a cabo con la función de pérdida con MSE (Error cuadrático medio), métrica que utiliza la red para calcular el promedio del cuadrado de errores, esta métrica se hace diferenciable en todos los puntos y por lo tanto, es ideal para el entrenamiento con *backpropagation*, además penaliza más los errores grandes y por eso ayuda al modelo a aprender mejor, al tratar de reducirlo en cada época, además se incluye la métrica MAE (Error absoluto medio), para tener otra referencia en el comportamiento de los datos, esta métrica al no penalizar los datos, muestra el error que está teniendo la red al predecir los pesos, esta no afecta

en el entrenamiento, pero da otro punto de vista con respecto a los datos históricos. En la Figura 20, se observa el gráfico de pérdida del entrenamiento de los 3 modelos.

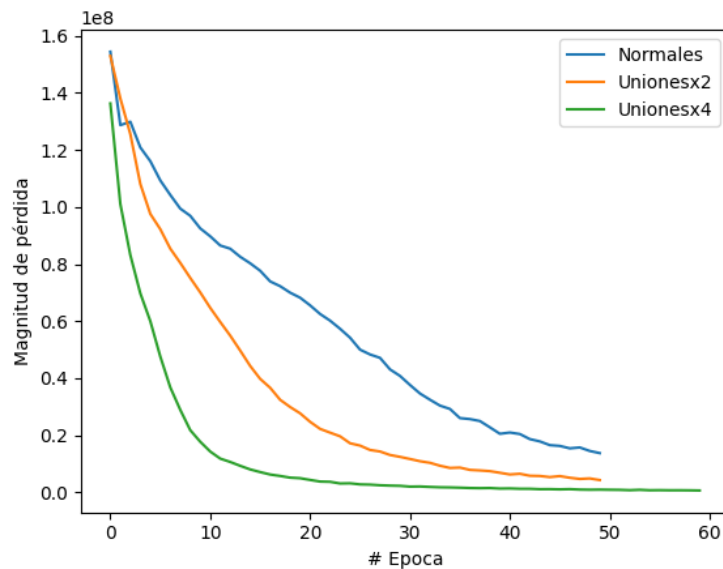


Figura 20. Gráfico de pérdida por modelos entrenados

El entrenamiento de los 3 modelos que se muestra en la Figura 20, tiene un comportamiento muy diferente al comparar los 3 modelos en 50 épocas, tras tener presente que la magnitud de pérdida se encuentra en una escala $1e8$, se comprende que la diferencia con respecto al error de cada modelo al final del entrenamiento es muy elevado, donde el modelo 1 se encuentra próximo al valor 0.2 de error, que corresponde a 13760477 kg^2 de error MSE como último dato histórico, en el caso del modelo 2, se observa errores más bajos y se empieza a notar una función exponencial negativa, al mostrar en la gráfica suavidad en sus cambios, en este caso el último valor histórico es 4321344 kg^2 de error MSE, aún muy elevado, por último, el modelo con mejores resultados en el entrenamiento en función de su pérdida es el modelo 3, al mostrar que disminuye su error MSE más rápido que los otros dos modelos, entre las épocas 10 y 20, de ahí, hasta la época 50 que se acerca más al valor 0 de manera constante, sin embargo sigue teniendo un error elevado, que no se puede visualizar por la escala de la gráfica, dado que su último valor del histórico es 665443 kg^2 en error MSE, que al comparar con el modelo 2, representa un 84.6% menos de error. Ahora, evidenciando que el modelo 3 tuvo mejores resultados con respecto a la pérdida de error en la métrica MSE, también se analiza el comportamiento en base a la métrica MAE, mismo que se ve en la Figura 21.

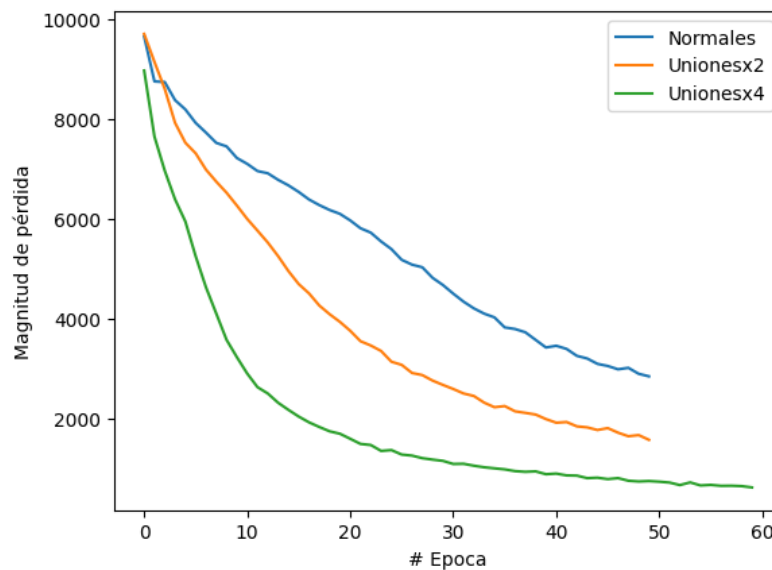


Figura 21. Gráfico histórico de la métrica MAE a lo largo de las épocas.

El gráfico con la métrica MAE, que muestra la Figura 21, tiene un comportamiento muy similar a la Figura 20, pero en este caso, los valores son más pequeños, al no penalizar los errores más grandes, obteniendo información muy importante con respecto a su comportamiento, hasta la época 50, en el caso del modelo con imágenes normales, el error es de 2838.48 *kg*, siendo este un error demasiado elevado, en el caso del modelo con imágenes unidas en pares, reduce a 1567.08 *kg*, y el último modelo, al que se unieron 4 imágenes en una, se obtuvo el error más bajo, además que este tuvo un entrenamiento a 60 épocas, su valor más bajo en el historial es de 614.03 *kg*, reduciendo en 78.3 % con respecto al modelo 2, aun así, el error sigue siendo demasiado elevado, eso tomando en cuenta que las básculas camioneras tienen un error permisible de ± 30 *kg*.

Con estos gráficos, se tiene en cuenta, que el modelo tiene un mejor entrenamiento al modificar las imágenes, en las imágenes normales, se le da a la red imágenes con todos los puntos de apoyo, para que analice ambas partes, y en el modelo 3 el conjunto de imágenes que toman lo más importante de cada imagen, hace que la red, este forzada a encontrar los patrones relacionados con el peso más rápido, siendo así, este modelo el más eficiente con respecto a los entrenamientos.

En el entrenamiento de los 3 modelos, se aprovechó la mayor cantidad de recursos de la computadora, pues en cada uno de ellos, el trabajo de la GPU fue muy elevado, tal como se muestra en la Figura 22.

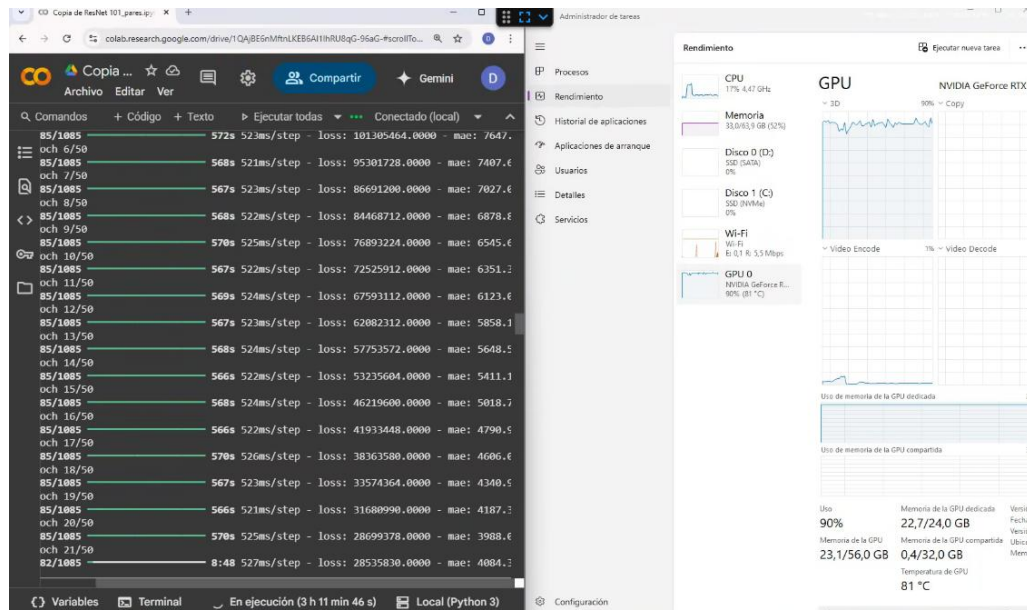


Figura 22. Entrenamiento y consumo de recursos de computador

En la Figura 22 se presenta el consumo de memoria se aprovechó al máximo, gracias al precargado de las imágenes, teniendo listo el *dataset* a una resolución de imágenes baja, este comportamiento se miró en los 3 entrenamientos, aunque la imagen representa más específicamente del modelo 2, siendo la optimización del *dataset* muy bueno, muestra que precarga los datos necesarios a la memoria RAM y así la GPU aprovecha su máximo potencial. Con esto, también se obtuvo más datos sobre el entrenamiento de los modelos, que se pueden analizar en la Tabla 23.

Tabla 23. Características del entrenamiento por modelo.

Modelo	Parámetros totales	Parámetros entrenables	Parámetros no entrenados	Parámetros optimizados	Tiempo de entreno
Modelo 1	64,720,516 (246.89 MB)	32,337,793 (123.36 MB)	44,928 (175.50 KB)	32,337,795 (123.36 MB)	Promedio (épocas: 2 → 11) ≈ 246 seg. Total Seg ≈ 12 300 Horas ≈ 3.4
Modelo 2	64,773,764 (247.09 MB)	32,364,417 (123.46 MB)	44,928 (175.50 KB)	32,364,419 (123.46 MB)	Promedio (épocas: 2 → 11) ≈ 569 seg. Total Seg ≈ 28 450 Horas ≈ 7.9
Modelo 3	64,773,764 (247.09 MB)	32,364,417 (123.46 MB)	44,928 (175.50 KB)	32,364,419 (123.46 MB)	Promedio (épocas: 2 → 11) ≈ 649 seg. Total Seg ≈ 38 940 Horas ≈ 10.8 (60 épocas)

En la Tabla 23, se muestra que el modelo 2 contiene la misma cantidad de parámetros totales que el modelo 3 con un peso de 247.09 MB, estos parámetros se basan en la

configuración de la red neuronal ResNet y la resolución de las imágenes, que a pesar de ser diferente el tamaño de los datos de entrada, en la capa de *average pooling* se obtiene mapas de activación de 5 x 5, esto porque se reduce el tamaño de las imágenes a la mitad en cada *stage*, mostrando que la resolución entre las imágenes del modelo 2 y 3, entran en un rango donde el modelo tendrá la misma cantidad de parámetros, mientras que la resolución usada para el modelo 1 es más baja, obteniendo una menor cantidad de parámetros. Sin embargo, con respecto al tiempo, este varía porque depende tanto de la resolución de las imágenes, como de la cantidad, en este aspecto el modelo 3 es más eficiente en el entrenamiento al demorarse por cada época en promedio 80 segundos menos que el modelo 2, pero 403 segundos más con respecto al modelo 1, esto principalmente por tener una resolución más alta, siendo la cantidad de imágenes no representativa, al ser entrenado el modelo 3 con 142 imágenes menos.

Considerando los modelos 2 y 3 los que tuvieron mejor rendimiento, este último es más liviano y se entrena en un menor tiempo, pues si hubiese sido entrenado en las mismas 50 épocas su duración sería de aproximadamente 7.6 horas, lo que representa 0.3 horas o 18 minutos menos que el modelo 2. Al usar 50 épocas no tiene demasiado peso, pero como se miró en las gráficas de pérdida de error, era una función exponencial negativa, que aún no llegaba a un valor aceptable, cercano a cero, por lo tanto, al aumentar el número de épocas significa mayor tiempo de entrenamiento.

Al finalizar los entrenamientos, se usó el siguiente código para descargar los modelos: `model.save('modelo_resnet_unidosx4.keras')`, permitiéndolo guardar en un formato nativo de *tensorflow*.

4.1.4. Evaluación del rendimiento del programa

Para evaluar el rendimiento del programa, primero se definió cual es el modelo que mejores resultados tenía con el conjunto de pruebas, datos que los modelos no han visto en el entrenamiento, para esto se usó las mismas métricas de rendimiento que se empleó en los entrenamientos, pues permiten saber que tan alejadas están las predicciones con respecto a los pesos reales.

Primero, se carga los *datasets* de prueba, tal como se hizo con los datos de entrenamiento, solo que, en este caso, no se llama a la imagen con el *label*, si no, con el nombre de la imagen, esto con el código que permite extraer datos de archivos en formato *TfRecord*.

Luego se carga el modelo y se guarda en una variable con el código de *tensorflow*: `modelo = load_model(r"ruta/ modelo_resnet_unidosx4.keras")`, para usarlo y predecir en los datos de los *dataset* de entrenamiento con los modelos usando el código: `predicciones = modelo.predict(test_dataset)`.

Esto da una variable con las predicciones del *dataset* y se obtiene otra variable que corresponde al nombre, esto con el código:

```
imagen = []
for image, name in test_dataset.take(test_size):
    for i in range(len(image)):
        imagen.append(name[i])
prediccion = list(prediccion)
```

Ya con las dos variables listas, se genera un *dataset* uniendo estas dos variables y se convierten en un Excel, como se ha desarrollado anteriormente con otras bases. Así, permite en el Excel de los datos de prueba relacionarse con los datos predichos mediante el nombre de la imagen. Obteniendo una base como la que se muestra en la Tabla 24.

Tabla 24. Excel con datos de la imagen y predicciones

Id	N_vehiculo	Pesounico	N_fotos	Angulo	Visibilidad	Lugar	tiempo	predicción	Id_tipo_veh	tipo
131	24	31220	0	di	si	TransBolivariana		25722,66	4	1
132	24	31220	0	dd	si	TransBolivariana		25642,46	4	1
133	24	31220	0	dd	si	TransBolivariana		23000,22	4	1
134	24	31220	0	ti	si	TransBolivariana		33824,01	4	1
135	24	31220	0	ti	si	TransBolivariana		32444,28	4	1
136	24	31220	0	td	si	TransBolivariana		37164,69	4	1
137	24	31220	0	td	si	TransBolivariana		41487,38	4	1
138	24	31220	8	td	si	TransBolivariana		22625,37	4	1
177	30	27678	0	ti	no	TransBolivariana		39627,31	6	0
:	:	:	:	:	:	:	:	:	:	:
14249	1389	7520	8	ti	si	Midecar		20428,25	1	1

La Tabla 24. representa una base ya relacionada con todos los datos, donde se encuentra el Id de la imagen de la base de prueba, el registro de vehículo al que pertenece, el peso real, en este caso el ángulo, pues la tabla de ejemplo representa la base con imágenes normales, el lugar de captura de la imagen, el id del tipo de vehículo, y la clasificación solamente para los dos tipos.

Este proceso de predicción y generación de bases con los resultados, se lo realiza a cada modelo, lo cual permite su análisis e interpretación con las métricas MAE y MSE.

4.1.4.1. Interpretación de los resultados

Con las bases de cada uno de los resultados, el procedimiento para obtener las métricas es leer los archivos Excel, seleccionar ambas columnas, la de predicción y la real para implementar el código:

```
import numpy as np
df = pd.read_excel(r'ruta_archivo.xlsx')
y = df["Peso"].values
yhat = df["prediccion"].values
x = list(range(len(y)))
d = y - yhat
mse_f = np.mean(d**2)
mae_f = np.mean(abs(d))
print("Resultados de modelo")
print("MSE: ", mse_f)
print("MAE: ", mae_f)
```

Para el caso de segmentar los datos, el dataframe es filtrado con el código:

```
df = df[df["Tipo vehículo"] == "Camiones"]]
```

Esto antes del anterior código, y el procedimiento se realiza con cada uno de los *datasets* de entrenamiento, para obtener los resultados de las métricas. Con esto mencionado, el resultado general, sin filtrar los datos y por los 3 modelos se presenta en la Tabla 25.

Tabla 25. Estimación del peso de los vehículos por tipo de modelo.

Modelo	MAE (kg)	MSE (kg ²)
Modelo 1	6591.49	88615248.92
Modelo 2	6698.32	79141742.40
Modelo 3	5345.16	57277464.00

El modelo 3, es el que tiene menor error con respecto al valor MAE, mostrado en la Tabla 25, pues con respecto al modelo 1, tiene una diferencia en un 18.9%, mientras que, con respecto al modelo 2, tiene una diferencia del 20.2%, entonces, este modelo con el resultado general es aquel que mejor resultados dio, aunque no es funcional para implementar en un sistema de pesaje. De ahí, mirando el error MAE que tiene el modelo 1 con respecto al modelo 2, es menor, lo cual significaría que las imágenes que no toman en cuenta ambos lados son mejores, pese a ello, esta suposición es cancelada por el error MSE, pues este si es mayor al del modelo 2, esto debido a que los errores más grandes se han penalizado, indicando que las predicciones tienen

errores similares, mientras el modelo 1, tiene mucha variación, con predicciones que tienen equivocaciones más grandes. También se analiza los datos por la clasificación de los vehículos en camiones y tractocamiones para saber cuál es el modelo que mejor predice y conocer el comportamiento con cada clase, esto en las Figuras 23 y 24.

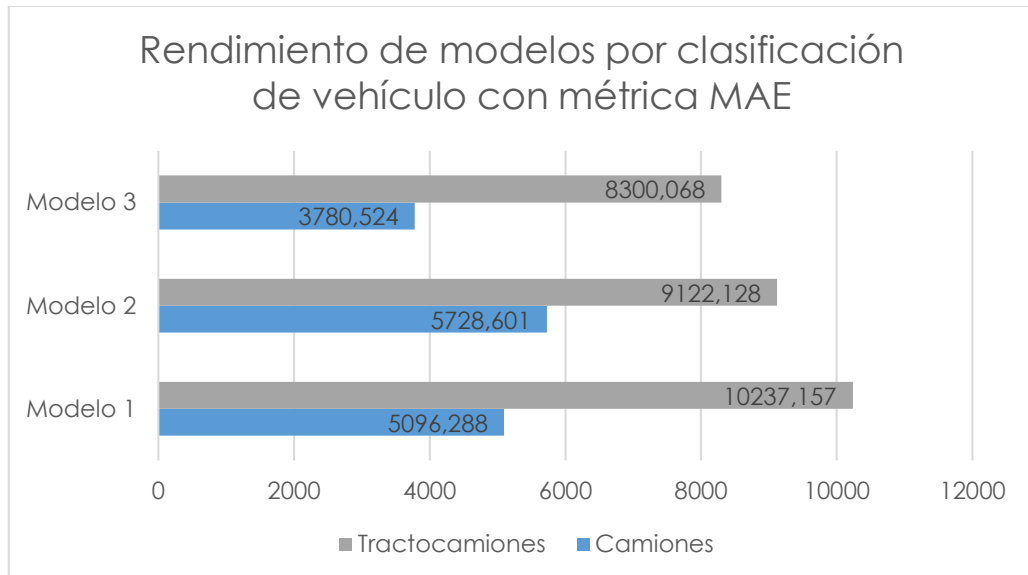


Figura 23. Rendimiento con métricas MAE por modelo y clasificación de vehículo

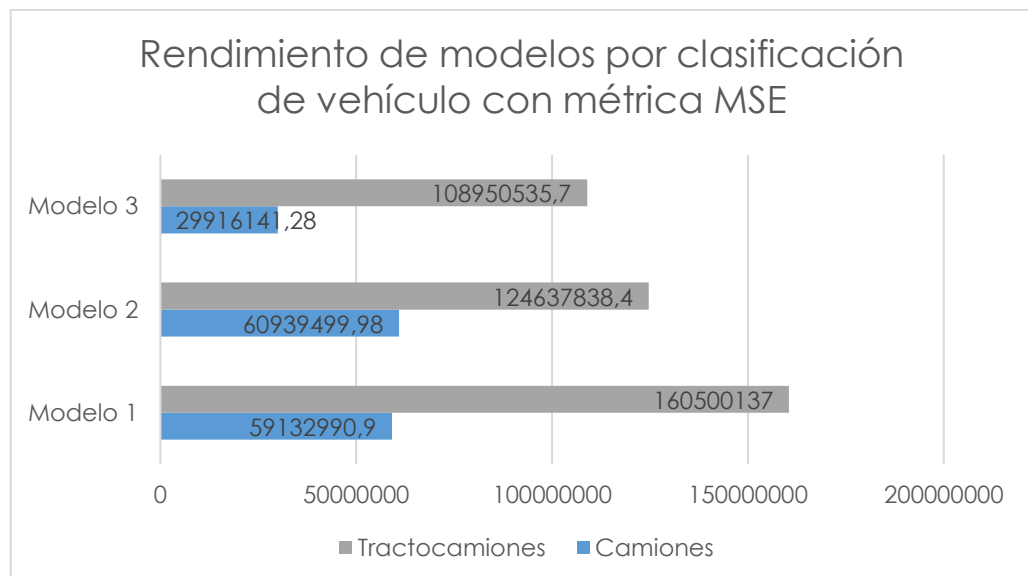


Figura 24. Rendimiento con métricas MSE por modelo y clasificación de vehículo

Mediante la observación al momento de tomar las imágenes de los vehículos, se notó que los tractocamiones cuentan con una estructura del vehículo más robusta, por lo tanto, no sufren alteraciones visibles al aumentar en poca medida su peso, esto puede hacer que para estos vehículos sea mucho más complicado estimar el peso con un error bajo, además, al cargar mayor cantidad de peso, el error debe ser demasiado bajo, menor al 1%. En la Figura 23, se evidencia este comportamiento,

teniendo los 3 errores más altos en tractocamiones, siendo el menor de estos el error MAE de 8300.06 *kg* en el modelo 3, de igual manera, el modelo 3 da mejor resultado en los camiones, obteniendo un error de 3780.52 *kg*. Además, en la Figura 24, con respecto al valor MSE, muestra errores más elevados respecto al valor real, en el modelo 3 que tiene mejores resultados, el error con respecto a los camiones es de 29916141.28 *kg*², mientras en los tractocamiones llegan a superar los 100000000 *kg*², por ende, muestra que al modelo le cuesta predecir mucho más este tipo de vehículos.

Otro punto importante es el lugar donde se tomaron las fotografías, para conocer si el lugar tiene algún efecto sobre el modelo y más aun tomando en cuenta las dificultades que se tuvo en la bodega Transbolivariana, estos resultados se miran en la Figura 25.

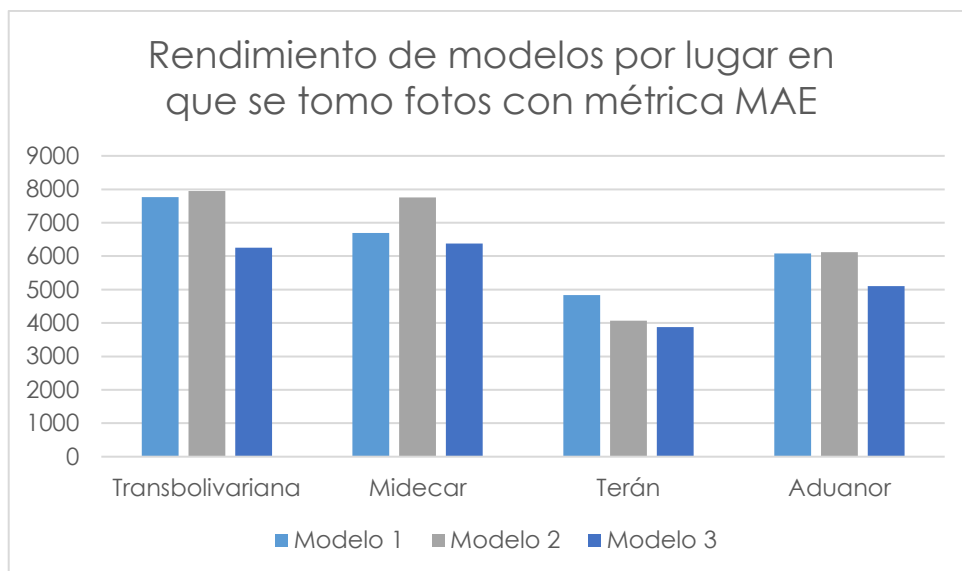


Figura 25. Rendimiento con métrica MAE por modelo y lugar de captura de imágenes

Con base en esta gráfica por lugares, de la Figura 25, que representa la métrica MAE, se aprecia que los 3 modelos en Transbolivariana tienen errores muy altos, siendo el mayor de 7949.32 *kg* de error con el modelo 2, Midecar también presenta errores elevados, siendo el del modelo 2 el más notable con 7759,17 *kg*. En Aduanor el modelo 1 y 2 dan errores altos, próximos a los 6000 *kg*, un valor demasiado elevado. El lugar que genera menos errores es Terán, siendo inferiores a los 5000 *kg*, en el que el modelo 3 representa el menor con 3873.85 *kg*, así mismo este modelo es el que genera menos errores de los 4 lugares.

La figura siguiente (Figura 26), muestra el rendimiento de modelos por lugar en que se recopilaron las fotos en base a la métrica MSE.

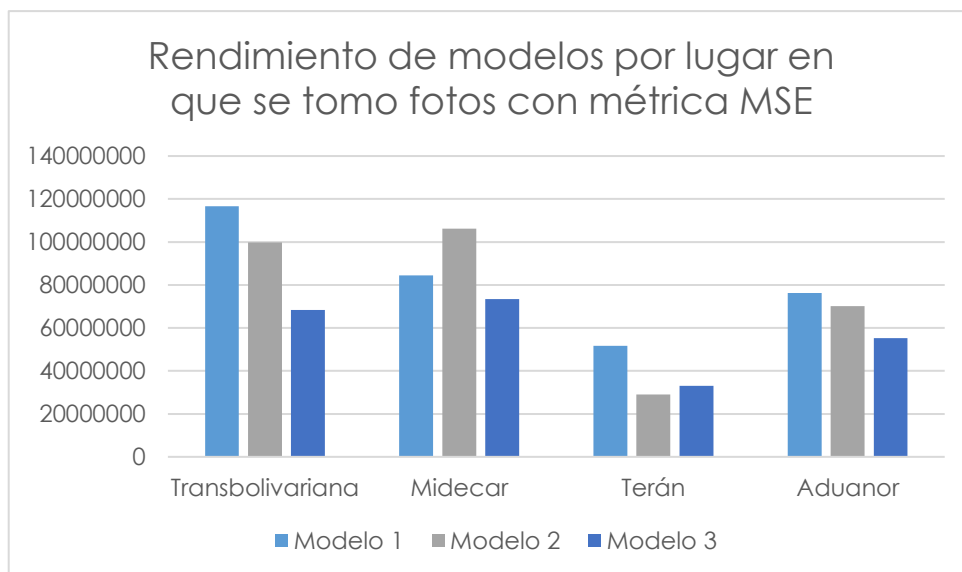


Figura 26. Rendimiento con métricas MSE por modelo y lugar de captura de imágenes

En el caso de la métrica MSE, de la Figura 26, los valores son similares, mostrando errores elevados, en el modelo 1 y 2 en Transbolivariana, mientras el modelo 3 es mucho menor, con 68347896.32 kg^2 . El modelo 2 en Midecar, tiene errores superiores a los otros dos modelos inclusive al modelo 2 en Transbolivariana. En comparación con los 2 lugares mencionados, Aduanor baja ese error en los 3 modelos, siendo el más bajo en el modelo 3 con 55238339.35 kg^2 y por último en Terán tiene los errores más bajos entre los 4 lugares, reduciendo en un 55.66 % de error con respecto al modelo 3.

El modelo 2 y 3 en bodegas Terán muestran un mejor rendimiento, que, a pesar de seguir teniendo errores, se observa que en este espacio los modelos encuentran características relacionadas al peso. Del mismo modo, aunque las métricas no son precisas, se puede atribuir las malas predicciones a las imágenes que no cuentan con buenos ángulos de captura de imagen, como fue el caso de Transbolivariana, que en todos los resultados salieron los errores más elevados.

Ahora sí, habiendo analizado el comportamiento de los errores, se selecciona el modelo 3, en el que las imágenes se unieron por los 4 ángulos para generar una imagen y así forzar a que la red aprenda las características de estas desde una imagen.

Pero antes de tomar esta red para analizarla un poco más, es importante saber si los ángulos en que se tomaron las imágenes afectan al comportamiento, para lo cual, se toma los resultados en camiones, que fueron los vehículos que permitieron tomar

imágenes lateralmente, además, se toma en cuenta el modelo 2 que fue entrenado con las combinaciones de imágenes de esta forma y se obtuvieron los resultados de la Tabla 26.

Tabla 26. Resultados con métricas MAE y MSE, del modelo 2 por el tipo de ángulo.

Métrica	Imágenes delanteras	Imágenes traseras	Imágenes laterales
MAE (<i>kg</i>)	6262.96	5601.08	3990.17
MSE (<i>kg</i> ²)	70362917.83	59081586.52	28568854.3

Con base en la Tabla 26, el ángulo con el que se toma la imagen si afecta al momento que el modelo busca las características, siendo las imágenes laterales, aquellas con mejores resultados, con respecto al valor MAE, logró un error de 3990.17 *kg*, de ahí el segundo ángulo con mejores resultados es de aquellas que se tomaron desde la parte trasera de los vehículos, con un error de 5601.08 *kg* y por último las imágenes tomadas desde la parte delantera tuvieron un error de 6262.96 *kg*, teniendo una diferencia de 28.76% de error entre las imágenes de las partes trasera y la lateral, en el caso de la métrica MSE, el cambio es del 51.64%. Según los datos se muestra que los errores tienen relación, de lo contrario serían penalizados por la métrica MSE y esta tendría un comportamiento diferente a la métrica MAE, por lo tanto, el error de las imágenes laterales si disminuye con respecto a los otros dos ángulos.

Una vez analizado este resultado, se toma el modelo 3 que tiene mejores métricas y se mira si en este modelo, también afecta el tomar las fotografías de manera lateral, al tener las imágenes unidas de los 4 ángulos, de igual manera, tomando datos de los camiones para poder comparar con los anteriores datos, esto se muestra en la Tabla 27.

Tabla 27. Resultados con métricas MAE y MSE, del modelo 3 por el tipo de unión.

Métrica	Imágenes Unificadas	Imágenes laterales	Laterales (modelo 2)
MAE (<i>kg</i>)	3814.656	3534.386	3990.17
MSE (<i>kg</i> ²)	30963115.273	22365936.778	28568854.3

Con los datos de la Tabla 27, se obtiene que las imágenes unificadas dan mejor rendimiento que las imágenes laterales del modelo 2 con respecto al valor MAE, pero no reduciendo en el valor MSE, pues este se mira que ha aumentado, caso contrario pasa con las imágenes laterales del modelo 3, mejorado tanto en valor MAE y MSE, reduciendo en un 11.42% y un 21.71% respectivamente, esto con respecto a las imágenes laterales del modelo 2, aunque estas imágenes sean las mismas en ambos

modelos, el entrenar con el *dataset* de las imágenes unificadas por cada ángulo, puede afectar positivamente en su entrenamiento.

Por lo tanto, se ha definido que el modelo 3, en el que las imágenes son manipuladas y unificadas por las imágenes correspondiente a cada esquina del vehículo, da los mejores resultados, no los óptimos, pero si muestra un mejor entrenamiento y menor error en los datos de prueba, además, el lugar en el que se toma las imágenes no se mira que afecte directamente en el rendimiento, siempre y cuando las imágenes sean tomadas correctamente, a una distancia que le permita a la cámara capturar todo el vehículo, si el lugar ofrece un espacio propicio y que el nivel del piso sea óptimo y liso, el rendimiento será bueno, al tener una buena distribución de la carga en el vehículo, debido a que la forma de las llantas sobre alteraciones en el piso, como baches o degradación del suelo, se vuelve muy irregular, igualmente se mira que las imágenes laterales pueden llegar a tener buenas métricas de rendimiento, y por eso, un vehículo debe tener todos los ángulos, en el caso de los camiones, pues en tractocamiones, es mucho más difícil capturar el vehículo lateralmente.

4.1.4.2. Ajuste de curva para históricos de entrenamiento

En el modelo 3, al ser el mejor modelo entre los 3 entrenados, mostró que tiene fallas muy elevadas en los datos de prueba, y como se mencionó, en la Figura 20, de los históricos, se ajustó los datos de magnitud de pérdida para mostrar todo el proceso de pérdida, por lo que, se toma estos datos de pérdida que representan la métrica MSE, y se calcula el valor raíz del error cuadrático medio (RMSE) obteniendo una mejor interpretación de los datos, esto se mira en la Figura 27.

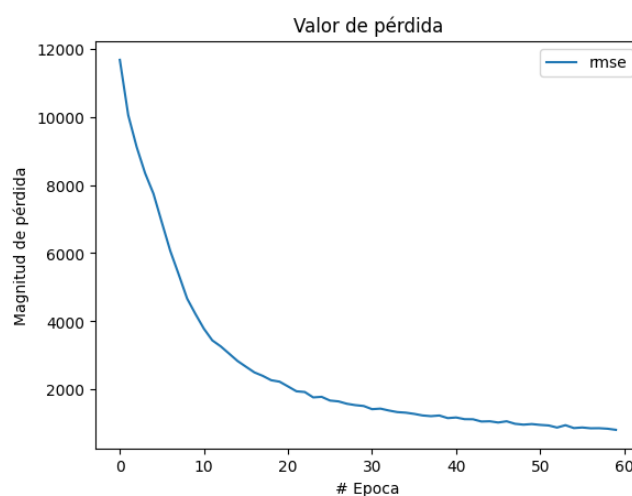


Figura 27. Valor de pérdida por estadístico RMSE.

En este caso en la Figura 27 , ya se mira la magnitud de pérdida en valores por kilogramo, lo que muestra que hasta la época 60, llego a tener un error menor a 1000 *kg* con respecto a los datos reales, por lo tanto, se mira la posibilidad de mejorar estas métricas tras añadir más épocas, sin embargo, añadirlas sin haber analizado las necesarias, representa gastar recursos computacionales, debido a que otras 60 épocas significa entrenar el modelo 10.8 horas más, por eso, al mirar los datos de entrenamiento, se observaría que no eran necesarias tantas épocas, entonces, se analizó que la gráfica representa una función exponencial negativa, y las razones se deben a que, primero el error disminuye drásticamente por cada época, y de a poco, el error va disminuyendo en menor medida.

Por lo tanto, se buscó realizar un ajuste de curva que permita obtener una función exponencial negativa para predecir en que punto el error llegará a aproximarse a cero, con base en la función:

$$y = ae^{bx}$$

donde, a , es un valor inicial cuando $x = 0$, y , la variable dependiente que para este caso es el valor de perdida, x , la variable independiente correspondiente al número de épocas y b , la tasa de decaimiento para saber que tan rápido disminuye y .

Para el cálculo, se utiliza el valor de *loss* y MSE, pues es el que necesita ver que tanto disminuye según el ajuste de la curva. Para esto se utiliza el siguiente código, que permite obtener los datos a y b :

```
import numpy as np
from scipy.optimize import curve_fit
df_3 = pd.read_excel(r'ruta_archivo\grafic_resnet_unionex4.xlsx')
# Función a ajustar
def modelo_exp(x, a, b):
    return a * np.exp(-b * x)
# Parámetros iniciales para el proceso de fiteo
parametros_iniciales=[df_3['loss'][0], 0.1]
x_datos = df_3['epoca']
y_datos = df_3['loss']
#Hacer el ajuste con curve_fit
popt, pcov = curve_fit(modelo_exp, x_datos, y_datos, p0=parametros_iniciales)
# en la variable popt se guardan los resultados óptimos de a y b, .
# y pcov la matriz de covarianza de los parámetros hallados.
```

```

x_modelo = np.linspace(0, 60, 1000)

plt.figure()
plt.plot( x_datos,          y_datos, 'o', label='datos')
plt.plot(x_modelo, modelo_exp(x_modelo, *popt), 'r-', label='modelo ajustado')
plt.legend(loc='best')
plt.xlabel('x')
plt.ylabel('y')
plt.tight_layout()

```

Este código arroja la gráfica mostrada en la Figura 28, donde se miran los valores de los datos y el modelo ajustado.

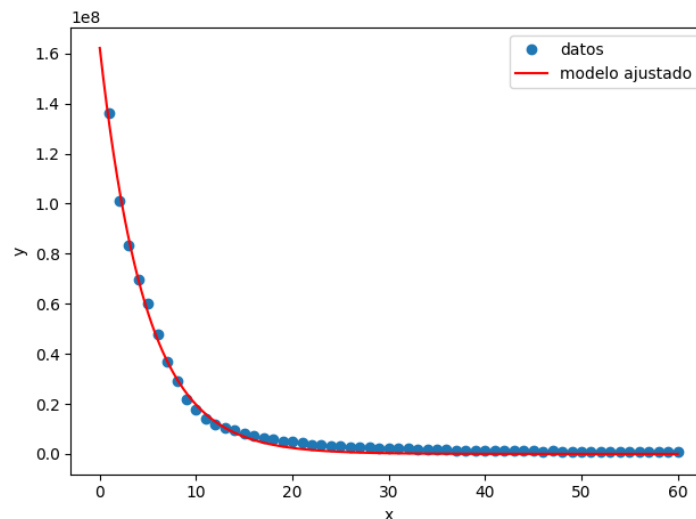


Figura 28. Ajuste de curva de función exponencial negativa

En esta Figura 28, se logra contemplar que el modelo ajustado sigue muy bien a los datos de pérdida, mostrando una diferencia entre las épocas 20, en que el modelo sobresaleara un poco con respecto a los datos reales, dando una perspectiva sobre cómo se puede comportar el modelo tras aumentar más épocas, además, con el anterior código se obtuvo los datos óptimos de a y b , obteniendo la siguiente función:

$$y = (1.62200920e+08) * e^{(2.10945906e-01)x},$$

Ahora, se ingresó datos en la variable x , para saber el comportamiento desde la época 60 hasta la época 100, para esto, se usó el código:

```

x_pred = np.array(list(range(61, 101)))
y_pred = modelo_exp(x_pred, *popt)

```

```

plt.xlabel("# Epoca")
plt.ylabel("Magnitud de pérdida")
plt.plot(y_pred)
plt.legend(["loss proyectada"])
plt.ylim(0,100)
plt.show()

```

Este permitió realizar las predicciones para épocas posteriores, y a su vez, graficar estos datos para analizarlos, este se lo presenta en la Figura 29.

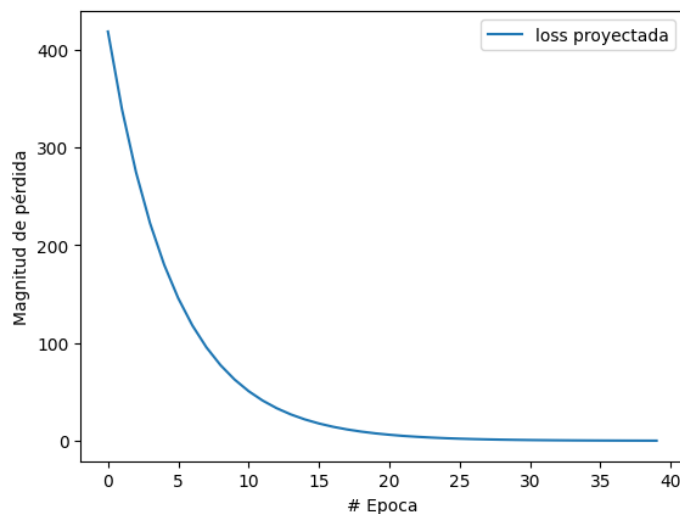


Figura 29. Predicción de magnitud de pérdida a 40 épocas posteriores al entrenamiento.

En la Figura 29, se da un comportamiento muy diferente al esperado, pues según esto en la última época del entrenamiento del modelo debería tener una magnitud de pérdida MSE menor a 500 kg^2 , mientras que en realidad fue de 665443 kg^2 , por lo tanto, no se puede estimar un número de épocas con el cuál se espere que el error disminuya hasta un valor esperado. Debido a que los datos no tienen sentido con respecto a los datos reales hasta la época 60, donde según este modelo ajustado, entre la época 20 y 25 el error estaría muy próximo a tener un error de cero, pero como se mencionó, estos datos no son precisos y por eso no fiables, por lo que, no se puede determinar un número de épocas en el que se pueda reducir el error hasta un valor óptimo.

4.1.4.3. Interpretación visual de las capas convolucionales de las redes.

En esta parte, tras interpretar los datos de los resultados, se identifica que al menos en el modelo 3, tras el entrenamiento la red, esta logra identificar características del peso, pues sus resultados mostraron menores errores, por lo tanto, también se miró a

través de una capa convolucional en el medio de la red, para observar que es lo que analizan en las imágenes, la capa seleccionada pertenece al "stage 3", esto se hace porque en las capas principales las convoluciones buscan características más generales, en las centrales busca características más específicas y a la vez permite observar una imagen aún interpretable al ojo humano, pues más allá los mapas de activación se vuelven más pequeños que resultan en patrones que solo la red neuronal entiende, por lo que, se escoge en específico la capa denominada "res3d_branch2c".

Para poder observar las convoluciones que suceden en esta capa, se usó el siguiente código:

```
modelo = load_model("ruta_del_modelo"),
# Obtener una imagen del dataset de imágenes de prueba
img = []
for image, name in test_dataset.skip(150).take(1):
    for i in range(1):
        # Acceso al único elemento de la predicción array antes de convertirlo a int
        img.append(tf.expand_dims(image[i], axis=0))

# Selecciona una capa de la red
layer_name = "res3d_branch2c"
intermediate_layer_model = Model(inputs=modelo.input,
outputs=modelo.get_layer(layer_name).output)

# Obtiene las activaciones
feature_maps = intermediate_layer_model.predict(img)

# Visualiza los primeros filtros
n_filters = 18 # número de filtros a mostrar
rows, cols = 3, 6 # matriz de visualización

fig, axes = plt.subplots(rows, cols, figsize=(12, 6))
fig.suptitle(f'Activaciones de la capa: {layer_name}', fontsize=16)
```

```

for i in range(rows):
    for j in range(cols):
        index = i * cols + j
        if index < n_filters:
            axes[i, j].imshow(feature_maps[0, :, :, index], cmap='viridis')
            axes[i, j].axis('off')
        else:
            axes[i, j].axis('off') # Desactiva si no hay imagen para ese slot
plt.tight_layout()
plt.show()

```

El funcionamiento de este código empieza cargando un *dataset* de pruebas, desde un archivo *TfRecord* como se lo realizó anteriormente, y cargando el modelo con `load_model()`, luego, de este *dataset* se escoge un elemento al azar, se extrae del modelo la capa especificada, se escoge el número de filtros que se quiere observar y se mira en que formato se verá la matriz de imágenes, para posteriormente aplicar estos filtros a la imagen y mostrarlos en pantalla. Este mismo proceso se repite para cada modelo.

Primero se empezó a observar los filtros (convoluciones) en el modelo 3, pues fue el que dio mejores resultados, estos filtros se pueden observar en la Figura 30,

Activaciones de la capa: res3d_branch2c



Figura 30. Filtros aplicados a una imagen del *dataset* de prueba del modelo 3.

Las imágenes mostradas en la Figura 30, son muy representativas, pues en cada filtro resalta las partes importantes que definen el peso de un vehículo, como lo son las

llantas, la forma del vehículo y los espacios que se encuentran entre las llantas y el chasis, de ahí, las partes que representa el piso y parte superior del vehículo muestra como no los toma en cuenta al eliminarse las características en los filtros, con esto, reafirma la idea, sobre que la red neuronal si busca las características que se relacionan con el peso del vehículo. Luego, se mira el comportamiento de los filtros en el modelo 2, que tuvo menor rendimiento, pero fue mejor que el modelo 1, esto por el tipo de unión lateral que tuvieron las imágenes, dando como resultados los filtros mostrados en la Figura 31.



Figura 31. Filtros aplicados a una imagen del *dataset* de prueba del modelo 2.

Según lo observado en la Figura 31, en el modelo 2 se logra identificar que la capa denominada "res3d_branch2c", no ha generado filtros óptimos para buscar características, porque se evidencia que opaca la imagen de entrada, en este caso se evidencia la eficiencia de ResNet, al tener un error en la red no permite que toda la red deje de aprender, pues por su aprendizaje pasa a otras capas, las cuales si aprenden los patrones, para este caso, se cambió la capa con otra del mismo step, denominada "res3d_branch2a" y se las presenta en la Figura 32.

Activaciones de la capa: res3d_branch2a

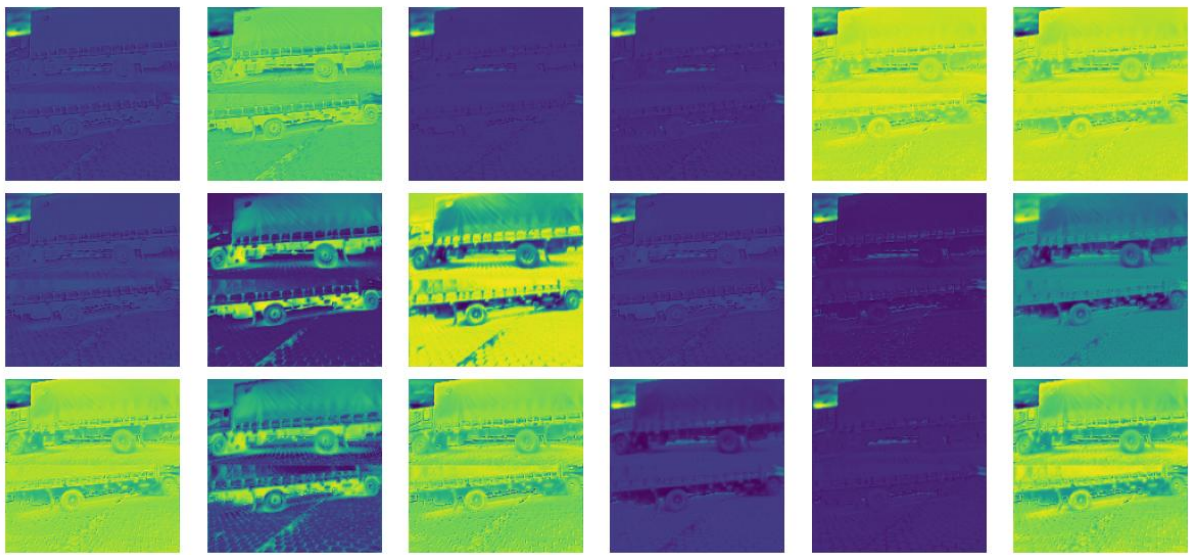


Figura 32. Filtros aplicados con distinta capa a una imagen del *dataset* de prueba del modelo 2.

Ahora, en la Figura 32, ya se visualiza como los filtros aplicados resaltan al vehículo, donde en algunos trata de ocultar el contorno y dejar resaltado al vehículo, en este caso no se mira que resalta partes en específico, pero si entiende que la relación del peso está asociada con el vehículo, y por lo tanto, en capas posteriores buscará estas características más importantes. Otro aspecto relevante a considerar es que el funcionamiento de las CNNs no se basan en tomar una parte específica de la imagen, si no, de ir reduciendo la imagen completa, por lo tanto, el contorno de los vehículos se mantiene a lo largo en la profundidad de la red, por lo que, este *dataset* no es muy eficiente en cuestión de recursos computacionales, al tener 64773764 parámetros totales en el modelo, los cuáles estén analizando estas partes no relevantes para el peso, que corresponde al fondo de las imágenes, como lo es el cielo o el piso.

El modelo 1, al ser el que cuenta con más errores y fue cargado con imágenes que solo representaban un ángulo por imagen, no se lo analiza. De ahí, observando los filtros de estos dos modelos, se logra destacar dos aspectos importantes, en el modelo 3 se ve que al unificar 4 imágenes, cada una representativa de una esquina, el modelo se centra en estas partes específicas, evitando estar analizando el contorno que no tiene relación con el peso, forzando a la red a aprender esos datos, volviendo más eficiente el modelo para entrenar y con mejores resultados, este aspecto fue lo contrario en el modelo 2, pues se mira que hasta la capa 3 encuentra al vehículo como el objeto a analizar, pero por el contorno que tienen las imágenes, donde sale

en gran medida el cielo o el piso, también son analizados estos parámetros, lo que se traduce en consumo de recursos y eficiencia en el entrenamiento del modelo, sin embargo, al final muestra que si aprende, al hacer notorio que resalta las características del vehículo relacionadas con su peso.

4.1.5. Cotización de equipos computacionales

En el entrenamiento de modelos con CNNs, es fundamental contar con equipos computacionales de gran capacidad, para el procesamiento de datos y obtener un buen rendimiento en los entrenamientos, siendo la GPU, el componente que los acelera en modelos de IA, esto acompañado de una memoria RAM de gran capacidad, para mantener las imágenes cargadas en memoria cache y así estén a la disposición de la GPU, logrando aprovechar su máxima capacidad.

En el caso, que no se cuente con suficiente memoria RAM, el procesador debe tener la capacidad suficiente para precargar lotes de imágenes eficientemente para que la GPU aproveche sus recursos, evitando cuellos de botella. Al entrenar los modelos, la cantidad de imágenes del *dataset* saturaba la memoria RAM, optando por el preprocesamiento de datos, al redimensionar y mezclar las imágenes en un archivo *TfRecord*, permitiendo que en los entrenamientos el procesador sea capaz de precargar lotes de imágenes necesarios para que la GPU sea usada en su máxima capacidad.

En este punto se hace necesario realizar una cotización a partir de las características de la computadora que se utilizó, misma que es propiedad de la Universidad Politécnica Estatal del Carchi, tomada como referencia para estimar un presupuesto con el que se pueda adquirir un equipo con capacidades y características similares a las que se mostraron en la Tabla 11, llegando así a tener un costo referencial con el que se debería haber contado sí el proyecto no se hubiera realizado con equipos universitarios. Por eso, en la Tabla 28, se muestra una cotización de un equipo con las características del computador utilizado, esta se la realiza a mes de junio de 2025.

Tabla 28. Cotización de equipo computacional con características al equipo usado en los entrenamientos.

Componentes	Especificaciones	Precio local estimado (USD)
CPU	AMD Ryzen 9 5950X, 16 núcleos	≈ 765
GPU	NVIDIA RTX 3090, 24 GB (VRAM)	≈ 3 199
Motherboard (AM4)	B550 / X570 AM4	≈ 215
RAM – 64 GB	DDR4-2666 (4×16 GB)	≈ 225

Componentes	Especificaciones	Precio local estimado (USD)
SSD – 1 TB NVMe (sistema)	Kingston NV2 NVMe	≈ 175
SSD – 4 TB SATA	Samsung 870 EVO	≈ 375
Fuente de poder – 750 W	Corsair CV750 Bronze / NZXT C750 Gold / Gamemax	≈ 100
Cooler CPU	Noctua NH-U12S, Corsair H100)	≈ 95
Demas	Componentes faltantes	≈ 200
Total		≈ 5349

En la Tabla 28, se mira que el componente más caro es la GPU, la cual permite realizar entrenamientos de Inteligencia Artificial maximizando el rendimiento, por lo tanto, una tarjeta gráfica es el componente más potente con el que cuenta un computador, pues ni la CPU tiene un costo tan elevado al compararlo, representando un 29.9% del costo de la GPU, y en comparación con la cotización total, representa un 14.30% del costo, mientras que la tarjeta gráfica de NVIDIA RTX 3090, representa el 59.8% del total del costo de la computadora.

Con esto, tomando en cuenta que la GPU es la que consume más recursos, mismo que se pudo evidenciar en el proceso de entrenamiento de los modelos, se buscó el estimado del consumo en watts de este componente al estar sometido a trabajo, para poder realizar un estimado sobre el consumo eléctrico y el costo que conlleva entrenar un modelo de CNNs, considerando las tarifas de Ecuador, estos datos se muestran a continuación en la Tabla 29.

Tabla 29. Consumo en watts de la GPU y tarifa de energía eléctrica.

	GPU	Observación	Referencia
TDP (Potencia de diseño térmico)	350 W = 0.350 kW	- Consumo en vatios bajo máxima carga.	TechPowerUp (2025)
Consumo por kWh.	0.35 kWh	- Consumo/segundo	
Tarifa kWh	0.10 USD		Tapia (2024)
Precio por hora	0.35 kWh x 0.10 USD = 0.035 USD		
Consumo por entrenamiento de modelo 3	0.035 USD X 10.8 horas ≈ 0.378 USD		

Con base en la Tabla 29, se determinó que el modelo 3 tuvo un costo de USD 0.38 en las 10.8 horas de entrenamiento, tomando en cuenta el consumo de la GPU utilizada,

este dato muestra que al tener un equipo de este tipo, el entrenamiento de un modelo de las mismas características tiene un precio relativamente bajo, pues tras casi 11 horas se puede cargar un modelo, el cuál en caso de funcionar, se puede usar en cualquier otro equipo, por lo tanto, el uso de este representa un costo bajo comparado con el beneficio.

4.1.6. Comparación entre sistema por báscula y programa de pesaje.

Se tiene en cuenta que el mejor modelo entrenado, no cumple con la precisión requerida para el pesaje, pero si se mira la posibilidad que mejorando los *datasets* sea posible llegar a mejorar este aspecto, esto al haber analizado la importancia de las básculas, pues su uso en el transporte llega a ser muy necesario, inclusive obligatorio. Por eso, se analiza las básculas comparándolas con el programa de pesaje con IA, en un escenario en el que el modelo hubiese dado resultados precisos, esto se describe en la Tabla 30, en diferentes categorías.

Tabla 30. Comparación entre básculas y programa de pesaje con IA

Categoría	Báscula	Sistema de Pesaje con Imágenes
<i>Infraestructura física</i>	- Requiere un terreno preparado, rampas, losas de asentamiento, caseta de control, canales de evacuación para lluvia, tubería interna, ductos, vigas, etc.	Su infraestructura se basa en un punto de captura de imágenes que tenga un piso de preferencia nivelado y liso con espacio para moverse.
<i>Componentes tecnológicos</i>	Celdas de carga (más comunes tipo botella), sistema eléctrico de toma a tierra e indicador digital.	Cámaras (puede ser un celular que permita capturar imágenes con resolución de 4000 x 3000 píxeles a RGB), computador con GPU potente y software de IA (modelo ya entrenado).
<i>Costo estimado</i>	Aproximadamente entre 30 000 y 35 000 USD	Aproximadamente 5349 USD un computador con especificaciones como GPU para modelos de IA, y cámara o celular, aproximadamente 200 USD, total 5549 USD.
<i>Capacidad máxima de pesaje</i>	60 000 kg	Depende de <i>dataset</i> con el que haya sido entrenado. Si no tiene suficientes datos con pesos altos no funcionaría.
<i>Exactitud</i>	El laboratorio de Certificación Metrológica, CERTMETROL, realiza las calibraciones y permite un error máximo de más o menos 30 kg con respecto al valor real, en pesos superiores o iguales a 20780 kg.	El programa base muestra la posibilidad de tener buena exactitud, por lo tanto, no existe en el mercado, no teniendo aún posibilidad de obtener certificaciones.
<i>Escalabilidad</i>	Difícil, requiere replicar toda la infraestructura en cada implementación.	Alta, pues tras entrenado el modelo en un computador potente, el programa se puede replicar en diferentes

Categoría	Báscula	Sistema de Pesaje con Imágenes
Mantenimiento	Requiere de revisiones de celdas, limpiezas, recalibraciones, cambio de partes físicas, generando costos elevados.	computadores de menores características. Tiene bajo costo, mantenimiento en cámaras, en el caso de celular depende del uso, y actualizaciones del programa periódicamente.
Consumo Eléctrico	110V para equipo principal y consumo eléctrico de la infraestructura.	110V–220V dependiendo del equipo de cómputo, y necesidades de la IA.
Portabilidad	No es portátil, se instala fijamente y el proceso es complejo.	Con una base de datos o un entorno controlado se vuelve potencialmente portátil.
Tiempo de implementación	Tiempo aproximado de 30 días laborables.	La recolección de datos no se puede definir en un parámetro en concreto. Una vez con el modelo entrenado, la implementación es rápida, si se cuenta con la infraestructura mínima (cámaras, espacio para las fotos y conectividad).
Recolección de datos	Anotación manual, desde la caseta de pesaje.	Altamente susceptible a automatizarse e incluir modelos como revisión de placas o tipo de vehículo.
Interacción con el sistema	Indicador digital con control manual y operador	Interfaz automatizada o remota desde un sistema que albergue a la IA.

En la Tabla 30, se observa que entre los dos sistemas, el que más destaca es el programa con IA, pues elimina todos los sistemas complejos de infraestructura e implementación, cambiando por un *software* adaptable a computadoras con características básicas, en este caso se ven mejoras desde el precio, pues el costo de un computador para entrenar los modelos, representa entre el 17% y 20% con respecto a una báscula camionera, además, requieren de una infraestructura para implementar una nueva, mientras la computadora, genera los modelos (actualizaciones) y este se puede implementar en cualquier lugar con las especificaciones mínimas, aunque uno de los inconvenientes es la certificación con la que no cuenta esta clase de programas de pesaje basados en IA, de ahí, en flexibilidad el programa vuelve a tener ventaja, pues es más fácil de escalar y no se requiere de infraestructura compleja. Todo esto se puede apreciar en un escenario en que los modelos, fueran precisos, pues una de las limitaciones es el *dataset*, al no tener una cantidad de imágenes amplia, y el tiempo para poder desarrollar los modelos.

4.1.7. Hipótesis

Para la prueba de hipótesis primero se debe tener en cuenta, las predicciones del modelo, y luego ver si mejora el proceso de pesaje.

La Hipótesis general es:

H_0 : Un programa de pesaje basado en imágenes no permite optimizar el proceso de pesaje de vehículos de carga pesada.

H_1 : Un programa de pesaje basado en imágenes permite optimizar el proceso de pesaje de vehículos de carga pesada.

Para validar la hipótesis sobre que un programa de pesaje basado en imágenes optimiza el proceso de pesaje, se plantea un análisis secuencial en dos fases:

4.1.7.1. Fases

4.1.7.1.1. Fase 1: Se evalúa la precisión del modelo de inteligencia artificial mediante una prueba estadística que compara los pesos reales con los predichos. Si no existe una diferencia significativa entre ambos, se considera que el modelo es técnicamente válido.

4.1.7.1.2. Fase 2: Si el modelo demuestra ser preciso, se realiza una segunda prueba estadística para verificar si el programa de pesaje contribuye a mejorar el proceso, evaluando indicadores como tiempo, eficiencia o flujo operacional.

Solo si ambas fases se superan, se puede afirmar que el programa permite optimizar el pesaje de vehículos de carga pesada.

4.1.7.2. Fase 1. Formulación de hipótesis:

Para hacer la prueba de hipótesis T de *Student* y buscar si existe diferencia significativa de medias entre los datos reales de peso y predichos por el programa, se tiene en cuenta que los datos tengan normalidad, donde, debido a que en esta prueba se tienen 2851 datos, con el Teorema Central del Límite, se determina que no existe inconveniente al usar la prueba T de *Student*, porque para muestras mayores o iguales a 30 tienden a seguir una distribución normal,

Considerando otro aspecto, se analiza que las variables tengan correlación entre sí, debido a que ambos representan el peso, sin embargo, unos pesos fueron tomados en básculas y los otros son estimados por el programa, por lo tanto, se busca la fuerza de relación que tienen y la variación de una variable con relación a la variación de la otra variable. Estos datos se obtienen a partir del siguiente código en *Python*:

```

import numpy as np
from scipy.stats import pearsonr
# Datos
df = pd.read_excel(r'ruta_archivo.xlsx')
y = df["Peso"]
y_pred = df["prediccion"]
# Coeficiente de correlación
corr_matrix = np.corrcoef(y, y_pred)
# Coeficiente de determinación
re_f = 1-(sum((y - y_pred)**2)/sum((y-np.mean(y))**2))
print("Coeficiente de correlación (r)", corr_matrix[0, 1])
print("Coeficiente de determinación (R2): ", re_f)

```

Obteniendo de resultado $r = 0.75902$ y $R^2 = 0.57452$, mismos que dan una mejor comprensión sobre la relación entre las variables, esto junto a un diagrama de dispersión, el cual muestra el comportamiento de los datos gráficamente, presentado en la Figura 33.

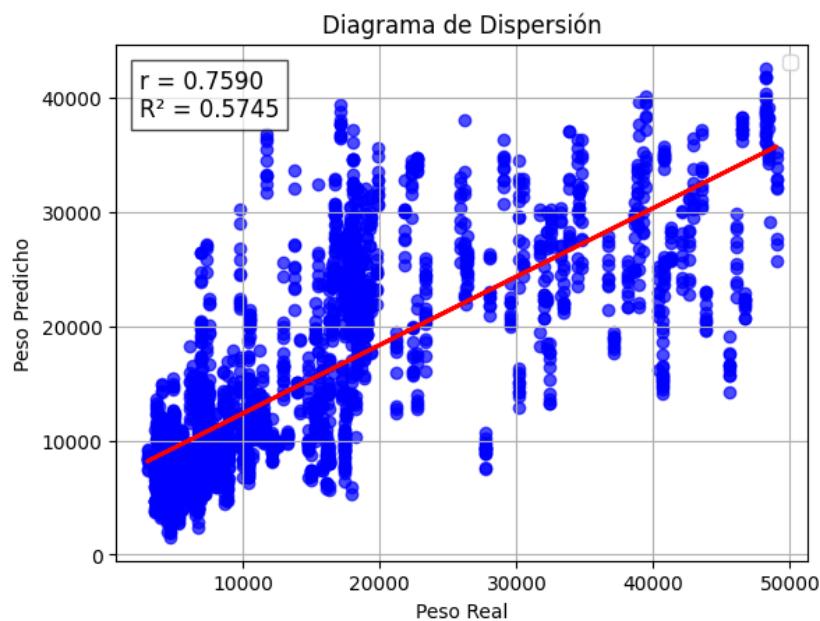


Figura 33. Diagrama de dispersión entre el peso real y peso predicho por el programa.

Con base en la Figura 33, se observa datos dispersos entre sí, pero con cierta relación, en la que si el peso real es bajo las predicciones tienden a ser bajas y si los datos son altos también dan predicciones altas, que al tener un $r = 0.7590$, corresponde a una correlación positiva fuerte, al ser un valor intermedio entre $+0.5$ y $+1$, y con el

coeficiente de determinación se puede decir que el 57.45% de la variación en el peso predicho está representado por la variación en el peso real. Por medio de este análisis se tiene que los datos si representan una relación, misma que se da porque se busca que el programa llegue a dar los mismos resultados, con esto, se continúa analizando si existe diferencia significativa entre estas variables.

H_0 : No existe diferencia significativa entre el peso real del vehículo y el peso predicho por el programa.

H_1 : Existe diferencia significativa entre el peso real del vehículo y el peso predicho por el programa.

Para comprobar la hipótesis se plantea el siguiente contraste:

$$H_0 : \mu_d = 0 \quad \gamma \quad H_1 : \mu_d \neq 0$$

Donde, μ_d es la media de las diferencias entre los pesos reales y predichos.

4.1.7.2.1. El método de prueba seleccionado es la prueba T de Student apareada.

Se selecciona esta prueba estadística, al tener datos emparejados, peso real y peso predicho de la imagen de los vehículos.

4.1.7.2.2. Nivel de significancia

El nivel de confianza es del 95%, por lo tanto, el nivel de significancia es:

$$alfa = 1 - 0.95$$

4.1.7.2.3. Criterio de decisión

Si el valor p (valor de probabilidad) resulta menor que $alfa$ (0.05), se rechaza la hipótesis nula y la hipótesis alternativa es acertada.

4.1.7.2.4. Tamaño de muestra

Para esta prueba se utiliza el *dataset* de prueba relacionado con el que se entrena el modelo 3, porque es este del que se realizan las predicciones, este corresponde a 2851 datos, que son imágenes que no han ingresado en el entrenamiento del modelo, por lo tanto, se realizan las predicciones a estas, teniendo en cuenta que cada imagen cuenta con su valor real, misma razón de ser datos emparejados.

4.1.7.2.5. Prueba estadística T de Student en Python

Para realizar la prueba T de Student, se utilizó el código que se detalla a continuación:

```

# Instalar librerías necesarias
import pandas as pd
from scipy import stats
# Cargar los datos desde un archivo Excel
df = pd.read_excel('ruta_archivo.xlsx')
# Realizar prueba T apareada
t_stat, p_value = stats.ttest_rel(df['Peso'], df['prediccion'])

# Imprimir resultados de prueba T
print(f'Estadístico t: {t_stat:.4f}')
print(f'p_valor: {p_value:.4f}')

# Decisión
alpha = 0.05
if p_value < alpha:
    print(f""Decisión: Se rechaza H_0
          Conclusión: H_1 es cierta
          Existe evidencia necesaria para decir que hay diferencia significativa entre el
          peso real del vehículo y el peso predicho por el programa, con un nivel de
          confianza del 95%.""")
else:
    print(f""Decisión: No se rechaza H_0
          Conclusión: H_1 es falsa
          No existe evidencia necesaria para decir que hay diferencia significativa entre
          el peso real del vehículo y el peso predicho por el programa, con un nivel de
          confianza del 95%.""")

```

4.1.7.2.6. Resultados de prueba T de Student

Estadístico t : - 2.1800

Valor p : 0.0293

Decisión: Se rechaza H_0

Conclusión: H_1 es cierta

Existe evidencia necesaria para decir que hay diferencia significativa entre el peso real del vehículo y el peso predicho por el programa, con un nivel de confianza del 95%.

4.1.7.2.7. Interpretación

Estadístico $t = -2.18$:

Para saber si existe una diferencia entre las medias de los datos, se toma en cuenta los valores t críticos, y así establecer una zona en la que se puede rechazar o no a la hipótesis nula, para este caso la zona de rechazo cuenta con dos colas, al tener diferencia en las medias positivas y negativas, los valores t críticos se los encontró con el siguiente código:

```
from scipy import stats
# La cantidad de datos es 2851
n = 2851
alpha = 0.05
gl = n - 1 # Grados de libertad
# t crítico bilateral
t_critico_1 = stats.t.ppf(1 - alpha/2, df=gl)
t_critico_2 = t_critico_1*(-1)
print(f"t crítico_1: {t_critico_1:.4f}")
print(f"t crítico_2: {t_critico_2:.4f}")
```

Obteniendo como resultado, que el valor $t_{crítico1}$ es 1.96 que corresponde al lado derecho y el valor $t_{crítico2}$ es -1.96 que corresponde al lado izquierdo. Esto gráficamente lo podemos ver en la Figura 34.

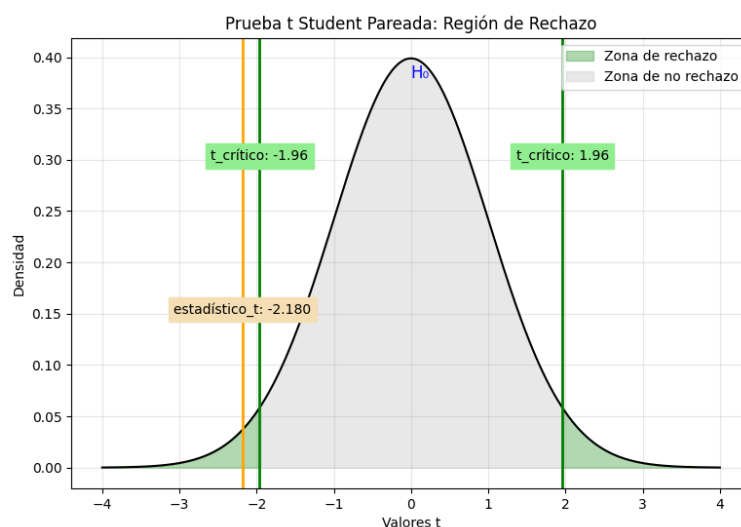


Figura 34. Región de rechazo para el estadístico t en prueba T de Student

En base a la gráfica de la figura 34, se observa que el valor t cae en la zona de rechazo de H_0 , por lo tanto, muestra que existe diferencia significativa entre los pesos reales y los predichos por el programa.

Valor $p = 0.0293$

El valor de 0.0293 es inferior a 0.05, lo que hace rechazar la hipótesis nula, obteniendo que existe diferencia significativa entre los pesos reales y predichos, por ende, el modelo no es válido y al no ser preciso, no se requiere proseguir a la fase 2, pues al rechazar la hipótesis nula respecto a la diferencia significativa de pesos, esto tanto con el estadístico t y el valor p , aceptando así la hipótesis nula general, al no cumplir con el objetivo principal del programa.

4.1.7.2.8. Decisión de hipótesis general

Decisión: No se rechaza H_0

Conclusión: H_1 es falsa

Existe evidencia necesaria para decir que un programa de pesaje basado en imágenes no permite optimizar el proceso de pesaje de vehículos de carga pesada.

4.2. DISCUSIÓN

El uso de inteligencia artificial para el procesamiento de imágenes, abre las puertas a investigaciones relacionadas con la solución a diversos problemas, tal como tratar de analizar imágenes de vehículos de carga pesada para poder obtener el peso que tienen en el momento en el que se les toma una foto, donde no se tiene parámetros definidos sobre cómo se desarrolla un programa que resulte funcional, por eso, se hace importante comprender el funcionamiento de estos modelos con respecto a problemas similares y generar una base para posteriores investigaciones.

Tras el desarrollo de los modelos y encontrar que el mejor, es aquel que fue entrenado con imágenes unificadas de otras 4 imágenes, cada una con un ángulo en específico del vehículo, para que el modelo sea forzado a encontrar las características de las partes más relevantes, que son la zona de las llantas, evitando que analice zonas de las imágenes con información irrelevante con respecto al peso, se obtiene un resultado en el que existen errores muy elevados con respecto al valor MAE, pues obtuvo un error de 5345.16 kg con respecto a las imágenes de prueba, pero se encontró que las redes identifican las características relevantes que se asocian al peso, así mismo, se observa comportamientos en la red, que no permitieron arrojar

errores más bajos, por lo tanto, estos aspectos se toman como base para ser mejorados y así en posteriores investigaciones se obtengan mejores resultados.

De igual manera, se llevó a cabo la comprobación de hipótesis con la prueba T de *Student*, para analizar si existe diferencia significativa en las medias entre los pesos reales y los predichos por el modelo 3, con 2851 datos en las muestras, donde se obtuvo que el estadístico $t = -2.18$ cae en la región de rechazo de la H_0 al ser menor que el valor $t_{crítico} = -1.96$, y también que el valor $p = 0.0293$ es menor al nivel de significancia de 0.05, por lo tanto, se rechaza H_0 y se concluye que existe la evidencia necesaria para decir que hay diferencia significativa entre el peso real del vehículo y el peso predicho por el programa, con un nivel de confianza del 95%. Con esto se manifiesta que el programa no es funcional, al existir diferencia entre los datos, es un resultado no esperado, debido a que se busca que los errores sean los permisibles para las básculas, que corresponde a más o menos 30 kg. Estos resultados permiten hacer la comprobación de la hipótesis de la investigación, en la que se acepta la H_0 , al existir evidencia necesaria para decir que un programa de pesaje basado en imágenes no permite optimizar el proceso de pesaje de vehículos de carga pesada. De igual manera, resulta fundamental comprender cómo estas nuevas tecnologías pueden mejorar los sistemas de transporte, con el fin de optimizar recursos empresariales, tratando de implementar un sistema de pesaje sin la complejidad estructural que significa usar básculas camioneras. Y así, se desarrolla la discusión con respecto a diferentes comportamientos dependiendo el problema y complejidad, lo que permite ver aspectos que permitan mejorar los modelos.

Siendo así, se analiza lo mencionada por Chen et al. (2022), el cual plantea que, en la investigación realizada para pesar los cadáveres de patos de Pekín en una línea de producción obtuvo mejores resultados tras tomar 150 imágenes de 50 patos, logrando una precisión de ± 58.8 gramos (MAD) con CNNs, mientras un modelo de regresión lineal de área de píxeles tenía un valor de ± 123.5 gramos, mejorando el sistema de pesaje, pues es más fácil y barato de implementar en las líneas de producción, demostrando que la riqueza visual mejora considerablemente la estimación. Lo mismo, que se realizó en este estudio, al tomar entre 2 y 3 fotos a un mismo vehículo en un mismo ángulo, pero tomando en cuenta que el presente estudio aborda un entorno más complejo, con variabilidad en los tipos de vehículos, escenarios no controlados y condiciones lumínicas naturales, un aspecto que en la línea de patos no tuvieron, donde para evitar el ruido, las imágenes fueron tomadas

tras poner un fondo azul y además al usar más imágenes de los mismos especímenes obtiene un amplio *dataset*, por lo tanto, el pesaje de patos resultó en un modelo eficiente y preciso, paralelamente, la complejidad de la CNN es menor a la usada en el problema de pesaje de vehículos, misma que comprende con 3 capas convolucionales cada una seguida de otra de agrupación, seguidas de dos capas completamente conectadas y finalmente una neurona en la capa de salida. Sin embargo, a pesar de usar una red neuronal ResNet, la precisión alcanzada no es óptima para la industria aún, debido a la complejidad del problema que es considerablemente mayor, pero con la estrategia de unificación de imágenes por ángulo, y la de tomar varias imágenes de un ángulo, en diferentes posiciones, han mostrado resultados relevantes que pueden ser mejorados, al entrenar más tiempo los modelos y aumento del *dataset*.

Con respecto a Ormeño y Zapata (2024), aplicaron la red Mask R-CNN para estimar el peso de cuyes, obteniendo una precisión de $R^2 = 80.2 \%$, donde, de 225 cuyes, obtuvieron 6244 imágenes al aumentar datos, tomando entre 3 y 5 imágenes por cuy, registrando el peso dado por una balanza, de ahí, redimensionando las imágenes de 3000 x 3000 píxeles, a 640 x 640 píxeles, simultáneamente, generaron imágenes girándolas o modificando levemente los píxeles, logrando un aplicativo con el que pueden dar seguimiento a los cuyes de manera accesible. En este caso, se aprecia un aumento de datos representativo, permitiendo tener imágenes con una cantidad limitada de cuyes, y se mira que la resolución utilizada fue más que suficiente para ese problema en específico, caso contrario, al problema del pesaje de vehículos, donde existe un error elevado, pero destacan las condiciones en las que se tomaron las imágenes, pues fueron más complejas, incluyendo diversidad de tipos de vehículos, iluminación y entornos no controlados. Además, a diferencia del enfoque basado en segmentación (Mask R-CNN), aquí se priorizó la regresión directa del peso, sin requerir segmentación explícita del objeto, lo que permitió simplificar el flujo de inferencia a partir de una imagen compuesta y también desataca un mayor consumo de recursos computacionales, pues la cantidad de píxeles usados en el presente proyecto corresponden a 1400 x 1400, con esto, se puede destacar que para obtener mejores resultados se debe tratar de obtener un *dataset* con una cantidad de datos lo suficientemente grande para generalizar bien y tomar las imágenes de manera más específica, donde aquellas tomadas con la visión lateral obtuvieron mejores resultados, 3534.38 *kg* en MAE, y así tratar de controlar de mejor

manera como en la toma de fotos de los cuyes, que pusieron cámaras en un lugar específico, además para este proyecto, es preferible identificar un lugar, en el que exista un piso nivelado y liso con amplitud para poner cámaras y que ahí se tomen las imágenes. Ambos estudios coinciden en demostrar el potencial de la visión artificial como herramienta no invasiva para estimar peso, pero difieren en el grado de precisión alcanzado debido al contexto de aplicación. En conjunto, estos resultados refuerzan la idea que, con un procesamiento adecuado de imágenes y un modelo ajustado al entorno, es posible reemplazar parcialmente métodos tradicionales de pesaje, optimizando tiempo y esfuerzo.

Posteriormente Maldonado (2021), desarrolló un sistema de predicción del peso de cerdos, a través de un sistema controlado, en el que obtiene imágenes cenitales infrarrojas, y está relacionada a datos como peso, chip, hora de pesaje, etc., a través de un sistema automatizado, logrando así obtener una cantidad de imágenes considerables, pues los cerdos debían pasar de manera obligatoria por la zona en que se toma las fotos, donde cuentan con una báscula, una cámara, computador y lector RFID, que actúa automáticamente una vez el peso se estabiliza, de ahí segmenta las imágenes, recortando a los cerdos, eliminando información redundante relacionada con el peso, inclusive eliminando la cabeza, ya que su posición suponía que en imágenes aparecía oculta, siendo las imágenes en que el cerdo aparecía centrado y sin cabeza, las que tenían un error MAE más bajo de 3.6 *kg* mediante el uso de CNNs, el entrenamiento de este modelo se llevó a cabo con 600 imágenes segmentadas. En contraste, el presente estudio abordó la estimación del peso de vehículos de carga pesada mediante imágenes a escala de grises, con captura de múltiples ángulos, pero sin condiciones controladas, algo que el investigador destaca en su investigación, donde automatizó el proceso, además, se tuvo un proceso similar de la segmentación que utilizó, debido a que en las imágenes de los vehículos, se tomó imágenes por cada ángulo, se tomó la parte más importante de cada uno y estas se unieron, evitando información irrelevante, también se utilizó una red ResNet de 68 capas, y así como en los cerdos, no todos los ángulos de imágenes fueron igual de eficientes, pues el modelo 3, que fue forzado a buscar las características necesarias, igualmente, mostró un desempeño superior con relación a las imágenes laterales, requiriendo únicamente dos, una por cada lado, este resultó en un rendimiento inferior a 3534.38 *kg* en MAE, sin embargo la complejidad para conseguir imágenes sigue siendo alta. Y a pesar de tener

resultados en el desempeño diferente, se coinciden en validar el potencial de la inteligencia artificial para estimar el peso de cuerpos físicos sin contacto directo, con datasets más grandes para que puedan generalizar de mejor manera, en esta comparación destaca los sensores usados en la granja, la calidad del *dataset* y que el entorno de captura influyen directamente en la precisión del modelo. Futuras investigaciones podrían evaluar la posibilidad de incorporar cámaras con sensores en el pesaje vehicular, y aumentar la capacidad del modelo con una base más grande con datos estructurados (como tipo de eje o largo del vehículo), tal como se hizo con los metadatos en el estudio de Maldonado.

El trabajo de Arbañil (2020), propone un sistema de pesaje por conjunto de ejes basado en plataformas metálicas multimodulares y celdas de carga POWERCELL PDX, integradas con sensores de posicionamiento, terminales de pesaje programables y tecnología RFID para la identificación automática de camiones. Este sistema busca eliminar errores del pesaje manual, mejorar la distribución de carga por eje, reducir derrames de mineral y agilizar el flujo de transporte en la minera Volcán. A diferencia del enfoque físico-electromecánico mencionado, este estudio plantea una solución basada en visión computacional, utilizando imágenes a escala de grises y redes neuronales convolucionales para estimar el peso total del vehículo sin contacto físico. Ambos enfoques buscan optimizar el proceso de pesaje, pero difieren en su infraestructura y grado de automatización. Mientras Arbañil se enfoca en mejorar el sistema tradicional mediante sensores y controladores industriales (como el terminal IND780 de Mettler Toledo), la propuesta busca reducir el hardware requerido, aprovechando únicamente cámaras y *software* de procesamiento de imágenes. Aunque el sistema por ejes tiene mayor precisión, implica altos costos de instalación, mantenimiento, calibración, y requiere una infraestructura fija. En cambio, el sistema basado en imágenes a pesar de tener errores puede mejorar con base en un *dataset* más robusto, tomando en cuenta la forma en la que se toma las fotos e implementar un entorno controlado, para así obtener un proceso mucho más eficiente, en el que no se requiera tener toda una infraestructura física tan compleja como la que Arbañil automatizo con más sensores. En conjunto, ambos estudios demuestran distintas estrategias para abordar la optimización del pesaje en el transporte de carga. Mientras el sistema de Arbañil destaca por su precisión y control detallado por eje en entornos industriales cerrados, la solución por IA ofrece una alternativa más flexible, económica y sin intervención física, adecuada para contextos donde no se puede

instalar infraestructura pesada. A futuro, la combinación de sensores físicos y visión por computador podría brindar un sistema híbrido que maximice tanto precisión como eficiencia operativa.

El trabajo de López (2020) se enfoca en optimizar el proceso de pesaje mediante el desarrollo de un sistema de *software* que aplica patrones de diseño para reducir los tiempos de registro, asegurar la integridad de los datos y permitir la escalabilidad a distintas plataformas. Su enfoque se centra en la eficiencia del sistema desde el punto de vista del desarrollo y mantenimiento del *software*, implementando estructuras reutilizables y desacopladas sin modificar los mecanismos físicos de pesaje. A través de encuestas y pruebas funcionales, el estudio comprobó que el uso de patrones de diseño mejora significativamente la respuesta del sistema y la experiencia de los usuarios. En cambio, el presente trabajo plantea un enfoque diferente: reemplazar o complementar el sistema tradicional de pesaje físico mediante visión por computador e inteligencia artificial, utilizando una red neuronal convolucional profunda (ResNet) para estimar el peso de vehículos a partir de imágenes con varios ángulos. Aunque el sistema no alcanzó una precisión suficiente para sustituir a las básculas en la industria, pues se obtuvo un error MAE de 3534.38 *kg* respecto a imágenes tomadas lateralmente, siendo las de menor error, sí demostró ser más ágil y menos dependiente de infraestructura física, destacando como una solución de bajo costo operativo y con potencial de mejora. A diferencia del trabajo de López, que optimiza el *software* sobre un sistema físico existente, este proyecto propone una alternativa disruptiva basada en visión por computadora y aprendizaje automático. En ambas investigaciones, se tiene como objetivo optimizar este proceso, llegando a ser complementarias, uno desde la arquitectura del sistema para garantizar estabilidad y escalabilidad del sistema tradicional, y otro desde la exploración de nuevas tecnologías que permitan automatizar y flexibilizar el proceso sin contacto físico. Evidenciando que la eficiencia del pesaje no solo puede lograrse mejorando el *software* que opera las básculas, sino también innovando en el modo de pesar, lo cual abre la posibilidad a sistemas híbridos que integren patrones de diseño con inteligencia artificial para una solución más robusta y adaptable.

V. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Este trabajo desarrolló un sistema para estimar el peso de vehículos pesados mediante visión por computador e inteligencia artificial mediante una red neuronal convolucional profunda (ResNet). Los resultados mostrados por los modelos entrenados no dieron mediciones esperadas con respecto a la estimación del valor real de un vehículo, pero si permitió, entender cuál es el comportamiento de las CNNs con respecto a este problema, donde se encontró que las imágenes que fueron tomadas en lugares mejor controlados son aquellas que dan mejores predicciones, así mismo, el ángulo que mejores resultados da es aquel que se toman las imágenes de manera lateral al vehículo, esto con base en el modelo 2 y 3, en la que aquellas imágenes unidas entre laterales comprendían errores más bajos, en el modelo 3 obtuvo el menor error MAE, siendo de 3534,38 *kg*.
- Además, se identifica que el uso de básculas hoy en día ya es un aspecto obligatorio dentro de las actividades comerciales, pues estas deben cumplir normativas que especifiquen la cantidad comercializada, sin embargo, las básculas son sistemas con estructuras complejas, costosas de implementar y mantener, pero muy fiables, que, luego de ser calibradas deben tener errores en las pruebas de más o menos 30 *kg*.
- El modelo 3, es aquel que da mejores resultados, trata de forzar a la CNN a encontrar las características, esto al modificar las imágenes, tomando 4, una por cada esquina, obteniendo la zona más visible de cada una y uniéndolas en una sola imagen, eliminando el fondo y mostrando a un vehículo desde diferentes ángulos, pero en una sola imagen. A pesar de ser el mejor modelo, tuvo un error MAE de 5345.16 *kg* a nivel general, donde el menor error de todos fue con este modelo y tomando a vehículos de 2 ejes, es decir camiones, teniendo un error MAE de 3780.52 *kg* pues son más susceptibles de mostrar cambios al aplicar peso encima, caso contrario pasa con tractocamiones, que, por su estructura más robusta, evita que se noten cambios al no aumentar una gran cantidad de peso, siendo el menor error de estos de 8300.06 *kg*.

- Se determinó que los filtros de los modelos 2 y 3, si identifican al vehículo como objeto al que se quiere medir el peso, en el caso específico del modelo 2, se observó que una de sus capas analiza demasiados parámetros dentro de las imágenes que no aportan al aprendizaje del modelo, haciendo la red neuronal más pesada y menos eficiente en los entrenamientos. En el caso del modelo 3, al aplicar los filtros, se evidenció que la manipulación de imágenes fue de gran ayuda para el modelo, pues en cada filtro de la capa seleccionada, se observó resaltadas las características que sufren variaciones por el peso, como llantas, el espacio entre estas con el chasis y la forma del vehículo, evitando las partes superiores de estos.
- La precisión depende mucho de la calidad de las imágenes, pues en el depósito Transbolivariana C.A., las imágenes fueron tomadas encima de la báscula y para eso, la distancia entre las esquinas de los vehículos era diferente, puesto que se tomó un área en específico, y la forma de las imágenes cambia dependiendo la posición que se pongan los vehículos, esto con respecto a la métrica MAE superó a todos los modelos llegando a tener 7949.32 *kg* de error.
- A pesar de no haber obtenido los resultados esperados, se mostró que el modelo tiene gran potencial al comparar con las básculas tradicionales, pues se elimina por completo la infraestructura física compleja con las que operan, se requiere de una sola computadora potente que cargue los modelos y el costo de esta representa entre un 17% y 20% del costo total de una báscula nueva, considerando que los modelos si identifican las características necesarias, se hace evidente su relevancia en aplicaciones posteriores.

5.2. RECOMENDACIONES

- El programa tiene potencial para funcionar, pero requiere de un *dataset* más robusto y que los datos tengan ciertos parámetros específicos, primero, la cantidad de datos debe ser mayor, pues para esta investigación se tuvo la cantidad de 1390 registros fotográficos de vehículos, para evitar este problema, se requiere de invertir más tiempo en el proceso de recolección de datos, donde se deberá tomar una imagen por ángulo de cada vehículo que se pese, y enfocando las imágenes en las zonas de las llantas. Además, los modelos deben ser entrenados por tiempos más prolongados, mayores a las 10.8 horas del tiempo entrenado en el modelo 3 con 60 épocas.

- El espacio para la captura de imágenes debe ser controlado, para que las distancias entre la cámara y el vehículo sea cercanas, tomando en cuenta capturar imágenes por ángulos, al ser más eficientes para entrenar modelos, de igual manera que cuente con un piso nivelado para que los vehículos tengan una buena distribución de la carga, y que sea liso, pues en pisos con alteraciones las llantas llegan a afectar su forma con gran irregularidad.
- Un proceso de pesaje automatizado es más óptimo para recolectar datos, con un sistema de cámaras que capture la imagen y anote el peso correspondiente, así se evita tener a una persona tomando las fotos, con esto, se logra obtener una mayor cantidad de datos, pues se obtiene imágenes de todos los vehículos que lleguen a una báscula, sin importar el día o el clima, siendo considerablemente más eficiente, al no tener que rodear al vehículo para tomar imágenes ángulo por ángulo y anotando los pesos manualmente.

VI. REFERENCIAS BIBLIOGRÁFICAS

- Alvarado, M. (2015). Una mirada a la Inteligencia Artificial. *Revista Ingeniería, Matemáticas y Ciencias de la Información*, 2(3), 27-31. <https://dialnet.unirioja.es/servlet/articulo?codigo=7894426>
- Amazon Web Services. (2024). ¿Qué es Python? - Explicación del lenguaje Python - AWS. Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/python/>
- Arbañil, V. (2020). *Diseño de un sistema de pesaje para conjunto de ejes para el control de registro de mineral en camiones Bitren* [Tesis de grado, Universidad César Vallejo]. <https://hdl.handle.net/20.500.12692/53162>
- Arias, J. y Covinos, M. (2021). *Diseño y metodología de la investigación* (1.ª ed.). Enfoques Consulting EIRL.
- Ayala, M., y Campos, C. (2013). *Evolución del peso bruto vehicular en Colombia y su incidencia en los pavimentos de la red vial de carreteras* [Especialización en Ingeniería de Pavimentos]. Universidad Militar Nueva Granada. <https://repository.umng.edu.co/bitstreams/6f4a80ba-ac66-499d-82ad-8aebfc0f04e4/download>
- Barragán, X. (2023). Situación de la inteligencia artificial en el Ecuador en relación con los países líderes de la región del Cono Sur. *FIGEMPA: Investigación y Desarrollo*, 16(2), 23-38. <https://doi.org/10.29166/revfig.v16i2.4498>
- Bascosta (2023). Impacto ambiental del transporte y el papel de las básculas camioneras en la sostenibilidad. *BASCOSTA BLOG*. <https://www.bascosta.com/blog/impacto-ambiental-del-transporte-y-el-papel-de-las-basculas-camioneras-en-la-sostenibilidad/>
- Bascosta (2023, 2 agosto). Tipos de básculas camioneras y sus características. *Bascosta - Soluciones inteligentes de*

pesaje. <https://www.bascosta.com/blog/tipos-de-basculas-camioneras-y-sus-caracteristicas/>

Bascosta (2024). La importancia de las básculas camioneras en la industria manufacturera. *BASCOSTA BLOG*. <https://www.bascosta.com/blog/la-importancia-de-las-basculas-camioneras-en-la-industria-manufacturera/#:~:text=Las%20b%C3%A1sculas%20permiten%20una%20r%C3%A1pida,m%C3%A1s%20r%C3%A1pida%20de%20los%20veh%C3%ADculos.>

Bascosta (2024, 13 abril). La importancia del pesaje preciso en el transporte de mercancías: optimizando la eficiencia logística. *Bascosta - Soluciones inteligentes de pesaje*. <https://www.bascosta.com/blog/la-importancia-del-pesaje-preciso-en-el-transporte-de-mercancias-optimizando-la-eficiencia-logistica/#:~:text=Se%20refiere%20a%20la%20medici%C3%B3n,y%20el%20cumplimiento%20de%20regulaciones.>

Basogain, X. (2008). *Redes neuronales artificiales y sus aplicaciones*. Escuela Superior de Ingeniería de Bilbao, EHU. https://ocw.ehu.eus/file.php/102/redes_neuro/Course_listing.html

Bernal, C. (2010). *Metodología de la Investigación* (3.a ed.). PEARSON EDUCACIÓN. <http://librodigital.sangregorio.edu.ec/librosusgp/B0061.pdf>

Bertoglio, J. (1993). *Introducción a la Teoría General de Sistemas* (Octava reimpresión). Grupo Noriega Editores. <https://camilos03.wordpress.com/wp-content/uploads/2015/08/1-introduccion-a-la-teoria-general-de-sistemas-oscar-johansen2-libre.pdf>

Camarillo, J., Saavedra, A., y Ramos, C. (2013). Recomendaciones para seleccionar índices para la validación de modelos. *TecnoLógicas*, 109-122. <https://www.redalyc.org/pdf/3442/344234341009.pdf>

Chen, R., Zhao, Y., Yang, Y., Wang, S., Li, L., Sha, X., Liu, L., Zhang, G. & Jung Li, W. (2022). Online estimating weight of white Pekin duck carcass by computer

vision. *Poultry Science*, 102(2), 102348.
<https://doi.org/10.1016/j.psj.2022.102348>

Chollet, F. (2018). *Deep Learning with Python*. Manning Publications
<http://cds.cern.ch/record/2301910>

Comunidad Andina. (2001). *Decisión 491. Reglamento Técnico Andino sobre límites de pesos y dimensiones de los vehículos destinados al transporte internacional de pasajeros y mercancías por carretera*. Comisión de la Comunidad Andina.

Comunidad Andina (2001). DECISIÓN 491. GACETA OFICIAL del Acuerdo de Cartagena.

Correa, M. (2008). *Fundamentos de la teoría de la información* (1.ª ed.). Instituto Tecnológico Metropolitano. <https://repositorio.itm.edu.co/bitstream/handle/20.500.12622/2019/Fundamentos%20de%20la%20teor%C3%ADa%20de%20la%20informaci%C3%B3n.pdf?sequence=1&isAllowed=y>

Fauland, J. (2018). *Inteligencia artificial en logística: Términos, aplicaciones y perspectivas*. SSI Schafer IT Solutions GmbH. <https://www.ssi-schaefer.com/resource/blob/1077946/21597041464333c25cf325b51f64c06d/inteligencia-artificial-en-log%C3%ADsticawhitepaper-dam-download-en-17151--data.pdf>

González, P. (2013). *Transporte y Logística Internacional*. Universidad de Las Palmas de Gran Canaria. https://accedacris.ulpgc.es/bitstream/10553/11886/4/Transporte_logistica_internacional.pdf

Gutiérrez, M., y Polo, E. (2023). INTELIGENCIA ARTIFICIAL DENTRO DE LA CADENA DE SUMINISTROS [AREANDINA Fundación Universitaria Del Área Andina]. En ARENDINA. <https://dig itk.areandina.edu.co/server/api/core/bitstreams/99c0f602-d71e-4e99-b8d4-1037b4c35e3a/content>

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Computer Vision Foundation*. https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html
- Hernández, R., Fernández, C., y Baptista, P. (2014). Metodología de la Investigación (6.a ed.). McGraw Hill. <https://www.esup.edu.pe/wpcontent/uploads/2020/12/2.%20Hernandez,%20Fernandez%20y%20BaptistaMetodolog%C3%ADa%20Investigacion%20Cientifica%206ta%20ed.pdf>
- Hitawala, S., & Cheriton, D. (2018). Evaluating ResNeXt Model Architecture for Image Classification. *University Of Waterloo*, 1805.08700v1. <https://arxiv.org/abs/1805.08700>
- Huang, G., Liu, Z., Maaten, L., & Weinberger, K. (2018). Densely connected convolutional networks. *Cornell, Tsinghua University And Facebook AI Research*, 1608.06993v5. <https://arxiv.org/abs/1608.06993v5>
- INEN (2023). Guía Práctica Metrológica. En *Servicio Ecuatoriano de Normalización*. Ley Del Sistema Ecuatoriano de la Calidad (2021). Ley 76 de la Ley de Sistema Ecuatoriano de la Calidad. *Presidencia del Congreso Nacional*.
- Lind, D., Marchal, W., & Wathen, S. (2011). *Estadística aplicada a los negocios y la economía*. McGraw Hill. <https://gc.scalahed.com/recursos/files/r161r/w25633w/Estadistica%20para%20Administracion%20y%20Negocios.pdf>
- López, B. (2020). *Desarrollo e implementación de un sistema basado en patrones de diseño para optimizar los servicios de pesaje en la empresa Balanzas Vegasystems SAC* [Tesis de grado, Universidad Peruana de Ciencias e Informática]. <http://repositorio.upci.edu.pe/handle/upci/120>
- Malave, S. (2018). La Seguridad del Transporte Terrestre: Retos y Perspectivas. *Ciencias Sociales y Económicas*, 2(1), 38-54. <https://doi.org/10.18779/csye.v2i1.265>

- Maldonado, M. (2021). *Data Science: Estimación del peso de un cerdo* [Tesis de grado, Universidad Autónoma de Barcelona]. <https://ddd.uab.cat/record/257805>
- Marín, M., Herrera, M., y Acero, A. (2023). La inteligencia artificial en la generación de imágenes: consideraciones desde el diseño, la comunicación y el arte. *South Florida Journal Of Development*, 4(9), 3713-3728. <https://doi.org/10.46932/sfjdv4n9-028>
- Martínez, P. (2006). *El transporte terrestre de mercancías. Responsabilidad por averías, faltas y retrasos*. Díaz de Santos. <https://www.editdiazdesantos.com/wwwdat/pdf/9788479787851.pdf>
- Mettler-Toledo (2020). *Pesaje de Vehículos. Pesaje y Medición Industriales.*, 12.
- Ministerio de Transporte y Obras Públicas (MTO) (2016). Acuerdo Ministerial No. 018-2016 (Ecuador). https://www.obraspublicas.gob.ec/wp-content/uploads/downloads/2016/08/STI_C_AM_MTOP_018_2016_Pesos-y-Dimensiones.pdf
- Ministerio de Transporte y Obras Públicas (MTO) (2015). *Reforma Reglamento A la Ley de Caminos de la República del Ecuador* (Ecuador).
- Moreno, E. (2004). El sobrepeso en el autotransporte de carga: elementos para su estudio y control. *Secretaría de Comunicaciones y Transporte (SCT)*, No 250. <https://imt.mx/archivos/Publicaciones/PublicacionTecnica/pt250.pdf>
- Ormeño, Y., y Zapata, A. (2024). Aplicación de la red convolucional Mask R-CNN para la estimación del peso corporal del cuy. *Revista Científica de Sistemas E Informatica*, 4(1), e614. <https://doi.org/10.51252/rcsi.v4i1.614>
- Pinto, E. (2021). *Propuesta de un modelo de gestión de logística verde, caso de estudio Cooperativa de Transporte Pesado Los Andes* [Trabajo de titulación]. http://dspace.esPOCH.edu.ec/bitstream/123456789/15169/1/11_2T0197.pdf
- Qichuang. (2024). Errores comunes en el uso de escamas de camiones - Peso de control de calidad. QC

Weighing. <https://www.qcweighing.com/es/common-mistakes-in-using-truck-scales.html>

Ramírez, J. y Chacón, M. (2011). Redes neuronales artificiales para el procesamiento de imágenes, una revisión de la última década. *RIEE&C: Revista de Ingeniería Eléctrica, Electrónica y Computación*, 9(1). https://itson.mx/publicaciones/rieeyc/Documents/v9/art2vf_redes_neuronales_artificiales_para_el_procesamiento_de_imagenes-una_revision_de_la_ultima_decada.pdf

Rouhiainen, L. (2018). Inteligencia Artificial. 101 cosas que debes saber hoy sobre nuestro futuro. *Planeta*, S.A. https://planetadelibrosar0.cdnstatics.com/libros_contenido_extra/40/39307_Inteligencia_artificial.pdf

Ruano, L., Montenegro, J., y León, R. (2019). Inteligencia artificial e innovación: campos de aplicación para la industria del Ecuador. *Visión Empresarial*, (9), 163-172. <https://doi.org/10.32645/issn.1390-6852>

Salguero, M. (2003). *Implementación y modernización en los sistemas de pesaje, para el mejor control de unidades vehiculares con carga pesada que transitan sobre la red vial de Guatemala*. [Trabajo de graduación]. Universidad de San Carlos de Guatemala.

Tan, M., & Le, Q. (2020). EfficientNet: Rethinking model scaling for convolutional neural networks. *arXiv*. <https://doi.org/10.48550/arXiv.1905.11946>

Tapia, E. (2024). ¿Cuál es el costo real de la energía eléctrica en Ecuador? <https://www.primicias.ec/economia/precio-real-energia-ecuador-subsidio-cortes-luz-81293/>

TechPowerUp (2025). NVIDIA GeForce RTX 3090. TechPowerUp. <https://www.techpowerup.com/gpu-specs/geforce-rtx-3090.c3622>

TensorFlow (2024). *Introducción a TensorFlow*. TensorFlow. <https://www.tensorflow.org/learn?hl=es-419>

Teuwen, J., & Moriakov, N. (2020). Chapter 20 - Convolutional neural networks. En *Handbook of Medical Image Computing and Computer Assisted*


Intervention (pp. 481-501). The Elsevier and MICCAI Society Book Series.
<https://doi.org/10.1016/B978-0-12-816176-0.00025-9>

TYGOM. (2023). *Peligros del Exceso de Carga en un Camión* - Tygom. Tygom.
<https://www.tygom.es/peligros-exceso-carga-camion/>

Walpole, R., Myers, R., y Myers, S. (2012). *Probabilidad y estadística para ingeniería y ciencias* (9.ª ed.). PEARSON EDUCACIÓN.
https://bibliotecavirtualaserena.wordpress.com/wp-content/uploads/2017/05/libro_probabilidad-y-estadistica-para-ingenerc3ada-y-ciencias-ronald-e-walpole-mayers.pdf

VII. ANEXOS

Anexo 1. Acta de la sustentación de Predefensa del TIC




UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI

FACULTAD DE COMERCIO INTERNACIONAL, INTEGRACIÓN, ADMINISTRACIÓN Y ECONOMÍA EMPRESARIAL

CARRERA DE LOGÍSTICA Y TRANSPORTE

ACTA

DE LA SUSTENTACIÓN ORAL DE LA PREDEFENSA DEL TRABAJO DE INTEGRACIÓN CURRICULAR



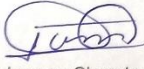
ESTUDIANTE: Herrera Montenegro Diego Joel	CÉDULA DE IDENTIDAD: 0450167978
PERIODO ACADÉMICO: 2025B	
PRESIDENTE TRIBUNAL: Ph.D. Montenegro Obando Blanca Liliana	DOCENTE TUTOR: MSc. Mafla Bolaños Iván Gabriel
DOCENTE: MSc. López Ruano Juan Carlos	
TEMA DEL TIC: "Diseño de un programa para el pesaje de vehículos de carga pesada a partir de imágenes con Inteligencia Artificial"	

No.	CATEGORÍA	Evaluación cuantitativa	OBSERVACIONES Y RECOMENDACIONES
1	PROBLEMA - OBJETIVOS	9,67	Revisar la redacción del objetivo general (Optimizar)
2	FUNDAMENTACIÓN TEÓRICA	9,00	Colocar las definiciones de la parte estadística, como hipótesis, correlación de variables
3	METODOLOGÍA	9,33	
4	RESULTADOS	8,33	Justificar la prueba hipótesis, que significan, cuando habla de correlaciones colocar el marco teórico, explicar la diferencia de fotos (1390 vs 19000). Explicar bajo que criterio se aceptaría Ho.
5	DISCUSIÓN	9,67	Incluir el análisis de la comprobación de la hipótesis
6	CONCLUSIONES Y RECOMENDACIONES	10,00	
7	DEFENSA, ARGUMENTACIÓN Y VOCABULARIO PROFESIONAL	9,00	Mejorar el vocabulario técnico con respecto a los resultados
8	FORMATO, ORGANIZACIÓN Y CALIDAD DE LA INFORMACIÓN	9,00	Revisar estilo, redacción en todo el documento

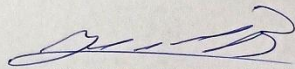
Obteniendo una nota de: **9,20** Por lo tanto, **APRUEBA** ; debiendo el o los investigadores acatar el siguiente artículo:

Art. 36.- De los estudiantes que aprueban el informe final del TIC con observaciones.- Los estudiantes tendrán el plazo de 10 días para proceder a corregir su informe final del TIC de conformidad a las observaciones y recomendaciones realizadas por los miembros del Tribunal de sustentación de la pre-defensa.

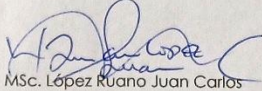
Para constancia del presente, firman en la ciudad de Tulcán el **jueves, 4 de septiembre de 2025**



Ph.D. Montenegro Obando Blanca Liliana
PRESIDENTE TRIBUNAL



MSc. Mafla Bolaños Iván Gabriel
DOCENTE TUTOR



MSc. López Ruano Juan Carlos
DOCENTE

Anexo 2. Certificado del abstract por parte de idiomas



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI FOREIGN
AND NATIVE LANGUAGES CENTER

ABSTRACT- EVALUATION SHEET				
NAME: Herrera Montenegro Diego Joel				
DATE: Miércoles, 17 de septiembre de 2025				
Topic: "Diseño de un programa para el pesaje de vehiculos de carga pesada a partir de imágenes con Inteligencia Artificial"				
MARKS AWARDED		QUANTITATIVE AND QUALITATIVE		
VOCABULARY AND WORD USE	Use new learnt vocabulary and precise words related to the topic	Use a little new vocabulary and some appropriate words related to the topic	Use basic vocabulary and simplistic words related to the topic	Limited vocabulary and inadequate words related to the topic
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input checked="" type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
WRITING COHESION	Clear and logical progression of ideas and supporting paragraphs.	Adequate progression of ideas and supporting paragraphs.	Some progression of ideas and supporting paragraphs.	Inadequate ideas and supporting paragraphs.
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
ARGUMENT	The message has been communicated very well and identify the type of text	The message has been communicated appropriately and identify the type of text	Some of the message has been communicated and the type of text is little confusing	The message hasn't been communicated and the type of text is inadequate
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
CREATIVITY	Outstanding flow of ideas and events	Good flow of ideas and events	Average flow of ideas and events	Poor flow of ideas and events
	EXCELLENT: 2 <input type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
SCIENTIFIC SUSTAINABILITY	Reasonable, specific and supportable opinion or thesis statement	Minor errors when supporting the thesis statement	Some errors when supporting the thesis statement	Lots of errors when supporting the thesis statement
	EXCELLENT: 2 <input checked="" type="checkbox"/>	GOOD: 1,5 <input type="checkbox"/>	AVERAGE: 1 <input type="checkbox"/>	LIMITED: 0,5 <input type="checkbox"/>
TOTAL/AVERAGE	9 - 10: EXCELLENT 7 - 8,9: GOOD 5 - 6,9: AVERAGE 0 - 4,9: LIMITED		TOTAL 9	



**UNIVERSIDAD POLITÉCNICA ESTATAL DEL
CARCHI- FOREIGN AND NATIVE LANGUAGES
CENTER**

**Informe sobre el Abstract de Artículo Científico
o Investigación.**

Autor: Herrera Montenegro Diego Joel

Fecha de recepción del abstract: Miércoles, 17 de septiembre de 2025

Fecha de entrega del informe: Miércoles, 17 de septiembre de 2025

El presente informe validará la traducción del idioma español al inglés si alcanza un porcentaje de: 9 – 10 Excelente.

Si la traducción no está dentro de los parámetros de 9 – 10, el autor deberá realizar las observaciones presentadas en el ABSTRACT, para su posterior presentación y aprobación.

Observaciones:

Después de realizar la revisión del presente abstract, éste presenta una apropiada traducción sobre el tema planteado en el idioma Inglés. Según la rúbrica de evaluación de la traducción en Inglés, ésta alcanza un valor de 9; por lo cual se valida dicho trabajo.

Atentamente



MA. Martha Viveros
Docente responsable del
CIDEN

Anexo 3. Fichas de observación



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE COMERCIO INTERNACIONAL, INTEGRACIÓN, ADMINISTRACIÓN Y ECONOMÍA EMPRESARIAL
CARRERA DE INGENIERÍA EN LOGÍSTICA Y TRANSPORTE

FICHA DE OBSERVACIÓN PARA EL DESARROLLO DEL PROGRAMA DE PESAJE DE VEHÍCULOS DE CARGA PESADA CON INTELIGENCIA ARTIFICIAL


El objetivo de este instrumento es recopilar información para evaluar el desempeño del programa de pesaje de vehículos de carga pesada a través de imágenes. Los datos obtenidos serán utilizados exclusivamente con fines académicos.

Ficha 1. Caracterización

Observador		
Fecha		
DESCRIPCIÓN	OBSERVACIÓN	UNIDAD DE MEDIDA
Conjunto de datos		
Cantidad de imágenes		Total
Resolución de Imágenes		N, Pixeles
Ángulos de cámara		
Dimensiones de área de tomado de imágenes		m ²
Calidad de luz en la zona de tomar imágenes		
Infraestructura para las instalaciones de básculas		
Dimensiones de báscula		m ²
Nivel de báscula		
Tipo de báscula		
Capacidad máxima de báscula		Toneladas

Ficha 2. Tiempos

Observador									
Fecha									
Proceso de pesaje con Programa con IA									
N. Observación	Tipo de vehículo	Tiempo	Condiciones de captura de imágenes				Ángulos de captura de imágenes		
			Soleado	Nublado	Nocturno	Contraluz	Frontal y Trasera	Laterales	Diagonales
1									
2									
3									
...									
n									
Proceso de pesaje con báscula									
N. Observación	Tipo de vehículo	Tiempo	OBSERVACIONES						
1									
2									
3									
...									
n									


MSc. Mafla Bolaños Iván Gabriel
Tutor



Validador

Anexo 4. Fichas documentales



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE COMERCIO INTERNACIONAL, INTEGRACIÓN, ADMINISTRACIÓN Y
ECONOMÍA EMPRESARIAL



CARRERA DE INGENIERÍA EN LOGÍSTICA Y TRANSPORTE

FICHA DOCUMENTAL PARA EL DESARROLLO DEL PROGRAMA DE PESAJE DE VEHÍCULOS DE CARGA PESADA CON INTELIGENCIA ARTIFICIAL

El objetivo de este instrumento es recopilar información para evaluar el desempeño del programa de pesaje de vehículos de carga pesada a través de imágenes. Los datos obtenidos serán utilizados exclusivamente con fines académicos.

Ficha 1. Resultados de entrenamiento

Investigador					
Fecha					
Indicadores de entrenamiento del modelo					
Formulas					
Precisión	$PRECISIÓN = \frac{\text{Número de Predicciones con Error} < 100}{\text{Total de Predicciones}} * 100$				
Exactitud	$MAE = \frac{1}{n} \sum_{i=1}^n Peso Real_i - Peso Predicho_i $				
	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Peso Real_i - Peso Predicho_i)^2}$				
	$R^2 = 1 - \frac{\sum_{i=1}^n (Peso Real_i - Peso Predicho_i)^2}{\sum_{i=1}^n (Peso Real_i - Media del Peso Real)^2}$				
Resultados en el entrenamiento del modelo para el cálculo de indicadores					
Imagen	Peso Real	Peso Predicho	Error Absoluto Peso Real - Peso Predicho		
			Error < 100 kg	Error 100-300 kg	Error > 500 kg
1.					
2.					
3.					
...					
n					

Ficha 2. Costos Operativos de Básculas Tradicionales

Categoría	Elemento a Observar	Descripción / Observación	Costo Aproximado (USD)	Frecuencia
Adquisición	Precio de compra	Costo inicial de la báscula		Pago único
	Garantía	Duración y cobertura de la garantía		Pago único
Mantenimiento	Calibración	Costo por calibración periódica		Semestral
	Reparaciones	Costo por reparaciones específicas		Eventual
	Repuestos	Precio de repuestos más comunes		Eventual
Operación diaria	Energía eléctrica	Consumo eléctrico del sistema		Mensual
	Personal operario	Sueldo del personal que opera la báscula		Mensual
Infraestructura	Base de concreto	Costo de construcción e instalación		Pago único
	Espacio ocupado	Renta o valor del espacio donde opera la báscula		Anual
Costos indirectos	Certificaciones	Gastos por certificaciones normativas		De 3 a 5 años
	Multas por incumplimiento	Multas asociadas a fallas en operación o incumplimiento normativo		Eventual


MSc. Mafía Bolaños Iván Gabriel
Tutor


Validador

Anexo 5. Cuestionario



UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
FACULTAD DE COMERCIO INTERNACIONAL, INTEGRACIÓN, ADMINISTRACIÓN Y
ECONOMÍA EMPRESARIAL
CARRERA DE INGENIERÍA EN LOGÍSTICA Y TRANSPORTE

CUESTIONARIO DIRIGIDO AL PRESIDENTE DE LA ASOCIACION DE TRANSPORTE PESADO DEL CARCHI

El objetivo de este instrumento es recopilar información para evaluar el desempeño del programa de pesaje de vehículos de carga pesada a través de imágenes. Los datos obtenidos serán utilizados exclusivamente con fines académicos.

Encuestado: _____

Fecha: _____

1. ¿Cómo describiría el impacto general de las básculas en el funcionamiento de las empresas de transporte, especialmente en la optimización de sus operaciones logísticas?
2. ¿Considera que los transportistas están suficientemente informados y conscientes sobre las normativas de la CAN relacionadas con el pesaje de vehículos?
3. ¿Cómo afecta la falta de básculas al costo operativo de las empresas de transporte?
4. ¿Qué impacto tiene el uso de básculas en la prevención y gestión de multas por sobrepeso según las normativas de la CAN?
5. ¿Qué tan efectivas son estas normativas, considera que son un desafío o una ventaja para las empresas de transporte?
6. ¿Cómo cree que tecnologías como la inteligencia artificial pueden complementar o mejorar los sistemas de pesaje tradicionales?
7. ¿Considera que un programa basado en inteligencia artificial para estimar el peso podría ser una alternativa viable a las básculas tradicionales?

MSc. Maffa Bolaños Iván Gabriel

Tutor

Validador

Anexo 6. Solicitud para levantamiento de información en Transbolivariana C.A.



Oficio N° UPEC-FCIAEE-2024-0349-Of
Tulcán, 07 de octubre de 2024

Ingeniero
Johann Benítez Castro
SUB-GERENTE
DEPÓSITO TEMPORAL ADUANERO TRANSBOLIVARIANA S.A. TULCÁN
Presente

Asunto: Solicitud de información para desarrollar el trabajo de Integración Curricular – HERRERA MONTENEGRO (LYT)

De mi consideración:

Reciba un atento y cordial saludo de la Universidad Politécnica Estatal del Carchi UPEC, especialmente de la Facultad de Comercio Internacional Integración, Administración y Economía Empresarial – FCIAEE; a la vez que le deseamos éxitos en las funciones que usted acertadamente desempeña.

Por medio del presente solicito de la manera más comedida se autorice a quien corresponda el ingreso a las Instalaciones del Depósito Temporal Aduanero Transbolivariana S.A. Agencia Tulcán; al señor HERRERA MONTENEGRO DIEGO JOEL, portador de la cédula de ciudadanía N° 0450167978, estudiante de octavo nivel de la carrera de Logística y Transporte con la finalidad de que se le dé el permiso necesario para tomar imágenes, características de los vehículos de carga pesada en el área de pesaje, datos investigativos que servirán al estudiante para desarrollar el Trabajo de Integración Curricular denominado **"Diseño de un programa para el pesaje de vehículos de carga pesada a partir de imágenes con inteligencia artificial"**.

En virtud de lo antes mencionado me permito indicar que la información obtenida se utilizará con fines estrictamente académicos.

Por la atención que se digna dar al presente anticipo mi agradecimiento.

Atentamente,



MSc. Marcelo Cahuasquí

**DECANO DE LA FACULTAD DE COMERCIO INTERNACIONAL, INTEGRACIÓN,
ADMINISTRACIÓN Y ECONOMÍA EMPRESARIAL**
"Educamos para transformar el mundo"

Trans Bolivariana
DEPÓSITO TEMPORAL
info: transbolivariana.ec
Tel: 0957020040
ECL 8140

Calle Antisana y Av. Universitaria
Telf: (06) 2980837 - 2984435
info@upec.edu.ec
www.upec.edu.ec
Tulcán - Ecuador

Anexo 7. Solicitud para levantamiento de información a SENA E



Oficio Nro. SENA E-DDT-2025-0140-OF

Tulcán, 18 de febrero de 2025

Asunto: Respuesta a Solicitud de Información para el Desarrollo de TIC HERRERA (LYT) - DISTRITO TULCÁN

Señor Economista
Guillermo Marcelo Argoti Enríquez
ADUANOR SERVICE

Licenciada
Narciza de Jesús Montenegro López
Gerente General
MIDECAR MOVIMIENTOS INTERNOS DE CARGA CIA. LTDA.

Ingeniera
Paola Lorena Terán Obando
COMPANIA BODEGAS PRIVADAS TERAN

Señor Magíster
Edwin Marcelo Cahuasquí Cevallos
Decano FCIAEE
UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI
En su Despacho

De mi consideración:

Reciba un cordial y atento saludo de quien suscribe el presente oficio, a la vez deseándole éxitos en las funciones que acertadamente desempeña.

En atención al Oficio Nro. UPEC-FCII-2025-0007-OF de fecha 31 de enero de 2025 y suscrito por el Mgs. Edwin Marcelo Cahuasquí Cevallos - DECANO FCIAEE de la UPEC, signado institucionalmente con documento Nro. SENA E-DDT-2025-0030-E; mediante el cual señala: "...Por medio del presente solicito muy comedidamente se autorice al señor HERRERA MONTENEGRO DIEGO JOEL, portadora de la cédula N° 0450167978, estudiante de noveno nivel de la Carrera de Logística y Transporte, el ingreso a las Instalaciones de los depósitos temporales ADUANOR CIA.LTDA., BODEGAS PRIVADAS TERÁN CIA LTDA. Y MIDECAR MOVIMIENTOS INTERNOS DE CARGA CIA LTDA ; con la finalidad de tomar imágenes de los vehículos de carga pesada que ingresan a las áreas de pesaje de los depósitos, registrar sus pesos y las características de los mismos, información necesaria para el entrenamiento del modelo de inteligencia artificial, en un periodo de dos meses quince días, información para el desarrollo de su trabajo de integración curricular cuyo tema es: "Diseño de un programa para el pesaje de vehículos de carga pesada a partir de imágenes con inteligencia artificial"".

Amparado en la cláusula CUARTA y SÉPTIMA del Convenio Interinstitucional SENA E – UPEC, firmado el 22 de abril de 202, el cual en su parte pertinente manifiesta: "...CUARTA.- OBJETO.- El presente convenio tiene por objeto la cooperación interinstitucional mediante el intercambio de información, capacitación, acompañamiento y apoyo de gestión, tendiente a promover el conocimiento y manejo de las operaciones de comercio exterior y aduanas, estableciendo para el efecto los nexos de coordinación y cooperación necesarios entre estas dos entidades del Estado, en cumplimiento de lo dispuesto en la Constitución de la República del Ecuador..." (...)"...SÉPTIMA.- COMPROMISOS.- 7.1. Por parte del SENA E se compromete a: a) Intercambiar la información requerida para el desarrollo de proyectos conjuntos, de conformidad con lo establecido en la cláusula de Intercambio de información del presente Convenio; b) Trabajar conjuntamente con LA UNIVERSIDAD en los proyectos generales de interés común en base a lo remitido por el Equipo Interinstitucional...".

Servicio Nacional de Aduana del Ecuador
Dirección: Av. 24 de Mayo y Alfonso Mena (Sector El Capulí)
Código postal: 040702 / Tulcán-Ecuador. Teléfono: +593-9 299 4051

EL NUEVO
Ecuador

1/2

* Documento firmado electrónicamente N° 0140-2025



Oficio Nro. SENAE-DDT-2025-0140-OF

Tulcán, 18 de febrero de 2025

Al respecto esta Dirección Distrital de Tulcán, en base al "Convenio de Cooperación Interinstitucional entre el Servicio Nacional de Aduana del Ecuador y la Universidad Politécnica Estatal del Carchi"; **AUTORIZA** al Sr. **HERRERA MONTENEGRO DIEGO JOEL** portador de la cédula N° **0450167978**, estudiante de noveno nivel de la Carrera de Logística y Transporte, el ingreso a las Instalaciones de los depósitos temporales **ADUANOR CIA.LTDA., BODEGAS PRIVADAS TERÁN CIA LTDA. Y MIDECAR MOVIMIENTOS INTERNOS DE CARGA CIA LTDA.**, con el fin únicamente de tomar imágenes de los vehículos de carga pesada como la obtención de los pesos y las características de los mismos. Dicha autorización será de 2 meses 15 días a partir de la suscripción de la presente.

Los datos recabados serán netamente de carácter académico (estadísticos), y está prohibida su difusión aquella información de carácter confidencial o reservada de cualquier naturaleza (placas, números de identificación, otras) en los términos previstos en la Ley.

Se solicita a los depósitos temporales: **ADUANOR CIA.LTDA., BODEGAS PRIVADAS TERÁN CIA LTDA. Y MIDECAR MOVIMIENTOS INTERNOS DE CARGA CIA LTDA.**; brindar las facilidades correspondientes por el plazo señalado.

Particular que comunico para los fines pertinentes.

Atentamente,

Documento firmado electrónicamente

Mgs. Xavier Olmedo Arias Sepulveda
DIRECTOR DISTRITAL TULCAN

Referencias:

- SENAE-DDT-2025-0030-E

Copia:

Señor Magíster
Argenis Lissander Heredia Campaña
Director de la Carrera de Logística y Transporte
UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI

Señor Ingeniero
Joffre Leodan Guerron Aguilar
Asistente Administrativo/a
UNIVERSIDAD POLITÉCNICA ESTATAL DEL CARCHI

Señorita Ingeniera
Andrea Magaly Obando Andrade
Directora de Despacho y Control Zona Primaria Tulcán

Señorita
Kimberly Dayana Acosta Paucar
Asistente de Abogacía

kdap/AMOA



EL NUEVO ECUADOR
KAVIER OLMEDO ARIAS
SEPULVEDA

Servicio Nacional de Aduana del Ecuador
Dirección: Av. 24 de Mayo y Alfilina Niemi (Sector El Capal)
Código postal: 040102 / Tulcán-Ecuador. Teléfono: +593-9 288 4261

* Documento firmado electrónicamente en www.senaec.gub.ec



2/2



Resolución Nro. INEN-INEN-2023-0007-RES

Quito, D.M., 09 de mayo de 2023

SERVICIO ECUATORIANO DE NORMALIZACIÓN

DIRECCIÓN EJECUTIVA

CONSIDERANDO:

Que, la Constitución de la República del Ecuador publicada en el Registro Oficial 449, del 20 de octubre de 2008 en su artículo 52 determina que: *"Las personas tienen derecho a disponer de bienes y servicios de óptima calidad y a elegirlos con libertad, así como a una información precisa y no engañosa sobre su contenido y características. La ley establecerá los mecanismos de control de calidad y los procedimientos de defensa de las consumidoras y consumidores; y las sanciones por vulneración de estos derechos, la reparación e indemnización por deficiencias, daños o mala calidad de bienes y servicios, y por la interrupción de los servicios públicos que no fuera ocasionada por caso fortuito o fuerza mayor"*;

Que, el artículo 226 de la Constitución de la República del Ecuador establece: *"Las instituciones del Estado, sus organismos, dependencias, las servidoras o servidores públicos y las personas que actúen en virtud de una potestad estatal ejercerán solamente las competencias y facultades que les sean atribuidas en la Constitución y la ley. Tendrán el deber de coordinar acciones para el cumplimiento de sus fines y hacer efectivo el goce y ejercicio de los derechos reconocidos en la Constitución"*;

Que, el artículo 227 de la Constitución de la República del Ecuador señala que: *"La administración pública constituye un servicio a la colectividad que se rige por los principios de eficacia, eficiencia, calidad, jerarquía, desconcentración, descentralización, coordinación, participación, planificación, transparencia y evaluación"*;

Que, el artículo 320 de la Constitución de la República del Ecuador indica: *"La producción, en cualquiera de sus formas, se sujetará a principios y normas de calidad, sostenibilidad, productividad sistémica, valoración del trabajo y eficiencia económica y social"*;

Que, el literal e) del artículo 77 de la Ley Orgánica de la Contraloría General del Estado, determina: *"Los Ministros de Estado y las máximas autoridades de las instituciones del Estado, son responsables de los actos, contratos o resoluciones emanados de su autoridad. Además, se establecen las siguientes atribuciones y obligaciones específicas: (...) e) Dictar los correspondientes reglamentos y demás normas secundarias necesarias para el eficiente, efectivo y económico funcionamiento de sus instituciones (...)"*;

Que, el artículo 1 de la Ley del Sistema Ecuatoriano de la Calidad tiene como objetivo

Anexo 9. Código de CNN ResNet -68

```
def identity_block(X, f, filters, stage, block):  
  
    #defining name basis  
  
    conv_name_base = "res" + str(stage) + block + "_branch"  
  
    bn_name_base = "bn" + str(stage) + block + "_branch"  
  
    #Retrieve Filters  
  
    F1, F2, F3 = filters  
  
    #Save the input value. We'll need this later to add back to the main path  
  
    X_shortcut = X  
  
    #First component of main path  
  
    X = Conv2D(filters = F1, kernel_size = (1,1), strides = (1,1), padding = "valid", name  
= conv_name_base + "2a", kernel_initializer = glorot_uniform(seed=0))(X)  
  
    X = BatchNormalization(axis = 3, name = bn_name_base + "2a")(X)  
  
    X = Activation("relu")(X)  
  
    #Second component of main path  
  
    X = Conv2D(filters = F1, kernel_size = (f,f), strides = (1,1), padding = "same", name =  
conv_name_base + "2b", kernel_initializer = glorot_uniform(seed=0))(X)  
  
    X = BatchNormalization(axis = 3, name = bn_name_base + "2b")(X)  
  
    X = Activation("relu")(X)  
  
    #Third component of main path  
  
    X = Conv2D(filters = F3, kernel_size = (1,1), strides = (1,1), padding = "valid", name  
= conv_name_base + "2c", kernel_initializer = glorot_uniform(seed=0))(X)  
  
    x = BatchNormalization(axis = 3, name = bn_name_base + "2c")(X)  
  
    #Final step: Add shortcut value to main path, and pass it through a RELU actvation  
  
    X = Add()([X, X_shortcut])  
  
    X = Activation("relu")(X)  
  
    return X
```

```

def convolutional_block(X, f, filters, stage, block, s=2):

    # defining name basis

    conv_name_base = "res" + str(stage) + block + "_branch"
    bn_name_base = "bn" + str(stage) + block + "_branch"

    #Retrieve Filters

    F1, F2, F3 = filters

    # Save the input value

    X_shortcut = X

    ##### MAIN PATH #####

    # First component of main path

    X = Conv2D(F1, (1, 1), strides = (s,s), name = conv_name_base + '2a',
kernel_initializer = glorot_uniform(seed=0))(X)

    X = BatchNormalization(axis = 3, name = bn_name_base + '2a')(X)

    X = Activation('relu')(X)

    # Second component of main path

    X = Conv2D(filters=F2, kernel_size=(f, f), strides=(1, 1), padding='same',
name=conv_name_base + '2b', kernel_initializer=glorot_uniform(seed=0))(X)

    X = BatchNormalization(axis=3, name=bn_name_base + '2b')(X)

    X = Activation('relu')(X)

    #Third component of main path

    X = Conv2D(filters=F3, kernel_size=(1,1), strides=(1,1), padding = "valid", name =
conv_name_base + "2c", kernel_initializer= glorot_uniform(seed=0))(X)

    X = BatchNormalization(axis = 3, name = bn_name_base + "2c")(X)

    ##### SHORTCUT PATH #####

    X_shortcut = Conv2D(F3, (1, 1), strides = (s,s), name = conv_name_base + '1',
kernel_initializer = glorot_uniform(seed=0))(X_shortcut)

```

```
X_shortcut = BatchNormalization(axis = 3, name = bn_name_base  
+ "1")(X_shortcut)
```

```
# Final step: Add shortcut value to main path, and pass it through a RELU  
activation
```

```
X = Add()([X, X_shortcut])
```

```
X = Activation("relu")(X)
```

```
return X
```

```
def ResNet68(input_shape = (64,64,3), classes = 3):
```

```
#Define the input as a tensor with shape input_shape
```

```
X_input = Input(input_shape)
```

```
# Zero-Padding
```

```
X = ZeroPadding2D((3,3))(X_input)
```

```
#Stage 1
```

```
X = Conv2D(64, (7,7), strides = (2,2), name = "conv1", kernel_initializer =  
glorot_uniform(seed=0))(X)
```

```
X = BatchNormalization(axis = 3, name = "bn_conv1")(X)
```

```
X = Activation("relu")(X)
```

```
X = MaxPooling2D((3,3), strides=(2,2))(X)
```

```
#Stage 2
```

```
X = convolutional_block(X, f = 3, filters = [64, 64, 256], stage = 2, block = "a", s = 1)
```

```
X = identity_block(X, 3, [64,64,256], stage = 2, block = "b")
```

```
X = identity_block(X, 3, [64,64,256], stage = 2, block = "c")
```

```
#Stage 3
```

```
X = convolutional_block(X, f = 3, filters = [128, 128, 512], stage = 3, block = "a", s =  
2)
```

```
X = identity_block(X, 3, [128,128,512], stage = 3, block = "b")
```

```
X = identity_block(X, 3, [128,128,512], stage = 3, block = "c")
```

```
X = identity_block(X, 3, [128,128,512], stage = 3, block = "d")
```

```
#Stage 4
```

```
X = convolutional_block(X, f = 3, filters = [256, 256, 1024], stage = 4, block = "a", s =  
2)
```

```
X = identity_block(X, 3, [256,256,1024], stage = 4, block = "b")
```

```
X = identity_block(X, 3, [256,256,1024], stage = 4, block = "c")
```

```
X = identity_block(X, 3, [256,256,1024], stage = 4, block = "d")
```

```
#Stage 5
```

```
X = convolutional_block(X, f = 3, filters = [256, 256, 1024], stage = 5, block = "a", s =  
2)
```

```
X = identity_block(X, 3, [256,256,1024], stage = 5, block = "b")
```

```
X = identity_block(X, 3, [256,256,1024], stage = 5, block = "c")
```

```
X = identity_block(X, 3, [256,256,1024], stage = 5, block = "d")
```

```
#Stage 6
```

```
X = convolutional_block(X, f = 3, filters = [256, 256, 1024], stage = 6, block = "a", s =  
2)
```

```
X = identity_block(X, 3, [256,256,1024], stage = 6, block = "b")
```

```
X = identity_block(X, 3, [256,256,1024], stage = 6, block = "c")
```

```

X = identity_block(X, 3, [256,256,1024], stage = 6, block = "d")

#Stage 7
X = convolutional_block(X, f = 3, filters = [512, 512, 2048], stage = 7, block = "a", s =
2)
X = identity_block(X, 3, [512, 512, 2048], stage = 7, block = "b")
X = identity_block(X, 3, [512, 512, 2048], stage = 7, block = "c")

# AVGPOOL.
X = AveragePooling2D((2,2), name = "avg_pool")(X)

# output layer
X = Flatten()(X)
X = Dense(classes, activation = "linear", name = "fc" + str(classes), kernel_initializer
= glorot_uniform(seed=0))(X)

# Creat model
model = Model(inputs = X_input, outputs = X, name="ResNet68")

return model

model = ResNet68(input_shape = (img_height, img_width, 1), classes=1)
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])

history = model.fit(train_dataset, epochs=60, steps_per_epoch = train_size)

```

