



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO
CARRERA DE INGENIERÍA ELECTRÓNICA

**DISMINUCIÓN DE RUIDO EN IMÁGENES SATELITALES EN EL SISTEMA DVB-
S2**

Trabajo de titulación previo a la obtención del

Título de Ingeniero Electrónico

AUTOR: ANGEL GUSTAVO LUGUAÑA SIMBAÑA

TUTOR: LUIS GERMÁN OÑATE CADENA

Quito - Ecuador
2021

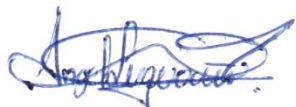
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Angel Gustavo Luguaña Simbaña con documento de identificación N° 1717644726, manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 20 de diciembre del año 2021

Atentamente,



Angel Gustavo Luguaña Simbaña

1717644726

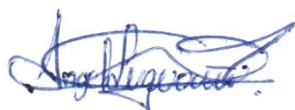
**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Angel Gustavo Luguaña Simbaña con documento de identificación No. 1717644726, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Artículo Académico: “ Disminución de ruido en imágenes satelitales en el sistema DVB-S2” el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 20 de diciembre del año 2021

Atentamente,



Angel Gustavo Luguaña Simbaña

1717644726

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Luis Germán Oñate Cadena con documento de identificación N° 1712157401, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISMINUCIÓN DE RUIDO EN IMÁGENES SATELITALES EN EL SISTEMA DVB-S2, realizado por Angel Gustavo Lugaña Simbaña con documento de identificación N° 1717644726, obteniendo como resultado final el trabajo de titulación bajo la opción Artículo Académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 20 de diciembre del año 2021

Atentamente,



Ing. Luis Germán Oñate Cadena MSc.
1712157401

Dedicatoria

Este trabajo se lo dedicó a mis padres y a mi hermana Noelia, que fueron mi apoyo incondicional en los momentos difíciles, sus consejos y sus reprimendas. A mis abuelitos que desde el cielo siempre me cuidan y bendicen día a día.

Angel Lugaña

Agradecimiento

A Dios, Santa Anita y San Joaquín, ya que sin su bendición nada de esto sería posible.

A mis padres por siempre confiar en mi a pesar de muchas adversidades.

A mi hermana Noelia por no dejarme solo y siempre motivarme a seguir adelante por cumplir mi objetivo.

A los docentes que a lo largo de mi trayectoria como estudiante me impartieron la mejor formación académica.

Y finalmente al Ingeniero Luis Oñate, por su guía, apoyo y paciencia en el desarrollo de este trabajo de titulación.

Disminución de ruido en imágenes satelitales en el sistema DVB-S2

Luguaña Simbaña Angel, Oñate Cadena Luis

Universidad Politécnica Salesiana

Quito, Ecuador

aluguana@est.ups.edu.ec

lonatec@ups.edu.ec

Resumen— En este artículo se realiza la simulación del estándar DVB-S2 (Transmisión de video digital por satélite versión 2) implementado en las herramientas de simulación Simulink y Matlab para la eliminación de ruido en imágenes satelitales, utilizando una red neuronal convolucional. Se empleó la modulación QPSK (Modulación por desplazamiento de fase en cuadratura) dentro del diagrama de bloques del estándar DVB-S2, para agregar ruido a las imágenes, siendo este valor modificado en el canal AWGN de -2.75 [dB]. Para la determinación del nivel de ruido se comparan las imágenes de salida del sistema DVB-S2 con y sin filtro CNN (Red neuronal convolucional). Las redes neuronales convolucionales poseen un rendimiento superior en muchas tareas visuales, como clasificación y detección de objetos. Mediante el entrenamiento de la CNN se obtuvo una mejora del 6.49%, esto representa que con el uso de CNN se elimina de manera sustancial el ruido en imágenes.

Abstract— In this article, the simulation of the DVB-S2 standard (digital video broadcasting by satellite version 2) implemented in the simulation tools Simulink and Matlab for the elimination of noise in satellite images is carried out, using a convolutional neural network. QPSK (Quadrature Phase shift keying) modulation was used within the block diagram of the DVB-S2 standard, to add noise to the images, this value being modified in the channel AWGN of -2.75 [dB]. For noise level checking the output images of the DVB-S2 system are compared with and without CNN (Convolutional neural network) filter. Convolutional neural networks have superior performance in many visual tasks, such as classification and object detection. Through CNN training an improvement of 6.49% was obtained, this represents that with the use of CNN noise in images is substantially eliminated.

I. INTRODUCCIÓN

La demanda de servicios de televisión satelital va en aumento, por este motivo se busca la transmisión de

información de mejor calidad [1]. Uno de los problemas que presentan los satélites es la presencia de ruido, para lo cual se implementan técnicas para la eliminación de este en las imágenes con el propósito de recuperar una imagen con menor ruido, recuperando así imágenes semejantes a las originales [2]. Se emplea el modelo de transmisión de video digital por satélite DVB-S2 con un sistema de corrección de errores FEC (Corrección de errores hacia adelante), basado en LDPC (Códigos de verificación de paridad de baja densidad) con códigos BCH (Bose Chaudhuri Hocquenghem) y con ACM (Codificación y modulación adaptables) cuya función es de optimizar el canal [1]. Las aplicaciones de las redes neuronales artificiales son diversas que van desde la realización e implementación de filtros, reconocedores de plantas de control, predictores, procesamiento de imágenes, chatbots, entre otras [3].

En el artículo se desarrolla una técnica para la eliminación del ruido en imágenes que al pasar por un satélite la calidad de la imagen se reduce. Con la ayuda de redes neuronales se disminuye el ruido que se presenta en el canal de comunicación de los sistemas satelitales [1]. Este modelo utiliza QPSK junto con la codificación convolucional RS (Reed Solomon), para la transmisión de señales por transponder del satélite [4]. Se especifica el nivel de ruido para reducir los diferentes tipos de ruidos utilizando redes neuronales convolucionales, los niveles son seleccionados de acuerdo con el nivel de ruido de la imagen de entrada [5]. El estándar DVB-S2 es implementado por el software Matlab/ Simulink, además se realiza la programación de la red neuronal convolucional para generar la corrección del ruido obtenido en la imagen de salida del satélite DVB-S2.

En este artículo se tiene las siguientes secciones: en la sección II se describe el marco metodológico, presentando los fundamentos teóricos del modelo DVB-S2, redes neuronales y la disminución de ruido, la sección III muestra el diseño y la simulación del estándar DVB-S2 mediante el software Simulink y la implementación del algoritmo de la red neuronal convolucional en el software Phyton, en la sección IV se muestra los resultados obtenidos y en la sección V se presenta las conclusiones.

II. MARCO METODOLÓGICO

A. Modelo DVB-S2

Este modelo posee un adaptador de flujos de entrada flexible en el cual se puede operar con diferentes formatos de entrada, maneja un rango de $\frac{1}{4}$ hasta $\frac{9}{10}$ de tasa de código FEC. Se han desarrollado diversos tipos de arquitecturas de este modelo a lo largo de los últimos años, esto con base a los principales modulaciones usadas en diversas comunicaciones por satélite [1]. DVB-S2 tiene como objetivo proporcionar servicios directos al hogar cuyos receptores reciben emisiones adicionales de mayor capacidad y calidad, soporta 4 tipos de modulaciones, las cuales se clasifican en: modulaciones para aplicaciones QPSK y 8 PSK, las modulaciones que necesitan transpondedores no lineales 16 APSK y por último las que requieren transpondedores semilineales 32 APSK[4].

El estándar DVB-S2 utiliza códigos BCH basado en LDPC para la corrección de errores FEC, que utiliza una codificación y modulación adaptable ACM y el esquema de modulación proporciona señales piloto para la portadora de regeneración de señales [4].

B. Redes Neuronales Convolucionales (CNN)

Es un método de aprendizaje discriminativo simple que aprende una función de mapeo directo para modelar la imagen a priori implícitamente en lugar de modelar imágenes previas explícitamente. El uso de CNN para eliminar ruido de imágenes se remonta a una red de cinco capas con linealidad sigmoide. Posteriormente surgen los métodos basados en el codificador automático. El aprendizaje discriminativo simple ha mostrado un mejor rendimiento que el aprendizaje discriminativo guiado por inferencia MAP (Máxima A posteriori); sin embargo, los métodos de aprendizaje discriminativos existentes necesitan un aprendizaje de múltiples modelos para el manejo de imágenes con diferentes niveles de ruido. Las redes neuronales convolucionales deben poseer una velocidad rápida, flexibilidad y robustez [6].

Tienen un rendimiento bueno en muchas técnicas de procesamiento de imágenes, como clasificación y detección de objetos, además del aprendizaje discriminativo, la interpretabilidad del modelo es importante para las redes neuronales, sin embargo, este es uno de los problemas que tiene CNN siendo a la vez uno de los más grandes desafíos de esta técnica [7].

C. Disminución de ruido.

Varios tipos de ruidos no deseados puede influir en la visibilidad de las imágenes, dado que una imagen puede distorsionarse cuando es contaminada por el ruido durante los procesamientos de adquisición y transmisión, por este motivo la eliminación de ruido es una de las tareas más importantes en el procesamiento de señales e imágenes. Se han propuesto varios métodos para reducir el ruido en imágenes y así mejorar su calidad, como el filtro de contracción trivariante basado en Wavelets y basado en el espacio, el filtro bilateral articular, regresión de vectores de soporte, el enfoque bayesiano [6]. En tiempo real, el ruido puede ocurrir en cualquier etapa, que puede ser desde la captura hasta la recepción de la imagen [8].

Existen varios tipos de ruidos en una imagen, como el ruido gaussiano que surge por el sensor en mala iluminación, variaciones extremas de temperatura, envejecimiento de dispositivos electrónicos. Otro tipo de ruido los píxeles oscuros aparecen en regiones brillantes y aparecen píxeles brillantes en las regiones oscuras, este tipo de ruido se puede eliminar mediante el uso de sustracción de fotogramas e interpolación alrededor de píxeles oscuros/brillantes. El ruido de disparo es causado por fluctuaciones cuánticas estadísticas en los fotones detectados dado el nivel de exposición. El ruido uniforme es causado al cuantificar los píxeles de una imagen a un número de niveles discretos únicos [8].

III. DISEÑO

Se utilizó las siguientes herramientas: Simulink, Matlab 2020a, Sistema operativo Linux en la versión 2018.04 y Python 3.6.

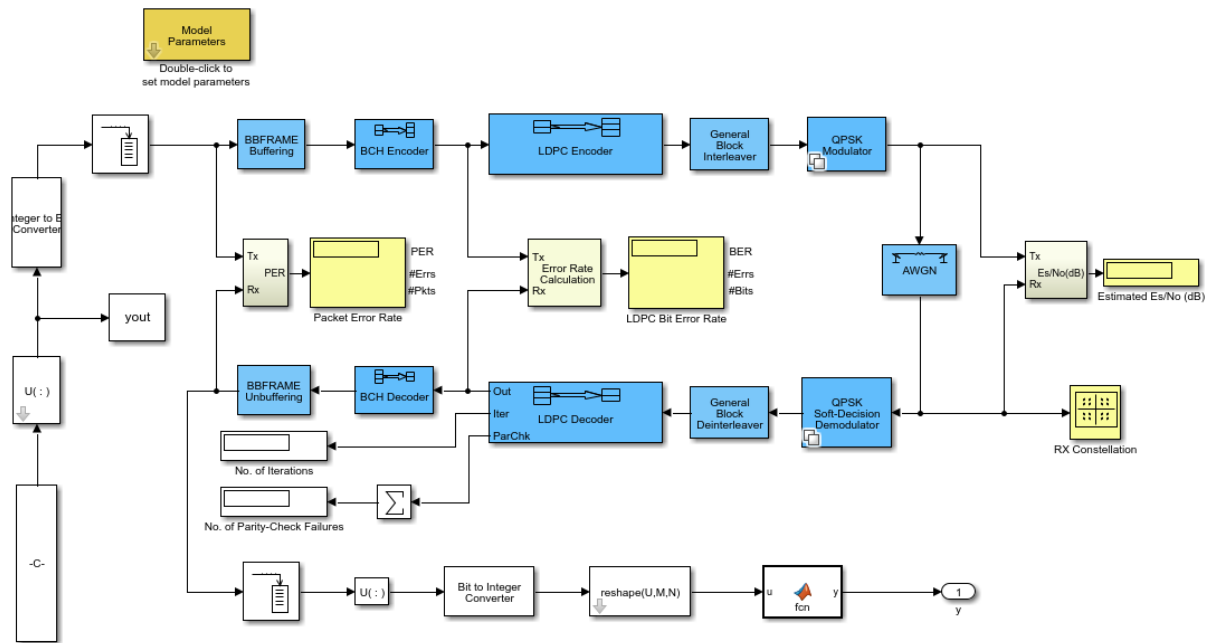


Fig. 1 Esquema de configuración de estándar DVB-S2 en diagrama de bloques en el software Ingeniería Simulink/Matlab

A. Modelo DVB-S2 en el software Simulink MATLAB

El diseño en bloques del modelo DVB-S2 se describe en la Figura 1, sus componentes principales son los siguientes bloques de procesamiento de señales:

Model Parametres: se utilizó el modelo QPSK $\frac{1}{4}$, donde cada símbolo representa una secuencia de dos bits así de las posibles combinaciones se obtienen cuatro símbolos de igual amplitud con diferente fase inicial, cuya separación de fase es de 90° entre símbolo [9].

BBFRAME Buffering: se utiliza para preparar BB (Curva de base) que sirven como macros de entrada para el BCH codificador, las tramas de datos de entradas son de 188 bytes o 1504 bit [4].

BCH Encoder: realiza un error de avance codificación de corrección, agrega bits redundantes que son utilizados para la corrección de errores por la transmisión a través de un canal inalámbrico propenso a errores [4].

LDPC Encoder: realiza un error interno codificación de corrección basada en el cálculo de bits de paridad y su inserción en la secuencia de bits de información. La codificación LDPC es el último bloque del procesamiento de corrección de errores [4].

General Block Interleaver: recibe la señal proveniente del bloque LDPC Encoder permitiendo la permutación de los símbolos de la señal de entrada, aumenta la capacidad de corrección con la creación de una distribución de errores más uniforme [10].

QPSK Modulator: realiza la modulación de la señal, donde se considera la frecuencia que se necesita o

desea para la transmisión de la señal, en función a la información que se desea transmitir se varía la señal a través de la modulación de modo proporcional [11].

AWGN: el bloque de ruido blanco gaussiano aditivo AWGN (Ruido gaussiano blanco aditivo) tiene como objetivo añadir ruido al sistema para hacer de referencia al canal de propagación de un sistema de comunicación real, se varía los niveles de ruido Gaussiano blanco en este bloque mediante el modo SNR, que en este caso se colocó -2.75 dB [10].

QPSK Soft-Decision Demodulator: realiza la demodulación de la señal, tiene como fin detectar el mensaje o la información original [11].

General Block Deinterleaver: este bloque reorganiza los elementos de su vector de entrada sin repetir u omitir ningún elemento [4].

LDPC Decoder: la arquitectura de estos decodificadores se clasifican en tres categorías, siendo estas completamente en paralelo, en serie y parcialmente en paralelo [12].

BCH Decoder: decodificador BCH.

BER: se compara el número de bits errados y el número de bits transmitidos [13], para este modelo se obtuvo una tasa de bit de errores de 0.0462, con una transferencia de bits de 4.102×10^7 , con un número de errores de 1.895×10^6 .

PER: es obtenido de la tasa de error de bits (BER) y el tiempo de colisión [14], para el sistema se obtuvo 0.896 de valor, con una transferencia de bits de 2.53210^4 , con un número de errores de 2.28610^4 .

La simulación diseñada por diagrama de bloques en Simulink cubre los aspectos básicos mecanismos de

procesamiento y transmisión de señales durante emisión de señal en el sistema DVB-S2.

El píxel se utiliza como datos de entrada y se procesa con corrección de errores, codificación, modulación, transmisión de canal inalámbrico, demodulación y decodificación de corrección de errores. Finalmente, la salida de la imagen se muestra en el extremo del receptor.

B. Algoritmo implementado en el software Python

La red neuronal convolucional (CNN) posee un aprendizaje profundo, son redes neuronales usadas para el procesamiento de imágenes, pueden aprender relaciones entrada-salida donde la entrada es una imagen, la convolución consiste en filtrar la imagen usando una máscara donde cada pixel de salida es una combinación lineal de los pixeles de entrada, los pesos son los valores en las máscaras que representan la conectividad entre las capas sucesivas [15].

Keras es una biblioteca, la cual proporciona una API de aprendizaje automático, posee bibliotecas para el procesamiento y manipulación de tensores o también llamados Backends como lo son TensorFlow, Theano y CNTK [15]. TensorFlow es un marco de software diseñado por el equipo de Google para la implementación del aprendizaje automático y profundo de una forma sencilla, combina el álgebra computacional y técnicas de optimización [16].

```
In [1]: 1 from tensorflow.keras.layers import BatchNormalization
2 from tensorflow.keras.layers import Conv2D
3 from tensorflow.keras.layers import Conv2DTranspose
4 from tensorflow.keras.layers import LeakyReLU
5 from tensorflow.keras.layers import Activation
6 from tensorflow.keras.layers import Flatten
7 from tensorflow.keras.layers import Dense
8 from tensorflow.keras.layers import Reshape
9 from tensorflow.keras.layers import Input
10 from tensorflow.keras.models import Model
11 from tensorflow.keras import backend as K
12 from skimage.measure import compare_ssim
```

Fig. 2 Importación de las librerías de TensorFlow Keras en el software Python.

En la figura 2 se puede observar parte del código realizado en el software Python que se requiere para la importación de varias librerías, estas son: BatchNormalization, conv2D, conv2DTranspose, LeakyReLU, Activation, Flatten, Dense, Reshape, Input, Model. Para la normalización de los lotes se importa BatchNormalization, pues se aplica una transformación que mantiene la salida media cercana a 0 y la desviación estándar de salida cercana a 1, funciona de manera diferente durante el entrenamiento y durante la inferencia. Se crean los filtros de cada capa con la ayuda de la librería CONV2D es una capa de convolución 2D, la cual crea un núcleo de convolución con la entrada de la capa que ayuda a producir un tensor de salidas [15]. Con la capa conv2DTranspose permite transponer la imagen directamente y forma parte del decodificador.

Se usa el optimizador Adam, que es un método de optimización estocástica que se basa en la estimación adaptativa de momentos de primer y segundo orden, es un método computacionalmente eficiente, posee pocos requisitos de memoria, invariante al reajuste diagonal de gradientes, con EPOCHS se define el número de veces que el algoritmo de aprendizaje funcionará en el entrenamiento, esto quiere decir, el número de iteraciones que se realizará [17].

C. Métricas de calidad de imagen

La calidad de imagen puede ser afectada por distorsiones durante la adquisición y procesamiento de la imagen. Se realiza una comparación entre la imagen original y la imagen resultante luego de las 400 iteraciones.

El MSE (Error cuadrático medio) de dos imágenes es una función de rendimiento para entrenar la red neuronal [18]. Es una medida de dispersión del error que se predice, se describe en la siguiente ecuación [19]:

$$MSE: \frac{\sum(\text{Valor real}-\text{Valor simulado})^2}{\text{Número de muestras}} \quad \text{Ec. (1)}$$

El pSNR (Relación señal pico a ruido) es una medida de calidad en la reconstrucción en el área de compresión de imágenes [20], es una métrica que mide la relación de señal a ruido, es medida en decibelios y se representa con la siguiente fórmula matemática [21]:

$$PSNR = 20 * \log_{10}\left(\frac{L^2}{MSE}\right) \quad \text{Ec. (2)}$$

Donde:

MSE: es el error cuadrático medio, que calcula los pixeles recibidos con respecto al original.

L: representa el rango de pixeles [21].

PSNR define la relación entre la energía máxima de la señal y la del ruido, usada para medir la calidad de una imagen reconstruida de la codificación [22].

El SSIM (Índice de similitud estructural) es una medida que cuantifica de forma numérica la calidad visual de una imagen para un observador [20], que se expresan en una escala de decibelios no lineales, se representa con la fórmula presentada a continuación [21]:

$$SSIM(x, y) = [l(x, y)]^\alpha [l(x, y)]^\beta [l(x, y)]^\gamma \quad \text{Ec. (3)}$$

IV. RESULTADOS

En la figura 3 se puede observar la imagen de entrada a la cual se aplicó la relación señal/ruido de -2.75 [dB] en el canal AWGN y el resultado de la imagen de salida con este ruido se muestra en la figura 4.



Fig. 3 Imagen de entrada.

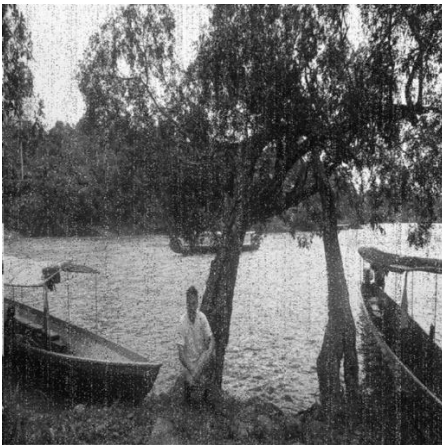


Fig. 4 Imagen de salida aplicado la relación señal/ruido de -2.75 [dB].

Con la imagen obtenida en la figura 4 se realizó la programación de la red neuronal convolucional. La red neuronal convolucional consta de 6 capas, el filtro máximo es de 128 bits y el menor de 16 bits, no se tomó el valor inferior a este por la pérdida de información, cabe mencionar que por cada filtro es una capa en cuanto subdivide a la imagen en píxeles.

El bucle que crea los filtros de cada capa son matrices de 3×3 para que de esta manera abarque toda el área dentro de la cual se crea las regiones. La función de activación que se utilizó es la sigmoide ya que introduce un grado de no linealidad en la CNN. Adicional se creó un método de normalización para el aprendizaje a través de imágenes dado que el valor entero de 8 bits es desde 0 a 255.

Como se generó la normalización el error con el que empezó la imagen es de 0.0849, este valor corresponde a la primera época de entrenamiento. Se optó por un entrenamiento de 400 épocas porque durante este procesamiento se fue reduciendo hasta encontrar el mejor número de época en la cual se obtuvo un error de 0.0200. Esto quiere decir que mientras más pequeño sea el error se obtendrá un mejor resultado como se muestra en la figura 5. Posterior al número de época de 400 el resultado se encontraba en valores aleatorios dando una respuesta no acorde a la estimada. Durante este entrenamiento se utilizó la pérdida de regresión MSE y el optimizador Adam. Esta función permite que luego de cada época realice un ajuste a la tasa de aprendizaje evitando que la red genere varias épocas con un mismo resultado sin haber mejoras al final, tal como se detalló anteriormente.



Fig. 5 Imagen aplicada la red neuronal convolucional (CNN).

En base a los resultados obtenidos se realizó la comparación entre la figura 3 y figura 4, para lo cual se empleó las métricas de calidad de referencia completa que se detallan a continuación. Se comparó el MSE, para esto se realizó la programación de la función en Python obteniendo un error de 0.0199, este valor representa la diferencia cuadrada promedio entre los valores de píxeles reales entre las 2 imágenes [18]. Se realizó el cálculo de pSNR donde se obtuvo un error de 16.9966, esto quiere decir que, si MSE tiene un valor bajo implica que no existe demasiado error en la reconstrucción de la imagen respecto a la original, con esto se tendrá un pSNR alto, y por lo tanto una mayor calidad de imagen [23]. Para el SSIM se importó la estructura *compare_ssim* a python donde se obtuvo un error de 0.3703 esto quiere decir que, evalúa la distorsión de dos imágenes en base a los valores de luminancia, contraste y textura que se calculan para cada pixel individual, sus valores oscilan entre 0-1 [21].

V. CONCLUSIONES

Los resultados obtenidos determinan que mediante la utilización de una red neuronal convolucional se disminuye el ruido en imágenes, con el entrenamiento de la CNN se obtuvo una mejora del 6.49% lo que permitió comparar los niveles de ruido mediante los algoritmos de referencia completa con la imagen de salida del estándar DVB-S2. Además, que un sobre entrenamiento de la CNN no es correcto para un análisis eficaz.

Mediante la programación en Python de las funciones MSE y pSNR se obtuvo los errores: 0.0199 y 16.9966. En cada época se modificó las ganancias de las matrices provocando que el valor de MSE sea más pequeño y este valor tienda a 0. En cambio, cuanto mayor sea el valor de pSNR más eficiente será el método. El error SSIM fue de 0.3703, este valor representa la diferencia de percepción entre las 2 imágenes.

El desarrollo de este estándar permitirá brindar apoyo para una investigación futura con una evolución de red neuronal convencional CNN diferente o el caso más robusto, permitiendo de esta manera mejorar el sistema para una optimización superior.

REFERENCIAS

- [1] L. Guerra, "Universidad Politécnica Salesiana Sede Quito," Tesis, pp. 1–65, 2017, [Online]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/14025/1/UPS-ST002992.pdf>.
- [2] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep CNN with batch renormalization," *Neural Networks*, vol. 121, pp. 461–473, 2020, doi: 10.1016/j.neunet.2019.08.022.
- [3] E. Esteban and N. Benavides, "Implementación de un sensor virtual mediante redes neuronales y filtrado adaptativo para un sistema hidráulico," 2018.
- [4] P. Baotic, M. Draganic, D. Bundalo, I. Kesegic, D. Tralic, and S. Grgic, "Simulation Model of DVB-S2 System," *Proc. ELMAR-2013*, no. September, pp. 227–231, 2013.
- [5] K. Isogawa, T. Ida, T. Shiodera, and T. Takeguchi, "Deep shrinkage convolutional neural network for adaptive noise reduction," *IEEE Signal Process. Lett.*, vol. 25, no. 2, pp. 224–228, 2018, doi: 10.1109/LSP.2017.2782270.
- [6] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-Based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, 2018, doi: 10.1109/TIP.2018.2839891.
- [7] Q. Zhang, Y. N. Wu, and S. C. Zhu, "Interpretable Convolutional Neural Networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 8827–8836, 2018, doi: 10.1109/CVPR.2018.00920.
- [8] K. S. Kumar, G. Sreenivasulu, and S. V. Rajan, "Block based SVD approach for Additive White Gaussian Noise level estimation in satellite images," *Proc. 10th INDIACom; 2016 3rd Int. Conf. Comput. Sustain. Glob. Dev. INDIACom 2016*, pp. 1464–1468, 2016, doi: 10.21884/ijmter.2016.3060.d28j2.
- [9] J. Acuña, "Simulación en Matlab de los métodos de acceso OFDMA y SC-FDMA para la tecnología LTE sobre un canal inalámbrico AWGN y un canal multitrayectoria con distribución Nakagami," p. 283, 2017.
- [10] W. Guapucal, "Universidad Politécnica Salesiana Sede Quito," Tesis, pp. 1–114, 2021, [Online]. Available: <http://dspace.ups.edu.ec/bitstream/123456789/5081/1/UPS-CYT00109.pdf>.
- [11] S. Ponce, "Diseño de un prototipo inhibidor de banda 4 (1700/2100 mhz) a través de radio definido por Labview," 2019.
- [12] K. W. Shin and H. J. Kim, "A multi-mode LDPC decoder for IEEE 802.16e mobile WiMAX," *J. Semicond. Technol. Sci.*, vol. 12, no. 1, pp. 24–33, 2012, doi: 10.5573/JSTS.2012.12.1.24.
- [13] L. Oñate, J. Gómez, and H. Lapo, "Análisis y comparación de ISDB-T utilizando modulaciones OFDM y CC-OFDM," *Ingenius*, no. 21, pp. 71–77, 2018, doi: 10.17163/ings.n21.2019.07.
- [14] S. Y. Shin, J. S. Kang, and H. S. Park, "Packet error rate analysis of ZigBee under interferences of multiple bluetooth piconets," *IEEE Veh. Technol. Conf.*, no. May, 2009, doi: 10.1109/VETECS.2009.5073765.
- [15] P. Loncomilla, "Deep learning: Redes convolucionales," *Recuper. https://ccc. Ina. mx/~ pgoomez/deep/presentations*, 2016.
- [16] Tutorials Point (I) Pvt. Ltd., "About the Tutorial Copyright & Disclaimer," *Tutorials Point*, pp. 1–13, 2017, [Online]. Available: <https://store.tutorialspoint.com>.
- [17] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.
- [18] J. D. Castillo Calderón, B. A. Solórzano Castillo, and J. A. Moreno Moreno, "Diseño de una red neuronal para la predicción del coeficiente de pérdidas primarias en régimen de flujo turbulento," *Ingenius*, no. 20, pp. 21–27, 2018, doi: 10.17163/ings.n20.2018.02.
- [19] K. Simba, "Estimación de parámetros de un motor trifásico de inducción a partir de mediciones en el arranque directo de corrientes transitorias del estator utilizando la herramienta de estimación de parámetros de MATLAB/SIMULINK," 2018.
- [20] M. Herrera, "Optimización binivel del parámetro de regularización con dependencia espacial del modelo de variación total generalizada para el filtrado de ruido de imágenes," 2017.
- [21] C. Mantilla, "Análisis del impacto de un ataque DoS en la Calidad de servicio de sistemas streaming multimedia en Redes SDN," 2021.
- [22] A. De la Cadena, "Estudio de las técnicas de visión estereoscópica para el estándar ISDB-T de televisión digital terrestre," p. 283, 2012.
- [23] F. Berrazueta, "Análisis de sistemas de videoconferencia en tiempo real sobre redes inalámbricas móviles AD HOC 802.11," 2008.