FORMATO 03

Aprobado según acta de Comité Curricular No. 020 de 2006

FACULTAD SEDE SOGAMOSO
ESCUELA DE INGENIERÍA ELECTRÓNICA

PROPUESTA DE PROYECTO DE TRABAJO DE GRADO, MODALIDAD: PARTICIPACIÓN ACTIVA EN UN GRUPO DE INVESTIGACIÓN RECONOCIDO POR LA DIN

FECHA DE PRESENTACIÓN: 05 - 08- 2024

IDENTIFICACIÓN DEL PROYECTO

TITULO: Diseño de un set-up para la evaluación de la confiabilidad de una Red Neuronal Convolucional (CNN) en alto nivel.

PROPONENTE(S) David Esteban Gonzalez Becerra **CÓDIGO:** 201812189

DIRECTORES: Ing. Luis Ariel Mesa Mesa, Ph.D(c).
Ing. Wilson Javier Pérez Holguín Ph.D

LÍNEA DE PROFUNDIZACIÓN: Electrónica Digital

GRUPO DE INVESTIGACIÓN: Grupo de Investigación en Robótica y Automatización Industrial - GIRA

NOMBRE P.I.I.I.: Reliability assessment of deep neural networks implemented in parallelized hardware platforms

LÍNEA DE INVESTIGACIÓN: Diseño y Test

1. ANTECEDENTES

1.1. Antecedentes

Redes Neuronales Convolucionales

La inteligencia artificial, en particular las Redes Neuronales Artificiales (*Artificial Neural Networks* - ANNs), han experimentado un crecimiento exponencial en su aplicación para diversos campos, que van desde el reconocimiento de imágenes hasta la toma de decisiones en sistemas de navegación autónoma. Aunque estas redes neuronales son altamente resilientes por sus características, pueden presentar vulnerabilidades que deterioran su confiabilidad, afectando su uso en aplicaciones como las denominadas aplicaciones de “misión crítica”. Una aplicación de misión crítica es aquella donde la ocurrencia de una falla no es tolerable ya que implica una posible afectación a la integridad del ser humano, el medio ambiente y/o el equipamiento mismo.

Las Redes Neuronales Convolucionales (*Convolutional Neural Networks* - CNNs) son una arquitectura de aprendizaje profundo (*Deep Learning* - DL) desarrollada por (LeCun & others, 1989) con el propósito de procesar información en forma de arreglos numéricos (Pérez Cerdeira, 2021). En relación a esto, se utilizan actualmente para identificar patrones, lo que las hace ideales para diversas tareas, como el reconocimiento, segmentación y clasificación de imágenes, entre otras. Su desempeño ha alcanzado niveles sobresalientes, impulsando avances significativos en diversas aplicaciones, tales como diagnóstico médico, análisis de datos científicos, control de procesos, navegación autónoma de vehículos, etc.

La arquitectura de una CNN se compone de múltiples capas que aplican filtros al conjunto de datos de entrada (dataset) para propagar los resultados a las capas siguientes. Esta técnica facilita la detección y extracción de características relevantes, lo que enriquece el proceso de aprendizaje y la representación efectiva de patrones en la red. Cada capa realiza operaciones de convolución y agrupación para capturar detalles importantes y reducir la dimensionalidad, permitiendo que la red aprenda de forma progresiva y aborde tareas complejas de manera eficiente. (Moreno, 2019) agrega que las capas convolucionales de una red neuronal dividen la matriz de entrada en subconjuntos más pequeños mediante operaciones matemáticas con un kernel o filtro de reducida dimensión. Las CNNs, según (Pérez Cerdeira, 2021), representan un enfoque revolucionario en las redes neuronales al reemplazar las multiplicaciones y los pesos unidimensionales utilizados en las redes convencionales, como las fully connected o perceptrón multicapa, por convoluciones y máscaras de filtros. De manera similar, (Moreno, 2019) describe las CNNs como un conjunto de capas diseñadas para emular el córtex visual del cerebro humano y detectar diversas características en las entradas. La figura 1, muestra una representación general de la arquitectura de una CNN.

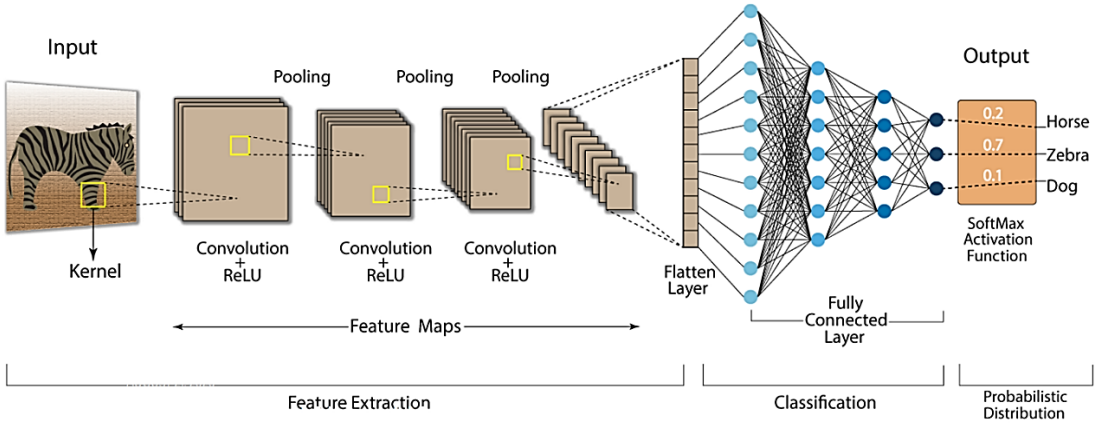

Figura 1. Representación de la arquitectura general de una CNN.
Fuente: https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5

El manejo de una red neuronal se divide en dos aspectos importantes, el entrenamiento y la inferencia. El entrenamiento es el proceso mediante el cual la red "aprende" a partir de un conjunto de datos etiquetados de entrada (*dataset)*, con la introducción de cada nuevo dato, la red ajusta progresivamente sus parámetros internos, denominados pesos y sesgos (weights & biases), con el objetivo de minimizar una función de pérdida. Este ajuste (minimización de la función de pérdida) se logra comparando la diferencia entre las predicciones de la red y los valores reales definidos por las etiquetas del dataset de entrenamiento. Por otro lado, la inferencia es el proceso en el que se pone en funcionamiento realmente la red entrenada. Allí se introducen nuevos datos de entrada a la red y esta debe realizar la respectiva predicción o clasificación de clases. Esto se puede visualizar en la figura 2.

Para la implementación de una CNN pueden existir dos niveles, el nivel de hardware y el nivel de aplicación. El primero involucra la plataforma de hardware que corre la red neuronal, mientras la otra incluye solo el nivel de aplicación.

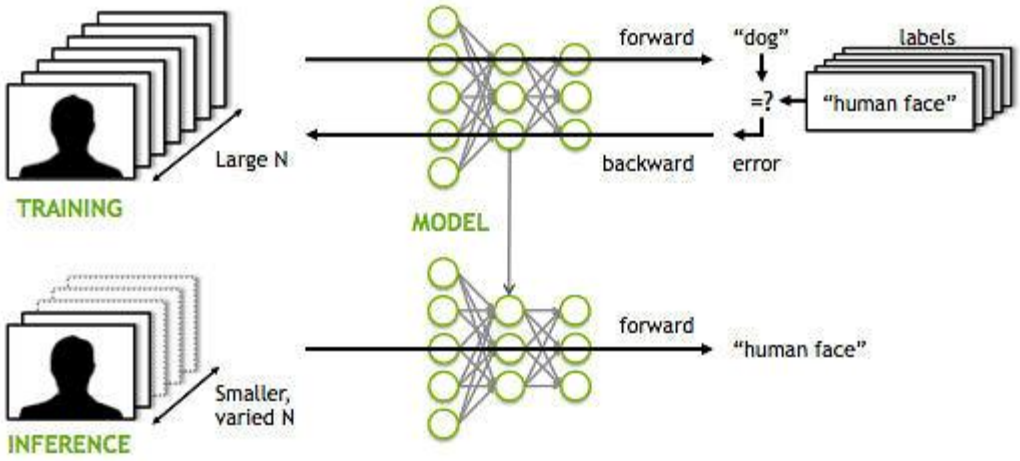

Figura 2. Procesos de entrenamiento e inferencia de una ANN.
Fuente: https://mitxpc.com/pages/ai-inference-applying-deep-neural-network-training

(Moreno, 2019) indica que, en el campo de la inteligencia artificial, existen diversos entornos de desarrollo, entre los que destacan Python y Matlab. Para el caso de Python existen plataformas como TensorFlow o Pythorch, que permiten construir, entrenar y evaluar modelos de aprendizaje automático, a través de una serie de librerías.

Confiabilidad de una Red Neuronal Convolucional

La confiabilidad en una red neuronal se refiere a la capacidad de la red para producir resultados consistentes y precisos en una amplia variedad de situaciones y datos de entrada, que pueden estar asociados con la aplicación. Para las aplicaciones de misión crítica este aspecto resulta fundamental ya que un fallo puede desencadenar efectos adversos no deseados (Hernández et al., 2022).

Las CNNs son por naturaleza, arquitecturas de aprendizaje profundo altamente resilientes gracias a sus capacidades de generalización, tolerancia al ruido y su aprendizaje adaptativo. Sin embargo, su confiabilidad puede verse vulnerada por eventos como errores en la programación, errores en los datos de entrenamiento, sobreajuste (overfitting), errores de preprocesamiento de datos, errores en la arquitectura de la red, hiperparámetros incorrectos, problemas de optimización, requisitos de hardware inadecuados, bug en el código, errores de hardware, ataques maliciosos, entre otros.

La evaluación de la confiablidad permite a los expertos y usuarios de estas herramientas, hacerlas más robustas ante sus vulnerabilidades, sin embargo, identificar las situaciones específicas que pueden hacerlas fallar no es una tarea fácil. Se pueden utilizar diferentes enfoques y técnicas que permiten inyectar fallos intencionados a la red con el propósito de descubrir los puntos más críticos. En este escenario normalmente se evalúa el desempeño de la red sometida a un ataque de fallos controlados, contra un modelo de la misma red sin fallos denominado modelo "*Golden Model*", esto se puede visualizar en la figura 3. Evaluar la confiabilidad puede tener además como objetivo, el nivel de aplicación (modelo de la CNN independiente de la tecnología) y el sistema completo que incluiría tanto el modelo de la red como la plataforma de hardware.

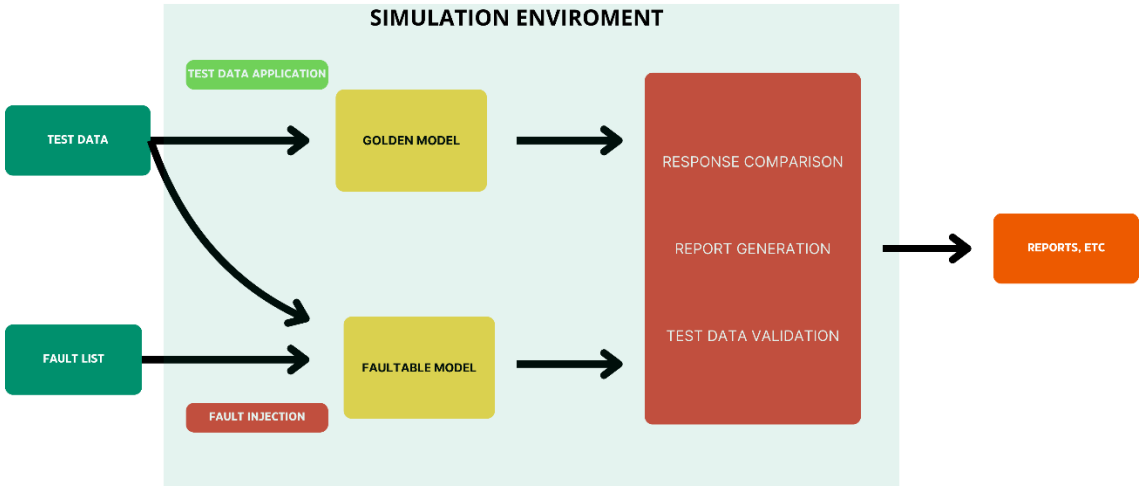

Figura 3. Esquema de inyección de fallos y evaluación de la confiabilidad.

Una de las técnicas para evaluar la confiabilidad de algún sistema es la inyección de fallos. En el contexto de las redes neuronales convolucionales (CNNs), esto implica introducir deliberadamente errores, ruido o condiciones inusuales en la entrada, el modelo o el proceso de entrenamiento para evaluar cómo responde la CNN a estas situaciones. Estas pruebas ayudan a identificar vulnerabilidades, mejorar la robustez y entender mejor el comportamiento de la CNN bajo condiciones adversas. Ejemplos de inyección de fallos en CNNs incluyen la introducción de ruido en los datos de entrada, perturbaciones en los pesos, eliminación de capas o nodos, alteración de la arquitectura, ataques adversarios, rotación o traslación de datos, interrupción de la conectividad de la red y simulación de condiciones de iluminación y entorno.

La literatura muestra que se han propuesto diferentes tipos de arquitectura de CNNs para atender a diferentes necesidades. Estas arquitecturas han considerado variaciones como el tamaño de la imagen de entrada, el tipo de operaciones, el número de capas, el tipo de funciones de activación, la representación de datos, entre otros. Por ejemplo, en (Bengio & Haffner, 1998) se propuso una arquitectura de CNN denominada LeNet-5, la cual para procesar imágenes de 32x32 pixeles, incluía siete capas, sin considerar la entrada, donde cada una contiene parámetros entrenables llamados pesos y sesgos. En (Krizhevsky et al., 2017), se propuso otro modelo de red llamado AlexNet, el cual procesa imágenes de 224x224 a través de una arquitectura de cinco capas convolucionales y tres fully connected. La salida de la última capa fully connected alimenta un clasificador softmax de 1000 salidas que produce una distribución sobre el mismo número de etiquetas de clase. Algunas arquitecturas de CNNs han propuesto incluir, además, un extenso número de capas convolucionales, como VGG16 y VGG19, donde el número representa la profundidad del modelo (Renza Torres & Ballesteros, 2023), buscando mejorar la extracción de características y la precisión. Por otro lado, otras iniciativas como el modelo MobileNet presentado en (Howard et al., 2017), han propuesto mejoras en el desempeño, basados en arquitecturas menos extensas y computacionalmente menos intensivas.

Otros enfoques han buscado optimizar el funcionamiento y reducir el tamaño de algunas arquitecturas de CNNs, aplicando técnicas como la eliminación de neuronas menos significativas y/o la cuantización de pesos y bias. En (Molchanov et al., 2017) y (Sarkar, 2020), se propone como método de optimización de CNNs, denominado poda (*pruning*). El *pruning* es una técnica que consiste en identificar y eliminar conexiones innecesarias en una red neuronal con el propósito de hacerla más simple. (Molchanov et al., 2017) se enfocó, en el concepto de transferencia de aprendizaje (*transfer learning*), con el fin de que redes previamente entrenadas puedan adaptarse a tareas especializadas y además se pueda reducir el uso de recursos como la memoria. (Sarkar, 2020) plantea por su lado una nueva técnica de poda de CNNs, basada en la técnica de poda incremental así logrando mayor precisión con menos entrenamientos previos, lo que conlleva a una reducción en tiempos que otras técnicas estudiadas anteriormente.

Considerando que el presente proyecto pretende abordar la evaluación de la confiabilidad de las CNNs a nivel de aplicación, resulta importante determinar los aspectos que se relacionan con este tema cuando se contempla el entorno de desarrollo. En (Kaur & Bahl, 2014) se brinda un panorama acerca de la confiabilidad, considerando las diferencias entre software y hardware. Para el caso del software, se debe tener presente que su complejidad tiende a ser alta debido a su carácter intangible y abstracto, lo que involucra que su comprobación y verificación no resulta fácil. Así mismo (Bengio & Haffner, 1998) presenta dos enfoques que permiten evaluar exhaustivamente la confiabilidad del software; el primero, se basa en la evaluación del plan de prueba, asegurando que el sistema contenga la funcionalidad especificada mientras que el segundo, se basa en la evaluación de la cantidad de errores y la tasa de detección y corrección.

En el Grupo de Investigación en Robótica y Automatización Industrial GIRA de la Universidad Pedagógica y Tecnológica de Colombia (UPTC) se han venido desarrollando diversos proyectos orientados a atacar nuevas tendencias en inteligencia artificial, teniendo como base, áreas de diseño y test de sistemas, robótica, diseño digital avanzado, biotecnología y aceleradores de hardware. En este contexto, el presente trabajo se encuentra enmarcado dentro de los proyectos “Desarrollo de una plataforma robótica móvil basado en FPGA para la enseñanza de robótica y sistemas embebidos” (SGI 2847) y “Reliability assessment of deep neural networks implemented in parallelized hardware platforms”. Este último proyecto está siendo ejecutado por el ingeniero Luis Ariel Mesa, y se pretende con este, apoyar su ejecución a través de la evaluación de la confiabilidad de una Red Neuronal Convolucional implementada en alto nivel. En este sentido, se pretende abordar solo el nivel de aplicación de la red neuronal, utilizando para el caso un entorno de desarrollo como Python o Matlab.

1.2. Estado del arte

Una de las técnicas para la evaluación de la confiablidad es la inyección de fallos, la cual consiste en introducir intencionalmente errores en el sistema para observar su comportamiento ante estas perturbaciones e identificar y cuantificar los modos de falla que no son detectables a través de pruebas tradicionales. En (Ruospo et al., 2020) y (Ruospo et al., 2021) se realizan estudios de inyección de fallos en CNNs mediante software utilizando dos tipos de representación de datos para los pesos, punto flotante y punto fijo. Uno de sus objetivos principales radica en identificar la combinación óptima entre tipo de datos, reducción de bits y confiabilidad. Los resultados de estos estudios indican que, para una CNN, el uso de datos en punto fijo proporciona la mejor relación entre uso de memoria y confiabilidad. En la figura 4 se observa de manera general el procedimiento de inyección de fallos hecho por (Ruospo et al., 2021).

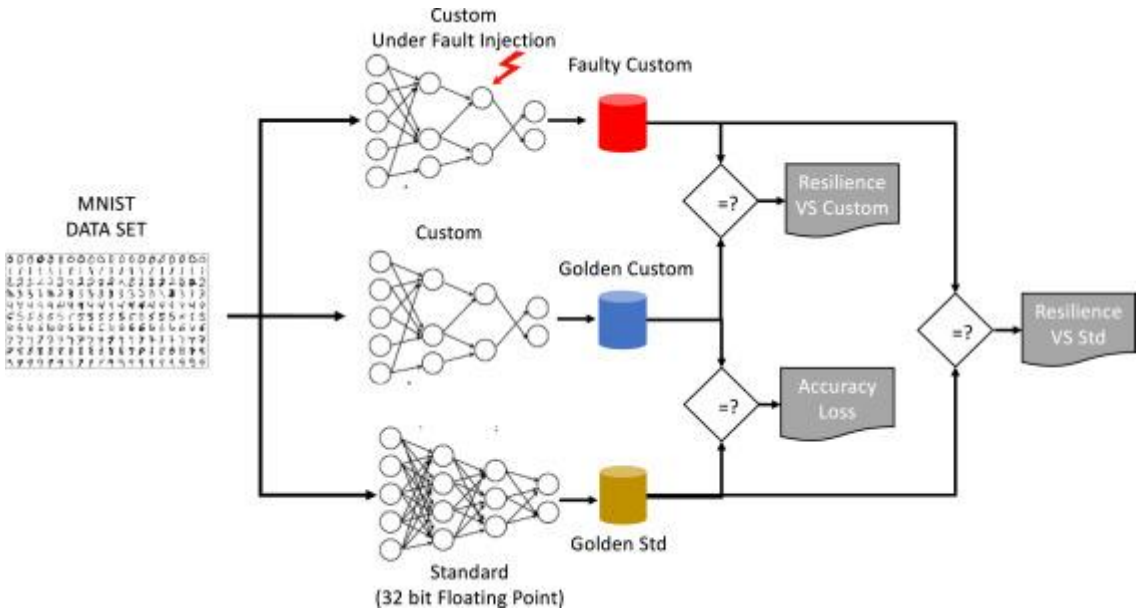

Figura 4. Escenario de inyección de fallos propuesto en el trabajo de (Ruospo et al., 2021)

En la literatura se encuentra disponibles algunas herramientas para evaluar la confiablidad de CNNs implementadas sobre lenguajes de alto nivel como Python, que emulan fallas de hardware y software. Estas herramientas se basan en inyectar errores en los modelos de aprendizaje automático con el fin de detectar potenciales vulnerabilidades. La inyección de fallos se puede realizar antes o durante el proceso de inferencia, específicamente en los operadores de aprendizaje. Algunas de las herramientas usadas con en este fin, son **TensorFI, PyTorch, ARES y Fidelity.**

Donde TensorFI es una herramienta de inyección de fallos (FI) altamente configurable, caracterizada por su flexibilidad, facilidad de uso y portabilidad. Esta herramienta se integra de manera sencilla a TensorFlow con el propósito de evaluar la resiliencia de CNNs ante posibles fallos (Chen et al., 2020). Otra herramienta a mencionar para esta tarea es la librería PyTorchFI, la cual permite a los usuarios realizar perturbaciones en las redes neuronales pesos y/o neuronas en operaciones convolucionales de una DNN de forma dinámica o estática (Mahmoud et al., 2020).

Con estas herramientas también se han desarrollado frameworks los cuales permiten, mejorar el rendimiento y desempeño de la inyección de fallos en una red neuronal convolucional, uno de estos marcos de trabajo llamado SCI-FI (Smart, aCcurate and unIntrusive Fault-Injector), presenta como se puede a llegar a reducir los tiempos de ejecución de fallos, empleando dos técnicas distintas, que se pueden llegar a emplear por separado o una combinación de ambas. La primera técnica llamada **Delayed Started**, consiste en guardar la ejecución completa de la Golden Network, y con ello si se desean aplicar fallos desde una capa determinada de la red, no es necesario ejecutar nuevamente toda la red, por el contrario, se ejecuta desde la capa donde se inyectaron el lote de fallos (Gavarini et al., 2023), un esquema de la técnica en ejecución se observa en la figura 5.

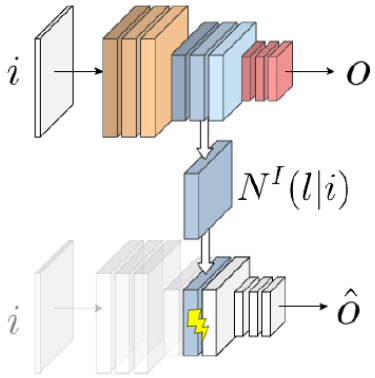

Figura 5. Uso de la técnica Delayed Started sobre una red neuronal convolucional. (Gavarini et al., 2023)

La segunda técnica llamada **Fault Dropping,** se refiere a la eliminación de fallos que podría ocurrir dentro de la red en plena ejecución, como ocurre cuando se usa la función de activación ReLU, la cual previene la propagación de pesos negativos. Este método consiste en ir comparando la Golden Network, con la Faultable network, capa por capa y si en algún momento los mapas de características de salida de las dos redes llegan a ser iguales, inmediatamente se detiene la inferencia, ya que se concluye que los fallos no van a tener impacto a la salida de la red neuronal (Gavarini et al., 2023). Una visualización de la técnica se observa en la figura 6.

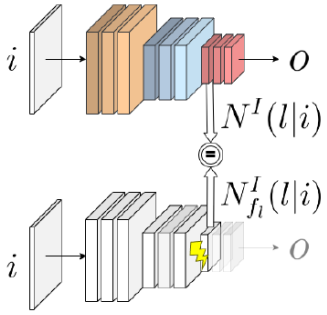

Figura 6. Uso de la técnica Fault Dropping sobre una red neuronal convolucional.(Gavarini et al., 2023)

2. IDENTIFICACIÓN DEL PROBLEMA

No obstante, a medida que el uso de sistemas basados en CNNs se ha expandido hacia aplicaciones críticas y entornos de seguridad, surge una creciente preocupación en torno a su confiabilidad y robustez. A pesar de sus éxitos, los modelos de aprendizaje profundo, incluyendo las CNNs, pueden enfrentar desafíos considerables en términos de su capacidad para mantener un rendimiento consistente y preciso en condiciones diversas y desafiantes.

Diversos factores pueden comprometer la confiabilidad de las CNNs, un ejemplo de ello es la presencia de datos no deseados, como errores en los datos de entrenamiento o incluso imprecisiones en los pesos, estos pueden desencadenar fallos graves en las predicciones, con posibles repercusiones significativas en aplicaciones críticas, como diagnósticos médicos erróneos o la navegación autónoma de vehículos. Además, los errores en el proceso de inferencia pueden derivar de diversos factores, que van desde fallas en el software hasta vulnerabilidades en el hardware utilizado(Hernández et al., 2022).

En este contexto, la investigación en torno a la confiabilidad y robustez de las CNNs se ha vuelto esencial para garantizar su aplicabilidad en aplicaciones de misión críticas. Abordar estos desafíos requiere no solo el desarrollo de modelos más precisos y resistentes, sino también la implementación de estrategias de evaluación y validación exhaustivas que permitan identificar y mitigar posibles debilidades. En última instancia, el avance en la comprensión y mejora de la confiabilidad de las CNN no solo contribuirá a su desempeño en tareas actuales, sino que también abrirá nuevas oportunidades para su implementación en campos donde la seguridad y la precisión son cruciales.

2.1. Pregunta de investigación.

¿Como evaluar la confiabilidad de una CNN implementada sobre un entorno de desarrollo basado en python considerando la arquitectura de red y la representación de datos?

2.2. Hipótesis

La inyección controlada de fallos a una red neuronal previamente entrenada tendrá un impacto significativo en su confiabilidad y rendimiento en aplicaciones de misión crítica. Se anticipa que la manipulación deliberada de factores como los pesos de las neuronas más sensibles resultarán en una disminución en la precisión y estabilidad de la red neuronal.

Adicionalmente, se formula la hipótesis de que tanto la magnitud como la naturaleza de los fallos inyectados influirán en el grado de deterioro de la confiabilidad de la red neuronal. Se espera que los resultados de este estudio exhiban diferencias sustanciales en el rendimiento entre la red con inyección de fallos y la red sin fallos, lo cual respaldaría la importancia de investigar la confiabilidad en contextos de aplicaciones críticas.

En última instancia, se proyecta que los descubrimientos de esta investigación proporcionen evidencia empírica que respalde la necesidad de desarrollar estrategias de mitigación de riesgos y mejorar la confiabilidad de las redes neuronales utilizadas en aplicaciones de misión crítica. Esto, a su vez, contribuirá al avance en la comprensión de los mecanismos de tolerancia a fallos en la inteligencia artificial.

3. JUSTIFICACIÓN

En este contexto, la presente propuesta de trabajo de grado tiene como objetivo, evaluar a alto nivel, la confiabilidad de una red neuronal mediante la inyección controlada de fallos y comparar su comportamiento con una red idéntica sin fallo. Para el caso se contempla la posibilidad del uso de entornos basados en Python como TensorFlow y Pytorch entre otros.

La creación de un setup de pruebas específico para una red neuronal convolucional (CNN), sería una gran herramienta para mejorar la robustez de cual red neuronal, de dicha arquitectura. Un entorno de pruebas bien diseñado permitirá una evaluación minuciosa de todos los componentes que conforman la red, desde el dataset, la arquitectura y características internas de la misma. Este setup proporcionará una plataforma sistemática para realizar pruebas exhaustivas en cada parte de la red neuronal, asegurando que cada aspecto, desde la integridad de los datos hasta la robustez de los parámetros del modelo, sea cuidadosamente analizado.

Mediante un proceso riguroso de inyección de fallos en una red neuronal previamente entrenada, se podrá comparar su rendimiento con el mismo modelo sin fallos. Este enfoque permitirá explorar diversos tipos de fallos, como la alteración de los pesos de las neuronas más sensibles, y evaluar su impacto. A través de un análisis detallado de métricas de rendimiento, como la precisión, se investigará cómo estos fallos afectan la confiabilidad de la red neuronal y su capacidad para mantener un rendimiento aceptable.

Los resultados de este estudio proporcionarán información valiosa para mejorar la confiabilidad de las redes neuronales en aplicaciones de misión crítica. Además, se espera que este trabajo contribuya al avance en la comprensión de los mecanismos de tolerancia a fallos en las redes neuronales y fomente el desarrollo de estrategias de mitigación de riesgos en sistemas basados en inteligencia artificial.

4. OBJETIVO GENERAL

Diseñar un set-up para la evaluación de la confiabilidad de una CNN implementada sobre un entorno basado en Python cuando se considera la arquitectura de la red y la representación de datos.

4.1. OBJETIVOS ESPECÍFICOS

1. Identificar tipos de arquitecturas de CNN, su representación de datos y mecanismos de evaluación de confiabilidad para sistemas de alto nivel.

2. Desarrollar un entorno de prueba que incluya implementación, entrenamiento y herramientas para evaluar la confiabilidad de las CNN.

3. Ejecutar pruebas para evaluar la confiabilidad de algunas CNN seleccionadas y recopilar los datos.

4. Analizar y validar los resultados obtenidos.

5. METODOLOGÍA

Este proyecto busca aplicar el conocimiento teórico y experimental para abordar un problema real relacionado con la confiabilidad de las CNNs, clasificándolo como investigación aplicada y experimental. En esta investigación, se realizarán pruebas controlados para evaluar la confiabilidad de las CNNs aplicándoles inyección controlada de fallos. Se analizará cómo la confiabilidad y el rendimiento de la red varían en función de diferentes condiciones de fallos. Esta metodología experimental nos permite establecer relaciones causales y medir de manera precisa el efecto de las variables independientes, en este caso, la inyección de fallos, sobre las variables dependientes, que son la confiabilidad y el rendimiento de la CNN.

6. ACTIVIDADES

Las actividades que permitirán la ejecución de los diferentes objetivos específicos de planten en la presente sección.

a. Para cumplir con el primer objetivo específico, “Identificar tipos de arquitecturas de CNN, su representación de datos y mecanismos de evaluación de confiabilidad para sistemas de alto nivel.”, se plantean las siguientes actividades:

1. Investigar los diferentes tipos de arquitecturas de redes neuronales existentes.

2. Indagar sobre la representación de datos que se pueden presentar en las redes neuronales y mirar sus efectos.

3. Revisar los diferentes métodos que permitan evaluar la confiabilidad de las redes neuronales convolucionales.

b. Para cumplir con el segundo objetivo específico, “Desarrollar un entorno de prueba que incluya implementación, entrenamiento y herramientas para evaluar la confiabilidad de las CNN”, se plantean las siguientes actividades:

4. Implementar 3 tipos de arquitecturas de red (Alexnet, Vgg, Lenet5).

5. Realizar el entrenamiento e inferencia de las diferentes arquitecturas elegidas.

6. Implementar el método “Pruning” en las arquitecturas ya entrenadas.

7. Investigar el formato con el que se entregan los datos de la red entrenada y realizar la visualización de estos.

c. Para cumplir con el segundo objetivo específico, “Ejecutar pruebas para evaluar la confiabilidad de algunas CNN seleccionadas y recopilar los datos”, se plantean las siguientes actividades:

8. Implementar un código que permita acceder a la información de cada capa y que permita modificar al menos un dato de estas.

9. Realizar las inferencias de las redes entrenadas Golden model y el modelo con fallas.

10. Recolectar la información arrojada en la inferencia de ambos modelos.

d. Para cumplir con el segundo objetivo específico, “Analizar y validar los resultados obtenidos”, se plantean las siguientes actividades:

11. Analizar los datos obtenidos y comparar los resultados obtenidos en el Golden model con los del modelo de fallas.

12. Concluir los resultados obtenidos.

5. CRONOGRAMA

Para lograr este objetivo, hemos diseñado una serie de actividades que se llevarán a cabo en etapas específicas del proyecto. Estas actividades están destinadas a abordar aspectos críticos de la confiabilidad de la red neuronal, incluyendo la evaluación de su rendimiento,

A continuación, presentamos una tabla1 de actividades que detalla los pasos clave de este proyecto, incluyendo fechas estimadas de inicio y finalización. Este plan se desarrollará a lo largo de un período determinado, y se espera que las actividades nos conduzcan a una evaluación completa y rigurosa de la confiabilidad de nuestra red neuronal.

Etapa	Actividad	2024															
		Mes 1				Mes 2				Mes 3				Mes 4			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
a	1	■	■														
	2			■	■												
	3					■											
b	4						■										
	5							■									
	6								■	■							
	7									■							
c	8										■	■					
	9												■				
	10												■	■			
d	11														■		
	12															■	■

Tabla1: Cronograma de actividades.

6. COSTOS DEL TRABAJO Y FUENTES DE FINANCIACIÓN

A continuación, se presenta una tabla2 que detalla los costos asociados con la realización y finalización del proyecto. Esta información es esencial para una planificación financiera efectiva y una gestión eficiente de los recursos:

RECURSO HUMANO	$/h	No. Horas	TOTAL	FUENTE FINANCIADORA
Investigador	$ 15.000,00	100	$ 1.500.000,00	Investigador
Director	$ 36.628,00	100	$ 3.662.800,00	Universidad Pedagógica y Tecnológica de Colombia
Asesor	$ 48.728,00	48	$ 2.338.944,00	Universidad Pedagógica y Tecnológica de Colombia
TOTAL TALENTO HUMANO	$ 100.356,00	248	$ 7.501.744,00	
COMPRA O ALQUILER DE MAQUINARIA Y EQUIPOS	COSTO UNITARIO	CANTIDAD	TOTAL	FUENTE FINANCIADORA
Computadores	$ 7.600.000,00	1	$ 7.600.000,00	Investigador
Software	$ -	1	$ -	Investigador
TOTAL MAQUINARIA Y EQUIPO	$ 7.600.000,00	2	$ 7.600.000,00	
FUNGIBLES	COSTO UNITARIO	CANTIDAD	TOTAL	FUENTE FINANCIADORA
Materiales de prueba	$ -	0	$ -	Investigador
Materiales del modelo	$ -	0	$ -	Investigador
Libros	$ -	0	$ -	Universidad Pedagógica y Tecnológica de Colombia
Papelería y otros	$ 80.000,00	15	$ 1.200.000,00	Universidad Pedagógica y Tecnológica de Colombia
TOTAL FUNGIBLES	$ 80.000,00	15	$ 1.200.000,00	
OTROS GASTOS	COSTO UNITARIO	CANTIDAD	TOTAL	FUENTE FINANCIADORA
Servicios públicos	$ 60.000,00	6	$ 360.000,00	Investigador
Viajes	$ 18.000,00	180	$ 3.240.000,00	Investigador
TOTAL OTROS GASTOS	$ 78.000,00	186	$ 3.600.000,00	Investigador
SUBTOTAL	$		19.901.744,00	
IMPREVISTOS (4 al 10%)	$		796.069,76	Investigador
COSTO TOTAL DEL PROYECTO	$		20.697.813,76	

Tabla2. Presupuesto del proyecto.

A continuación, se presenta la tabla3 de la distribución de los costos asociados con la realización y finalización del proyecto:

FUENTE	COSTO A CARGO	PORCENTAJE
Autor	$ 13.496.069,76	65,21%
Universidad	$ 7.201.744,00	34,79%
COSTO TOTAL PROYECTO	$ 20.697.813,76	100%

Tabla3: Distribución de los costos del proyecto.

Bibliografía

Bengio, Y., & Haffner, P. (1998). *Gradient-Based Learning Applied to Document Recognition*. *86*(11).

Chen, Z., Narayanan, N., Fang, B., Li, G., Pattabiraman, K., & DeBardeleben, N. (2020). Tensorfi: A flexible fault injection framework for tensorflow applications. *Proceedings - International Symposium on Software Reliability Engineering, ISSRE*, *2020-Octob*, 426–435. https://doi.org/10.1109/ISSRE5003.2020.00047

Gavarini, G., Ruospo, A., & Sanchez, E. (2023). SCI-FI: A Smart, aCcurate and unIntrusive Fault-Injector for Deep Neural Networks. *Proceedings of the European Test Workshop*, *2023-May*, 1–6. https://doi.org/10.1109/ETS56758.2023.10173957

Hernández, A. Z., Jesús, H., Santiago, J., & Lee, M. M. (2022). *Métricas de rendimiento para evaluar el aprendizaje automático en la clasificación de imágenes petroleras utilizando redes neuronales convolucionales . 2215*, 4624–4637.

Howard, A. G., Chen, B., & Wang, W. (2017). *MobileNets : Efficient Convolutional Neural Networks for Mobile Vision MobileNets : Efficient Convolutional Neural Networks for Mobile Vision Applications*. *April 2017*.

Kaur, G., & Bahl, K. (2014). *Software Reliability , Metrics , Reliability Improvement Using Agile Process*. *1*(3), 143–147.

Krizhevsky, B. A., Sutskever, I., & Hinton, G. E. (2017). *ImageNet Classification with Deep Convolutional Neural Networks*. https://doi.org/10.1145/3065386

LeCun, Y., & others. (1989). Generalization and network design strategies. *Connectionism in Perspective*, 143–155.

Mahmoud, A., Aggarwal, N., Nobbe, A., Vicarte, J. R. S., Adve, S. V., Fletcher, C. W., Frosio, I., & Hari, S. K. S. (2020). PyTorchFI: A Runtime Perturbation Tool for DNNs. *Proceedings - 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN-W 2020*, 25–31. https://doi.org/10.1109/DSN-W50199.2020.00014

Molchanov, P., Tyree, S., Karras, T., Aila, T., & Kautz, J. (2017). *P RUNING C ONVOLUTIONAL N EURAL N ETWORKS*. *2015*, 1–17.

Moreno, A. (2019). Autor: Álvaro Artola Moreno Tutor: José Antonio Pérez Carrasco. *Universidad de Sevilla*, 80. https://hdl.handle.net/11441/89506

Pérez Cerdeira, I. J. (2021). *Aceleración hardware para inferencia en redes neuronales convolucionales*.

Renza Torres, D., & Ballesteros, D. M. (2023). *Fundamentos de visión por computador utilizando aprendizaje profundo*. 1–166.

Ruospo, A., Bosio, A., Ianne, A., Sanchez, E., Torino, P., & Lyon, E. C. De. (2020). *Evaluating Convolutional Neural Networks Reliability depending on their Data Representation*. 672–679. https://doi.org/10.1109/DSD51259.2020.00109

Ruospo, A., Sanchez, E., Traiola, M., Connor, I. O., & Bosio, A. (2021). Microprocessors and Microsystems Investigating data representation for efficient and reliable Convolutional Neural Networks. *Microprocessors and Microsystems*, *86*(December 2020), 104318. https://doi.org/10.1016/j.micpro.2021.104318

Sarkar, S. (2020). *An Incremental Pruning Strategy for Fast Training of CNN Models*. 371–375.

COMENTARIOS DEL DIRECTOR

_____
_____
_____
_____
_____

VISTO BUENO DEL DIRECTOR:

_____

FIRMA PROPONENTE:

_____

_____