



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil en Informática

TIMBERAI: ALGORITMO DE CONTROL DE PATIO AUTÓNOMO PARA EMPRESAS FORESTALES A TRAVÉS DEL USO DE DEEP LEARNING.

Proyecto para optar al título de
Ingeniero Civil en Informática

PROFESOR PATROCINANTE:
JOSÉ LUIS DANIEL CARRASCO VENEGAS
INGENIERO CIVIL EN INFORMÁTICA

PROFESOR CO-PATROCINANTE
JORGE PATRICIO MATURANA ORTÍZ
INGENIERO CIVIL INFORMÁTICO
DOCTOR EN INFORMÁTICA

PROFESOR INFORMANTE
MATTHIEU VERNIER
DOCTOR EN INFORMÁTICA

FELIPE IGNACIO LAGOS MARTÍNEZ

VALDIVIA – CHILE
2023

DEDICATORIA

A la carrera llegué sin ganas.

Sólo seguía yendo a clases y esforzándome porque no había un “Plan B”.

El 3er año fue el que más me costó. Estaba peligrando en dos ramos. Si me los echaba esa vez, me “echaban de la U”.

Fue ahí donde se abrieron mis ojos.

Mi viejo sufrió desde chico problemas renales, y justo cuando entré a la carrera su condición llegó al límite. La coincidencia fue tragicómica. Pero aún así se esforzó para que no faltara la plata, al igual que mi mamá, para que pudiera terminar.

Los amigos que hice durante ese tiempo fueron los que me mantuvieron motivado a ir a clases, aunque sea a estar con ellos. Me motivaron a mantener el hábito.

Ese año aprobé todo por primera vez. Nunca más volví a bajar la cabeza desde ese punto e incluso llegué a agarrarle cariño a la carrera, donde la IA me atrapó definitivamente.

De principio a fin, mis viejos y mis amigos siempre estuvieron presente; hasta el final, aun cuando yo pude haber dado más de mí.

Sería faltarles el respeto a todos ellos, sobre todo a mi papá, si no les dedico este logro; les debo mucho. Este título es la primera cuota de mi deuda, porque comparado a todo lo que hicieron para yo estar aquí,

este título se queda corto.

ÍNDICE

ÍNDICE	I
ÍNDICE DE TABLAS	III
ÍNDICE DE FIGURAS	IV
RESUMEN	VI
ABSTRACT	VII
1. INTRODUCCIÓN	1
1.1 Definición de la Problemática.....	2
1.2 Definición de Objetivos	2
1.2.1 Objetivo general	2
1.2.2 Objetivos específicos	2
2. MARCO TEÓRICO	4
2.1 Estado del arte en empresas forestales	4
2.1.1 Tecnologías usadas en el presente para post-faena	4
2.1.3 Oportunidad de la solución propuesta	10
2.2 Estado del arte Inteligencia Artificial para Procesamiento de Imágenes	11
2.2.1 Breve historia de las Redes Neuronales	11
2.2.2 Segmentación Semántica	14
2.2.3 Redes neuronales Convolucionales (CNN)	16
2.2.4 Convolución	17
2.2.5 Arquitecturas destacables de CNN	19
3. DESARROLLO DE LA SOLUCIÓN	22
3.1 Equipo de proyecto	22
3.1.1 Mónica Fischer D.	22
3.1.2 Edinson Contreras	23
3.1.3 Ányelo Quisto.....	23
3.1.4 Juan Letelier	23
3.1.5 Rodrigo Miranda	23
3.1.6 Cristian Estrada	23
3.1.7 Staff	23
3.2 Requerimientos del Proyecto	24
3.2.1 Reemplazar al encargado de planta	24

3.2.2 Utilizar las cámaras de seguridad	24
3.2.3 Acelerar el proceso de cubicación de la carga de un camión	24
3.3 BPMN	24
3.4 Casos de Uso / Diagramas de Secuencia	27
3.4.1 Calcular el volumen de carga de un camión.....	27
3.5 GUI	28
3.5.1 Interfaz de llegada de camión.....	28
3.5.2 Interfaz de Reporte de medición en llegadas de camiones.....	31
3.6 Resumen del desarrollo.....	32
4. EVALUACIÓN EN ENTORNO SINTÉTICO.....	38
4.1 Metodología de evaluación	38
4.1.1 Optimizadores.....	38
4.1.2 Funciones de pérdida	42
4.1.3 Métricas para la evaluación del modelo	43
4.2 Resultados	44
5. EVALUACIÓN EN ENTORNO REAL.....	46
5.1 Metodología de evaluación	46
5.2 Resultados	46
6. CONCLUSIONES Y TRABAJOS FUTUROS	47
7. REFERENCIAS	49
8. ANEXOS.....	52
Anexo A	52
Anexo B	53

ÍNDICE DE TABLAS

Tabla	Página
Tabla 1: Resultados de la segmentación vs. cantidad real	5
Tabla 2: Comparación de volumen estimado por las cámaras vs. medición manual (caja rectangular).....	8
Tabla 3: Comparación de ventajas y desventajas de las soluciones de segmentación manual vs. soluciones láser.	10
Tabla 4: Resultados de las etimaciones en el entorno real.	46

ÍNDICE DE FIGURAS

Figura	Página
Figura 1: El encargado de la empresa cliente, midiendo la altura de la carga.	2
Figura 2: Foto de un camión con troncos, desde atrás.	4
Figura 3: Segmentación de secciones transversales de los troncos de la Figura 2, con su círculo asociado.	5
Figura 4: Esquema de configuración de cámaras para la reconstrucción en 3D de la carga, representada por el rectángulo al centro (vista desde el frente).	6
Figura 5: Nube de puntos que representan un camión en movimiento.	7
Figura 6: figuras de prueba utilizadas por Rundgren (2017)	8
Figura 7: Esquema que describe la instalación de la solución láser.	9
Figura 8: Lectura láser del camión.	9
Figura 9: Ejemplo de análisis láser aplicado a carga de troncos.	10
Figura 10: Estructura básica de una neurona.	12
Figura 11: Estructura de un Perceptrón.	13
Figura 12: Arquitectura de Mark I Perceptron.	13
Figura 13: Arquitectura de una red neuronal simple, que presenta capa oculta.	14
Figura 14: Ejemplo de Segmentación Semántica (persona, animales).	15
Figura 15: Arriba, imagen de una calle. Abajo, las múltiples etiquetas definidas para este caso.	16
Figura 16: Arquitectura genérica que conforma una red neuronal convolucional.	17
Figura 17: Arquitectura de una red neuronal convolucional, dedicada a la Segmentación de Imágenes.	17
Figura 18: Representación gráfica de la convolución, con pasos.	18
Figura 19: Las distintas etapas de convolución de una imagen de entrada. La parte azul oscura en la matriz g, y la zona verde el resultado de la convolución completa.	19
Figura 20: Esquema que describe la arquitectura de U-net.	20
Figura 21: Estructura y resultados de Mask R-CNN.	20
Figura 22: Organigrama del equipo de trabajo.	22
Figura 23: Diagrama BPMN proceso de cálculo manual.	25
Figura 24: Diagrama BPMN estimación autónoma de altura y posterior cubicación.	26
Figura 25: Caso de uso para estimar el volumen de la carga en un camión de forma autónoma.	27
Figura 26: Diagrama de secuencia para "Calcular el volumen de carga de un camión".	28
Figura 27: Interfaz completa para documentar la carga de un camión recién llegado.	28
Figura 28: Ingreso de datos que identifican al camión entrante.	29
Figura 29: Botón que llama al servicio de IA para predecir el volumen de madera.	29
Figura 30: Ventana de Espera a que la Red neuronal estime altura de las imágenes proporcionadas.	30
Figura 31: Resultados cubicados en la última etapa del reporte del camión recién llegado.	31

Figura 32: Reporte de Medición autónoma y manual de la carga de los camiones que entraron a la planta.	32
Figura 33: Columna de Diferencia porcentual autónomo versus manual.	32
Figura 34: Posición aproximada del camión y cámaras frente a la garita del encargado de planta.	33
Figura 35: Mejor resultado obtenido haciendo segmentación manual.....	34
Figura 36: Resultado en bruto de la estimación de la Red Neuronal del proyecto.	34
Figura 37: Una de las nuevas imágenes utilizadas para el entrenamiento de U-Net	35
Figura 38: De arriba hacia abajo. La etiqueta de la carga y la etiqueta de las estacas de una imagen de un camión.....	36
Figura 39: Algoritmo para calcular la altura de la carga con las dimensiones estimadas por la Red Neuronal.....	36
Figura 40: Algoritmo para obtener y filtrar alturas de estacas o de carga.	37
Figura 41: Ejemplos de funciones diferenciables.	39
Figura 42: Ejemplos de funciones no diferenciables.	39
Figura 43: Ejemplo de función convexa y no convexa.	39
Figura 44: Ejemplo de tasas de aprendizajes correctas e imprecisas.	40
Figura 45: Comparativa de convergencia entre diferentes optimizadores en un ejemplo de entrenamiento.	42
Figura 46: Resultados de la función Cross Entropy en los conjuntos de entrenamiento y validación.	44
Figura 47: Desempeño del modelo en el conjunto de validación con IoU.	45
Figura 48: La estimación de la Red Neuronal sobre la imagen de un camión.....	45
Figura 49: Distribución de porcentajes de diferencias entre estimación autónoma y manual.	47
Figura 50: Ejemplo de un falso positivo de la Red Neuronal considerando carga una ruma de troncos a la distancia.	48
Figura 51: Espacio de Color HSV.....	52
Figura 52: Cancha Arica, situada en Valdivia (-39.837347406912265, -73.25789258212504).....	53

RESUMEN

Una empresa forestal, cliente de West Ingeniería Ltda. desea mejorar su metodología de control del volumen de su madera que circula a lo largo del país; las empresas forestales deben mantener un nivel de control sobre sus faenas porque la ley 20283 determina una cantidad máxima de madera, tanto nativa como no nativa para faenar en cada una de las varias zonas que pueda tener una empresa forestal y, por lo mismo, estas empresas buscan sacarle el mayor provecho posible a materia prima.

Esta empresa cliente controla su producción calculando los metros cúbicos de la carga de cada camión que mueve su madera. Este cálculo se hace obteniendo las dimensiones de dicha carga; el largo y ancho son las dimensiones del camión, que son constantemente 6 y 2.4 metros respectivamente, lo que deja como única variable a la altura de la carga. Esta altura se calcula de manera manual; una persona se acerca al camión y con una regla mide la altura.

La empresa reconoce que esta metodología no solo es imprecisa, sino que es peligrosa para la persona encargada, por lo que acepta que esta tarea sea realizada por un algoritmo de Inteligencia Artificial. Este algoritmo utiliza las cámaras de seguridad que se utilizan en las instalaciones de la empresa cliente para alimentar una imagen a una Red Neuronal basada en U-net, una Red Neuronal Convolutiva. Esta Red retornará las dimensiones de la carga, de la cual se extrae solamente la altura, y se calcula los metros cúbicos con dicho resultado.

La Red Neuronal del algoritmo fue entrenada con una colección de alrededor de 600 imágenes etiquetadas, y los resultados son muy prometedores estimando satisfactoriamente las dimensiones de la carga de madera. Como la obtención de la altura depende fuertemente de la estimación de la Red Neuronal, los resultados de la altura (y posterior cubicación) de la carga de madera, comparado con los cálculos hechos por el encargado en planta, demuestran un error de cálculo de alrededor del 2%, además de ser estimaciones más estables.

ABSTRACT

A forestry company, a client of West Ingeniería Ltda., wishes to improve its methodology for controlling the volume of its wood that circulates throughout the country. Forestry companies must maintain a level of control over their operations because Law 20283 determines a maximum amount of native and non-native wood to be harvested in each of the various zones that a forestry company may have. As a result, these companies seek to maximize the use of their raw materials.

This client company controls its production by calculating the cubic meters of each truckload that moves its wood. This calculation is done by obtaining the dimensions of the load; the length and width are the dimensions of the truck, which are constantly 6 and 2.4 meters respectively, leaving only the height of the load as the variable. This height is manually calculated; a person approaches the truck and measures the height with a ruler.

The company acknowledges that this methodology is not only imprecise but also dangerous for the person in charge. Therefore, it agrees that this task be performed by an Artificial Intelligence algorithm. This algorithm uses the security cameras used in the client company's facilities to feed an image to a U-net based Convolutional Neural Network. This Network returns the dimensions of the load, from which only the height is extracted, and the cubic meters are calculated with this result.

The Neural Network of the algorithm was trained with a collection of around 600 labeled images, and the results are very promising in satisfactorily estimating the dimensions of the wood load. Since the height estimation heavily depends on the Neural Network's estimation, the results of the height (and subsequent cubic calculation) of the wood load, compared to the calculations made by the supervisor on site, demonstrate a calculation error of around 2%, as well as more stable estimations.

1. INTRODUCCIÓN

La madera como bien de consumo es uno de los productos más utilizados y, por lo tanto, más demandado del mundo. Los usos que tiene la madera son muy diversos; pueden ir desde el uso directo de esta madera (para la construcción de casas o muebles) hasta la creación de los derivados de esta, como por ejemplo el papel.

Por esta razón, han surgido una innumerable cantidad de empresas dedicada a la faena de árboles para la extracción de su madera. La organización de estas empresas es diversa, donde algunas talan los árboles y venden la materia prima directamente a sus clientes, otras empresas trasladan su madera a canchas de acopio, donde se procesa la materia prima en astillas, las cuales tienen una alta gama de usos, como pisos flotantes o muebles modernos.

Estas empresas existen en distintos tamaños; sin embargo, independiente de su tamaño, cada empresa faena una cantidad importante de árboles, en diversas zonas de explotación por empresa forestal a lo largo de todo un país; lo anterior, naturalmente, genera un impacto significativo en el ambiente, lo que ha llevado a las naciones que albergan estas empresas a tomar medidas para mantener una explotación controlada de este bien de consumo. En Chile no es la excepción; en el 11 de julio de 2008, se promulgo la Ley sobre Recuperación del Bosque Nativo y Fomento Forestal, o Ley 20283, la cual define, entre otras cosas, los límites de faena forestal renovable que varían dependiendo, principalmente, de la zona del país donde se pretende faenar, y del tamaño de la empresa forestal que solicite zonas de explotación al Ministerio de Agricultura. Las empresas tienen permitido faenar uno o más predios de un máximo de 200 hectáreas de madera renovable, o de 500 hectáreas cuando dichos predios se ubiquen en las regiones I y IV, incluida la XV; o de 800 hectáreas para predios ubicados en la comuna de Lonquimay, en la IX región, o en la provincia de Palena, en la X región. Independiente del sector, la tala de árboles nativos no puede exceder las 40 hectáreas anuales. (Biblioteca del Congreso Nacional de Chile, 2015).

A lo anterior, se le suma el hecho de que la madera, una vez faenada, comienza a secarse, lo que afecta paulatinamente el volumen de los troncos explotados. Esto afecta directamente a las empresas forestales, debido a que prometen movilizar una cantidad puntual de materia prima, y cualquier diferencia entre la promesa y la entrega final (ya sea a una cancha de acopio, a un cliente de manera directa) obliga a esta empresa a reponer la cantidad faltante de material, o asumir los ingresos reducidos de la venta en su condición de llegada.

Una empresa forestal cliente de West Ingeniería Ltda., de ahora en adelante empresa cliente, considera necesario mantener un control sobre el volumen de materia prima que circula entre las canchas de acopio a lo largo del país. En este momento, la empresa cliente mantiene un control de su madera midiendo la altura de la carga que llegan a las canchas de acopio, la cual se determina de manera manual (Figura 1).



Figura 1: El encargado de la empresa cliente, midiendo la altura de la carga.

1.1 Definición de la Problemática

Esta estrategia de medición manual saca a la luz dos dolores para la empresa cliente. El primer dolor es el hecho de que, en cada cancha de acopio, circulan numerosas unidades de maquinaria pesada, a lo largo y ancho de toda la cancha; es por esto que, para el personal y personas externas, se exige una capacitación extensa para garantizar la seguridad de las personas en las instalaciones, en donde se expresa de manera explícita no circular cerca de los camiones ni otra maquinaria. Por lo tanto, la metodología adoptada en la Figura 1 es una clara violación a la estrategia de seguridad estándar para las canchas de acopio, arriesgando la vida del encargado de la empresa cliente.

El segundo dolor es el error humano inducido en la medición de la altura de la carga misma. No se puede garantizar que la medición se haya hecho en las mejores condiciones, debido a posibles errores de estimación que el encargado de planta no tenga presente a la hora de ejecutar la tarea.

Es necesario subsanar estos dolores, tanto para que la empresa no arriesgue la vida del encargado de cancha, como para entregar una métrica más apegada a la realidad que sea eficiente y confiable.

1.2 Definición de Objetivos

1.2.1 Objetivo general

Apoyar la estimación de altura de la carga de manera precisa mediante la predicción de las cargas de camiones para lograr una precisión muy buena del volumen de madera explotada, a través del uso de un modelo de Deep Learning.

1.2.2 Objetivos específicos

- Estudiar acerca del estado del arte de las empresas forestales en Chile.

- Investigar acerca del estado del Arte de la Visión Computacional (desde algoritmos más sencillos hasta la Inteligencia Artificial)
- Crear un modelo de Inteligencia Artificial capaz de aplicar Segmentación Semántica en las imágenes de las cámaras, con la intención de detectar la carga y atriles del camión proyectado en la imagen.
- Implementar el algoritmo que utiliza el resultado de la Segmentación Semántica del modelo de IA para lograr el cálculo final de la altura de la carga del camión proyectado.
- Probar el modelo de IA, en una colección simplificada de imágenes para validar el concepto como viable.
- Verificar la calidad del algoritmo completo (IA + cubicación) para corroborar la precisión de la solución completa.

2. MARCO TEÓRICO

2.1 Estado del arte en empresas forestales

Las empresas forestales están al tanto de que la madera es un bien frágil, y por ello, una vez faenada, hay que estar al tanto del estado en que llega la madera a las canchas de acopio; el cómo se lleva a cabo ese control varía de empresa a empresa. A continuación, se explorará brevemente implementaciones que buscan mejorar el control de la madera que llega a cancha post-faena.

2.1.1 Tecnologías usadas en el presente para post-faena

La problemática expresada en el Capítulo 1 es sobre una metodología manual que busca abordar en control de madera después de que esta sea explotada desde el árbol. Sin embargo, existen diferentes aproximaciones para abordar la necesidad de una empresa forestal (u otro tipo de empresa que trabaje con camiones) de controlar el flujo de su cargamento en un predio.

Una de las metodologías para controlar cargas de troncos faenados, consiste en ver desde detrás del camión para hacer un conteo de los troncos individuales cargados, como lo plantean Rahman, Yella & Dougherty (2011). Su propuesta inicia con imágenes de camiones, como se ejemplifica con la Figura 2.



Figura 2: Foto de un camión con troncos, desde atrás.

A estas imágenes se le aplican un proceso de segmentación¹ manual. Esta segmentación utiliza un algoritmo hecho con Python, por ejemplo, haciendo uso de cambios de espacio de colores desde RGB a HSV, espacio que puede verse en la imagen del Anexo A. De esta manera, los colores del fondo de la imagen quedan en negro, luego de ir quitando rangos de colores, indicándolos por escrito. Así, la imagen es solamente de las secciones transversales de los troncos, destacándolas con círculos aproximados utilizando círculos como se muestra en la Figura 3.

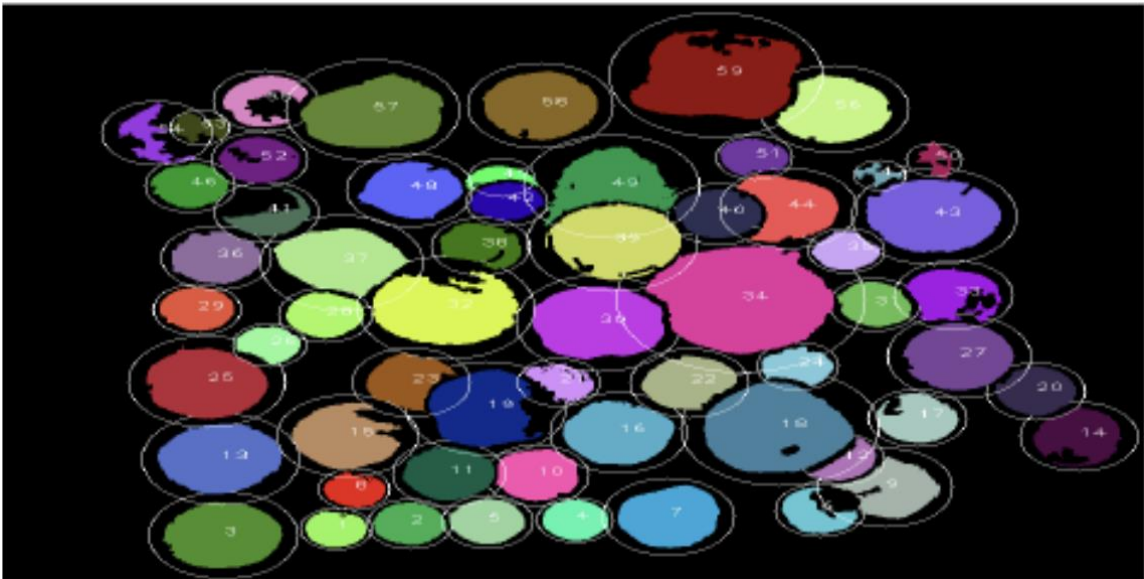


Figura 3: Segmentación de secciones transversales de los troncos de la Figura 2, con su círculo asociado.

Con estos círculos se puede, por ejemplo, hacer un conteo sobre la cantidad de troncos cargados en el camión, los cuales pueden compararse con un conteo real en terreno, como se muestra en la Tabla 1.

Tabla 1: Resultados de la segmentación vs. cantidad real

Imagen	Real	Segmentados	Tipo
1	58	59	Abeto
2	45	44	Abeto
3	78	73	Abeto
4	53	48	Abeto
5	65	61	Abeto
6	60	61	Abeto
7	53	54	Pino

¹ La segmentación subdivide una imagen en sus partes constituyentes u objetos, con el fin de separar las partes de interés del resto de la imagen.

Este control de cargas no tiene por qué ser necesariamente sobre imágenes 2D; también pueden usarse reconstrucciones 3D, como lo propone Rundgren (2017). Para esto se utilizan un conjunto de cámaras, instaladas como el esquema de la Figura 4.

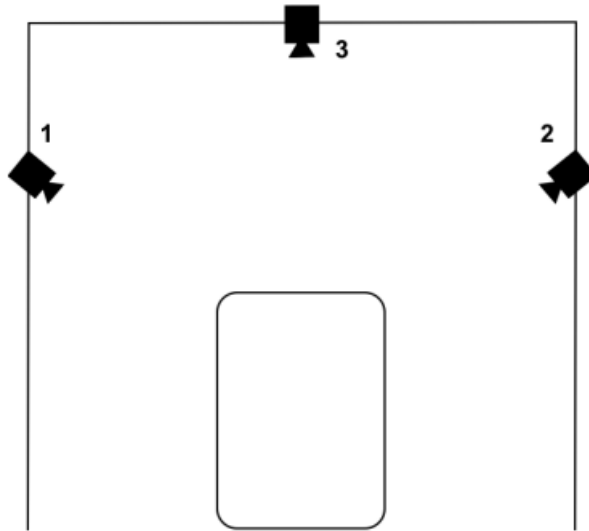


Figura 4: Esquema de configuración de cámaras para la reconstrucción en 3D de la carga, representada por el rectángulo al centro (vista desde el frente).

Con esta configuración de cámaras, se obtiene nubes de puntos que proyecten las dimensiones de la imagen, como ejemplifica la Figura 5.

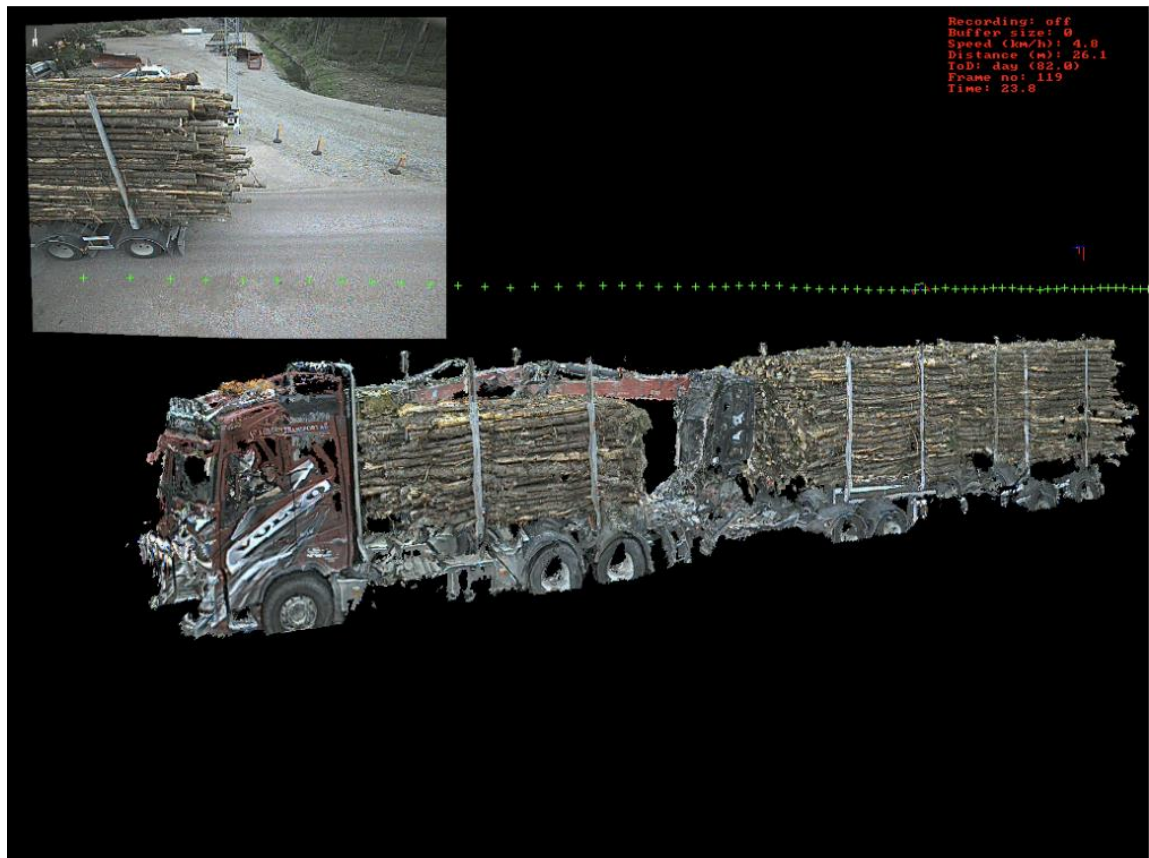
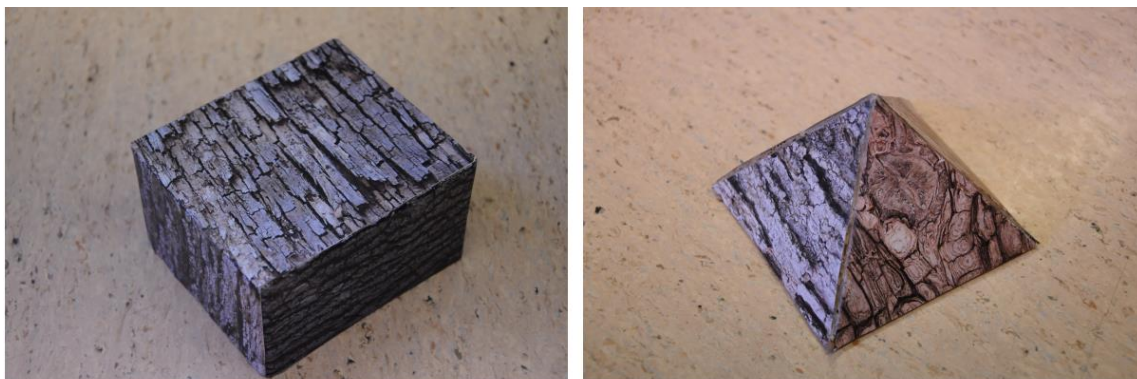


Figura 5: Nube de puntos que representan un camión en movimiento.

Con esta nube de puntos, una por cada cámara instalada, se procede a aplicar un algoritmo de Triangulación. De este modo, todas las nubes creadas por separado se unen en una sola gran nube, resultando en la figura 3D de una carga completa. Para probar la eficacia de su propuesta, Rundgren (2017) propuso calcular el volumen de dos figuras muy simples: la de una caja rectangular y una pirámide, mostradas en la Figura 6. En la Tabla 2, se entregan los datos de la posterior estimación de volumen de la figura rectangular, la cual es la más parecida a la carga de un camión, donde se comparan el volumen calculado de la nube completa de puntos, en comparación con el volumen real, calculado a mano, de dicha caja rectangular. En esta tabla, se puede destacar que el error del cálculo es relativamente constante respecto al volumen original. Sin embargo, el tiempo de ejecución del algoritmo aumenta enormemente a medida que se aumenta la cantidad de puntos que proyecta cada cámara de la configuración.



(a) Rectangular box.

(b) Pyramid.

Figura 6: figuras de prueba utilizadas por Rundgren (2017)

Tabla 2: Comparación de volumen estimado por las cámaras vs. medición manual (caja rectangular).

Resolución de puntos	Volumen real (cm ³)	Volumen estimado (cm ³)	Tiempo de ejecución (s)
80 x 64 x 32	581.175	578.667	81.623
160 x 128 x 64	581.175	578.592	605.587
320 x 256 x 128	581.175	578.596	5147.285

Hay otras tecnologías, sin embargo, que lograron automatizar el proceso de control de patio, con una mayor precisión. Tal caso es el de las soluciones que utilizan láser, como la que ofrece Macronline (2021), para dimensionar la carga de los camiones de manera precisa y en tiempo real. En la Figura 7, se muestra un esquema sencillo de la estructura necesaria para operar el láser, que cubre el camión completo, como se muestra en la Figura 8, y genera un espectro 3D como que se presenta en la Figura 9, utilizado para posteriores cálculos que requiera la empresa contratante, como por ejemplo el cálculo del volumen.

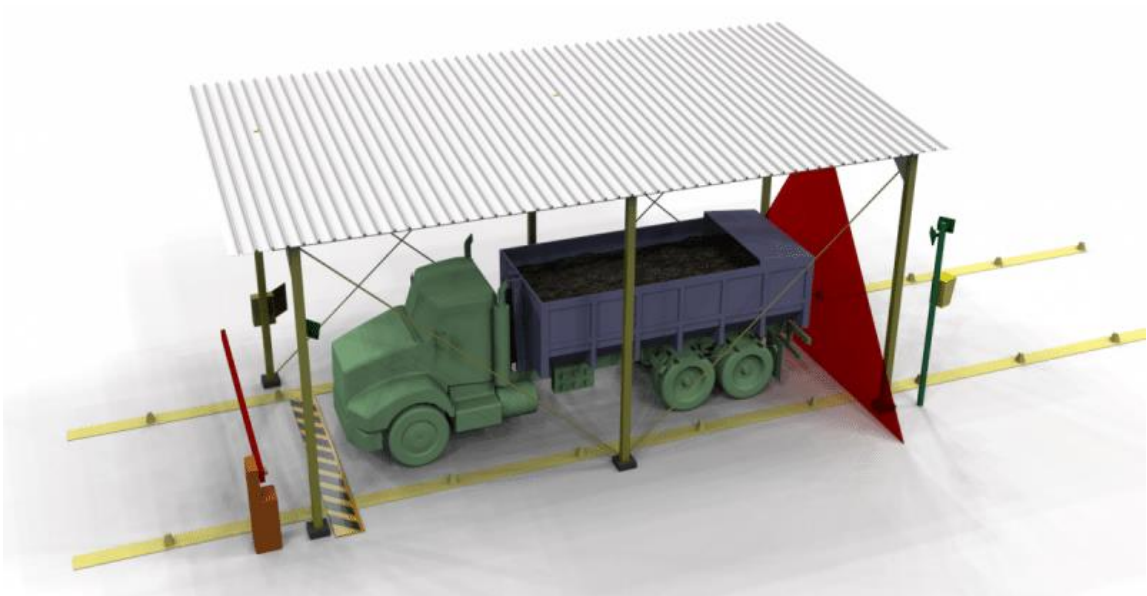


Figura 7: Esquema que describe la instalación de la solución láser.

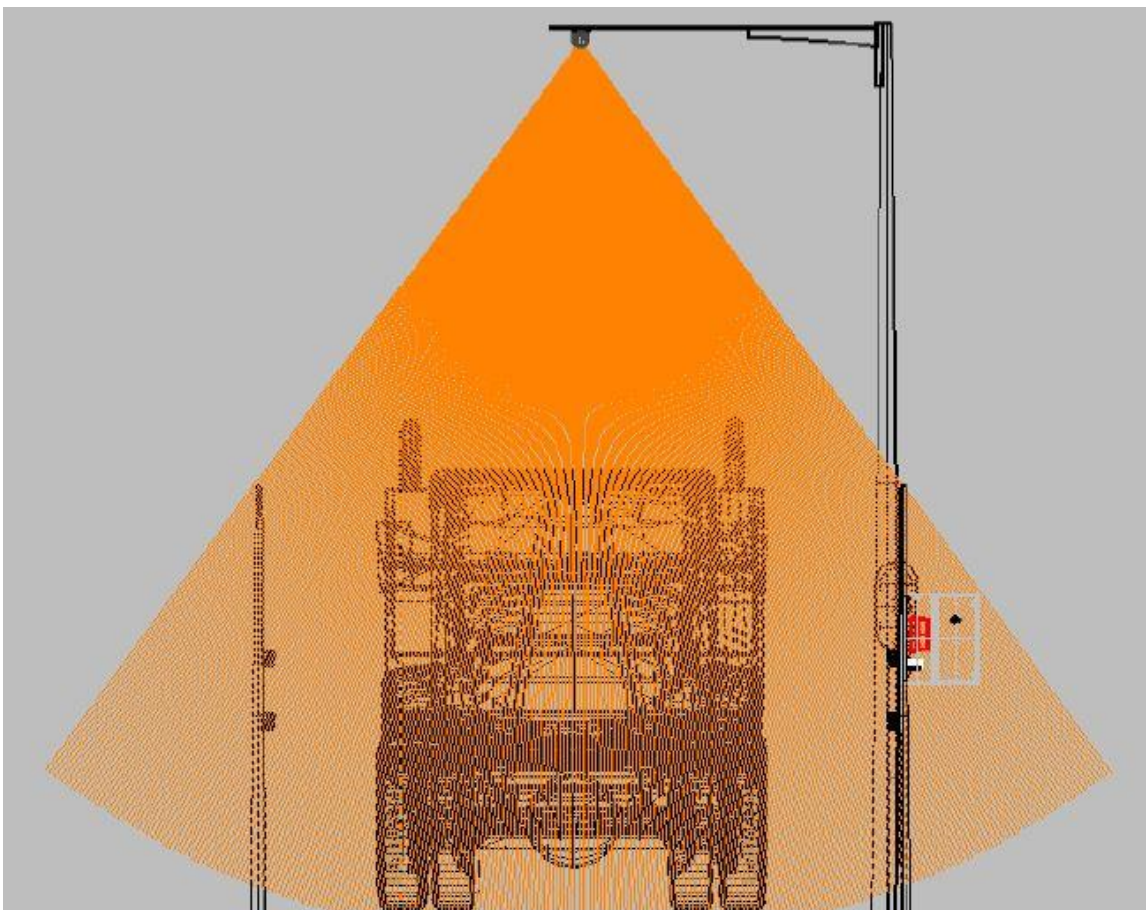


Figura 8: Lectura láser del camión.



Figura 9: Ejemplo de análisis láser aplicado a carga de troncos.

2.1.3 Oportunidad de la solución propuesta

La solución de este proyecto entra en directa competencia a las soluciones de segmentación manual, como la propuesta por Rahman et al. (2011) y las soluciones láser, como la propuesta y ofrecida por Macronline (2021) a empresas que buscan control de sus patios. Ambas soluciones mencionadas tienen sus ventajas y desventajas, las cuales se presentan en la Tabla 3. La segmentación manual tiene un techo de precisión que puede resultar poco atractivo para empresas que buscan tener la mejor estimación posible (en el Capítulo 3 se verá qué tan deficiente puede ser este método, en la primera aproximación que se tuvo con esta propuesta). Las soluciones láser, aunque presentan ventajas atractivas, requieren condiciones que pueden no hacer posible su implementación bajo ciertas circunstancias; en el caso de la empresa cliente, no sería posible implementar esta solución, porque la cancha de acopio no es de su propiedad, y esta presta servicio a más empresas forestales, por lo que implementar el láser, con toda la infraestructura que implica, significaría un cese de funciones de la cancha de acopio, lo que produciría pérdidas millonarias tanto para la empresa cliente como el resto de empresas forestales que utilizan diariamente el servicio de esta cancha, que mueve cientos de metros cúbicos de madera al día.

Tabla 3: Comparación de ventajas y desventajas de las soluciones de segmentación manual vs. soluciones láser.

	Ventajas	Desventajas
Segmentación Manual	<ul style="list-style-type: none"> • Facilidad de Implementación. • Facilidad de Uso. • Barato. 	<ul style="list-style-type: none"> • Baja precisión de dimensiones de cargas. • Poco resiliente a cambios en el entorno.
Láser	<ul style="list-style-type: none"> • Alta precisión de dimensiones de cargas. • No se ve afectado por cambios de entorno. 	<ul style="list-style-type: none"> • La estructura necesaria para la instalación es grande y poco flexible. • Alto costo (instalación, mantenimiento).

Habiendo expuesto las ventajas y desventajas de la segmentación manual y basada en láseres, la intención de esta propuesta es tomar las ventajas de ambas soluciones y combinarlas en una sola solución. TimberAI tiene como idea base que sea fácil y barato

de implementar, sea capaz de lograr una buena precisión y sea resiliente a múltiples ambientes, a la hora de estimar las dimensiones de la carga de los camiones de la empresa cliente. Esto se logrará aprovechando las cámaras de seguridad presentes en la cancha, con adaptaciones menores (por ejemplo, para cambiar de posición las cámaras) que no requerirían el cese de operaciones. De la precisión de la estimación de dimensiones estará encargado un algoritmo de Inteligencia Artificial, en concreto, de Deep Learning, que han saltado a la fama por lo eficaz que son en, entre otras cosas, en el procesamiento de imágenes, utilizando los principios que se verán a continuación.

2.2 Estado del arte Inteligencia Artificial para Procesamiento de Imágenes

La Inteligencia Artificial, hoy en día, tiene un amplio universo de aplicaciones, con resultados que se consideraban inesperados, y que ahora son un estándar en disciplinas como el pronóstico en la bolsa de valores (Florida International University, s.f). A continuación, se abordará en área de Procesamiento de imágenes, que en Deep Learning tiene un tipo de arquitectura ya estandarizada, la cual será el enfoque principal para la implementación de esta propuesta, utilizando el método de procesamiento de imágenes de Segmentación Semántica.

2.2.1 Breve historia de las Redes Neuronales

El concepto de neurona en la computación fue acuñado por McCulloch y Pitts (1943) donde se define a una neurona como nodos que se activan (o se prenden) si la suma de los datos de entrada, que son numéricos, supera cierto límite definido entre 0 y 1. En la Figura 10, se muestra el esquema sencillo propuesto por McCulloch et al. (1943), dando los fundamentos base para el nacimiento de las redes neuronales.

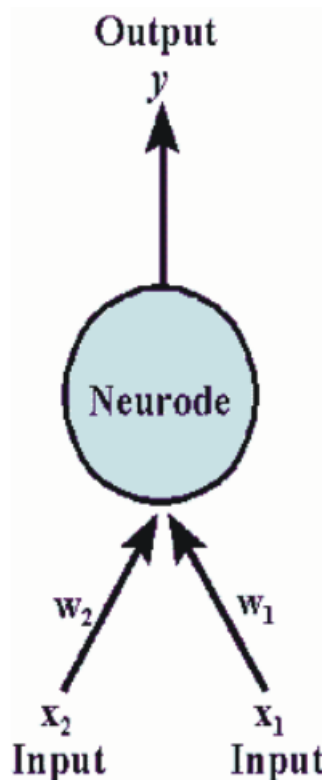


Figura 10: Estructura básica de una neurona.

Rosenblatt (1958) propuso el Perceptrón, presentado en la Figura 11. Con esta definición, en 1958 propuso el Mark I Perceptron, una de las primeras redes neuronales. Su arquitectura, se muestra en la Figura 12. Esta arquitectura poseía conexiones a una pantalla de 20×20 píxeles, con neuronas entrada, las cuales se unían de forma aleatoria a la pantalla. Estas neuronas después eran conectadas a todas y cada una de las unidades de respuesta, al final del sistema; estas unidades representaban neuronas de salida (Telefónica Tech, 2018).

Así como hizo Rosenblatt, las arquitecturas desde el año 1958 se volvieron más y más complejas. La falta de potencia del hardware de la época, eventualmente, hizo imposible la viabilidad de estas arquitecturas, y se descubrió tempranamente que debían de ser de un tamaño considerable para que obtengan buenos resultados. Estos motivos provocaron una serie de ceses en la implementación de redes neuronales, lo que se les llamó Invierno de la Inteligencia Artificial (BBC news, 2020). Estos inviernos fueron esporádicos, donde se destacan los de los años 1974-1980 y 1987-1993 (Actuaries Digital, 2018), que fueron especialmente largos.

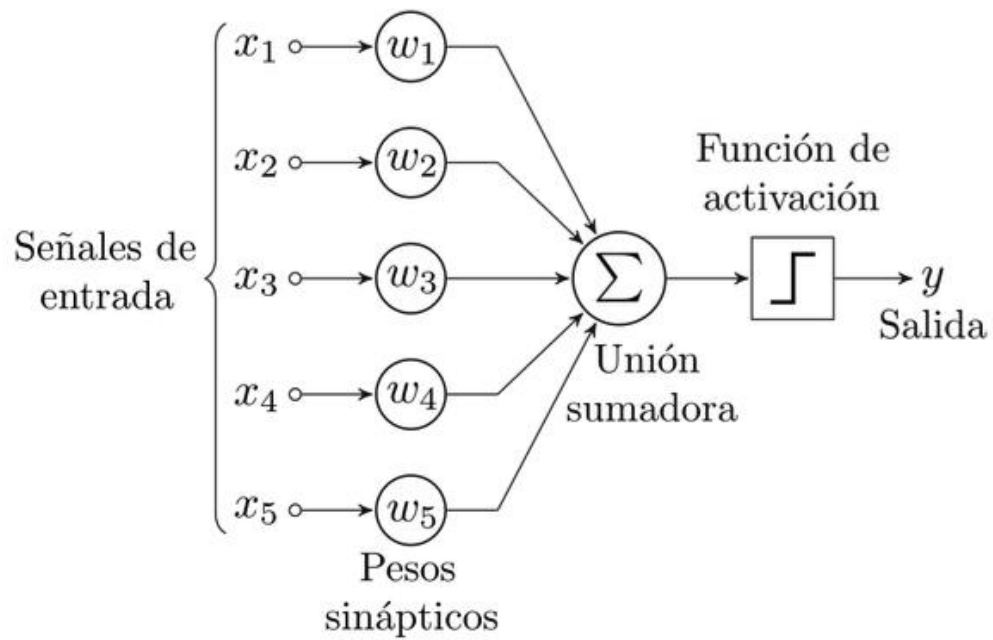


Figura 11: Estructura de un Perceptrón

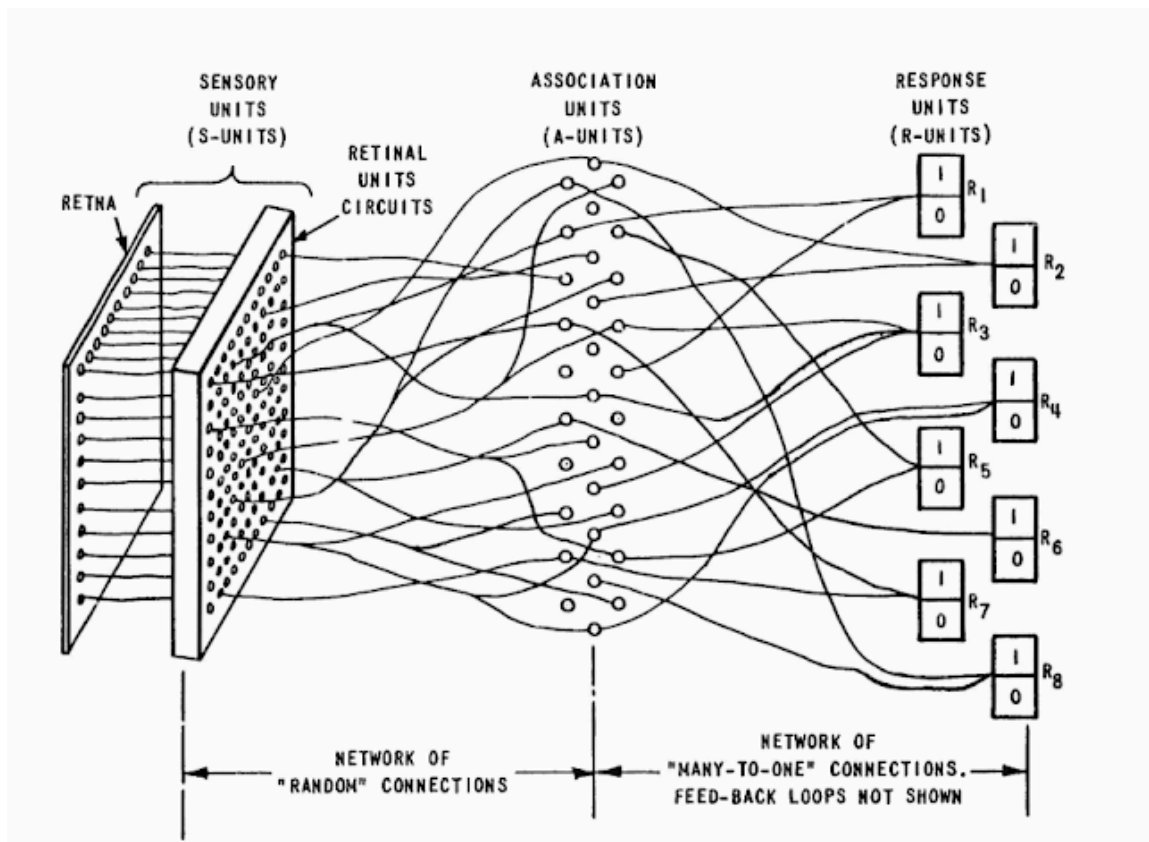


Figura 12: Arquitectura de Mark I Perceptron.

Con el pasar de los años, el hardware se puso al día. Su potencia hizo posible la continuación de las redes neuronales, y alrededor del año 2012, las GPUs² comenzaron a ser lo suficientemente potentes para poder trabajar con redes neuronales de gran cantidad de neuronas. Fue en este mismo año donde se descubrió que el aumento de neuronas mejoraba el desempeño de estas redes de manera importante. A estas neuronas se les organizó en capas, que se denominaron Capas Ocultas, debido a que pueden cambiar su configuración a elección del desarrollador, siempre y cuando las capas de entrada y salida se mantengan igual. En la Figura 13 se ejemplifica una arquitectura de la red neuronal simple.

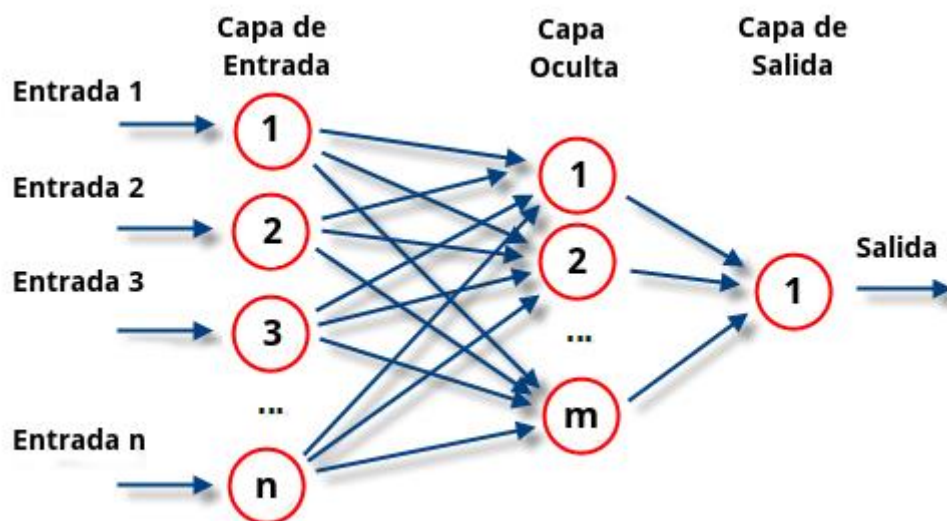


Figura 13: Arquitectura de una red neuronal simple, que presenta capa oculta.

2.2.2 Segmentación Semántica

La segmentación semántica es una tarea de visión computacional que utiliza etiquetas para distinguir regiones específicas de una imagen de entrada (Pytorch, 2022). A modo de comparación, la segmentación manual se hace eliminando los colores de la imagen de entrada, como lo ejemplifican Rahman et al. (2011). La segmentación semántica trabaja etiquetando píxel a píxel, independiente del color de este. Un algoritmo de Deep Learning bien preparado, va a ser capaz de etiquetar un grupo de píxeles que describan la figura de uno de los patrones para los cuales el algoritmo fue entrenado. En la Figura 14, se muestra una imagen de entrada (izquierda) y una imagen segmentada como resultado de un algoritmo de Deep Learning (derecha). La cantidad de patrones a etiquetar va a depender completamente del problema que se quiere solucionar; en la Figura 15, se puede ver que la cantidad de patrones es mayor que en el caso de la figura anterior, donde solo se etiquetan personas y animales, sin distinguir entre estos últimos; en esta figura, se diferencian entre, autos, veredas, calle, postes, edificios, entre otros.

² Graphic Processing Unit.

Los algoritmos de Deep Learning que son capaces etiquetar, píxel a píxel, utilizan arquitecturas que llevan el nombre de Redes Neuronales Convolucionales (CNN por sus siglas en inglés).



Figura 14: Ejemplo de Segmentación Semántica (persona, animales).



Figura 15: Arriba, imagen de una calle. Abajo, las múltiples etiquetas definidas para este caso.

2.2.3 Redes neuronales Convolucionales (CNN)

Desde sus inicios hasta ahora, las redes neuronales han encontrado nichos en diferentes áreas. Una de las áreas más célebres es en el procesamiento de imágenes; el procesamiento de imágenes es particularmente difícil de implementar porque, aunque para el ojo humano sean visiblemente iguales, tienen una sutil variación de valores.

Desde el año 1980, comenzaron a aparecer las redes neuronales convolucionales (CNN, de ahora en adelante), donde se establecen arquitecturas que se asemejan, hasta el día de hoy a la propuesta en la Figura 16. Las primeras capas de la red neuronal no tienen neuronas como tal, sino que se reemplazan por una sección pequeña de la imagen de entrada, de las cuales se obtienen características distintivas. Dependiendo de cómo se desea trabajar con la imagen, las arquitecturas pueden tener neuronas al final, como en esta figura, o se hace el proceso inverso, utilizando las características distintivas obtenidas en un comienzo para construir una nueva imagen, como la mostrada en la Figura 17. Este último tipo de arquitectura es capaz de generar Segmentación Semántica. Para obtener estas características, por cada sección pequeña de la imagen, se utiliza una función denominada convolución.

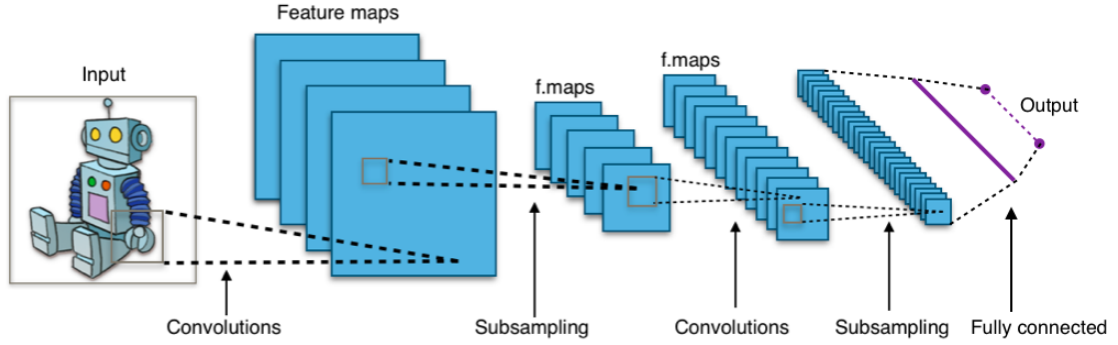


Figura 16: Arquitectura genérica que conforma una red neuronal convolucional.

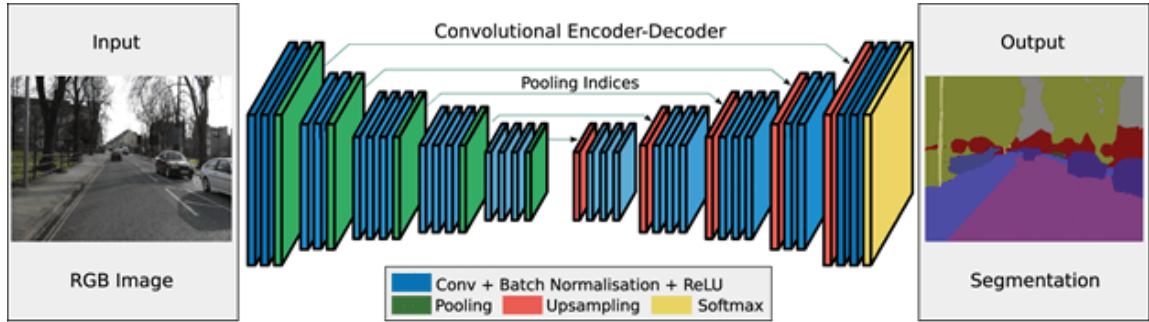


Figura 17: Arquitectura de una red neuronal convolucional, dedicada a la Segmentación de Imágenes.

2.2.4 Convolución

La convolución, matemáticamente, se define como la fórmula (1):

$$(f * g)(t) = \int_{-\infty}^{\infty} f(t)g(t - \tau) d\tau \quad (1)$$

Sean f y g funciones cualesquiera, en un instante de tiempo t , y un desfase τ . La convolución se entiende como la multiplicación de f , a lo largo de todo su dominio, con g , donde esta última se encuentra en constante desfase. El resultado de esta operación es la función f modificada en su dominio, como se muestra en la Figura 18.

Esta intuición se puede traducir a la perfección a las CNNs. Una imagen de entrada puede considerarse como una matriz de dimensiones $m \times n$, con m y n el largo y ancho de la imagen que representaría a f . La función g , en cambio, es representada por una matriz pequeña, generalmente de 3×3 , que aplica convolución a f como se ejemplifica en la Figura 19.

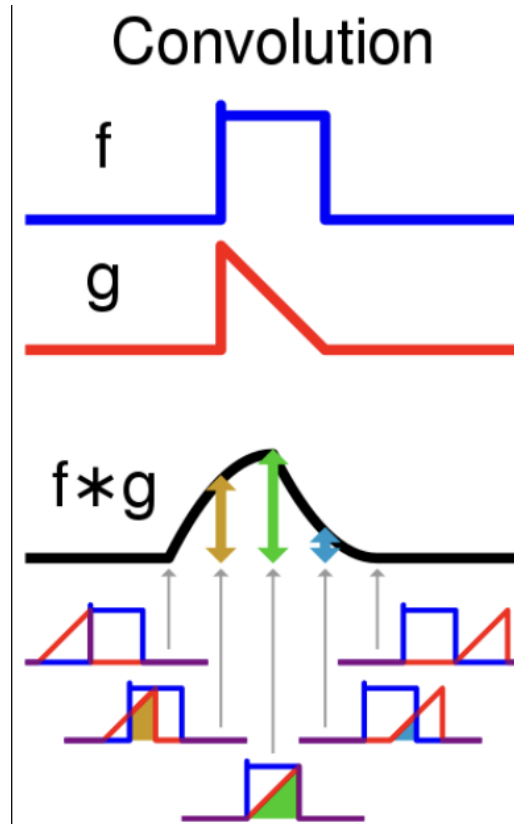


Figura 18: Representación gráfica de la convolución, con pasos.

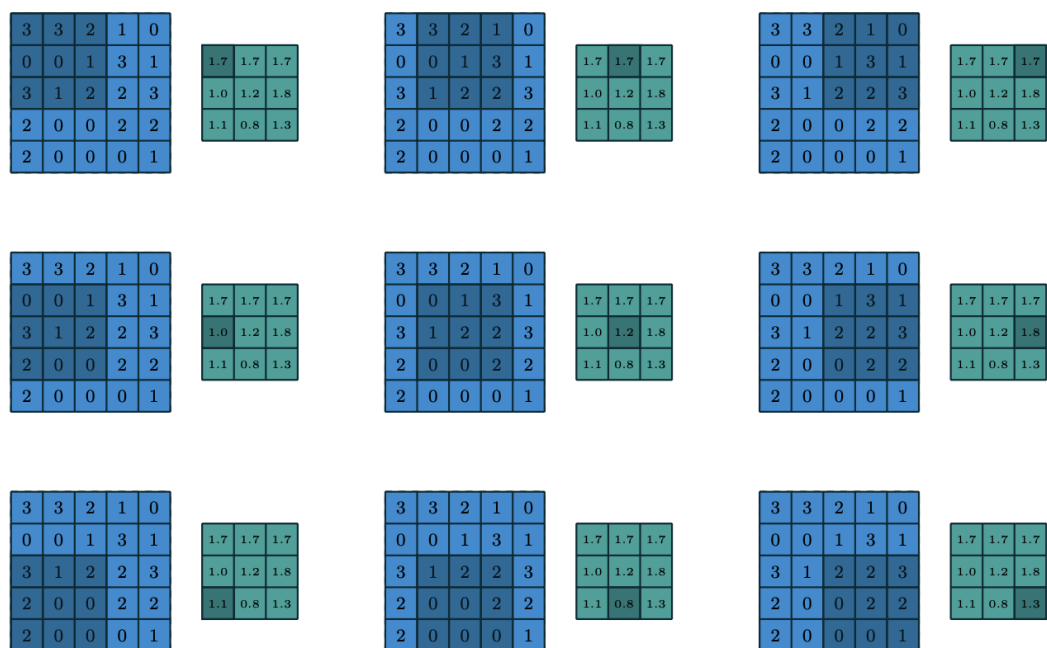


Figura 19: Las distintas etapas de convolución de una imagen de entrada. La parte azul oscura en la matriz g, y la zona verde el resultado de la convolución completa.

2.2.5 Arquitecturas destacables de CNN

Las arquitecturas de CNN, o de cualquier otro tipo de red neuronal, suelen ser de relativamente libre elección en cuanto a cómo debe estar estructurada la red desde dentro; la cantidad de capas ocultas puede variar según el problema a enfrentar.

Aun así, se destacan arquitecturas que han logrado una popularidad por sobre las demás, como es el caso de U-net (Uni-freigurb, s.f), inicialmente dedicada para la detección de tumores. Funciona utilizando imágenes en blanco y negro, representando una radiografía, y entrega otra imagen, blanco y negro, mostrando la posición exacta del potencial tumor alojado en el organismo de una persona. U-net ha logrado una gran popularidad, porque es una CNN dedicada a la Segmentación Semántica, que logra muy buenos resultados, en relativamente poco tiempo de entrenamiento (Neptune.ai, s.f). La estructura característica de U-net, la que le dio su nombre, se muestra en la Figura 20. Las flechas azules, representan una convolución aplicada a la imagen. Las flechas rojas, representan el rescate de las características de la convolución anterior. Las flechas verdes simbolizan una convolución inversa, que utiliza las características previamente rescatadas para generar una imagen. Las flechas grises son las que destacan a U-net; son las características de la convolución anterior que se concadenan a la imagen generada por la convolución inversa.

Otra arquitectura destacable es Mask R-CNN, la cual destaca por su capacidad de lograr Segmentación Semántica, separando los patrones repetidos como sus propias instancias.

Este tipo de segmentación se denomina Segmentación Instanciada. La estructura y resultados de Mask R-CNN se muestra en la Figura 21.

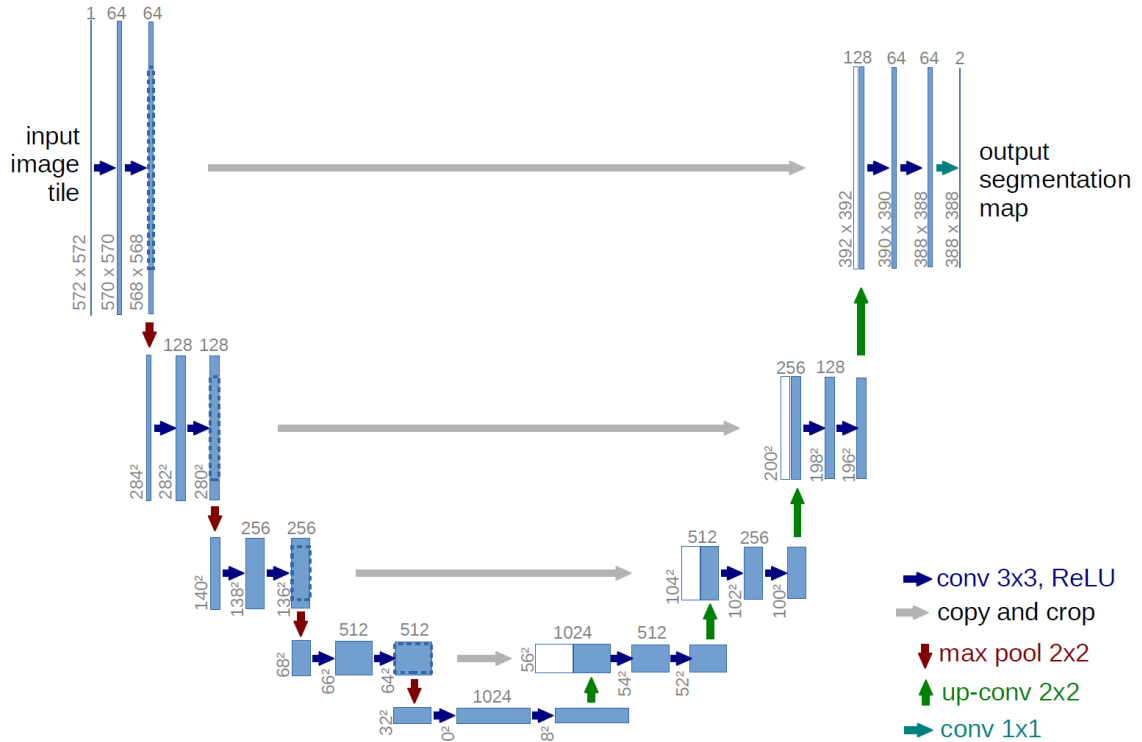


Figura 20: Esquema que describe la arquitectura de U-net.

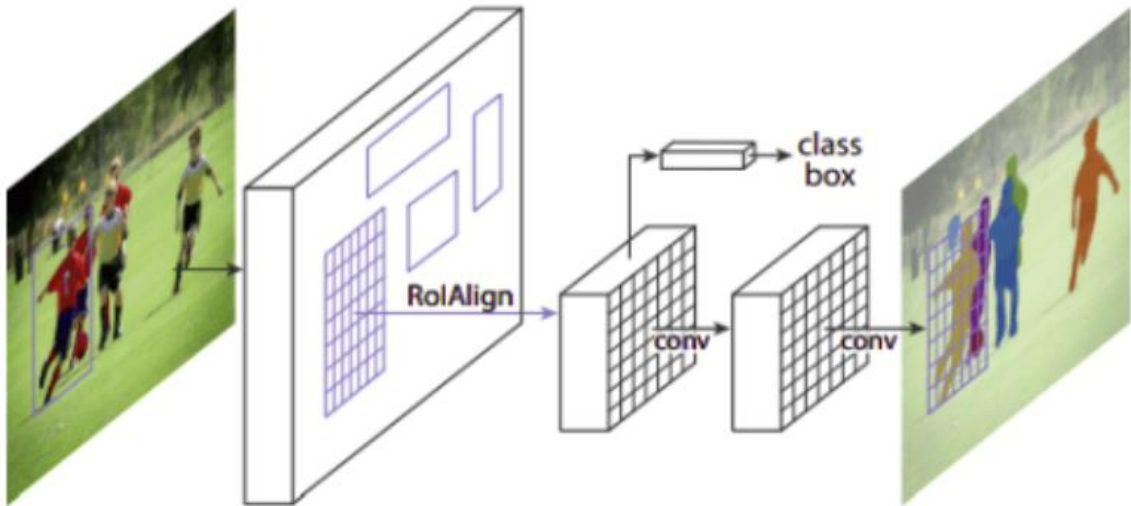


Figura 21: Estructura y resultados de Mask R-CNN

Así como las arquitecturas presentadas, hay muchas otras que han logrado fama, ya sea por su desempeño, como por sus complejidad y tamaño en constante aumento. De ellas solo pueden gozar las grandes corporaciones, como Google o OpenAI, para poder utilizar de manera óptima dichas arquitecturas.

En la propuesta de TimberAI, se decidió utilizar a U-net como base, haciendo cambios en la arquitectura para que esta sea capaz de entregar la cantidad de etiquetas necesarias para el correcto funcionamiento de la propuesta, además de las modificaciones pertinentes para poder utilizar imágenes de entrada a color. En el Capítulo 3, se darán los detalles exactos de la cantidad de etiquetas que fueron utilizadas, y el tipo de imágenes de entrada a utilizar.

3. DESARROLLO DE LA SOLUCIÓN

Este proyecto está a cargo de una empresa de desarrollo de software, la cual lleva 20 años trabajando para la comunidad. Por lo tanto, tiene una infraestructura establecida para llevar a cabo el desarrollo correcto de este y otros proyectos. A continuación, se hablará de las personas que conforman el equipo encargado del desarrollo de este proyecto en particular, los requisitos del software, diagramas que describen los casos de uso y su flujo de funcionamiento, interfaz de usuario y un resumen de cómo se llevó a cabo el desarrollo del proyecto.

3.1 Equipo de proyecto

Antes que el desarrollador se integrara a West Ingeniería Ltda., esta solución ya se encontraba en su fase de desarrollo y, por lo tanto, contaba con su equipo de trabajo. Las personas que conforman este equipo, ya con el desarrollador incluido, se representan en el organigrama de la Figura 22. A continuación, se hablará de cada persona en el organigrama, y su rol en este proyecto.

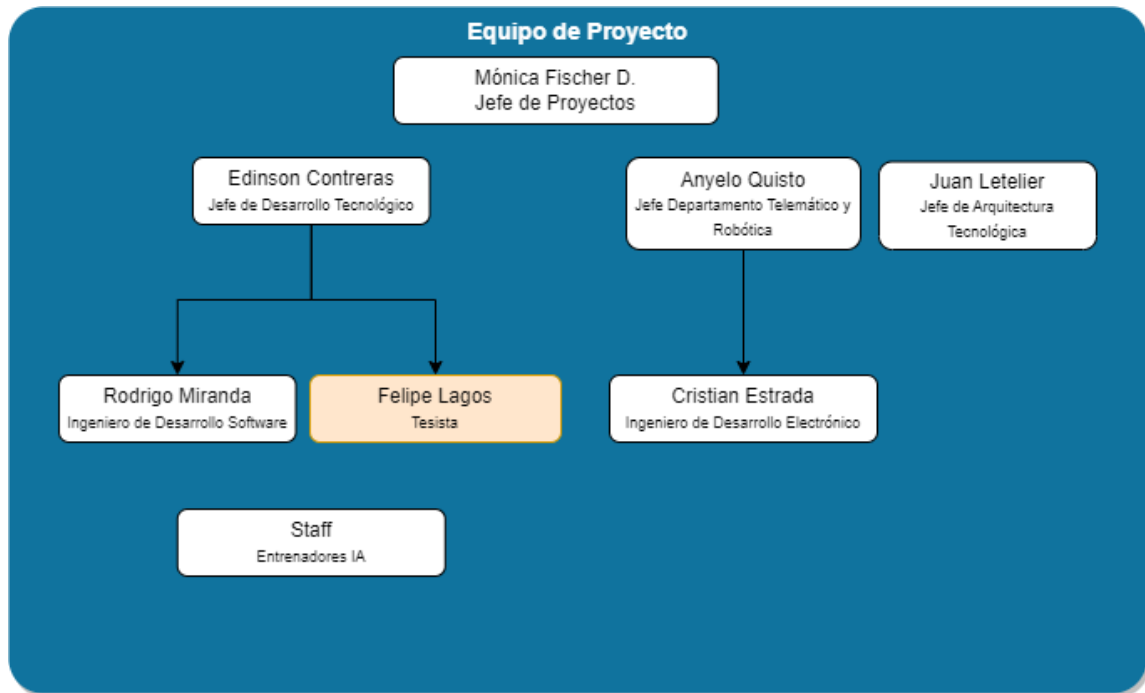


Figura 22: Organigrama del equipo de trabajo.

3.1.1 Mónica Fischer D.

Mónica es el contacto directo con la empresa cliente. Es la empresa encargada de reunirse de manera semanal con ellos, generalmente los jueves, para reportar el progreso del desarrollo e informarle de las decisiones que se han tomado en el camino (aquellas decisiones que resulten relevantes para el cliente). Además, recibe feedback del cliente, como por ejemplo estimar la altura de la carga de la cabina y del acoplado por separado, o la posibilidad de estimar cada altura con 2 cámaras, una opuesta de la otra para un total

de 4 cámaras. Es por esta comunicación directa que Mónica tiene la capacidad de pedir modificaciones al desarrollo, crear o detener aproximaciones para un desarrollo del proyecto que se adapte de mejor manera al cliente.

3.1.2 Edinson Contreras

Edinson es el jefe directo del desarrollador y de Rodrigo Miranda. Él conduce las reuniones scrum que se realizan de manera semanal para todo el Departamento de Desarrollo, el cual maneja otros proyectos, además del presente proyecto. En estas reuniones se reporta el progreso de este proyecto, y en función de eso organiza al personal para comprometer hitos durante la siguiente semana, hasta la siguiente reunión de Scrum.

3.1.3 Ányelo Quisto

Ányelo es la persona encargada de autorizar los cambios pertinentes a las cámaras utilizadas en este proyecto. Es la persona que autoriza, por ejemplo, mover las cámaras de posición. Se asegura que su funcionamiento sea ininterrumpido y cualquier problema que pueda ocurrir con estas, debe reportársele a él.

3.1.4 Juan Letelier

Juan es el encargado de levantar el servicio que provee este proyecto a los servidores que West Ingeniería Ltda. tiene en Amazon EC2 (Amazon EC2, 2022). Es con este levantamiento que el proyecto queda operativo y puede ser consultado por el cliente a través de la plataforma ofrecida por West, de la cual se hablará en la sección 3.5.

3.1.5 Rodrigo Miranda

Rodrigo era el anterior desarrollador de este proyecto, ya que él tenía conocimientos básicos de Inteligencia Artificial para intentar llevar a cabo el proyecto. Sin embargo, no tenía conocimiento de las redes neuronales convolucionales dedicadas a Segmentación semántica, porque su desarrollo estuvo lleno de tropiezos, hasta que el actual desarrollador llegó a la empresa. Después de que el desarrollador tomará el proyecto para su desarrollo, Rodrigo estuvo presente para dudas sobre el funcionamiento de la empresa y acercar al desarrollador al resto del personal de West en los primeros meses de llegado a la empresa.

3.1.6 Cristian Estrada

Cristian es el empleado encargado de instalar las cámaras utilizadas en este proyecto y, por lo tanto, responde directamente a Ányelo. Cristian, junto con el desarrollador, hacen visitas al terreno para que juntos, se posicionen las cámaras de la mejor manera posible, para que el camión y su carga se mantengan siempre a la vista en las imágenes obtenidas desde la cámara.

3.1.7 Staff

El desarrollador necesita ayuda para procesar la cantidad de imágenes que eventualmente llegan desde las cámaras del cliente, con el fin de preparar un entorno de entrenamiento controlado para que la Red Neuronal del proyecto aprenda de manera satisfactoria a detectar la carga del camión y las estacas que la mantienen en su lugar.

3.2 Requerimientos del Proyecto

Como este proyecto es un software que debe usar el cliente, este debe cumplir con ciertos requisitos, los cuales se han acordado entre la empresa y el cliente. A continuación, se detallan brevemente cuáles son estos requisitos y de qué trata cada uno de ellos.

3.2.1 Reemplazar al encargado de planta

Uno de los objetivos principales anteriormente descritos es que el encargado de planta se arriesga a sufrir un accidente que, dada las circunstancias, fácilmente puede acabar con su vida (un tronco puede caerse encima, puede tropezar y quedar frente a una máquina pesada que se encontraba circulando, etc.). Esta solución propone reemplazar a esta persona con las cámaras de seguridad que el cliente tiene instaladas en terreno.

3.2.2 Utilizar las cámaras de seguridad

La empresa cliente cuenta hoy, con 2 cámaras de seguridad genéricas; una para la carga detrás de la cabina del camión y otra para la carga en el acoplado de este. De esta manera, el software va a entregar la cubicación por separado de las cargas del camión, algo que especificó el cliente es necesario para ellos.

3.2.3 Acelerar el proceso de cubicación de la carga de un camión

Como se detalló antes, para efectuar el cálculo actualmente, la persona encargada tiene que desplazarse físicamente a la carga del camión y medir cada esquina, operación que toma, en promedio, 2 minutos. Se espera que con esta propuesta el tiempo que tome la operación sea, como mínimo, el mismo que toma actualmente. Cualquier aceleración en el proceso se verá como algo positivo.

3.3 BPMN

El proceso de obtener la altura de la carga para posteriormente cubicar la cantidad de madera en el camión es un proceso que se elabora en varios pasos. A continuación, se muestra el proceso efectuado por el encargado de planta de manera manual, en la Figura 23, y el mismo proceso, efectuado por el servicio proporcionado por la Red Neuronal, en la Figura 24.

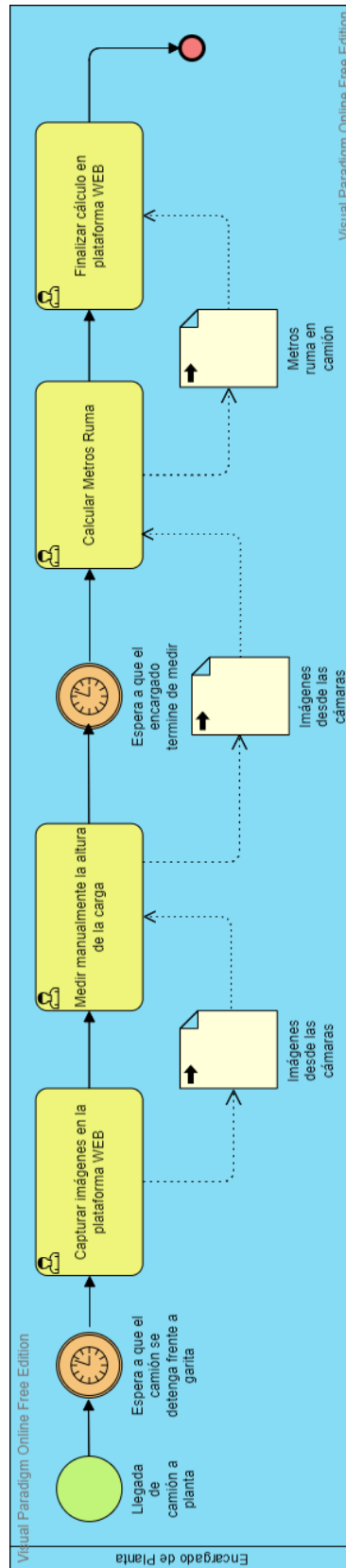


Figura 23: Diagrama BPMN proceso de cálculo manual.

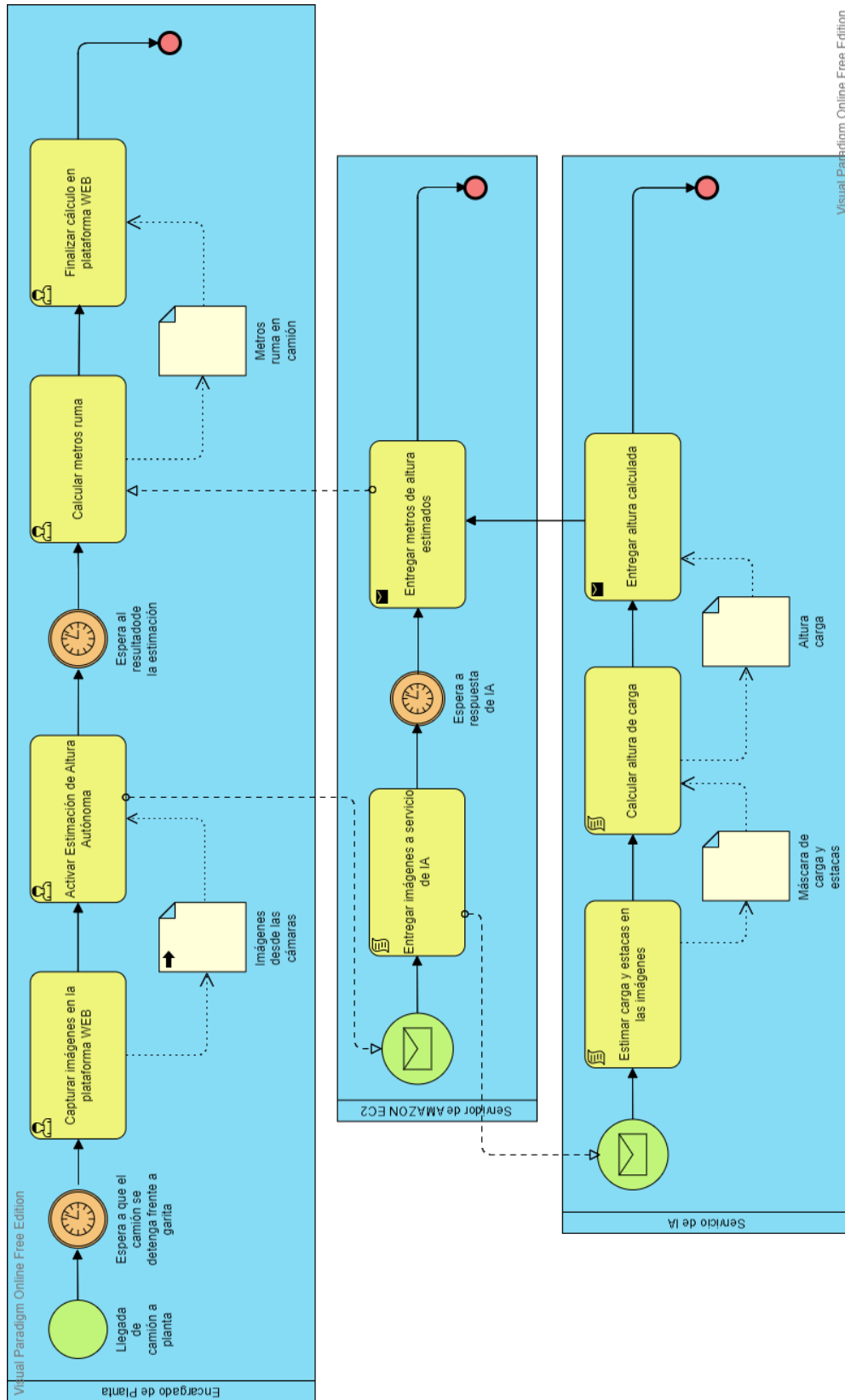


Figura 24: Diagrama BPMN estimación autónoma de altura y posterior cubicación

3.4 Casos de Uso / Diagramas de Secuencia

Como se describió anteriormente, las tareas del encargado necesarias para calcular el volumen de madera cargado de un camión son varios. A continuación, se detallan los casos de uso que busca aliviar este proyecto teniendo en cuenta los requisitos explicitados en la sección 3.2, con su correspondiente diagrama de secuencia detallando el proceso.

3.4.1 Calcular el volumen de carga de un camión.

El caso de uso que se busca aliviar se puede ver de manera sencilla en la Figura 25. Lo que necesita el cliente es que la plataforma web que utilizan actualmente, Wing (Wing, 2019), sea capaz de efectuar la tarea de calcular el volumen del camión, la cual momentáneamente es hecha por el encargado de planta, cuyos dolores se expusieron antes en el Capítulo 1.

Desde este caso de uso se puede detallar su secuencia de pasos en su correspondiente diagrama de secuencias, el cual se muestra en la Figura 26. El encargado usa la plataforma Wing para obtener una imagen tanto para el camión y su acoplado, mostrando su respectiva carga. Estas imágenes después son enviadas a través de la misma plataforma, la cual envía las imágenes al servicio de que proporciona ese proyecto, alojado en Amazon EC2. Este servicio usa la Red Neuronal para estimar las dimensiones de la carga en cada imagen, para posteriormente estimar la altura de dichas dimensiones. El resultado, ya calculado en metros, es devuelto a Wing, y desde Wing es entregado al cliente con el volumen ya calculado.

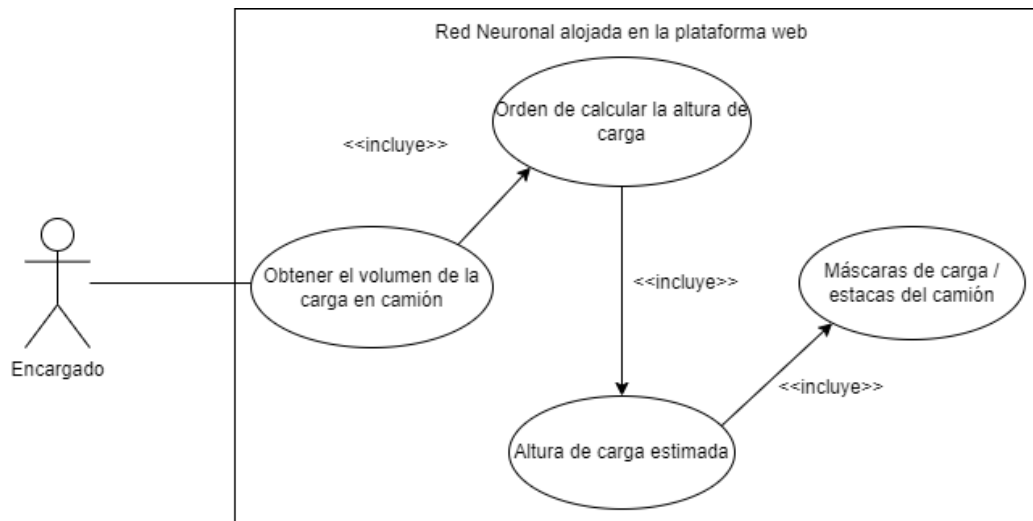


Figura 25: Caso de uso para estimar el volumen de la carga en un camión de forma autónoma.

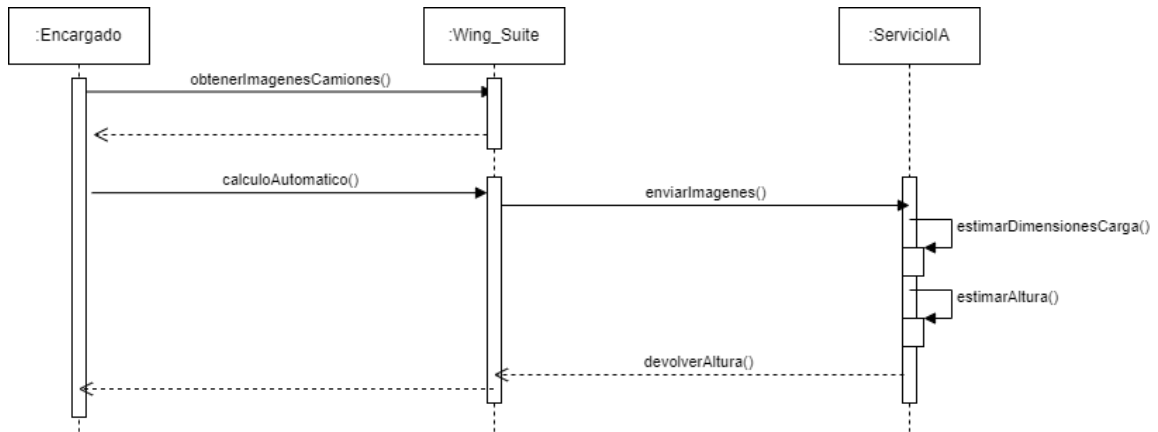


Figura 26: Diagrama de secuencia para "Calcular el volumen de carga de un camión".

3.5 GUI

El cliente ya se encuentra utilizando la plataforma Wing, desarrollada por West Ingeniería Ltda., por lo que esta empresa ya tiene conocimientos de cómo utilizar esta plataforma. A continuación, se detallará las interfaces utilizadas para documentar el volumen de la carga de los camiones del cliente; la primera para documentar cada camión conforme llegue a la planta, y la segunda para ver un reporte detallado de los camiones que pasaron por la planta en un rango de fechas, con detalles de la fecha de llegada, medidas estimadas de manera manual y automática, y el porcentaje de diferencia entre ellas; esta última interfaz se usa para probar el desempeño real del proyecto en el período de pruebas.

3.5.1 Interfaz de llegada de camión

La interfaz completa para documentar la llegada de un camión del cliente a la planta donde se descarga la madera se puede ver en la Figura 27. En esta plataforma, el encargado puede ingresar la patente del camión recién llegado y su guía asociada como se muestra en la Figura 28.

La interfaz de usuario para documentar la llegada de un camión incluye los siguientes elementos:

- Encabezado:** Funcionalidad: Cubbicación.
- Formulario de Datos:**
 - Camión:** Selección de una opción (dropdown).
 - Número de giro - guía:** Selección de una giro - guía (dropdown).
 - Ver guía:** Botón con ícono de lupa.
 - Código Producto:** Campo de texto.
 - Grúa:** Selección de una grúa (dropdown).
 - Ancho camión/carro (m):** Campo de texto con valor predeterminado 2.4.
 - Calculo Automático:** Botón con ícono de bombilla.
- Repetir foto:** Botón con ícono de cámara.
- Sector:** Selección de un sector (dropdown).
- Camión/Carro:** Selección de Camión/Carro (dropdown).
- Cantidad de bancos:** Selección de cantidad de bancos (dropdown).
- Fotos:** Dos imágenes de camiones cargados con madera, cada una con una fecha y hora de captura (10/01/2022 08:24:25).
- Medidas y Volumen:** Campos de entrada para Largo (m), Altura 1 (m), Altura 2 (m) y Volumen (MR) para cada banco.
- Resumen de Volumen:**
 - Volumen Camión (MR): 0
 - Volumen Carro (MR): 0
 - Volumen Total (MR): 0
 - Volumen Manual Total (MR): 0
- Botones de Acción:** Finalizar (naranja) y Cancelar (gris).

Figura 27: Interfaz completa para documentar la carga de un camión recién llegado.

The screenshot shows a web form titled 'Cubicación'. It has several input fields: 'Camión' with a dropdown menu open showing options 'LLKL24', 'JXZX33', 'JXZX34', 'JPFF41', and 'JPFF42'; 'Número de giro - guía' with a dropdown menu; and 'Cantidad de bancos' with a dropdown menu showing '1'. There is also a date and time stamp '13-01-2023 08:24' in the bottom right corner.

Figura 28: Ingreso de datos que identifican al camión entrante.

Este proyecto es integrado en esta pantalla con un botón de nombre “Cálculo Automático”, posicionado en la esquina superior derecha de la pantalla principal mostrado en la Figura 29. Es con este botón que la interfaz Web hace el llamado al servicio de IA desarrollada en este proyecto, entregándole las imágenes del camión para estimar las dimensiones de la carga en cada imagen.

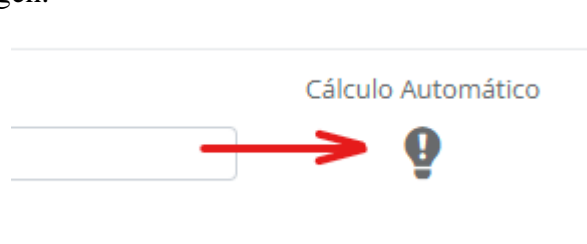


Figura 29: Botón que llama al servicio de IA para predecir el volumen de madera.

Cuando se oprime el botón “Cálculo Automático” la interfaz se bloquea momentáneamente y durante este tiempo, el servicio de IA hace la estimación de dimensiones y retorna la altura de la carga, informando al encargado de planta con una pequeña ventana flotante que aparece en pantalla, como se ve en la Figura 30.

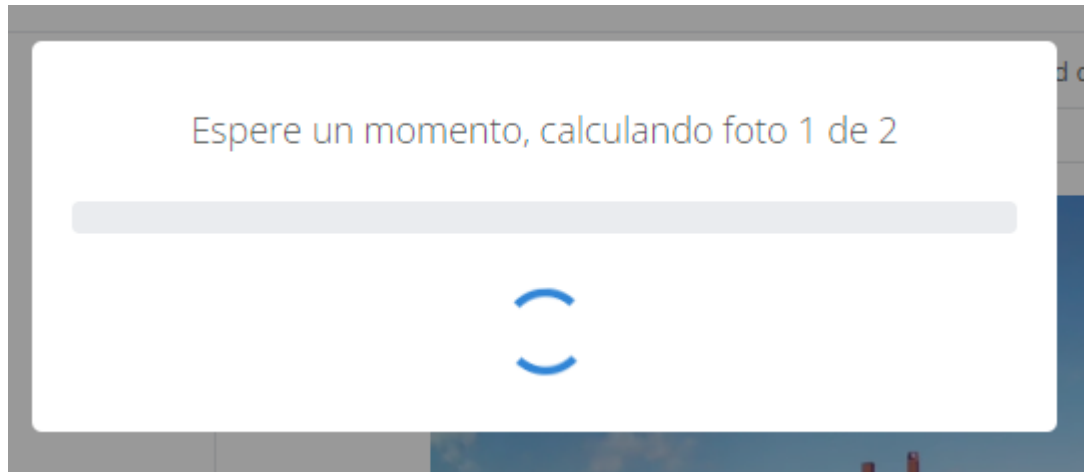


Figura 30: Ventana de Espera a que la Red neuronal estime altura de las imágenes proporcionadas.

Cuando la estimación de la Red Neuronal finaliza, la pantalla de espera desaparece y el encargado de planta vuelve a tomar el control de la plataforma; normalmente el proceso de cálculo ronda los 15 segundos. Al tomar el control, los resultados ya cubicados se encuentran en la sección a la derecha de la interfaz base, como se muestra en la Figura 31. Siendo esta la última etapa del proceso de reporte de llegada de un camión, en esta sección se puede enviar toda la información obtenida de la carga a la base de datos de Wing oprimiendo el botón “Finalizar” o, si se detectan errores en el proceso, ya sea por problemas del servicio o un error involuntario, el encargado puede reiniciar el proceso completo oprimiendo “Cancelar”. También se encuentra disponible la opción de ingresar el volumen calculado manual; esto para poder comparar el desempeño de este proyecto en la fase de pruebas.



Figura 31: Resultados cubitados en la última etapa del reporte del camión recién llegado.

3.5.2 Interfaz de Reporte de medición en llegadas de camiones

Para tener una noción de qué tan bien desempeña este proyecto, se compara sus resultados con los obtenidos con resultados manuales. Si bien se estableció anteriormente que la estimación manual es imprecisa, creemos que, conforme el entrenamiento mejore la precisión de la estimación autónoma de las dimensiones de la carga, el error entre autónomo y manual tenderá a estabilizarse; el encargado tiene su manera particular para estimar la carga, la cual por hábito debe tener una diferencia estable respecto al volumen real de madera. Para hacer esta comparación, al cliente se le habilitó un Reporte de Medición que se muestra en la Figura 32, la cual muestra un ejemplo de cuatro camiones reportados en ese instante del día; generalmente en un día se acercan alrededor de 38 camiones en esta cancha. Este reporte es sobre la cancha Arica, situada en Valdivia, en donde opera la empresa cliente; la ubicación de la cancha está disponible en el anexo B. En esta tabla, en la columna, se muestra el porcentaje de diferencia real de la estimación autónoma comparada con la estimación manual, mostrada en la Figura 33.

Reporte de medición entre las fechas 01-01-2023 00:00 y 13-01-2023 13:01

Mostrar 10 registros

Buscar:

Copiar Excel CSV

Guía	Fecha Registro	Vol Camión Manual	Vol Camión Automático	Altura Camión Sistema	Diferencia Porcentual Camión	Vol Carro Manual
96869	09-01-2023 21:05:02	3.14	3.85	1.602	22.61 %	3.67
96870	09-01-2023 20:09:59	3.19	4.13	1.723	29.47 %	3.62
96871	09-01-2023 20:59:20	3.19	4.32	1.8	35.42 %	3.74
96874	09-01-2023 21:32:49	3.72	3.56	1.484	-4.30 %	3.60

Mostrando registros del 331 al 334 de un total de 334 registros

Anterior 1 ... 30 31 32 33 34 Siguiente

Figura 32: Reporte de Medición autónoma y manual de la carga de los camiones que entraron a la planta.

Copiar Excel CSV

omático	Diferencia Porcentual Total
	-0.15 %
	6.61 %
	8.95 %
	2.46 %

Figura 33: Columna de Diferencia porcentual autónomo versus manual.

3.6 Resumen del desarrollo

Cuando al desarrollador se le asigna el desarrollo de este proyecto, inicialmente se tenía planeado utilizar una Red Neuronal especializada en Clasificar Imágenes. Estas Redes neuronales tienen una primera parte convolucional, para extraer las características principales de la imagen, seguida de capas de neuronas convencionales cuya función es que la Red Neuronal es entregar un número entero $i \in [0 \dots n]$, donde n es la cantidad de clases definidas previamente por el desarrollador. En este caso, existían tres clases de imágenes: día, noche y nublado. El fin de esta aproximación era realizar la segmentación semántica de la imagen de manera manual. Las cámaras y camiones están posicionadas en

la cancha Arica previamente mencionada, aproximadamente como es muestra en la Figura 34.

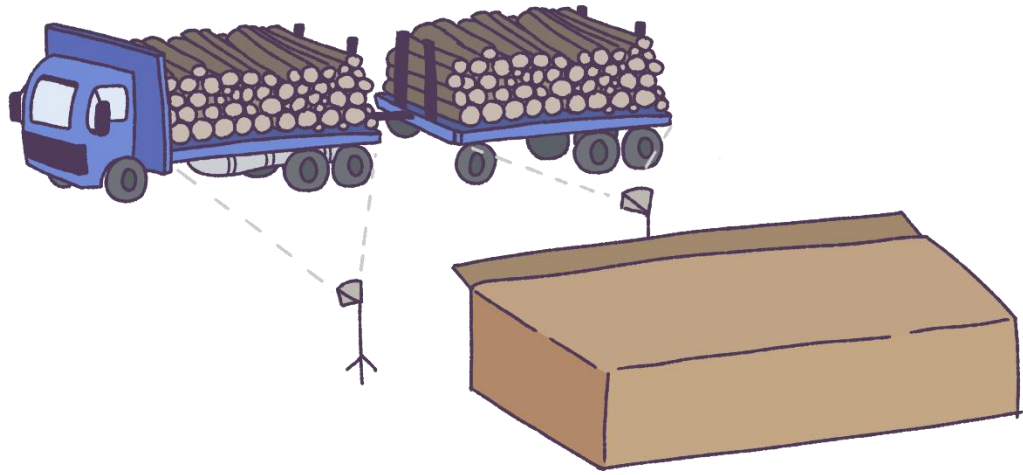


Figura 34: Posición aproximada del camión y cámaras frente a la garita del encargado de planta.

Lamentablemente esta aproximación demostró ser inviable debido a que segmentar la imagen de manera manual era ampliamente impreciso. En la Figura 35, se muestra el mejor resultado que se logró conseguir haciendo la segmentación manual. Ambas cámaras instaladas tienen efecto ojo de pez; como las cámaras son de seguridad, el efecto ojo de pez asegura mayor ángulo de visión; sin embargo, esto distorsiona la imagen, lo que termina afectando los resultados hechos por este proyecto. Debido a que lo que se necesita el cliente requiere de la mejor precisión posible, quedó claro tempranamente que esta aproximación era inviable.



Figura 35: Mejor resultado obtenido haciendo segmentación manual.

El desarrollador posteriormente utilizó una versión modificada de U-Net que permite ingresar imágenes a color, y permite salida con 4 tipos de colores: negro, para el fondo, y 3 tonos de grises para la carga, estaca y altura de carga; esta última es la combinación de los dos colores anteriores. Un ejemplo de la salida de la segunda aproximación se ve en la Figura 36.



Figura 36: Resultado en bruto de la estimación de la Red Neuronal del proyecto.

Para lograr estos resultados, primero fue necesario visitar presencialmente la planta donde llegan los camiones cargados, lugar donde se planea poner a trabajar este proyecto. Esta visita tenía como finalidad la obtención de nuevas imágenes de mejor calidad que las

utilizadas en la primera versión; antes de comenzar esta segunda aproximación, la calidad de las imágenes era deficiente. En la Figura 37, se muestra una de las nuevas imágenes que definieron un estándar de calidad para futuras imágenes para este proyecto. Las cámaras se posicionaron un poco más atrás de lo que estaban anteriormente, lo que disminuyó mucho el efecto de ojo de pez, y se logró una calidad similar a la calidad de la figura. Además, en la cancha se instaló una romana la cual se utiliza para documentar el peso del camión; esta romana es lo suficientemente angosta para estabilizar la distancia entre el camión y las cámaras, garantizando la calidad de las imágenes.



Figura 37: Una de las nuevas imágenes utilizadas para el entrenamiento de U-Net

Para que la Red Neuronal aprenda a distinguir las dimensiones de la carga del camión es necesario procesar manualmente cada imagen de entrenamiento para crear máscaras que describan precisamente dónde está la carga y las estacas; estas máscaras etiquetan cada elemento de la imagen de cada tipo. Para realizar esto se utilizan herramientas como label-studio (Label Studio, 2023) que son muy útiles al proporcionar herramientas sencillas para etiquetar cada elemento de interés en la imagen. La Figura 38 muestra dos de las tres máscaras necesarias para esta solución: carga y estacas; la tercera etiqueta, la altura de la carga, se obtiene al combinar las etiquetas de carga y estacas en una sola imagen, que es la que se utilizará para entrenar la Red Neuronal.

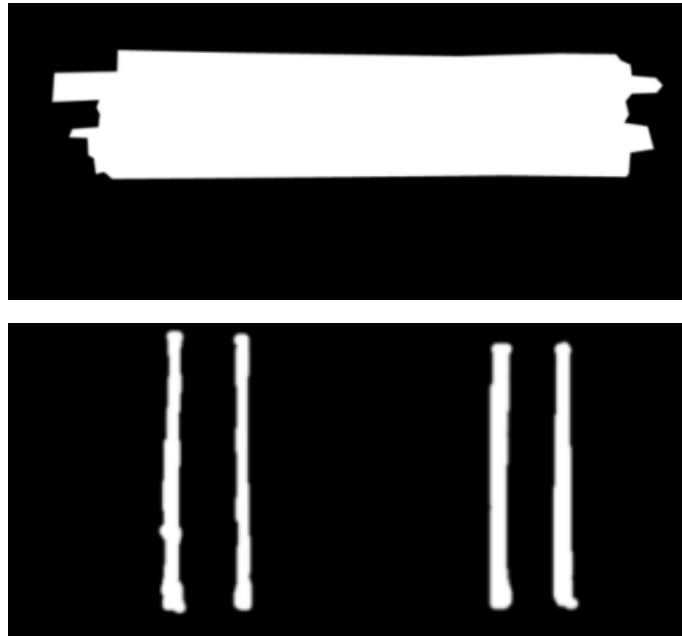


Figura 38: De arriba hacia abajo. La etiqueta de la carga y la etiqueta de las estacas de una imagen de un camión.

El siguiente paso del proyecto es utilizar el resultado obtenido de las dimensiones estimadas para calcular la altura de la carga del camión. Para obtener esta altura, se utilizan la predicción de la Red Neuronal *pred* y una altura de estacas referencial *_alt* en el algoritmo que se muestra en la Figura 39.

Algoritmo para calcular Altura	
	Entrada: <i>pred</i> : la etiqueta predicha por la Red Neuronal <i>alt</i> : la altura de las estacas en metros
	Salida: La altura estimada de la carga, en metros.
1	Inicialización de variables: se asigna a <i>i</i> y <i>j</i> matrices de ceros con las mismas dimensiones que <i>pred</i> .
2	<i>i</i> ← se pinta blanco los pixeles que coincidan con las estacas de <i>pred</i> .
3	<i>j</i> ← se pinta blanco los pixeles que coincidan con las alturas de carga de <i>pred</i> .
4	<i>pix2m</i> ← <i>alt</i> / la cantidad promedio de pixeles de altura que utilizan las estacas en <i>pred</i>
5	<i>altura</i> ← <i>pix2m</i> × la cantidad de pixeles de altura que utilizan las alturas de carga en <i>pred</i>
6	Entrega altura.
7	end

Figura 39: Algoritmo para calcular la altura de la carga con las dimensiones estimadas por la Red Neuronal

El primer paso es utilizar una referencia para transformar los valores de pixeles a metros; en este caso, la referencia utilizada son las estacas, que generalmente miden 2.08 metros. En el resultado de la Red Neuronal, las estacas están separadas en estacas y altura de carga,

por lo que hay que rescatar ambas etiquetas como una sola etiqueta auxiliar. En esta etiqueta, se aplica al algoritmo descrito en la Figura 40.

Algoritmo para obtener altura en pixeles	
	Entrada: <i>img</i> : la imagen, blanco y negro, donde el blanco describe estacas o altura, según corresponda.
	Salida: La altura estimada de zonas blancas, en pixeles.
1	conts \leftarrow contornos que describen los pixeles blancos de <i>img</i> .
2	heights \leftarrow la altura en pixeles de cada contorno obtenido de conts .
3	max_h \leftarrow la altura máxima de heights , utilizada para filtrar contornos erróneos.
4	filtered \leftarrow heights filtrado, aceptando alturas de por lo menos 90% de altura de max_h .
5	promedio \leftarrow promedio de pixeles de altura de filtered .
6	Entrega promedio.
7	end

Figura 40: Algoritmo para obtener y filtrar alturas de estacas o de carga.

Aquí, se calcula la altura en pixeles de cada estaca y se guardan. De las alturas guardadas, se filtran las alturas que son demasiado pequeñas; las estimaciones nunca son 100% limpias, por lo que es necesario limpiar posibles errores o estacas mal estimadas. Se asume con confianza que la mayor altura en pixeles es la estaca mejor estimada y se utiliza como referencia para descartar el ruido o estacas erróneas. Cualquier altura menor al 90% de la altura de la mejor estaca, es descartada, donde finalmente se retorna el promedio de pixeles de altura de las estacas de la imagen *img*.

El mismo proceso se aplica para la altura de la carga. La altura referenciada de las estacas (en metros) es dividida por el promedio de pixeles de altura de las estacas completas obtenidas anteriormente; esto retorna un ponderado de metros por píxel *pix2m*, el cual se multiplica al promedio de pixeles de altura de la carga para obtener y retornar finalmente el estimado de altura de carga del camión o acoplado en metros.

Es necesario hacer este proceso con cada imagen, porque los camiones, a pesar de tener una vía específica por la cual pasar frente a la cámaras, suelen posicionarse más cerca o más lejos de estas lo que obliga estimar la cantidad de pixeles de las estacas y altura de carga para tener una estimación precisa.

El proyecto puede separarse en dos partes: la parte de la estimación realizada por la Red Neuronal, y la parte de estimar la altura a partir de la misma realizada por el algoritmo *calc_altura*. Sin embargo, el desempeño depende fuertemente de la capacidad de la Red Neuronal de estimar correctamente las dimensiones del camión. En el Capítulo 4, se detallará la metodología para entrenar la Red Neuronal en un entorno sintético y los resultados obtenidos de esta rutina.

4. EVALUACIÓN EN ENTORNO SINTÉTICO

Previo a habilitar esta solución al cliente, la Red Neuronal debe pasar por un proceso de entrenamiento y, posteriormente, pasar a un proceso de validación; hay que corroborar que el desarrollo ha sido exitoso antes de exponer el producto al cliente. A continuación, se detallará la metodología que se utilizó para ver el progreso del aprendizaje de la Red Neuronal, junto con los resultados de este proceso.

4.1 Metodología de evaluación

Para que el proceso de entrenamiento sea exitoso, es necesaria la utilización de un optimizador y una función de pérdida. A continuación, se detallará qué son cada una de ellas.

4.1.1 Optimizadores

Un optimizador (Optimizers in Deep Learning, 2021) es un algoritmo utilizado en Deep Learning cuya función es minimizar el error que una Red Neuronal pueda tener en su tarea. Los optimizadores logran esta tarea adaptándose a cada uno de las neuronas y sus pesos del modelo de Deep Learning. Dependiendo del optimizador, la rutina que aplican es diferente; por lo mismo, los optimizadores varían dependiendo de la arquitectura y tarea que deba ejercer la Red Neuronal.

En esta ocasión, se utiliza el optimizador Adaptive Moment Estimation, o Adam para abreviar (Adam, 2021). Adam es uno de los algoritmos de optimización más populares y famosos debido a su facilidad de implementar y lo eficiente que es comparado con otras alternativas.

Adam se inspira fuertemente en la denominada Gradiente Descendente. La gradiente descendente (Gradient Descent Algorithm, 2021) es un algoritmo sencillo de optimización de primer orden dedicado a minimizar una función; es por esto que se usa de base para una gran parte de optimizadores, incluido Adam. La gradiente descendente exige que la función a la que se aplica cumpla con los siguientes requerimientos:

1. Que la función sea diferenciable,
2. Que la función sea convexa.

Que la función sea diferenciable significa que la función deba tener derivada en todo su dominio. Una función tiene dominio cuando es una función continua y sea suave. Ejemplos de una función continua se puede apreciar en la Figura 41, y ejemplos de funciones no diferenciables se pueden ver en la figura 42.

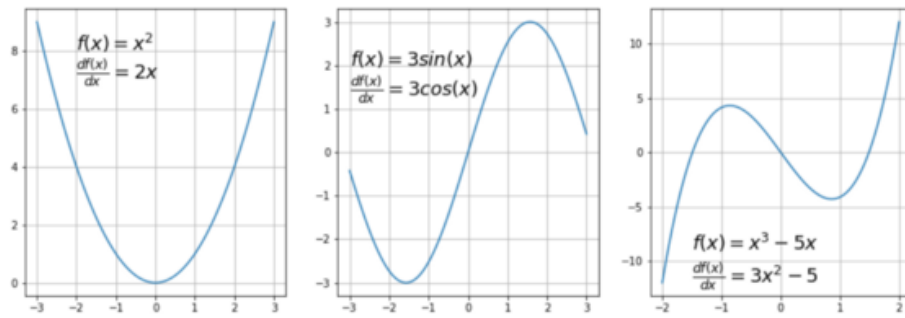


Figura 41: Ejemplos de funciones diferenciables.

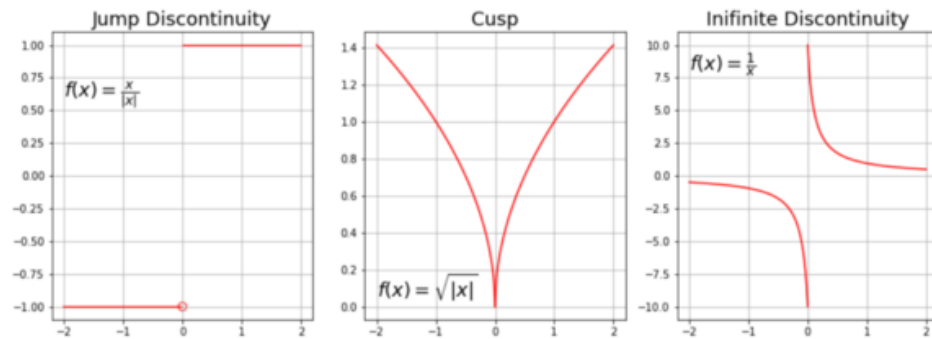


Figura 42: Ejemplos de funciones no diferenciables.

Que la función sea convexa significa que, siendo a y b puntos cualesquiera de la función, el segmento de línea que une a a y b no cruce a la función en ninguna parte. En la Figura 43, se puede ver un ejemplo tanto para una función convexa como para una función no convexa.

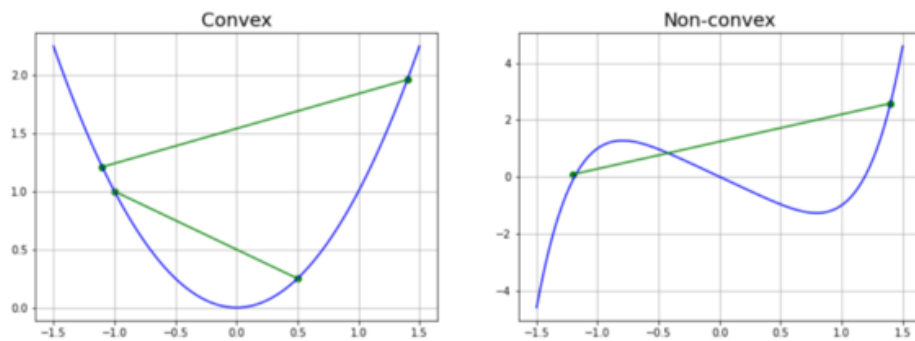


Figura 43: Ejemplo de función convexa y no convexa.

Las funciones en Deep Learning son n -dimensionales, con n la cantidad de neuronas de la Red Neuronal disponibles para entrenamiento. La gradiente en estos casos se define en la formula (2).

$$\nabla f(p) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(p) \\ \vdots \\ \frac{\partial f}{\partial x_n}(p) \end{bmatrix}, \quad (2)$$

donde p es un punto en el dominio de la función f y cada elemento del vector es la derivada parcial de la función en la neurona n , aplicada a p . Para aplicar la gradiente descendente, se aplica la fórmula (3),

$$p_{n+1} = p_n - \eta \nabla f(p_n), \quad (3)$$

donde η es la tasa de aprendizaje que define el desarrollador; generalmente se trabaja con una tasa de aprendizaje de 0.001 o 0.0001; esto debido a que los optimizadores en general son muy sensibles esta constante y un valor demasiado grande o muy pequeño impactan severamente el desempeño de un algoritmo de optimización. En la Figura 44 se ejemplifica

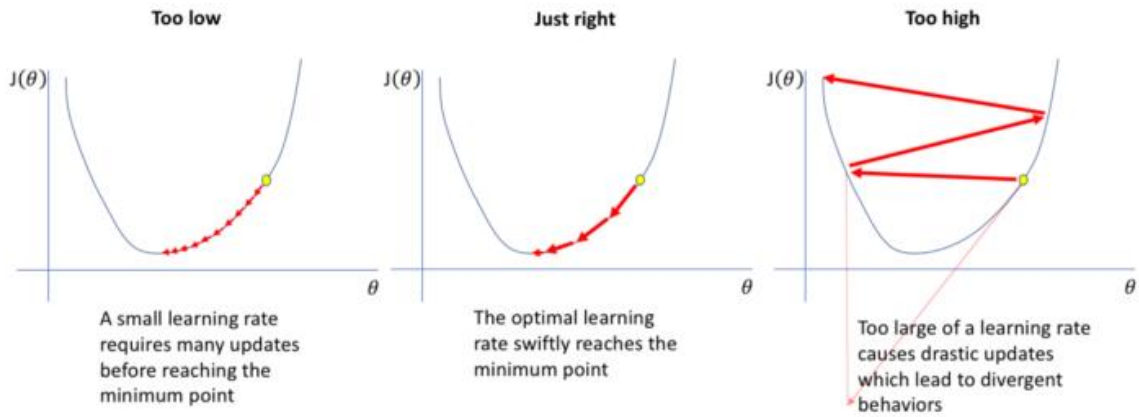


Figura 44: Ejemplo de tasas de aprendizajes correctas e imprecisas.

Adam aplica momentum a la gradiente descendente tomando en cuenta el promedio y pesos de las gradientes aplicadas a un punto p ; este proceso se hace de manera iterativa. Para esto, Adam define el momentum con la fórmula (4),

$$w_{t+1} = w_t - \alpha m_t, \quad (4)$$

donde

$$m_t = \beta m_{t-1} + (1 - \beta) \left[\frac{\delta L}{\delta w_t} \right] \quad (5)$$

m_t es la suma de la gradiente en el tiempo t ; inicialmente es cero. m_{t-1} es la suma de la gradiente en el tiempo $t - 1$. w_t son los pesos en el tiempo t . w_{t+1} son los pesos en el tiempo $t + 1$. α es la tasa de aprendizaje en el tiempo t . δL es la derivada de la función de pérdida. δw_t es la derivada de los pesos en el instante t . β es una constante de movimiento promedio, usualmente de valor 0.9.

Adam además define la Propagación Cuadrática de la Raíz del promedio (RMSP, por sus siglas en inglés) con la fórmula (6):

$$w_{t+1} = w_t - \frac{\alpha_t}{(v_t + \varepsilon)^{1/2}} * \left[\frac{\delta L}{\delta w_t} \right], \quad (6)$$

donde

$$v_t = \beta v_{t-1} + (1 - \beta) * \left[\frac{\delta L}{\delta w_t} \right]^2 \quad (7)$$

w_t son los pesos del gradiente en el instante t . w_{t+1} son los pesos del gradiente en el instante $t + 1$. α_t es la tasa de aprendizaje en el instante t . δL es la derivada de la función de pérdida. δw_t es la derivada de los pesos en el instante t . v_t la suma de los pesos anteriores al cuadrado; inicialmente es cero. β es una constante de movimiento promedio, usualmente de valor 0.9. ε es una pequeña constante positiva; generalmente es 10^{-8} .

De esta manera, Adam mantiene un control de la gradiente descendente, logrando superar mínimos locales en la función aplicada, con momentum inicialmente grandes (salvo el instante cero), y logra una convergencia precisa adaptando su momemtum a valores cada vez menores. En la Figura 45, se muestra el desempeño de Adam contra otros optimizadores, evidenciando su ventaja sobre ellos, justificando su elección.

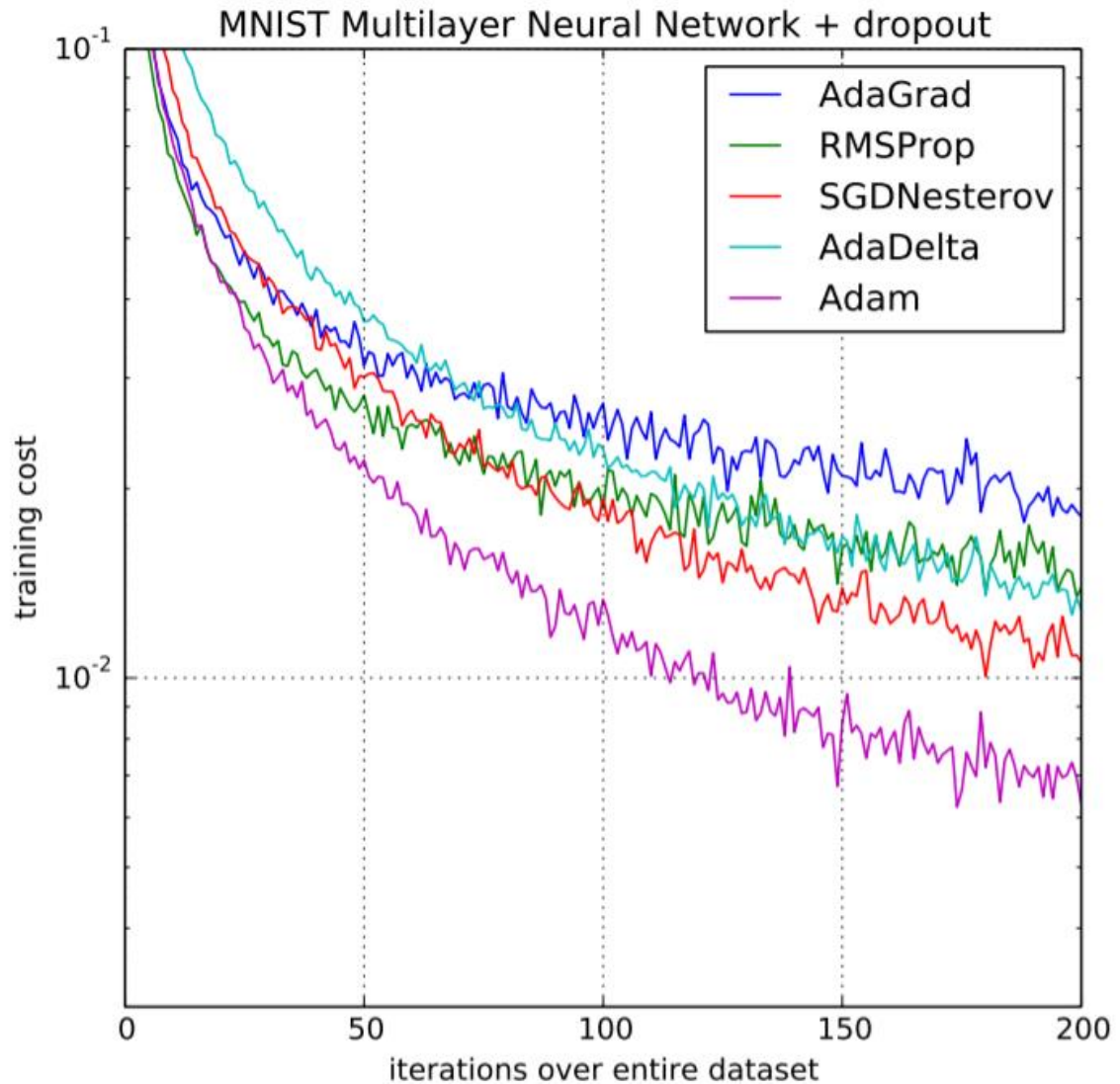


Figura 45: Comparativa de convergencia entre diferentes optimizadores en un ejemplo de entrenamiento.

4.1.2 Funciones de pérdida

Una función de pérdida, también conocida como función de coste o función objetivo, es una función que mide la diferencia entre el resultado previsto y el resultado real de un modelo de Deep Learning. El objetivo del entrenamiento de un modelo de aprendizaje automático es minimizar el valor de la función de pérdida. La importancia de una función de pérdida en el aprendizaje automático es que sirve de guía para que el modelo aprenda y haga predicciones precisas sobre datos nuevos y desconocidos. El optimizador toma como referencia la evaluación hecha por la función de pérdida y de manera iterativa minimizar resultado (o pérdida) que entrega esta función ajustando adaptativamente los pesos de cada una de las neuronas de la Red Neuronal; inicialmente, el ajuste sugerido por el optimizador a la Red Neuronal es mayor porque la Red Neuronal no ha aprendido nada,

y conforme el entrenamiento avance, el ajuste será progresivamente menor, al acercarse al mínimo de la función de pérdida.

Existen distintos tipos de funciones de pérdida según el tipo de problema y el tipo de modelo. En este caso, el problema es de índole multi clase, ya que tenemos que detectar carga, estacas y altura; al ser cosas muy diferentes y porque el problema lo necesita, se tratan como clases diferentes.

La función de pérdida Cross Entropy (Loss Functions in Machine Learning, 2021) una función muy común en problemas multi clase y la escogida para este problema. La función de pérdida Cross-entropy, también conocida como función de pérdida logarítmica, mide la diferencia entre la distribución de probabilidad de la predicción y la distribución de probabilidad de la etiqueta real.

La función Cross Entropy se define en la fórmula (8),

$$L = \frac{-1}{N} \times \sum_i [y_i \times \log(p_i) + (1 - y_i) \times \log(1 - p_i)], \quad (8)$$

donde y es la etiqueta real que, en esta ocasión, es una etiqueta de clase que puede ser $n \in [0,1,2,3]$. p es la probabilidad prevista para la etiqueta verdadera; como la imagen es un conjunto de pixeles y cada etiqueta tiene una cantidad finita presente en la imagen, se establece una probabilidad de que se presente dicha etiqueta en la imagen real. N es el número de imágenes por lote; el conjunto completo de imágenes se separa en lotes para optimizar el uso de recursos en el PC que ejecute el proceso de entrenamiento.

De esta manera, la función Cross Entropy castiga severamente una predicción incorrecta y recompensa al modelo por una predicción correcta. La presencia del logaritmo en la función garantiza que la penalización sea mayor que la recompensa por una predicción verdadera. El signo negativo al inicio establece el objetivo de minimizar esta función.

4.1.3 Métricas para la evaluación del modelo

El entrenamiento de la Red Neuronal pasa por una cantidad de épocas determinada por el desarrollador. En cada una de ellas la Red Neuronal pasa por cada una de las imágenes guardadas en el conjunto de datos de entrenamiento. Luego de pasar por ese conjunto, se evalúa el desempeño del modelo con un conjunto de imágenes que la Red Neuronal nunca haya visto; este conjunto se le denomina conjunto de validación. Además de evaluar el error con Cross Entropy, puede verse su desempeño aplicando otras métricas. En este caso, se utilizó el Índice Jaccard, o Intersection over Union (IoU por sus siglas en inglés).

IoU (Intersection over Union (IoU) for object detection, 2016) se define con la formula (9)

$$\frac{A \cap B}{A \cup B}, \quad (9)$$

donde A es la etiqueta predicha por la Red Neuronal y B es la etiqueta real de la imagen. El rango de valores de la fórmula (9) va desde cero a uno; por ello, IoU se puede considerar un porcentaje. El valor será cercano a cero cuando la etiqueta A sea una predicción muy pobre, y muy cercano a uno cuando A y B sean prácticamente iguales.

4.2 Resultados

Los resultados del entrenamiento se pueden ver en las Figuras 46 y 47. La pérdida del modelo usando la función Cross Entropy muestra claros símbolos de haber bajado de manera limpia, llegando un mínimo local muy cercano a cero. Por otro lado, IoU se ha acercado al 0.8, o 80%, lo que deja entender que la precisión de la estimación de la Red Neuronal es muy buena. En la Figura 49, se muestra un ejemplo de una imagen directamente desde una de las cámaras y la etiqueta predicha por la Red Neuronal sobre la imagen, con colores para una mejor visualización.

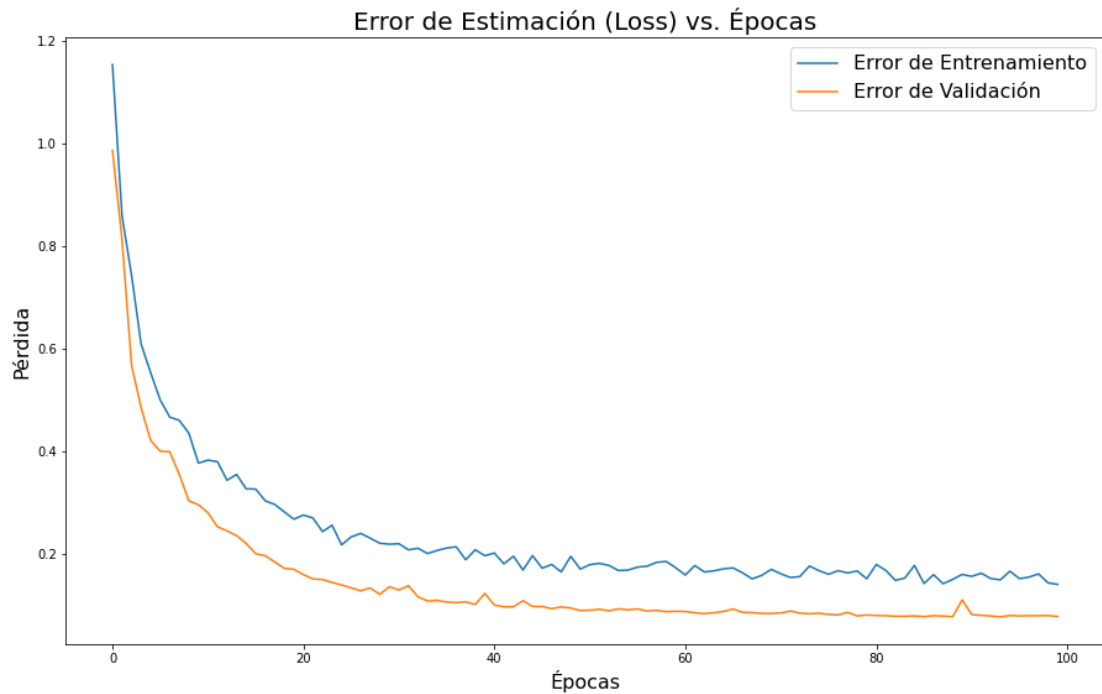


Figura 46: Resultados de la función Cross Entropy en los conjuntos de entrenamiento y validación.

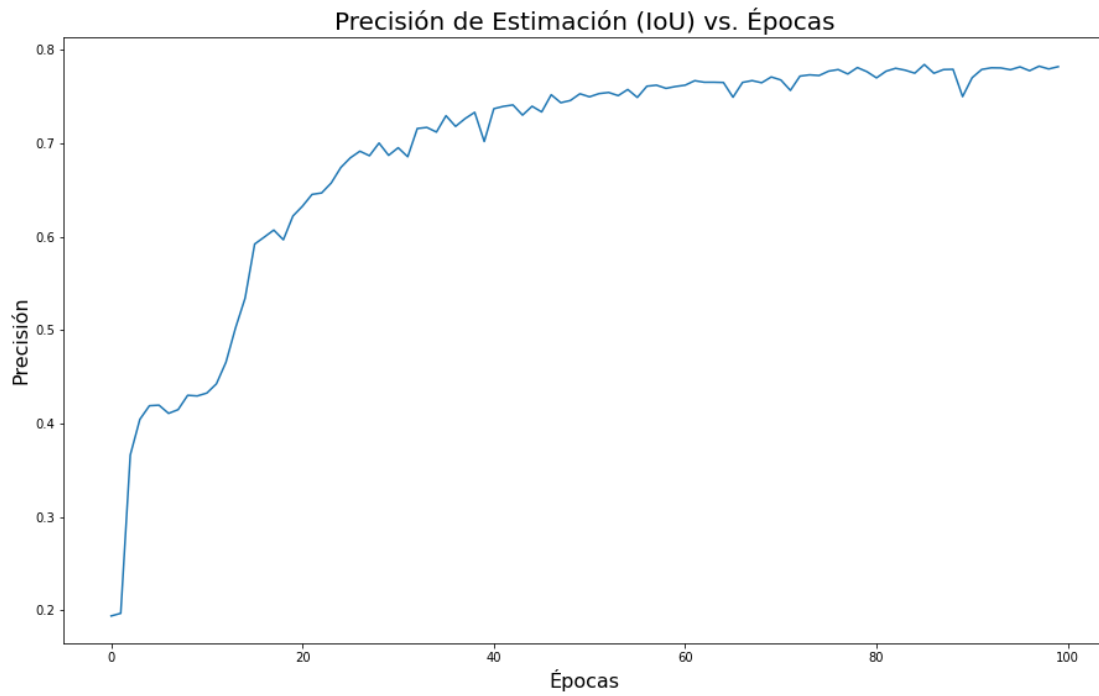


Figura 47: Desempeño del modelo en el conjunto de validación con IoU.



Figura 48: La estimación de la Red Neuronal sobre la imagen de un camión.

5. EVALUACIÓN EN ENTORNO REAL

El desempeño del modelo ha probado ser prometedor en el entorno sintético, por lo que ya se puede hacer pruebas de rigor con el entorno real, junto con el cliente. A continuación, se detallan la metodología de cómo se lleva a cabo esta evaluación, y los resultados obtenidos.

5.1 Metodología de evaluación

Para evaluar en el entorno real, se habilitó la interfaz descrita en la sección 3.5.2. En esta interfaz se documentará las alturas estimadas por el algoritmo, que toma la etiqueta estimada por la Red Neuronal, y las alturas estimadas por el encargado de planta. Además, se detalla el porcentaje de diferencia de camión y acoplado por separado, y un porcentaje de diferencia total. Las pruebas se efectuaron desde el día 16 de enero de 2023 hasta el 19 de enero de 2023; durante este rango, se estimaron un total de 233 camiones.

5.2 Resultados

Los resultados expresados en porcentajes se pueden ver en la Tabla 4. Las estimaciones de altura tienen un promedio de alrededor del 1%. Ha habido casos puntuales en donde los camiones suelen posicionarse demasiado cerca de las cámaras, distorsionando la imagen; la Red Neuronal estima las dimensiones sin tener en cuenta las distorsiones que pueda tener la imagen por estar demasiado cerca de la cámara. En la Figura 49, se detalla un histograma que evidencia que la mayoría de los casos se sitúan en unas diferencias - 3% y 1.18%.

El cliente esperaba una diferencia máxima de 3% para la estimación autónoma; por lo tanto, en los casos donde el etiquetado de la Red Neuronal es correcto, las expectativas se cumplen satisfactoriamente.

Tabla 4: Resultados de las estimaciones en el entorno real.

Promedio % error total	STD % error total
1.40	6.14

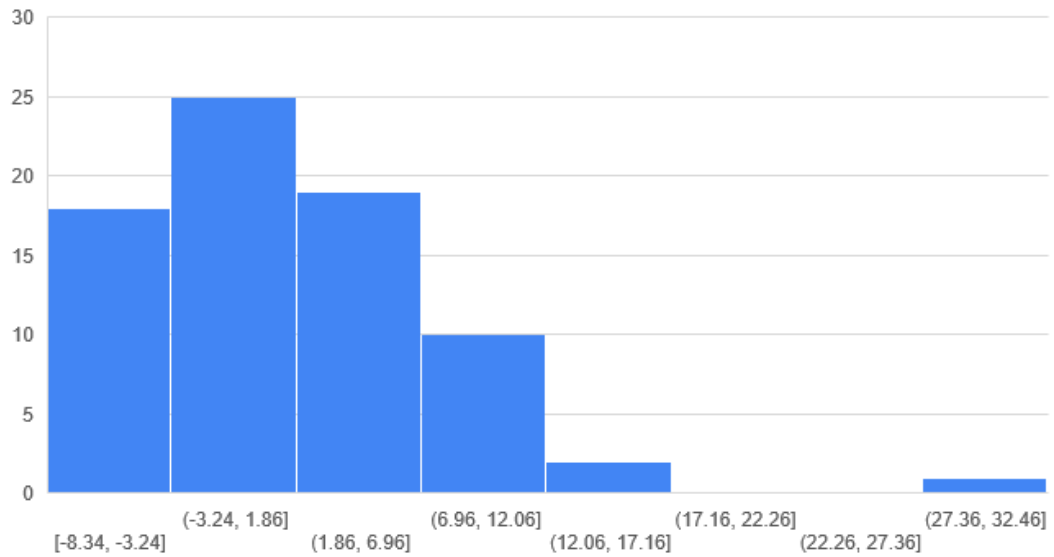


Figura 49: Distribución de porcentajes de diferencias entre estimación autónoma y manual.

6. CONCLUSIONES Y TRABAJOS FUTUROS

Por las pruebas efectuadas tanto en el entorno sintético como en el entorno real, y considerando el porcentaje máximo de fluctuación entre lo autónomo y manual, el proyecto se encuentra en una posición muy favorable para el cliente y para el desarrollador.

Sin embargo, es necesario puntualizar de que la Red Neuronal necesita mayor entrenamiento. Hay situaciones en donde existe madera detrás del camión, lejos en el predio, que la Red Neuronal considera carga, como se muestra en la Figura 50. Afortunadamente, este falso positivo se soluciona rescatando las imágenes brutas de la cámara de los casos en donde el modelo se equivoque. El cliente ha sido muy comprensivo con la situación, porque los resultados en donde las dimensiones están correctamente estimadas, el desempeño del proyecto es muy bueno.



Figura 50: Ejemplo de un falso positivo de la Red Neuronal considerando carga una ruma de troncos a la distancia.

7. REFERENCIAS

A Brief History of GPUs. Accedido el 18 de octubre, 2022, desde <https://medium.com/neuralmagic/a-brief-history-of-gpus-27122d8fd45#:~:text=GPUs%20began%20to%20be%20used,that%20require%20complex%2C%20simultaneous%20calculations>.

Adam. Accedido el 26 de enero de 2023, desde <https://optimization.cbe.cornell.edu/index.php?title=Adam>.

Amazon EC2. Capacidad de computación segura y de tamaño ajustable para prácticamente cualquier carga de trabajo. Accedido el 1 de enero, 2023, desde <https://aws.amazon.com/es/ec2/>

ARTIFICIAL INTELLIGENCE IN THE STOCK MARKET: HOW DID IT HAPPEN? Accedido el 17 de octubre, 2022, desde <https://business.fiu.edu/graduate/insights/artificial-intelligence-in-the-stock-market.cfm>

Gradient Descent Algorithm — a deep dive. Accedido el 27 de enero de 2023, desde <https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21>

Grilli, E., Menna, F. & Remondino, F. A review of Point Clouds Segmentation and Classification Algorithms, en The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, volumen XLII-2/W3, 2017, <https://dnb.info/1143876288/34>.

Historia de la IA: Frank Rosenblatt y el Mark I Perceptrón, el primer ordenador fabricado específicamente para crear redes neuronales en 1957. Accedido el 20 de octubre, 2022, desde <https://empresas.blogthinkbig.com/historia-de-la-ia-frank-rosenblatt-y-e/>

History of AI Winters. Accedido el 18 de octubre, 2022, desde <https://www.actuaries.digital/2018/09/05/history-of-ai-winters/>

History of the Perceptron. Accedido el 18 de octubre, 2022, <https://home.csulb.edu/~cwallis/artificialn/History.htm>

Image Segmentation: Architectures, Losses, Datasets, and Frameworks. Accedido el 18 de octubre, 2022, desde <https://neptune.ai/blog/image-segmentation>

IMAGE SEGMENTATION DEEPLABV3 ON IOS. Accedido el 18 de octubre, 2022, desde https://pytorch.org/tutorials/beginner/deeplabv3_on_ios.html#image-segmentation-deeplabv3-on-ios

Intersection over Union (IoU) for object detection. Accedido el 27 de enero de 2023, desde <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>

Label Studio. Open Source Data Labeling Platform. Accedido el 13 de enero de 2023, desde <https://labelstud.io/>.

Ley-20283 30-JUL-2008 MINISTERIO DE AGRICULTURA - Ley Chile - Biblioteca del Congreso Nacional. Accedido el 10 de octubre de 2022, desde <https://www.bcn.cl/leychile/navegar?idNorma=274894>.

Loss Functions in Machine Learning. Accedido el 27 de enero de 2023, desde <https://medium.com/swlh/cross-entropy-loss-in-pytorch-c010faf97bab>

Macronline, Sistema Láser para Calcular el Volumen de Carga de Camiones. Accedido el 18 de octubre, 2022, desde <https://www.macronline.cl/cubicacion-de-carga-y-volumen-de-camiones/>

McCulloch, W.S., Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5, 115–133 (1943). <https://doi.org/10.1007/BF02478259>

Optimizers in Deep Learning. Accedido el 26 de enero de 2023, desde <https://medium.com/mllearning-ai/optimizers-in-deep-learning-7bf81fed78a0>.

Qué es el "invierno de la inteligencia artificial" y por qué hay expertos que creen que estamos acercándonos a uno. Accedido el 20 de octubre, 2020, desde <https://www.bbc.com/mundo/noticias-51097189>.

Rahman, A., Yella, S. & Dougherty, M. IMAGE PROCESSING TECHNIQUE TO COUNT THE NUMBER OF LOGS IN A TIMBER TRUCK, en *Acta Press*, volumen 10.2316/P.2011, pp. 759-060, 2011, <https://www.actapress.com/PaperInfo.aspx?paperId=453153>.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408. <https://doi.org/10.1037/h0042519>

Rundgren, E. Automatic Volume Estimation of Timber from Multi-View Stereo 3D Reconstruction., en *Department of Electrical Engineering, Linköping University*, volumen LiTH-ISY-EX--17/5093—SE, 2017, <https://liu.diva-portal.org/smash/get/diva2:1153580/FULLTEXT01.pdf>

U-Net: Convolutional Networks for Biomedical Image Segmentation. Accedido el 18 de octubre, 2022, desde <https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/>

Wing Suite. Tecnología para movilizar de forma eficiente, organizada y segura su empresa. Accedido el 11 de enero de 2023, desde <https://wing.cl/>.

8. ANEXOS

Anexo A

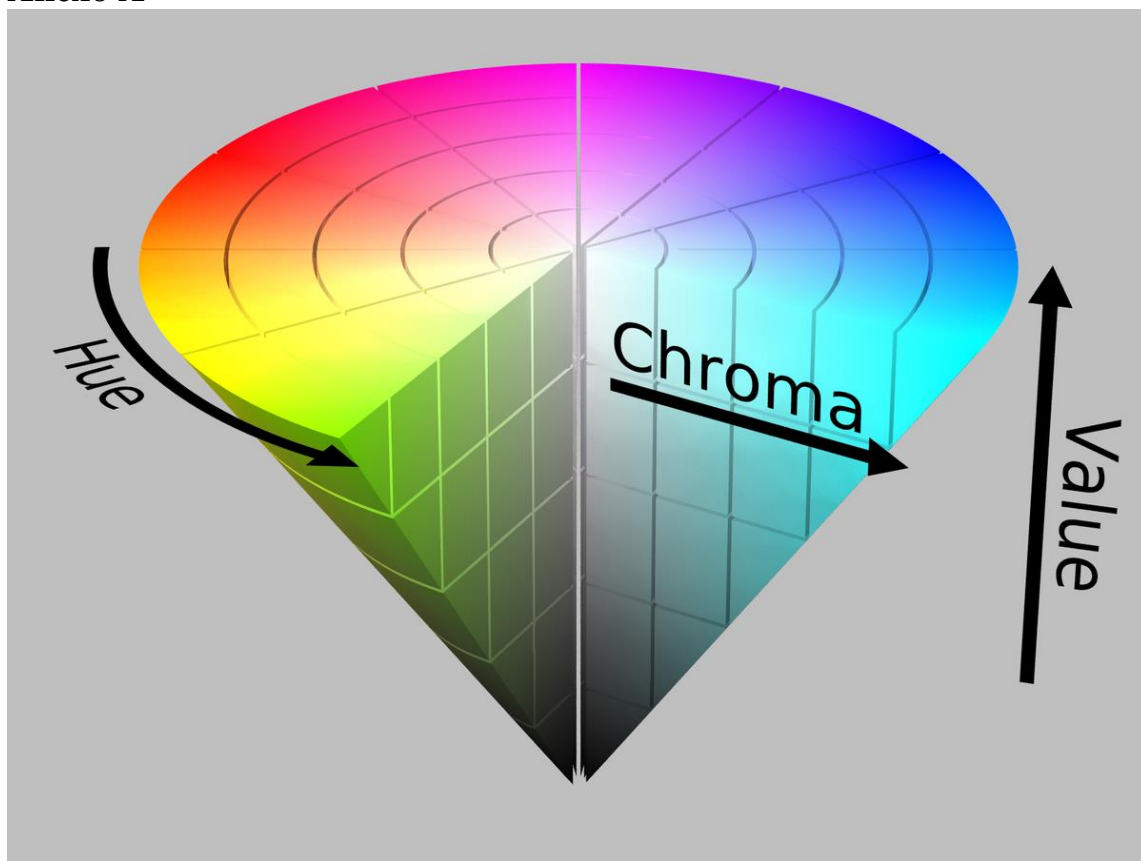


Figura 51: Espacio de Color HSV.

Anexo B



Figura 52: Cancha Arica, situada en Valdivia (-39.837347406912265, -73.25789258212504).