

UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



Uso de IA para estimación de calidad de
transmisión en señales ópticas.

POR

Jorge Javier Aravena Escobar

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil en Telecomunicaciones.

Profesor Guía

Gabriel Saavedra Mondaca

Concepción, 10 julio de 2024.

ÍNDICE

ACRÓNIMOS	IV
RESUMEN	V
AGRADECIMIENTOS	VI
1. INTRODUCCIÓN	7
2. REVISIÓN BIBLIOGRÁFICA	8
INTRODUCCIÓN	8
TRABAJOS PREVIOS	8
DISCUSIÓN	11
3. DEFINICIÓN DEL PROBLEMA	12
INTRODUCCIÓN	12
OBJETIVOS	12
OBJETIVO GENERAL	12
OBJETIVOS ESPECÍFICOS	12
ALCANCES Y LIMITACIONES	13
METODOLOGÍA	13
4. TEORÍA	15
4.1 MARCO TEÓRICO	15
5. DESARROLLO DE LA METODOLOGÍA	41
5.1 DESARROLLO DE SIMULACIÓN DE CANAL AWGN EN MATLAB	41
5.2 CREACIÓN DE BASES DE DATOS	41
5.3 CREACIÓN Y EJECUCIÓN DEL ALGORITMO CLASIFICADOR BINARIO CNN PARA LOS TIPOS DE MODULACIÓN.	43
5.4 CREACIÓN Y EJECUCIÓN DE ALGORITMOS CLASIFICADORES MULTICLASE CNN PARA DIFERENTES FORMATOS DE MODULACIÓN.	44
5.5 CREACIÓN Y EJECUCIÓN DE ALGORITMOS CLASIFICADORES MULTICLASE CNN QUE ESTIME LA CALIDAD ÓPTICA DE LA SEÑAL EN BASE AL SNR.	48
6. RESULTADOS	60
6.1 CLASIFICACIÓN BINARIA QAM Y PSK	60
6.2 CLASIFICADOR MULTICLASE CNN PARA DIFERENTES FORMATOS DE MODULACIÓN.....	64
6.2.1 Clasificador multiclase CNN para diferentes formatos de modulación de QAM.	64
6.2.2 Clasificador multiclase CNN para diferentes formatos de modulación de PSK.	66
6.3 CLASIFICADOR MULTICLASE CNN ESTIMADOR DE LA CALIDAD ÓPTICA DE LA SEÑAL EN BASE AL SNR.	69
6.3.1 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 16-QAM.	69
6.3.2 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 64-QAM.	72
6.3.3 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 256-QAM.	75
6.3.4 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 16-PSK.	78
6.3.5 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 64-PSK.	81
6.3.6 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 256-PSK.	84
7. ANÁLISIS DE RESULTADOS	88

7.1 ANÁLISIS CLASIFICADOR BINARIO QAM Y PSK.....	88
7.2 ANÁLISIS CLASIFICADOR MULTICLASE CNN PARA DIFERENTES FORMATOS DE MODULACIÓN	88
7.2.1 Análisis clasificador multiclase CNN para diferentes formatos de modulación de QAM.	88
7.2.2 Análisis clasificador multiclase CNN para diferentes formatos de modulación de PSK.	89
7.3 ANÁLISIS CLASIFICADOR MULTICLASE CNN ESTIMADOR DE LA CALIDAD ÓPTICA DE LA SEÑAL EN BASE AL SNR.	90
7.3.1 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 16-QAM.	90
7.3.2 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 64-QAM.	91
7.3.3 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 256-QAM.	91
7.3.4 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 16-PSK.	92
7.3.5 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 64-PSK.	92
7.3.6 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 256-PSK.	93
8. CONCLUSIONES	98
RESUMEN	98
CONCLUSIONES	98
TRABAJO FUTURO	100
9. BIBLIOGRAFÍA	101

Acrónimos

Mayúsculas

I.A.	: Inteligencia Artificial.
M.L.	: Machine learning.
I.P.	: Internet Protocol.
AWGN	: Additive White Gaussian Noise.
D.L.	: Deep Learning.
D.N.N.	: Deep Neuronal Network.
A.N.N.	: Artificial Neuronal Network.
C.N.N.	: Convolutional Neural Network.
S.N.R.	: Signal-to-Noise Ratio.
QoT	: Quality of Transmission.
O.P.M.	: Optical Performance Monitoring.
M.F.I.	: Modulation Format Identification.
C.P.U.	: Central Processing Unit.
G.P.U.	: Graphics Processing Unit.
I.Q.	: In-phase and Quadrature
S.V.M.	: Support Vector Machine
K-N.N.	: K- Nearest Neighbors
D.T.	: Decision tree
R.L.	: Reinforcement Learning
D.A.C.	: Digital to analog converted
A.D.C.	: Analog to digital converted
C.D.	: Chromatic dispersion
P.M.D.	: Polarization mode dispersion
P.N.	: Phase noise
S.P.M	:Self-phase modulation
X.P.M.	:Cross-phase modulation

Resumen

Las fibras ópticas y las redes ópticas son fundamentales para las comunicaciones globales. Hoy en día, su funcionamiento se logra mediante transmisores y detectores coherentes que utilizan algoritmos de procesamiento digital. Estos algoritmos, implementados en dispositivos como los Digital Signal Processors (DSP), permiten compensar las atenuaciones en las fibras ópticas.

Estos detectores permiten el uso de los Optical Performance Monitoring (OPM), que proporcionan un monitoreo continuo de los parámetros claves que afectan la calidad de una señal óptica. Dentro de los OPM se encuentran los Modulation Format Identification (MFI) los que logran una adaptabilidad dinámica en las redes ópticas.

El MFI es la base del desarrollo de esta memoria de título, que se enfoca principalmente en el desarrollo de algoritmos basados en inteligencia artificial, específicamente redes Convolutional Neural Network (CNN), los que clasifican información que estiman la calidad de señales ópticas. Para realizar esto se genera una base de datos de 217.800 imágenes que consisten en diagramas de constelación de señales ópticas producidos con el programa Matlab. Con la plataforma Google Colab se construyeron 9 códigos de redes CNN para diferentes tipos de clasificaciones.

Se creó una red CNN binaria para la clasificación de 2 tipos de modulación diferentes, phase-shift keying (PSK) y Quadrature amplitude modulation (QAM). También se crearon 2 algoritmos CNN multiclase para diferentes formatos de modulación, es decir, en un tipo de modulación, por ejemplo, QAM, clasificar por orden de modulación ya sea 16, 64 y 256 QAM. Finalmente se crearon 6 algoritmos de CNN multiclase para clasificar por SNR la calidad de cada tipo de señal modulada por medio de sus clases.

Se consiguieron precisiones generales por red de 100% para la CNN binaria, 92% para CNN 16/64/256-QAM, 91% para CNN 16/64/256-PSK, 73% para CNN 16-QAM, 88% para CNN 64-QAM, 58% para 256-QAM, 85% para CNN 16-PSK, 96% para CNN 64-PSK y 99% para CNN 256-PSK. Resultados que considerando las condiciones en las que se realizaron no son soluciones ni lineales ni esperadas para todos los casos, pero son óptimas de manera global. Esta investigación propone un acercamiento práctico/real a una mejor identificación, clasificación y toma de decisiones de formatos de modulación, teóricamente aplicables a dispositivos pertenecientes a los receptores coherentes de una comunicación óptica como los Digital Signal Processors.

Agradecimientos

A lo largo de todos estos años universitarios adquirí una cantidad enorme de experiencias más allá de lo que se podría encontrar en textos o los mismos conocimientos que ofrece la universidad, mucho de esto es gracias a mi papa y mama, quienes me forjaron muchos valores desde niño, quienes constantemente me apoyaban y motivaban en cualquier actividad que se me ocurriera incluso aunque no fuera en principio razonable. También un notable agradecimiento a mi hermanito quien es una persona increíble que siempre ha estado dispuesto a ayudarme. A todos mis familiares quienes cuando les contaba que estaba estudiando me expresaban constantemente que había tomado una buena decisión y se interesaban genuinamente en mí, pese a que no entendían casi nada de lo que les decía acerca de lo que estudiaba.

A todos los profesores que me hicieron clases en toda esta etapa estudiantil darle las gracias, de todos logre aprender cosas interesantes y útiles, en especial a mi profesor patrocinador Gabriel Saavedra quien siempre tenía consejos importantes que tome con humildad. A mis compañeros de carrera con quienes compartí momentos inolvidables tanto en trasnoches para certámenes o evaluaciones como para todas esas cosas extraprogramáticas típicas de un estudiante universitario.

A mis amigos y algunos conocidos quienes en varias ocasiones me daban palabras de aliento y me motivaban con pequeñas cosas que siempre sirvieron. Un agradecimiento a la Tuna y en concreto a la Tuna de la facultad de ingeniería de la universidad de Concepción quienes estuvieron y están presentes sobre todo en momentos especiales.

Un agradecimiento al financiamiento de Agencia Nacional de Investigación (ANID) a través del proyecto Fondecyt Regular 1231826, como tesis de pregrado año 2023.

1. Introducción

Las comunicaciones ópticas juegan un rol fundamental en las redes de comunicaciones de prácticamente cualquier tipo, estas se utilizan principalmente por su gran ancho de banda, baja latencia y su alta capacidad de anti-interferencia [1]. Además, conecta los servicios de capas superiores con los recursos físicos. Debido a que la creciente demanda de capacidad del tráfico de datos en las redes de comunicación ha crecido exponencialmente, ha provocado desde un aumento en el tráfico convencional por fibra óptica hasta la creación y utilización de nuevos servicios como las redes 5G y la comunicación en la nube, generando en consecuencia más complejidad en la red, el servicio y la gestión de recursos [1].

A nivel mundial se predijo un aumento del número total de usuarios de internet de 3,9 billones en 2018 a 5,1 billones en 2023, es decir, aproximadamente un 66% de la población mundial conectada a internet [2]. Es por esto que en los últimos años se han introducido conceptos como la transmisión coherente, modulación flexible y transceptores sintonizables, generando a su vez, un especial interés en el desarrollo de nuevas e innovadoras formas de mejorarla, como hacerla más flexible, asignar recursos donde correspondan y de manera inteligente.

Técnicas como el Optical Performance Monitoring (OPM) o el Modulation Format Identification (MFI) son útiles para la compensación de las complejidades de la red, pero no pueden adaptarse a diversas condiciones para satisfacer objetivos multitarea, lo que provoca aumentos en los costos operacionales generales de la red, además de no ser lo suficientemente eficientes en reconocer y detectar brechas en términos de seguridad de la red, debido a que las OPM proporcionan información continua sobre los parámetros ópticos por lo que cualquier ataque provoca cambios en la relación entre estos parámetros [3].

Bajo esta situación juega un papel importante las técnicas de Machine Learning (M.L.), subcampo de las inteligencias artificiales, las cuales son prometedoras soluciones a añadir adaptabilidad e inteligencia a los nodos de las redes ópticas, de una manera más sofisticada y menos dependiente del hardware, como lo es una mejorada estimación del Quality of Transmission (QoT), para su utilización en la viabilidad del trayecto luminoso o la supervisión de salud del mismo [4]. Es de vital importancia la elección de los algoritmos de M.L para el tipo de finalidad que se quiera lograr.

2. Revisión Bibliográfica

Introducción

Las referencias bibliográficas son en mayor parte artículos de investigación, los cuales sirven para dar un panorama más completo al tema abordado, puesto que no existen muchos trabajos similares que planteen de manera explícita la forma en que el tema se desarrollará más adelante. Además, se incluyen otros documentos útiles como libros, glosarios y páginas webs los cuales son de apoyo tanto para ejemplificar los aportes de las tecnologías relacionadas con sistemas de comunicación óptica como para poseer un vocabulario y herramientas más extenso en cuanto a las herramientas de M.L.

Trabajos Previos

En el mundo de las redes de comunicaciones ópticas existen medidores físicos de parámetros asociados a estos sistemas como el Optical line terminal (OLT) y el Optical time domain reflectometer (OTDR) cada uno para funciones específicas, siendo el más completo en grado y capacidad el OTDR [5]. 2 de los parámetros que en gran medida determinan el rendimiento de este último equipo son la resolución espacial que indica la distancia mínima entre dos puntos mensurables en la fibra, y varía desde unos pocos centímetros hasta decenas de metros. En esquemas simples en el dominio del tiempo, este parámetro depende del ancho del pulso, que a su vez tiene una relación directa con la relación señal-ruido (SNR) obtenida.

Aunque a lo largo de los años ha mejorado en características y funcionalidades, los que realmente se encargan de entregar una mejor calidad de señal son las técnicas de procesamiento de datos en la transmisión y recepción de un sistema de comunicación óptico, específicamente los detectores.

En un principio se tenían los detectores directos, los cuales como técnica se adaptaban bien a un formato de modulación binario, a medida que los sistemas trataban de emplear formatos de modulación espectralmente eficientes para aumentar su capacidad, la complejidad se hizo prohibitiva. Por lo que, la investigación se dirigió hacia enfoques más costosos, pero más avanzados como la detección coherente, puesto que el receptor lineal asociado no sufría los problemas de escalado de complejidad para formatos multinivel asociados a un receptor de detección directa, pese a esto la detección directa sigue siendo útil para comunicaciones de corta distancia, además de poseer un buen manejo en el consumo de energía [6]. La creciente globalización sobre todo en el aspecto tecnológico, conllevó a mayores flujos de datos y distancias más largas, se comenzaron a investigar y utilizar más ampliamente los detectores coherentes, los que, combinados con los DSP y sus algoritmos en el lado

receptor de la señal, consiguieron que la SNR de salida se pudiese estimar con buena precisión mediante simulaciones en el dominio del tiempo que consideran también los detalles del DSP. Los detectores coherentes a diferencia de los directos poseen: 1) Mejor sensibilidad, la que se puede utilizar para reducir el nivel de potencia del láser, para por ejemplo cambiar a un formato de modulación de mayor nivel para reducir el requisito de ancho de banda del componente, o un tener un margen de alta potencia para un enlace más largo. 2) Más tolerancia hacia los deterioros ópticos como la dispersión cromática. 3) Una mayor eficiencia espectral, que puede convertirse en un problema en el futuro. Con las continuas mejoras en tecnología, existe la posibilidad de que se utilice tecnología de detección coherente para las interconexiones dentro del centro de datos [7,8,9].

Un DSP genérico del lado receptor del sistema de comunicación posee varias funcionalidades entre ellas compensadores de desbalances de IQ, equalizadores, timing recovery, estimación de frecuencia, así como también portadoras de fase, y estimación de símbolo junto con decodificación. Estas funciones están formadas en su mayoría por técnicas de ortogonalización, variados filtros, algoritmos adaptativos y límites de decisión no rectangulares. Las tareas más importantes de estas funciones son compensar alguna ineficiencia, como las lineales y las no-lineales [9].

La estimación de la calidad de una señal óptica para un detector coherente es dependiente de múltiples factores desde pérdidas por ineficiencias físicas de la fibra, distancia y atenuaciones derivadas del proceso de conversión de señal. En estas últimas atenuaciones es donde se pone especial atención puesto que en esta área es donde juega un rol importante las propiedades del DSP, ya que, a mejor rendimiento de funcionalidades del DSP, se compensan más ineficiencias, lo que genera una mejor calidad de señal. Algunos estimadores de la calidad de una señal óptica en cuanto al SNR son Split-Symbol Moments Estimator, Maximum-Likelihood, Squared Signal-to-Noise Variance, Second- and Fourth-Order Moments y Signal-to-Variation Ratio [10].

Como se mencionó anteriormente las redes actuales suelen ser estáticas, usando técnicas como avanzados formatos de modulación, nuevas técnicas de multiplexación, transmisión flex-grid, o los reconfigurables optical add-drop multiplexer (OADM), las cuales son útiles, pero con un costo asociado a una mayor complejidad de la red. En la búsqueda de un mejor rendimiento general de la red, flexibilidad y eficiencia se encuentran técnicas, tales como el MFI y el OPM, las cuales ayudan a

identificar formatos de modulación y a el monitoreo con un trade-off entre la eficiencia espectral, calidad de la señal y la máxima distancia óptica (reach).

Actualmente existen investigaciones en las que se han aplicado el uso de machine learning en redes con técnicas de MFI y OPM en las señales ópticas, como, por ejemplo, los algoritmos Support vector machine (SVM), K- Nearest Neighbors (K-NN), Decision tree (DT), CNN, Reinforcement Learning (RL) entre otros.

En el trabajo realizado en [11] se implementaron los algoritmos CNN AlexNet, InceptionV1 y VGG16, que estiman el SNR en base a diagramas de constelación. Los cuales son algoritmos reconocidos eficientes y complejos que consumen altos recursos informáticos.

En términos generales estos se diferencian en el tipo de aprendizaje, etapa y el modo en que se implementan en el sistema de comunicación, interpretación de los datos y su precisión en cuanto a Control de red, gestión de recursos, monitoreo y supervivencia de la red [3].

Discusión

Debido al creciente tráfico de internet IP mundial y las diferentes maneras en las que se producen fluctuaciones de este mismo, producto de nuevas aplicaciones y servicios, es que la arquitectura general estándar estática de las redes ópticas no está a la altura de los cambios como, por ejemplo no poder compensar deficiencias como las atenuaciones no lineales, pese a los desarrollos de detectores coherentes los cuales actualmente y en el futuro cercano no serán suficientes para un óptimo rendimiento de la red [4]. Es por esto que se ve como una opción la utilización de algoritmos de M.L. para poder hacer más dinámicas, flexibles e inteligentes las redes ópticas, optimizando de mejor manera la demanda en el momento, además de prever en base al entrenamiento y testeo de las redes la manera en cómo se comportará la red y así tomar las medidas y decisiones necesarias para mejorar el rendimiento en ocasiones de manera automática, luego de haber pasado por el proceso de aprendizaje profundo.

Los algoritmos AlexNet, InceptionV1 y VGG16 utilizados en investigaciones de clasificación de imágenes de diagramas de constelación basados en SNR [11], en comparación al desarrollo de esta investigación son similares en que como todos son redes CNN, poseen capas de agrupamiento, convolución y totalmente conectadas, además poseen funciones de activación como ReLu, reducen las dimensiones de entrada de las imágenes y reducen la posibilidad de sobre entrenamiento (overfitting) añadiendo Dropout. En cuanto a las diferencias se tiene que los 3 algoritmos poseen diferentes configuraciones en cantidad y orden de todas las capas, poseen diferente cantidad de parámetros entrenables, poseen una parte offline y otra online, donde los entrenamientos de las redes se realizaron de manera offline, se utilizaron formatos de modulación más simples y se probaron en 3 canales diferentes incluyendo el Additive White Gaussian Noise (AWGN) [11]. Dentro de posibles mejoras se encuentra utilizar una base de datos más grande, utilizar formatos de modulación más complejos y el aprendizaje de los algoritmos en un entorno online, agregando más maleabilidad.

3. Definición del Problema

Introducción

La creciente demanda del tráfico de internet en conjunto de las variaciones constantes de parámetros debido a las nuevas tecnologías que ponen a prueba las arquitecturas de redes ópticas debido a su comportamiento estático frente a los nuevos servicios y aplicaciones que poseen un tráfico de datos con características cada vez más complejas [1]. Uno de los parámetros principales de las redes ópticas es la identificación de formatos de modulación, la cual usada en base a herramientas de machine learning, puede mejorar aspectos de la red como por ejemplo la flexibilidad, adaptabilidad, seguridad, reducción de costos de operación y mejor eficiencia general de la red [3].

Objetivos

Objetivo general

Implementación de algoritmos de machine learning CNN basados en inteligencia artificial utilizando diferentes formatos de modulación y base de datos de diagramas de constelación que estimen la calidad de señales ópticas en la plataforma Google Colab.

Objetivos específicos

1. Desarrollo de simulación de canal AWGN en Matlab.
2. Desarrollo de base de datos de 217.800 imágenes de diagramas de constelación para QAM y PSK.
3. Creación y ejecución de algoritmo clasificador binario CNN para los tipos de modulación.
4. Creación y utilización de algoritmos clasificadores multiclase CNN para diferentes formatos de modulación.
5. Creación y utilización de algoritmos clasificadores multiclase CNN que estimen la calidad óptica de la señal en base al SNR.

Alcances y Limitaciones

Se utilizará el programa Matlab, junto con el entorno de programación Google Colab, con los recursos de los servidores que dispone, principalmente sus tarjetas gráficas, además para las pruebas de clasificación se utilizarán formatos de modulación PSK y QAM con diferentes números de símbolos, cantidad máxima de símbolos y SNRs diferentes.

En cuanto a los recursos físicos se utilizará un notebook Asus ROG M16 y computadores del laboratorio de optoelectrónica para la realización de las pruebas en la universidad de concepción.

Metodología

Objetivo: Desarrollo de simulación de canal AWGN en Matlab.

Para alcanzar este objetivo, se utilizará el programa Matlab con funciones predefinidas como el AWGN, además del conocimiento de ramos de la carrera, en la que se creará un canal con ruido blanco gaussiano con un SNR dado, diferentes formatos de modulación, números de símbolos distintos y número máximo de símbolos a transmitir. Logrando como resultado diferentes diagramas de constelaciones con información importante acerca de cada combinación de variables mencionadas recientemente.

Objetivo: Desarrollo de base de datos de 217.800 imágenes de diagramas de constelación para QAM y PSK.

La creación de esta base de datos se logrará creando códigos de programación de Matlab para producir diferentes imágenes de constelaciones de los formatos QAM y PSK con la combinación de variables como el formato de modulación, ordenes de modulación, cantidad de símbolos diferentes, cantidad máxima de símbolos y el SNR de forma aleatoria, los cuales se almacenarán en archivos con un total de 217.800 imágenes en total.

Objetivo: Creación y ejecución de algoritmo clasificador binario CNN para los tipos de modulación.

La creación de este algoritmo se efectuará en el entorno de programación Google Colab, con base en algunos de sus algoritmos [10], en el cual se utilizarán 16.000 imágenes de la base de datos anteriormente creadas para el entrenamiento, validación y testeo. Para así, lograr clasificar si una modulación es QAM o PSK.

Objetivo: Creación y utilización de algoritmos clasificadores multiclase CNN para diferentes formatos de modulación.

Creación de 2 algoritmos que se realizarán en el entorno de programación de Google Colab, con una mayor complejidad, que clasificará diferentes ordenes de modulación tanto para QAM como para PSK a partir de 29.000 imágenes de la base de datos.

Objetivo: Creación y utilización de algoritmo clasificador multiclase CNN que estime la calidad óptica de la señal en base al SNR.

Creación de 6 algoritmos que se realizarán en el entorno de programación de Google Colab, con arquitecturas más compleja, clasificando imágenes en base al SNR de cada una, generando la estimación de cada clase asociada a su SNR a partir de 172.800 imágenes de la base de datos.

4. Teoría

4.1 Marco teórico

En esta sección se describirán diversos conceptos, aspectos generales y asociados de los contenidos a tratar, para la generación de algoritmos que permitan la clasificación de imágenes; para ello, es necesario un conocimiento básico de las comunicaciones ópticas, con especificaciones generales del funcionamiento de sus partes enfocadas en la identificación de formatos de modulación, estimación de bits de símbolos de constelaciones y el uso de la inteligencia artificial aplicada sobre un conjunto de base de datos con el objetivo de entrenar, validar y probar modelos en una red neuronal convolucional con las imágenes de constelaciones de modulaciones digitales, la cual permitirá la clasificación y estimación de calidad de una señal digital modulada.

Inteligencia Artificial (I.A.): Es un campo muy amplio que abarca muchas áreas del conocimiento relacionadas con el aprendizaje automático, aun así, una definición simple, aceptada y en el contexto de esta memoria es: Combinación de algoritmos planteados con el propósito de crear máquinas que presenten las mismas capacidades que el ser humano [13,14].

Machine Learning (M.L.): Traducido al español significa “aprendizaje automático”, se puede definir como un subcampo de la inteligencia artificial que proporciona a los computadores la capacidad de aprender sin ser explícitamente programados, es decir, sin que necesiten que el programador indique las reglas que debe seguir para lograr su tarea, sino que las hace automáticamente.

En otras palabras, el Machine Learning consiste en desarrollar para cada problema un “algoritmo” de predicción para un caso de uso particular. Estos algoritmos aprenden de los datos con el fin de encontrar patrones o tendencias para comprender qué nos dicen los datos y de esta manera construir un modelo para predecir y clasificar los elementos. Es importante tener en cuenta que, aunque todo M.L. es IA, no toda la IA es M.L. Un enfoque de las M.L. son las Deep Learning [13].

Deep Learning (D.L.): Es un subconjunto de machine learning, que es básicamente una red neuronal con tres o más capas. Estas redes neuronales intentan emular el comportamiento del cerebro humano, aunque lejos de igualar su capacidad. Estas le permiten "aprender" a partir de grandes cantidades de

datos. Aunque una red neuronal con una sola capa ya puede realizar predicciones aproximadas, las capas ocultas adicionales ayudan a optimizar y refinar la precisión [15].

A.N.N. (Redes neuronales artificiales): Son un subconjunto del M.L. que pertenecen a una parte central de los algoritmos de D.L. Su nombre y estructura están inspirados en el cerebro humano, imitando la forma en que las neuronas biológicas se envían señales entre sí. Se componen de capas de nodos que contienen una capa de entrada, una o más capas ocultas y una capa de salida. Cada nodo, o neurona artificial, se conecta con otro y tiene un peso y un umbral o límite asociado. Si la salida de cualquier nodo individual está por encima del valor umbral especificado, ese nodo se activa y envía datos a la siguiente capa de la red. De lo contrario, no se pasan datos a la siguiente capa de la red. [15]

D.N.N (Red neuronal profunda): Es una A.N.N. con varias capas ocultas entre las capas de entrada y salida. [16]

C.N.N. (Redes convolucionales neuronales): Son un tipo de redes neuronales jerárquicas cuyas capas convolucionales se alternan con capas de submuestreo, que recuerdan a las células simples y complejas de la corteza visual primaria. Estas varían en cómo se realizan las capas convolucionales, de submuestreo y en cómo se entrenan las redes, además reducen drásticamente la cantidad de parámetros que deben ajustarse. Se aprovechan los principios del álgebra lineal, en particular la multiplicación de matrices, para identificar patrones dentro de una imagen. Por lo tanto, las CNN manejan de manera eficiente la alta dimensionalidad de las imágenes en bruto. [15,16,17]

C.P.U. (Unidad de procesamiento central): Construida a base de millones de transistores, la CPU puede tener múltiples núcleos de procesamiento y comúnmente se la conoce como el "cerebro de la computadora". Es esencial para todos los sistemas informáticos modernos, ya que ejecuta las órdenes y los procesos que necesitan la computadora y el sistema operativo. La CPU también es importante para determinar la rapidez a la que pueden ejecutarse los programas, desde navegar por Internet hasta crear hojas de cálculo. [18]

G.P.U (Unidad de procesamiento gráfico): Diseñada para procesamiento paralelo, la GPU se utiliza en una amplia gama de aplicaciones, incluida la representación de gráficos y videos. Aunque

son más conocidas por sus capacidades en juegos, las GPU día a día se están volviendo más populares para su uso en producción creativa y en la utilización de variadas subáreas de la I.A. [19]

Google Colab: También conocido como “Colaboratory”, permite programar y ejecutar en Python en el navegador con ventajas como no requerir configuración, acceso a GPUs de estándar y alto rendimiento sin coste adicional, pero con ciertas limitaciones y permitir compartir contenido fácilmente. Facilita el trabajo para estudiantes investigadores de IA o científico de datos [20].

Matriz de confusión: Representa el número de clasificación correcta, así como la errónea, incluyendo dónde se equivoca.

SNR: La relación señal-ruido (SNR) es una medida que compara el nivel de la señal deseada con el nivel de ruido de fondo. Esta se define como la relación entre la potencia de la señal y la potencia del ruido, a menudo expresada en decibeles [21].

Constelación: Un diagrama de constelación o constelación es una representación de una señal modulada por un esquema de modulación digital como la modulación de amplitud en cuadratura o la manipulación por desplazamiento de fase. Muestra la señal como un diagrama de dispersión bidimensional en el plano complejo en instantes de muestreo de símbolos. En un sentido más abstracto, representa los posibles símbolos que pueden seleccionarse mediante un esquema de modulación dado como puntos en el plano complejo. Los diagramas de constelación medidos se pueden utilizar para reconocer el tipo de interferencia y distorsión en una señal [20].

OPM: La supervisión del rendimiento óptico OPM obtiene el rendimiento de la capa física relacionado con la señal óptica, los enlaces ópticos y los dispositivos. Proporciona los parámetros de rendimiento básicos de degradación y deterioro. Una vez obtenido el rendimiento subyacente, se puede estimar la QoT del trayecto luminoso de una fibra óptica para proporcionar información de referencia a la asignación de recursos y restauración del servicio [1].

MFI: La capacidad de los receptores de señales para identificar el tipo de señal, se conoce como identificación del formato de modulación (MFI), el que permite construir redes adaptables, eficientes y flexibles en las que el tipo de señal y el ancho de banda se determinan en función de las condiciones de la red [3].

DSP: Un digital signal processor toma señales del mundo real como voz, audio, vídeo, temperatura, presión o posición que han sido digitalizadas para manipularlas matemáticamente. Un DSP está diseñado para realizar funciones matemáticas como "sumar", "restar", "multiplicar" y "dividir" con gran rapidez. En el contexto de un sistema de comunicación óptica coherente están diseñados para ecualizar y mitigar atenuaciones de la fibra óptica como la Chromatic dispersion (CD), Polarization mode dispersion (PMD), Phase noise (PN) y atenuaciones no lineales como los efectos Kerr [22,23].

Canal: Medio por el cual viaja o se transmite información de señales que se intercambian entre emisor y receptor.

DAC: Digital to analog converted, convierte señales digitales en analógicas, por ejemplo, una secuencia de bits en corriente eléctrica.

Un sistema de comunicación óptica es un sistema de comunicación en el que se utiliza la luz como portadora para transmitir y recibir datos.

Con los avances tecnológicos se ha permitido la utilización de estos sistemas a gran escala de manera rápida para la comunicación global. Debido a que la velocidad de la luz es una de las mayores velocidades del mundo y sabiendo que la luz no se ve afectada por interferencias electromagnéticas, esta sería naturalmente un portador perfecto de señales de comunicación. Por lo tanto, la comunicación óptica ha sido un sistema atractivo para la transmisión de datos.

Otro factor interesante de los sistemas de comunicación óptica es el uso de las fibras ópticas, las cuales son el medio por el que se transportan las señales luminosas de un punto a otro. Al estar hecha de vidrio, es un medio relativamente barato, pequeño en sección transversal, resistente y flexible, además de su factor de distorsión pequeño que añade a los datos que se transmiten, lo que la convierte en un medio que compite (en variadas ocasiones es mejor) con los cables coaxiales y los hilos convencionales en muchas áreas de la comunicación como medio. Gracias a ello, los sistemas de comunicación por fibra óptica tienen una gran capacidad y muy pocas pérdidas de transmisión.

El enlace de comunicación por fibra óptica comparte similitudes con otros enlaces de sistemas de comunicación convencionales. La forma más básica consiste en un transmisor, un receptor y una fibra

óptica de vidrio o sílice como medio de transmisión [24]. A continuación, se presentará un sistema de comunicación digital básico, el que se describirá desde el punto de vista de un sistema de comunicación óptica.

Sistema de comunicación digital básico

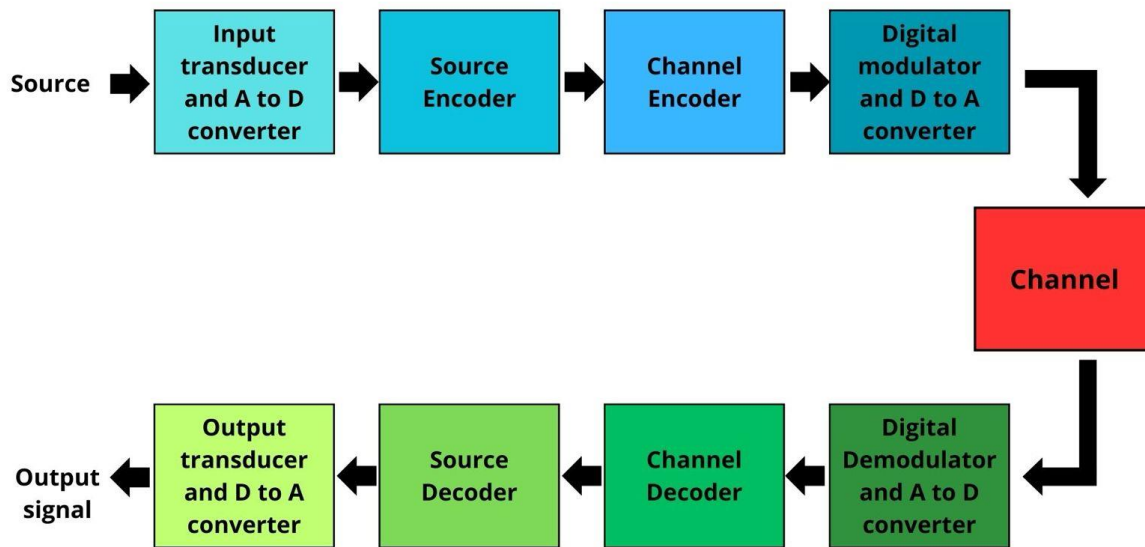


Fig. 1 Elementos básicos de un sistema de comunicación digital. [34]

Empezando por los bloques de arriba que conforman la parte de transmisión se tiene:

Transmisión

Source: La fuente puede ser una señal análoga como por ejemplo una señal de audio o video o señal digital es discreta en el tiempo y tiene un número finito de caracteres de salida.

Input Transducer and A to D converted: Este bloque comúnmente llamado Analog to digital converted (A.D.C.) consiste en convertir una señal análoga a una digital, siendo esta última representada por una secuencia binaria de información.

Source Encoder: Se comprime la información a un mínimo de bits, lo cual ayuda a una optimo uso del ancho de banda.

Channel Encoder: Realiza la codificación para la corrección de errores. Durante la transmisión de la señal, debido al ruido en el canal la señal puede alterarse, por lo que para evitar esto, el channel encoder agrega algunos bits redundantes a los datos transmitidos.

Digital modulator and D to A converted: La señal es modulada por una portadora. Además, se convierte la señal de digital a analógica, la Digital to analog converted (D.A.C.) a partir de la secuencia digital de bits [25].

En una comunicación óptica las principales funciones del transmisor son:

- Codificar la información: con source encoder y channel encoder.
- Modulación: Crear una forma de onda capaz de propagarse en el canal.
- Multiplexar información para transmitir datos.

Un transmisor óptico está conformado por:

Bloques DSP: Maneja los mensajes de forma digital, este es el encargado de la codificación y formas de pulsos.

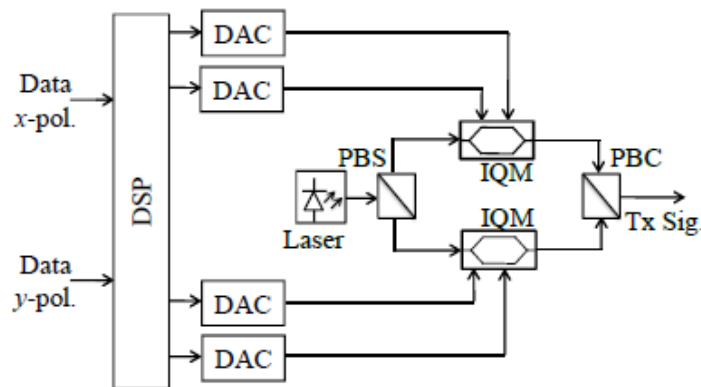


Fig. 2 Configuración de DSP basado en el transmisor. [26]

Se procesa en el DSP 2 secuencias de datos independientes de polarizaciones x e y, que puede incluir codificación, modulación, pre compensación de deficiencias lineales y no lineales y filtros pulse shaping. Luego estas señales procesadas son convertidas a señales analógicas por los 4 DACs correspondientes a fase y cuadratura de los componentes para las polarizaciones en x e y, las cuales son controladas por 2 moduladores IQM (IQ modulators). A las salidas de estos últimos se les añade un único laser polarization beam splitter (PBS). Después van a la entrada de un polarization beam combiner (PBC) y finalmente se transmiten a través de la fibra.

DAC: Transforma la señal digital en señal analógica de corriente/voltaje.

Fuente óptica: Emite luz, actúa como portadora

Moduladores: Modifica propiedades de la señal óptica proporcional a la señal eléctrica. Como, por ejemplo, intensidad, fase y polarización.

Channel: El canal o medio permite que la señal analógica se transmita desde el transmisor al receptor.

Un problema común en la transmisión de señales a través de cualquier canal es el ruido aditivo. Este es generado internamente por componentes como resistencias y dispositivos de estado sólido utilizados para implementar el sistema de comunicación, en ocasiones a esto se le llama ruido térmico.

Cuando estos ruidos usen la misma banda de frecuencia que la señal deseada, su efecto puede minimizarse mediante el diseño entre la señal transmitida y su demodulador en el receptor, además existen otros tipos de degradaciones que afectan a la señal que pueden encontrarse en la transmisión sobre el canal como la atenuación de la señal, la amplitud y la distorsión de fase. Algunos de estos ruidos pueden minimizarse aumentando la potencia en la señal transmitida, teniendo en cuenta las condiciones del equipo y las limitaciones prácticas del nivel de potencia de la señal transmitida.

Matemáticamente se puede representar el diseño del codificador y modulador del canal en el transmisor y del demodulador y decodificador del canal en el receptor. El modelo más simple para un canal de comunicación es el canal de ruido aditivo.

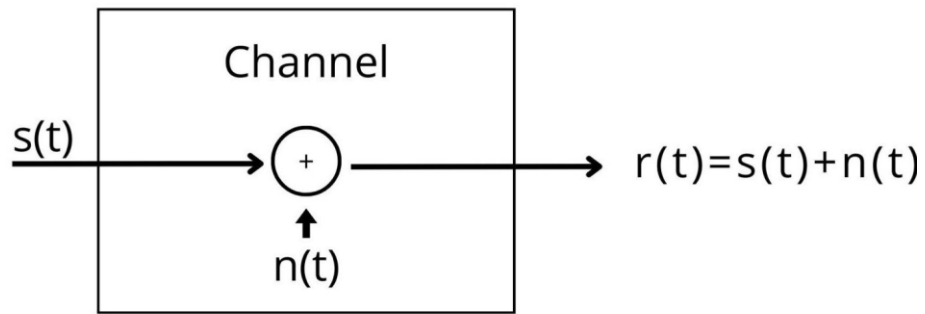


Fig. 3 Diseño matemático de canal de ruido aditivo. [34]

En este modelo la señal transmitida $s(t)$ está corrompida por un proceso de ruido aleatorio aditivo $n(t)$. Físicamente, el proceso de ruido aditivo puede provenir de componentes electrónicos y amplificadores en el receptor del sistema de comunicación o de interferencias encontradas en la transmisión (como en el caso de la transmisión de señales de radio).

Si el ruido es introducido principalmente por componentes electrónicos y amplificadores en el receptor, puede caracterizarse como ruido térmico.

Este tipo de ruido se caracteriza estadísticamente como un proceso de ruido gaussiano. Por lo tanto, el modelo matemático resultante para el canal es usualmente llamado “additive Gaussian noise channel” (canal de ruido gaussiano aditivo). Cuando la señal sufre atenuación en la transmisión a través del canal, la señal recibida es:

$$r(t) = \alpha s(t) + n(t)$$

Donde α es el factor de atenuación [25].

Este es un tipo de ruido de canal con propiedades similares al AWGN en cuanto a que sus funciones de densidad responden a una distribución normal.

En una comunicación óptica el canal está formado por:

- Fibra óptica: Guía la luz, esta deteriora algunas propiedades de la luz.

- Amplificadores: Amplifica la intensidad de la luz entrante.
- Compensadores de dispersión: Compensan óptica/electrónica/digitalmente la dispersión cromática de la fibra óptica.

En un sistema común y corriente de largas distancias como los “repetidos” entre cada fibra se encuentra un amplificador.

Los bloques de abajo conforman la parte de recepción, en esta parte se tiene:

Receptor

Digital demodulator and A to D converted: Este es el primer paso en el receptor. La señal recibida se demodula y se convierte de analógica a digital. Se reconstruye la señal.

Channel Decoder: El decodificador de canal, tras detectar la secuencia, realiza algunas correcciones de errores. Las distorsiones que puedan producirse durante la transmisión se corrigen añadiendo algunos bits redundantes. Esta adición de bits ayuda a la recuperación completa de la señal original.

Source Decoder: La señal resultante se digitaliza de nuevo mediante muestreo y cuantificación para obtener una salida digital pura sin pérdida de información. El decodificador de la fuente recrea la salida de la fuente.

Output Transducer and D to A converted: Es el último bloque que convierte la señal en la forma física original que tenía a la entrada del transmisor. Convierte la señal eléctrica en salida física (Ejemplo: altavoz).

Output signal: Salida que se produce después de todo el proceso. Por ejemplo: la señal de audio recibida.

En una comunicación óptica las principales funciones del receptor son:

- Transformar la señal óptica en una eléctrica y después digital.
- Procesa la señal digital (ecualización)
- Demodulación y decodificación: Con channel decoder y source decoder.

Un receptor óptico (coherente) está conformado por:

- Detector coherente
- A.D.C.
- Bloque de DSP.

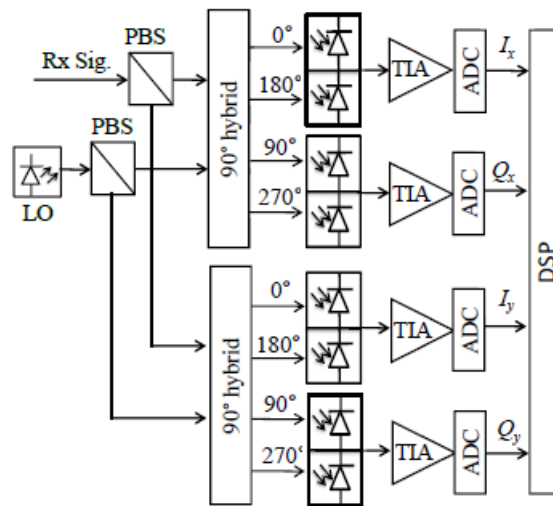


Fig. 4 Configuración de DSP basado en el receptor. [26]

Se usa generalmente para mapear linealmente la señal óptica entrante en cuatro señales eléctricas correspondientes a los componentes de campo en fase y en cuadratura de las dos polarizaciones. Se usan 2 PBSs y un par de 90° hybrid, uno para cada componente de polarización. La salida de estos bloques ópticos está dada por la siguiente ecuación:

$$\begin{bmatrix} I_x \\ Q_x \\ I_y \\ Q_y \end{bmatrix} \propto \begin{bmatrix} \text{Re}\{E_x E_{lo}^*\} \\ \text{Im}\{E_x E_{lo}^*\} \\ \text{Re}\{E_y E_{lo}^*\} \\ \text{Im}\{E_y E_{lo}^*\} \end{bmatrix} \quad (4.1)$$

Donde Re e IM representan la parte real e imaginaria de los componentes E_x y E_y , que denotan los campos eléctricos de la señal óptica de polarización dual entrante con componentes de polarización x e y. En su salida se detectan con 4 fotodiodos para cada bloque de los 90° hybrid cuya función es dividir la señal entrante en 2 salidas con una diferencia de 90° entre ellas. LO es el oscilador local que tiene la función de generar una frecuencia local para mezclarla con la señal óptica recibida y llevarla a una frecuencia más manejable para su procesamiento. Luego las señales son amplificadas con los transimpedance amplifiers (TIA) para después ser digitalizadas con los ADC y finalmente llegar al bloque DSP, donde la señal recibida se vió afectada por deficiencias del canal y ruido. Las deficiencias del canal pueden ser de efecto lineal como el CD o PMD o efectos no lineales de la fibra como el Self-phase modulation (SPM), o el Cross-phase modulation (XPM).

Las operaciones que realiza el DSP pueden variar, mientras que un procedimiento genérico de un DSP y sus funcionalidades pueden verse en el siguiente esquema:

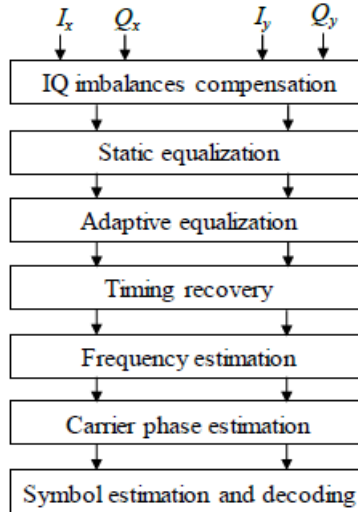


Fig. 5 Configuración de DSP basado en el receptor. [26]

IQ imbalances compensation: Como dice el nombre este compensa los desajustes de la fase/ganancia de los puertos I y Q de la señal recibida que pueden surgir en el transmisor, por ejemplo, de la polarización incorrecta del modulador IQ o en el receptor, como la imperfección en cualquiera de los

componentes de los híbridos ópticos de 90 fotodiodos equilibrados o TIA. Se pueden utilizar técnicas conocidas para la ortogonalización de dos vectores no ortogonales como la ortogonalización de Gram-Schmidt (GSOP).

Static equalization: Si bien, en principio la ecualización de todas las deficiencias lineales podría realizarse con un filtro digital, suele ser dividir el problema en un ecualizador estático y otro dinámico. Se usan filtros compensadores de la CD, como filtros Finite Impulse Response (FIR) y técnicas para ecualizadores en el dominio del tiempo, frecuencia y sub-banda. Además de Nyquist pulse shaping para satisfacer el criterio de Nyquist, generando que se pueda operar sin inter-symbol interference (ISI). También, se compensan deficiencias no lineales de la fibra como los efectos Kerr.

Adaptive equalization: Se utiliza para compensar las deficiencias dinámicas de los canales. Generalmente, un filtro FIR estructurado de entrada múltiple y salida múltiple (MIMO) de dos por dos se utiliza para estimar la matriz de Jones inversa del canal dinámico. Se emplean algoritmos de adaptación, como el constant modulus-algorithm (CMA), configuración MIMO 4x4 que permite la compensación del desequilibrio IQ y filtrado adaptativo en bloque.

Timing Recovery: Estos corrigen la fase y la frecuencia offset entre los relojes del transmisor y el receptor. Puede realizarse de manera retroalimentada donde la señal de error del timing fase es detectada a partir de las señales muestreadas, las cuales son usadas para controlar el oscilador controlador de voltaje o para accionar un interpolador digital o en un enfoque de retroalimentación en el que el desplazamiento de tiempo es estimado a partir de bloques de muestras, y luego se corrige mediante interpolación digital. El método de Gardner es bastante utilizado debido a su sencillez e independencia de la fase portadora.

Frequency offset estimation: En un receptor coherente intradino, el oscilador local libre de operación y los láseres transmisores no están sincronizados en frecuencia. Esto genera un desplazamiento de frecuencia residual en la señal recibida, que debe estimarse y compensarse en el DSP. Cuando la señal de entrada tiene la forma $X_i(n) = X_{sym}(n)\exp(j[\theta(n) + n2\pi\Delta fT_{sym}])$, la tarea del sistema es estimar Δf . Los métodos de estimación de frecuencia pueden clasificarse en métodos de blind y training-aided.

Carrier phase recovery: De manera similar a los algoritmos de estimación de desplazamiento de frecuencia, los algoritmos de recuperación de la portadora pueden clasificarse como técnicas de estimación blind o dataaided. Además, los algoritmos pueden implementarse en modo feedforward o en estructura de realimentación. Cuando la señal tiene la forma $X_i(n) = X_{sym}(n)\exp(j\theta(n) + w(n))$, la tarea del sistema es estimar θ .

Symbol estimation and decoding: Luego de la recuperación de la portadora, el receptor externo puede decodificar la señal, para esto puede utilizar una corrección de errores hacia adelante (FEC) de decisión suave con un campo de Galois correspondiente al alfabeto de símbolos o una estimación de símbolos seguida de una FEC de decisión dura.

Para decodificar datos binarios con decisión dura, es necesario estimar los símbolos y decodificar los bits. Para constelaciones rectangulares, como las QAM, esto puede lograrse aplicando una serie de umbrales de decisión a los componentes en fase y en cuadratura por separado, y puede hacerse de una manera eficiente desde el punto de vista del hardware utilizando circuitos lógicos. Aunque esta técnica funciona óptimamente para un sistema limitado por AWGN, en presencia de ruido de fase no lineal los límites de decisión no rectangulares pueden mejorar el rendimiento con un gasto de complejidad de hardware.

Para solucionar este problema, se han investigado varias técnicas como los límites de decisión basados en ML o el detector de aprendizaje automático basados en otras herramientas de algoritmos de ML [26].

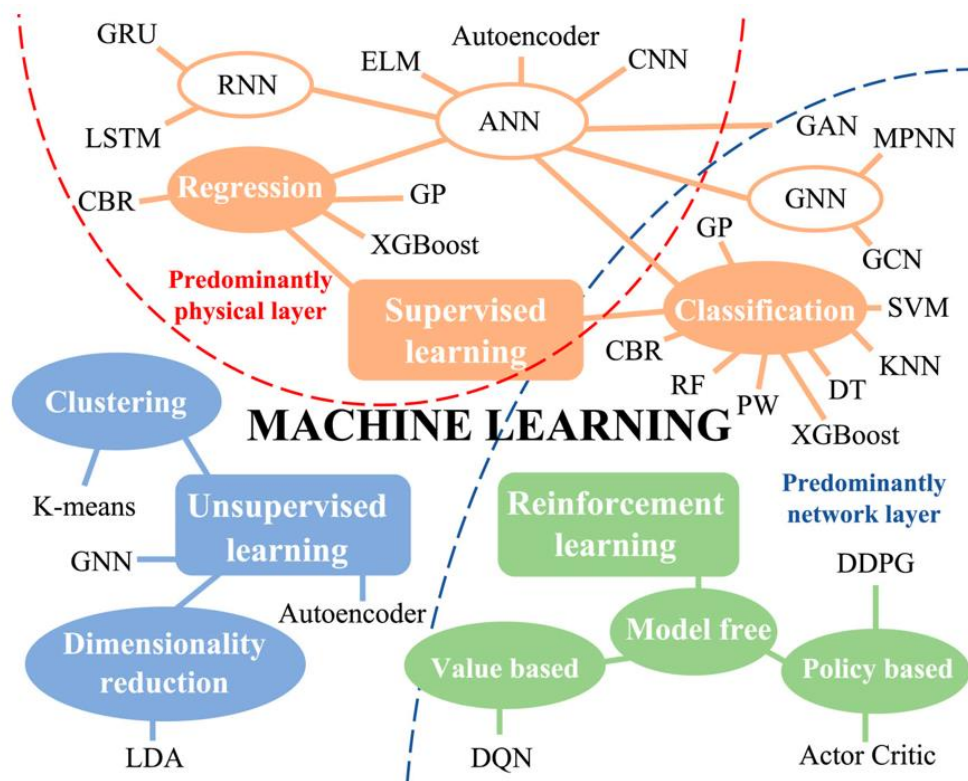


Fig. 6 Conceptos y técnicas de machine learning. [27]

En la Fig. 6 se puede apreciar algunas técnicas de machine learning donde se refleja el hecho de que el aprendizaje supervisado se usa más comúnmente en las comunicaciones ópticas que Reinforcement learning y el aprendizaje no supervisado, además el aprendizaje no supervisado es la clase de algoritmos menos utilizado [27]. Estos tipos de M.L. separados globalmente son:

Supervised learning (Aprendizaje supervisado): Este cuenta con un aprendizaje previo basado en un sistema de etiquetas, las que son parte de la solución deseada, asociadas a unos datos los cuales se entrenan y se les permiten tomar decisiones o hacer predicciones.

Unsupervised learning (Aprendizaje no supervisado): Este no cuenta con un aprendizaje previo, es decir, los datos de entrenamiento no incluyen etiquetas y será el propio algoritmo el que intente organizar o clasificar según sea su función la información.

Reinforcement learning (Aprendizaje por refuerzo): Su objetivo es que el algoritmo aprenda a partir de la propia experiencia. El modelo se implementa en forma de un agente que deberá explorar

un espacio desconocido y determinar las acciones a llevar a cabo mediante prueba y error: aprenderá por sí mismo gracias a las recompensas y penalizaciones que obtiene de sus acciones. El agente debe crear la mejor estrategia posible (políticas) para obtener la mayor recompensa en tiempo y forma. Este aprendizaje permite ser combinado con otros tipos, y está ahora mismo muy de moda puesto que el mundo real presenta muchos de estos escenarios [28,14].

Algunas características de estos algoritmos de M.L. de tipo aprendizaje supervisado se pueden observar en la Tabla 1.

Tabla 1. Comparación de diferentes algoritmos de M.L. de aprendizaje supervisado [3].

	Algoritmo de M.L.	Ventajas	Desventajas
Algoritmos de M.L. supervisados	k-NN	- Fácil de implementar. - Admite funciones de mapeo no-lineales.	- Sensible a la calidad de los datos y valores atípicos. - Requiere gran memoria de almacenamiento.
	SVM	- Eficaz en espacios de alta dimensión. - Uso eficiente de la memoria gracias a un uso de un subconjunto de puntos de entrenamiento.	- Selección de la función de kernel es heurística. - Incapacidad para utilizar datos alejados del límite de la clase.
	ANN	- Admite un amplio espectro de funciones. - Capaz de manejar datos ruidosos.	- Difícil de interpretar el modelo entrenado. - Limitado a pocas capas debido a desaparición de los gradientes.
	RNN	- Adecuado para el manejo de datos secuenciales. (adecuado para tareas que dependen del tiempo)	- El procesamiento recursivo puede ser difícil de optimizar. - Vulnerable a la explosión o desaparición de gradientes.
	DT	- Rápido y fácil de implementar. - Admite funciones de mapeo no-lineales.	- Críticamente sensible a las características y a los umbrales.
	Random forest	- Admite funciones de mapeo no-lineales. - Menos propenso a problemas de sobre ajuste.	- Complejo y lento en comparación con el DT.
	CNN	- Ofrece el rendimiento más avanzado en el tratamiento de entradas bidimensionales.	- Alta complejidad computacional.

A raíz de estos algoritmos, primeramente, se describirá y mencionará de una forma más intuitiva, para luego profundizar desde el punto de vista digital algunas estructuras y propiedades de las redes neuronales.

Redes neuronales:

Las redes neuronales artificiales son modelos matemáticos que imitan el comportamiento del cerebro humano, pero no replican exactamente su funcionamiento. Aunque están inspiradas en la biología, no se condicionan a emular el funcionamiento de las redes neuronales biológicas, ya que simplifican enormemente el proceso.

Una neurona biológica es una célula especializada en procesar información. Está compuesta por el cuerpo de la célula (soma) y dos tipos de ramificaciones: el axón y las dendritas. La neurona recibe las señales (impulsos) de otras neuronas a través de sus dendritas y transmite señales generadas por el cuerpo de la célula a través del axón. En la figura 7 se muestran los elementos de una red neuronal natural [29].

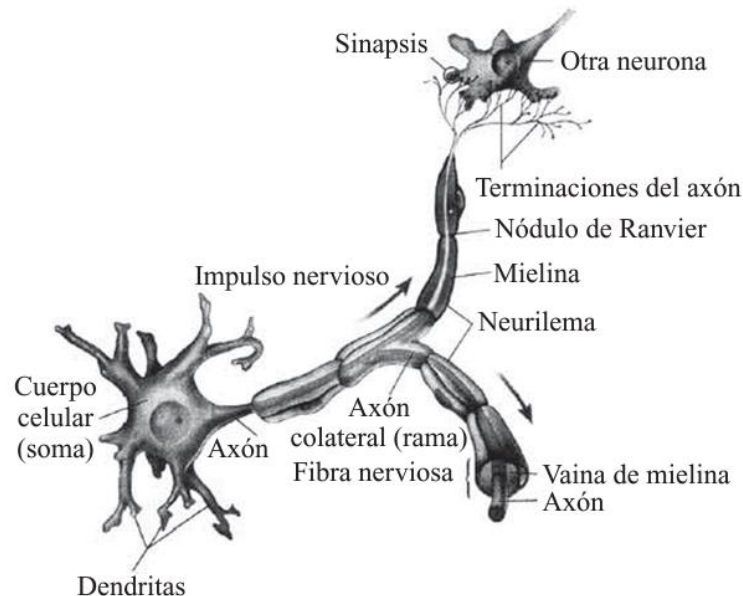


Fig. 7 Esquema de los elementos de una red neuronal natural [29].

Estos impulsos nerviosos viajan por toda la neurona empezando por las dendritas, hasta llegar al otro extremo, donde se pueden conectar con otra neurona; esta conexión entre una neurona y otra se llama sinapsis.

Para establecer una similitud directa entre la actividad sináptica y la analogía con las redes neuronales artificiales, se puede considerar que las señales que llegan a la sinapsis son las entradas a la neurona; estas entradas son ponderadas (atenuadas o simplificadas) a través de un parámetro denominado peso,

asociado a la sinapsis correspondiente. Estas señales de entrada pueden excitar a la neurona (sinapsis con peso positivo) o inhibir (sinapsis con peso negativo). El efecto, es la suma de las entradas ponderadas. Si la suma es igual o mayor que el umbral de la neurona, entonces la neurona se activa (da una salida). Esta es una situación binaria de todo o nada; cada neurona se activa o no se activa [29].

Una de las estructuras más simples de una red neuronal artificial es el perceptrón, inventada por el psicólogo Frank Rosenblatt. Éste consiste en una suma de las señales de entrada, multiplicadas por unos valores de pesos escogidos inicialmente en forma aleatoria. En la fase en la que éste aprende, la entrada se compara con un patrón preestablecido para determinar la salida de la red. Si en la comparación la suma de las entradas multiplicadas por los pesos es mayor o igual que el patrón preestablecido, la salida de la red es uno (1); en caso contrario la salida es cero (0). El perceptrón es un dispositivo que, en su configuración inicial, no está en capacidad de distinguir patrones de entrada muy complejos, sin embargo, mediante un proceso de aprendizaje puede adquirir esta capacidad. En esencia, el entrenamiento implica un proceso de refuerzo a través del cual los pesos que codifican las sinapsis se incrementan o se disminuyen. En la Fig. 8 se muestra la estructura básica de un perceptrón.

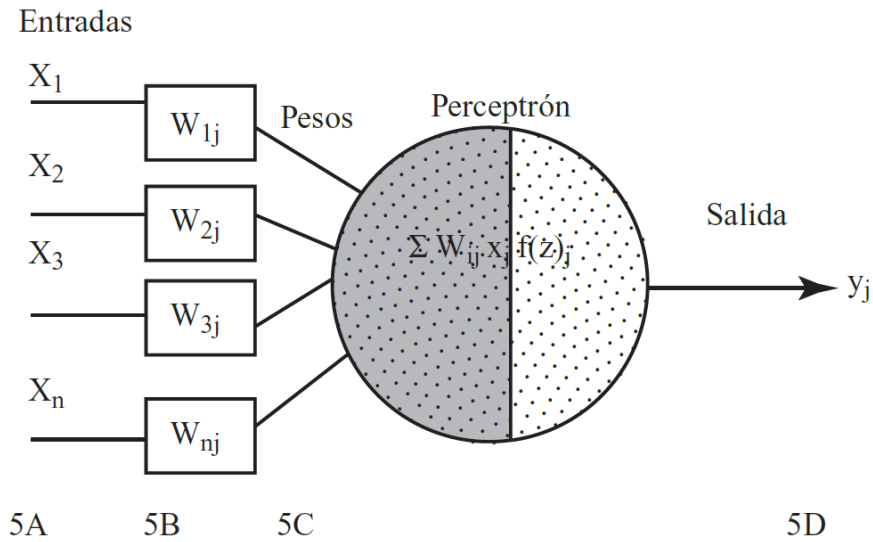


Fig. 8 Estructura del perceptrón [29].

Las entradas de la neurona (5A), los pesos aleatorios (5B), la sumatoria de la multiplicación de los pesos por sus respectivas entradas (5C) y la salida que es el cálculo de todos los pesos y sus entradas (5D) [29].

Con el tiempo se mejoró esta estructura a un perceptrón multicapa, el cual a diferencia del simple posee mejor rendimiento en la clasificación de problemas, además agrega capas intermedias llamadas capas ocultas que se encuentran entre la capa de entrada y la de salida, como se puede ver en la Fig. 9 con las capas hidden layer 1 y hidden layer 2.

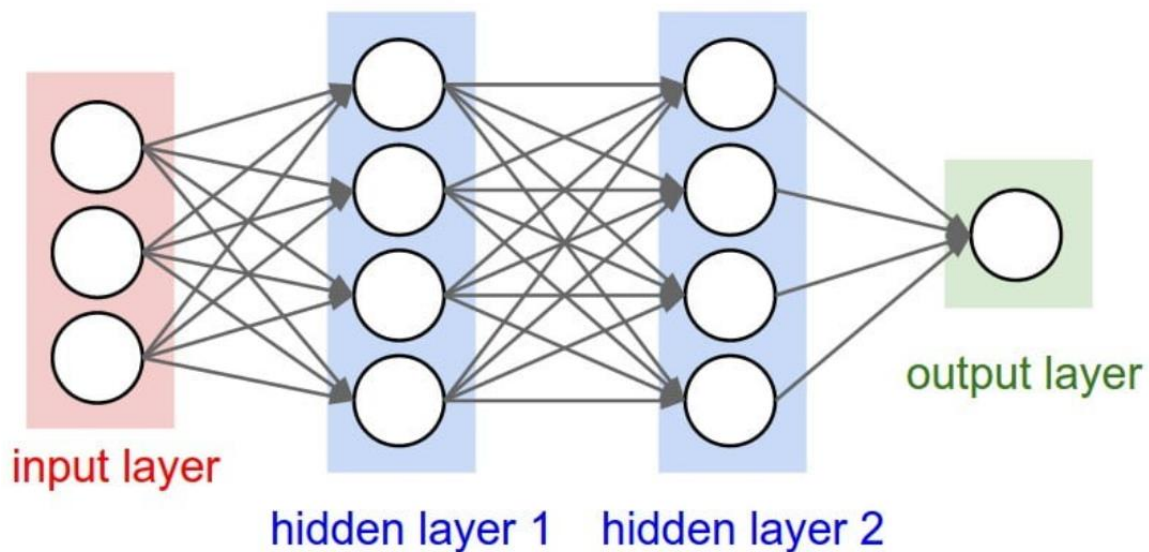


Fig. 9 Estructura básica de un perceptrón multicapa [30].

Con esta noción de las redes neuronales ahora se describirá el funcionamiento y algunas características de las redes convolucionales neuronales.

Redes neuronales convolucionales:

Se les denomina así puesto que poseen capas convolucionales que operan con la información. Esta principalmente compuesta por 3 capas, la capa de convolución, capa de agrupación o pooling layer y la capa totalmente conectada o Fully Connected (FC) que se pueden observar en la Fig. 10.

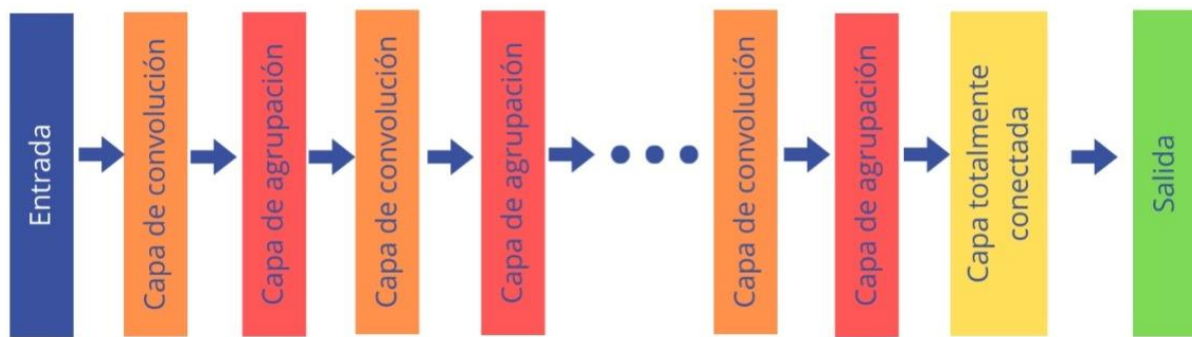


Fig. 10 Esquema de modelo general de CNN [34].

Capa de convolución

Esta es la capa núcleo de la arquitectura de las redes neuronales convolucionales, su objetivo es aprender representaciones de características (features [31]) de la información de entrada. Contiene un set de kernels convolucionales, también llamados filtros, que se convolucionan con las entradas de imágenes N-dimensionales para generar una salida de mapa de características [31] o feature maps (Un mapa de variables de entrada para un modelo de aprendizaje automático [32]). En general las capas convoluciones operan sobre tensores de 3D, con dos ejes espaciales de altura y anchura (height y width), además de un eje de canal (channels) también llamado profundidad (depth). Para una imagen de color Red Green Blue (RGB), la dimensión del eje depth es 3, porque la imagen tiene tres canales: rojo, verde y azul (red, green y blue) [28]. Un ejemplo de esto es el siguiente:

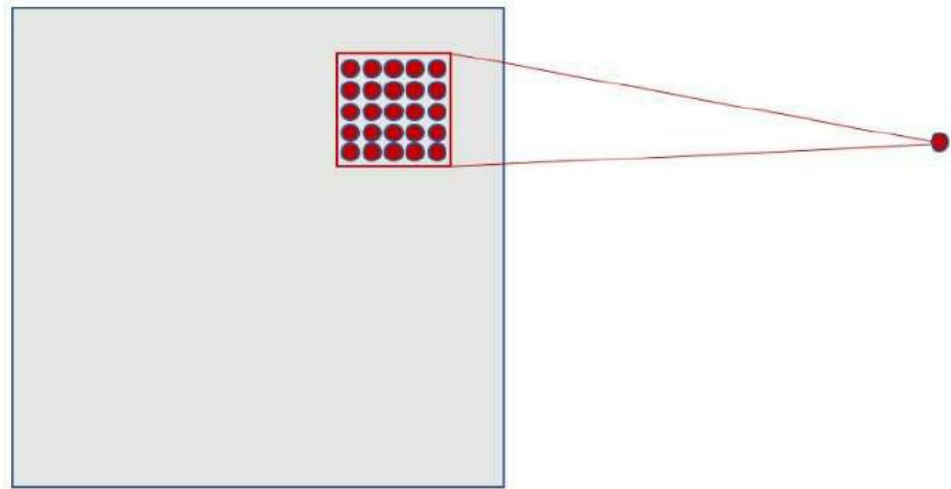


Fig. 11 Convolución de imagen 28x28 con kernel de 5x5 [28].

En la Fig. 11 se tiene una entrada o imagen de dos dimensiones 28×28 (height=28, width=28, depth=1) que se convolucionará con un kernel de tamaño de 5×5 . Intuitivamente, se puede pensar en una ventana de tamaño de 5×5 que va recorriendo toda la matriz de 28×28 de entrada que contiene la imagen. Esta ventana va deslizándose a lo largo de toda la imagen. Visualmente, se empieza con la ventana en la esquina arriba-izquierda de la imagen, y esto le da la información necesaria a la primera neurona o pixel de la capa oculta. Después, deslizamos la ventana una posición hacia la derecha para “conectar” las 5×5 neuronas de la capa de entrada incluidas en esta ventana con la segunda neurona de la capa oculta. Y así, sucesivamente, se va recorriendo todo el espacio de la capa de entrada, de izquierda a derecha y de arriba abajo. Analizando un poco el caso concreto que se ha propuesto, si se tiene una entrada de 28×28 píxeles y una ventana de 5×5 esto nos define un espacio de 24×24 neuronas en la primera capa del oculta, debido a que solo podemos mover la ventana 23 neuronas hacia la derecha y 23 hacia abajo antes de terminar con el lado derecho (o inferior) de la imagen de entrada.

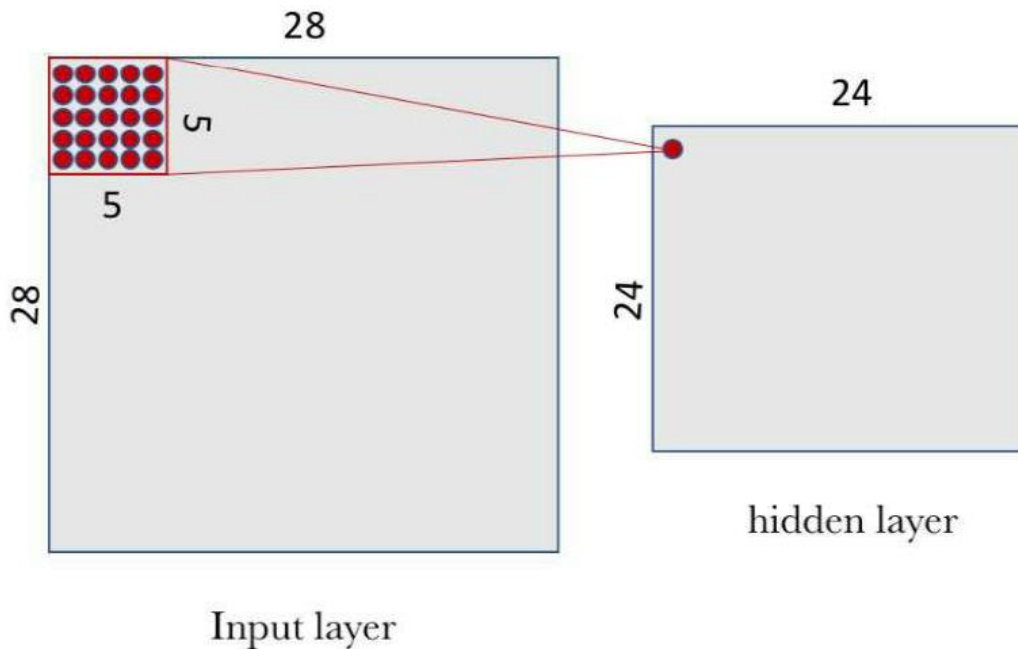


Fig. 12 A la izquierda capa de entrada de 28x28 con kernel de 5x5 y a la derecha un mapa de características 24x24 [28].

Se hace la suposición de que el avance de los movimientos de la ventana es de 1 pixel de distancia, tanto en horizontal como en vertical cuando empieza una nueva fila. Por ello, en cada paso la nueva ventana se solapa con la anterior excepto en esta línea de píxeles que hemos avanzado. En redes neuronales convolucionales se pueden usar diferentes longitudes de pasos de avance que se conoce como el parámetro “stride” [32]. También se puede aplicar una técnica de relleno de ceros alrededor del margen de la imagen para mejorar el barrido que se realiza con la ventana que se va deslizando. Este parámetro para definir el relleno se denomina padding [32]. En la Fig. 12 el valor de cada punto de la capa oculta corresponde al producto escalar entre el filtro y el kernel de 25 píxeles (5×5) de la capa de entrada (28x28), como se puede notar se reduce la dimensión de la imagen, de 28x28 a 24x24 [28].

Capa de agrupación (Pooling layer)

Las capas pooling se utilizan para submuestrear los mapas de características (producidos después de operaciones de convolución), es decir, toma los mapas de características de mayor tamaño y los reduce a mapas de características de menor tamaño. Al reducir los mapas de características, siempre se preservan las características más dominantes (o información). La operación de agrupación se realiza especificando la región agrupada el tamaño y el stride de la operación, similar a la operación de convolución. Hay diferentes tipos de técnicas de agrupación que se pueden utilizar en diferentes capas de agrupación, como agrupación máxima, agrupación mínima, agrupación promedio, agrupación cerrada, agrupación de árboles, etc. Max Pooling es el más popular. El principal inconveniente de la capa de agrupación es que a veces disminuye el rendimiento general de CNN. La razón detrás de esto es que la capa de agrupación ayuda a la CNN a encontrar si una característica específica está presente en la imagen de entrada dada o no sin preocuparse por la posición correcta de esa característica.

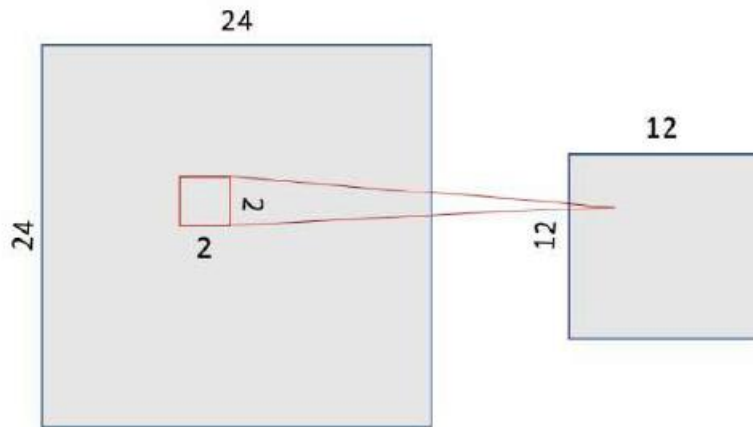


Fig. 13 Reducción de dimensión de una entrada 24x24 a 12x12 por una pooling layer de 2x2 [28].

En la Fig. 13 se tiene a la izquierda una imagen de 24x24, a la que se le aplicará una capa de pooling de 2x2 convirtiéndola a una imagen de 12x12, es decir, el tamaño de la salida se redujo 4 veces el tamaño original. Es importante remarcar que con la transformación de este pooling se mantiene la relación espacial.

Capa totalmente conectada

Generalmente la última parte (o la última de cada capa) de todas las arquitecturas de CNN (usada para clasificación) consiste en la capa totalmente conectada, donde cada neurona adentro de la capa está conectada con cada neurona de la capa previa. La última capa totalmente conectada es usada como la capa de salida (en el caso de un clasificador) de la arquitectura de una CNN.

Las capas totalmente conectadas son un tipo de ANN de retroalimentación y sigue el principio de la red neuronal perceptrón multicapa tradicional (MLP). Las capas FC toman la información de la capa convolucional o de la pooling layer, que tiene la forma de un conjunto de métricas (mapas de características) y esas métricas son “flattened”, es decir, se convierten todas las matrices bidimensionales resultantes de mapas de características agrupados a un único vector lineal largo y continuo. Este vector luego se alimenta en la capa FC para generar la salida final de CNN como se puede ver en la Fig. 14.

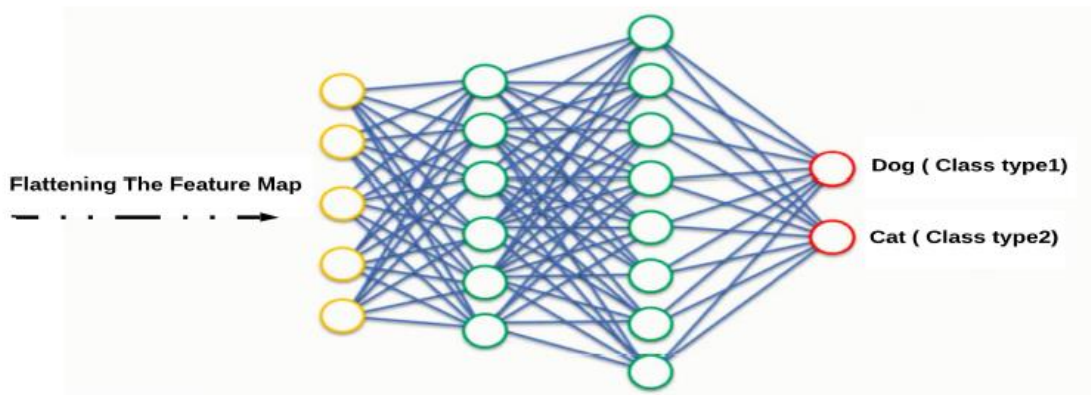


Fig. 14 Arquitectura de la capa totalmente conectada [31].

Funciones de activación:

La tarea principal de cualquier función de activación en cualquier modelo basado en redes neuronales es mapear la entrada a la salida, donde el valor de entrada se obtiene calculando la suma de los pesos de las neuronas entrada y agregando aún más sesgo (si es que existe un sesgo). En otras palabras, la función de activación decide si una neurona se activa o no para una entrada determinada produciendo la correspondiente salida. La función de activación sirve para introducir la no linealidad en el modelo de la red. En la arquitectura CNN, después de cada capa que pueda aprender (capas con pesos, es decir, convolucionales y capas FC) se utilizan capas de activación no lineales. Este comportamiento no lineal de esas capas permite al modelo CNN aprender cosas más complejas y lograr asignar las entradas a las salidas de forma no lineal [31]. Existen muchas funciones de activación a continuación se mencionarán algunas de ellas.

Linear:

Una relación entre dos o más variables que se puede representar únicamente mediante la suma y la multiplicación. El gráfico de una relación lineal es una línea recta que pasa por el origen [32].

Sigmoid:

Una función matemática que "comprime" un valor de entrada en un rango restringido, normalmente de 0 a 1 o de -1 a +1. Es decir, puede pasar cualquier número (dos, un millón, mil millones negativos, lo que sea) a un sigmoide y la salida seguirá estando en el rango restringido [28]. Su expresión matemática es:

$$f(x) = \frac{1}{1 + e^{-x}}$$

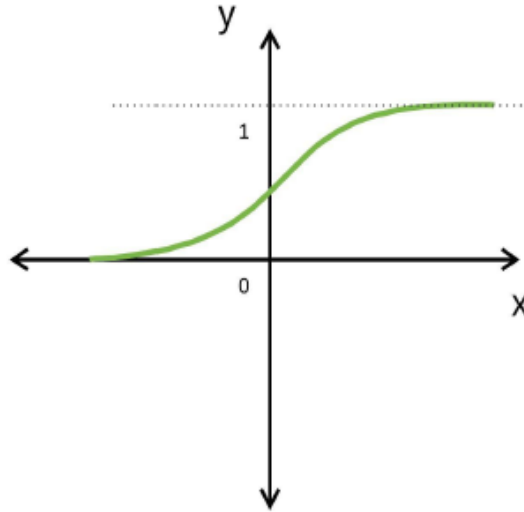


Fig. 15 Gráfico de una función sigmoide [31].

Tanh:

Representa la relación entre el seno hiperbólico y el coseno hiperbólico: $\tanh(x) = \sinh(x)/\cosh(x)$. A diferencia de la función sigmoide, el rango normalizado de tanh está entre -1 y 1, que es la entrada que le va bien a algunas redes neuronales. La ventaja de tanh es que puede tratar más fácilmente con números negativos [28]. Su expresión matemática es la siguiente:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

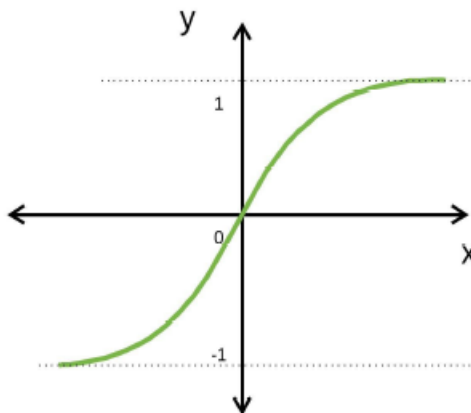


Fig. 16 Gráfico de una función Tanh [31].

ReLU:

La Unidad Lineal Rectificadora (ReLU) es la función de activación más comúnmente utilizada en Redes neuronales convolucionales. Se utiliza para convertir todos los valores de entrada a números positivos. La ventaja de ReLU es que requiere una carga informática mínima en comparación con otros. Su expresión matemática es la siguiente [31]:

$$f(x) = \max(0, x)$$

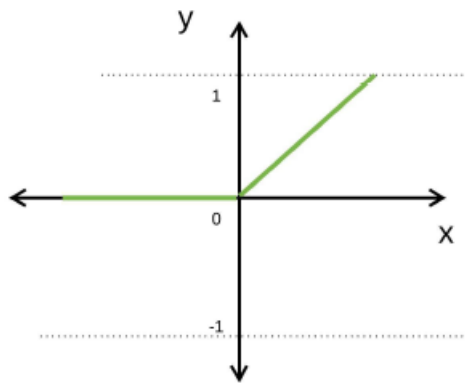


Fig. 17 Gráfico de una función ReLU [31].

Softmax:

Convierte un vector de valores en una distribución de probabilidad. Los elementos del vector de salida están dentro del rango [0,1] y suman 1. Cada vector de entrada se maneja de forma independiente. Se utiliza a menudo como activación de la última capa de una red de clasificación porque el resultado podría interpretarse como una distribución de probabilidad [33].

5. Desarrollo de la metodología

5.1 Desarrollo de simulación de canal AWGN en Matlab

Para empezar, se deben generar los datos a procesar, los cuales son imágenes correspondientes a las constelaciones de los diferentes tipos y ordenes de modulaciones. Se utilizó la herramienta computacional Matlab, en la que gracias a varios comandos pre-instalados se pudo generar la modulación de las señales con canales AWGN.

Se uso el comando “qamod” para la modulación en QAM y el “pskmod” para la modulación en PSK, además con “awgn” se puede añadir ruido gaussiano al canal. Con esto se procede a crear un código para ambas modulaciones en las que se generan constelaciones que otorgan información acerca del canal. Además, se itera el código para entregar diferentes bases datos de imágenes para cada modulación.

5.2 Creación de bases de datos

Para cada base de datos de imágenes se crean 3 algoritmos, uno para el conjunto de entrenamiento (train), otro para el conjunto de validación (validation) y otro para el conjunto de testeo (test), se especifica la cantidad total de imágenes, los rangos de SNRs en [dB] que van del 5-20. El orden de modulación (M) y número de bits x símbolo (N). De estos se utilizaron los órdenes de modulación 16, 64 y 256; mientras que para N se usaron 4, 6 y 8. Salvo el caso del clasificador binario también se añadió un factor F de rango [2 – 25]. Para poseer más variación y aleatoriedad en la cantidad de puntos a graficar en la constelación.

El número de bits por símbolo que se transmite viene dado por:

$$k \text{ (Bits por símbolo)} = \log_2 M, \text{ con } M = \text{cantidad diferente de símbolos.} \quad (5.1)$$

El número de bits de entrada (bits input) variable depende del valor de M, N y F. Con una configuración de entre 8-100 puntos por símbolo, por ese motivo el rango del factor $F \in [2 - 25]$. Esto con el motivo de hacer más interesante el entrenamiento en la clasificación de los algoritmos, ya que no sería útil y/o visualmente tangible si son muy pocos puntos o al revés, muchos puntos para cada constelación.

La ecuación que representa la cantidad de bits totales a transmitir es:

$$\text{Min y max(bits totales)} = N_{\frac{\text{Bits}}{\text{Sym}}} * M_{\text{Sym}} * F \text{ (en base a } M) \quad (5.2)$$

F = Su valor se encuentra entre [2-25], con saltos de 0,25 entre cada unidad discreta.

Para cada valor de M se generará un rango diferente, por ejemplo, para M =16 se tiene:

$$\text{Min y max(bits totales)} = N_{\frac{\text{Bits}}{\text{Sym}}} * M_{\text{Sym}} * F \text{ (en base a } M) \quad (5.3)$$

$$\text{Min y max(bits totales)} = 4 * 16 * F \text{ (en base a } M) \quad (5.4)$$

$$\text{Min y max(bits totales)} = 64 * F \text{ (en base a } M) \quad (5.5)$$

$$\text{Min y max(bits totales)} = 64 * F \in [2 - 25] \quad (5.6)$$

$$\text{Min y max(bits totales)} = [128 - 1600] \quad (5.7)$$

De manera general se puede visualizar en la siguiente tabla.

Tabla 2. Parámetros de los algoritmos para la creación de bases de datos [34].

Modulations				
M - QAM	M - PSK	N° bits x Symbol (N)	N° bits input	SNR [dB]
16 - QAM	16 - PSK	4	128-1600	5 - 20
64 - QAM	64 - PSK	6	512-6400	5 - 20
256 - QAM	256 - PSK	8	4096-51200	5 - 20

En los códigos para generar las imágenes existe un comando presente en todos llamado “scatterplot” el cual grafica un diagrama de dispersión dependiente de la señal que irá a graficar, en este caso una señal modulada del tipo QAM y PSK, además se tuvo cuidado en realizar de la manera más sobria posible la gráfica de imágenes, es decir, sin adornos, puntos referenciales o nombres, con el fin de que la CNN no aprenda a partir de estos y se enfoque en la información bruta de las constelaciones. Con el comando “randperm” se genera un índice aleatorio, con este índice se accede a un elemento específico de los vectores que contienen el SNR y el factor F. Con todos estos parámetros se genera la señal modulada, se itera con ciclos for y while, se almacenan constelaciones y errores de símbolos en carpetas para mejor análisis. La generación de imágenes de todos los algoritmos es aleatoria. Se debe tener especial cuidado al momento de guardar las imágenes, debido a que, si poseen el mismo nombre se sobre escriben en la carpeta contenedora, por lo que para evitar esto se agregó al final de cada imagen un nombre diferente, este nombre corresponde al número de imagen en el orden que se generó.

5.3 Creación y ejecución del algoritmo clasificador binario CNN para los tipos de modulación.

Para crear el algoritmo se utilizaron 4.000 imágenes de entrenamiento, 2.000 imágenes de validación y 2.000 imágenes de test, tanto para modulaciones QAM como PSK. Se importaron varias librerías, entre ellas las numpy, matplotlib inline y numerosos tensorflow.keras.

Se montaron las bases de datos de Google drive, por medio de archivos con extensión .zip, para un manejo más óptimo en los tiempos de espera para subir archivos a Google drive.

Se juntaron las carpetas contenedoras de las bases de datos, se mostró explícitamente la cantidad de imágenes y algunas muestras de ellas.

Se crea el modelo, utilizando 3 módulos principales, la convolución, función de activación ReLu y capa MaxPooling. Se deja la entrada de imágenes a 150x150 pixeles, luego empiezan las capas de la siguiente manera:

1era etapa: Capa convolucional de 16 filtros, con una capa MaxPooling 2D de 2x2.

2da etapa: Capa convolucional de 32 filtros, con una capa MaxPooling 2D de 2x2.

3era etapa: Capa convolucional de 64 filtros, con una capa MaxPooling 2D de 2x2.

Lo siguiente es aplanar el mapa de características a 1 dimensión de tensor, se crea la Fully connected layer con función de activación ReLu y se finaliza con la salida que posee una función de activación sigmoid, debido a que este es un problema de clasificación binario, por lo que, la salida será un único escalar entre 0 y 1, es decir, será una modulación QAM o PSK la que se clasifique.

Finalmente se configuran generadores de datos que leerán las imágenes de las carpetas, uno para entrenamiento y el otro para validación produciendo lotes de 20 (batch size) de tamaño 150x150, con sus etiquetas (binarias), además se procesan las imágenes normalizando los valores de los pixeles para que estén en el rango [0,1]. Se evalúa la precisión y perdida del modelo tanto para datos de entrenamiento como de validación. En cuanto a los recursos de Google utilizados se utilizó solamente el acelerador por hardware “CPU”. En la Fig. 18 se puede observar un resumen esquemático del diseño de la CNN binaria.

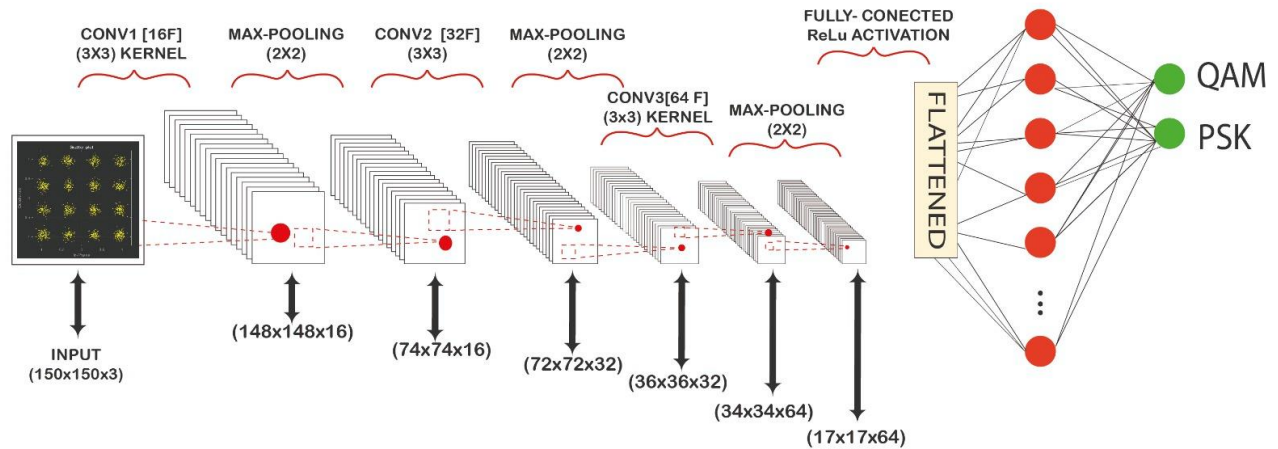


Fig. 18 Diseño de CNN binaria QAM y PSK [34].

5.4 Creación y ejecución de algoritmos clasificadores multiclase CNN para diferentes formatos de modulación.

En esta parte se crearon 2 algoritmos, uno para las modulaciones 16-64-256-QAM y otro para las modulaciones 16-64-256-PSK, la idea es que la CNN clasifique para cada orden de modulación diferente en cada tipo de modulación de manera separada.

1) Modulación 16-64-256-QAM:

Se utilizaron 4.000 imágenes de entrenamiento para los formatos de modulación, 2.000 imágenes de validación para los formatos de modulación y 2.000 imágenes de test para los formatos de modulación. de la misma forma que en el clasificador binario se agregan las librerías, se montaron las bases de datos en Google drive y se unieron las carpetas.

Luego se redimensionan las imágenes a 150x150 pixeles, definiendo un batch size de 20. Se aplica “augmentation” que es una técnica para expandir artificialmente el tamaño de los datasets, mediante la creación de versiones modificadas de imágenes del conjunto de datos de entrenamiento. Además se utiliza un “ImageDataGenerator” y “train_datage.flow_from_directory” con especificaciones del tamaño de imágenes, el tamaño del batch, normalización de los pixeles dejándolos en rango de [0,1] y el tipo de clase, en este caso una “categorical” puesto que se está trabajando con multiclase. Se

etiquetan los datos predefinidos por la forma en cómo se organizaron las carpetas con los nombres de cada clase al momento de subir la base de datos.

Se crea el modelo, utilizando 3 módulos principales, la convolución, función de activación ReLu y capa MaxPooling. La arquitectura queda de la siguiente manera:

1era etapa: Capa convolucional de 256 filtros, con un kernel de 5x5, una capa MaxPooling 2D de 2x2 y una normalización del batch.

2da etapa: Capa convolucional de 128 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

3era etapa: Capa convolucional de 32 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

El motivo de realizar el modelo con esta estructura decreciente de filtros, es porque en las primeras capas de una CNN, los filtros tienden a aprender características locales simples, como bordes y texturas, por lo que se abstrae información de manera más general, además, reducir la cantidad de filtros en capas subsiguientes ayuda a reducir la dimensionalidad del mapa de características. Esta disminución en la cantidad de filtros puede considerarse como una forma de construir una jerarquía de características, lo que puede hacer que el modelo sea más eficiente computacionalmente y ayude a prevenir el sobreajuste (overfitting).

Se aplanan con la función “Flatten”, luego se utiliza la función de activación ReLu, con un “Dropout” de 0,5 y se termina el modelo especificando la cantidad de clases (3) que en este caso son con una función de activación “softmax”.

Se usa un optimizador y un “callback” (objeto que puede realizar acciones interesantes como reducir la tasa de aprendizaje cuando una métrica deja de mejorar), se compila el modelo y se entrena con epochs igual a 50. Se evalúa la precisión y pérdida del modelo tanto para datos de entrenamiento como de validación.

No existe una única arquitectura de CNN que funcione para todos los casos, y es común que sea necesario ajustar la arquitectura según las características del conjunto de datos y los objetivos del problema. Debido a esto, tanto para esta sección (5.4) como para la sección 5.5 los modelos de CNN se diseñarán de manera en que puedan tener arquitecturas similares, con fines comparativos, pero al mismo tiempo que logren ejecutarse lo más fluidamente, por lo tanto, la elección particular de cada

uno de estos modelos fue a modo de prueba y error, siempre teniendo en consideración lo anteriormente mencionado. Además, se prueba con los datos de test para finalmente crear una matriz de confusión a partir de los resultados de las clases originales vs las clases predichas por el modelo. En la Fig. 19 se puede observar un resumen esquemático del diseño de la CNN multiclase para formatos de modulaciones QAM.

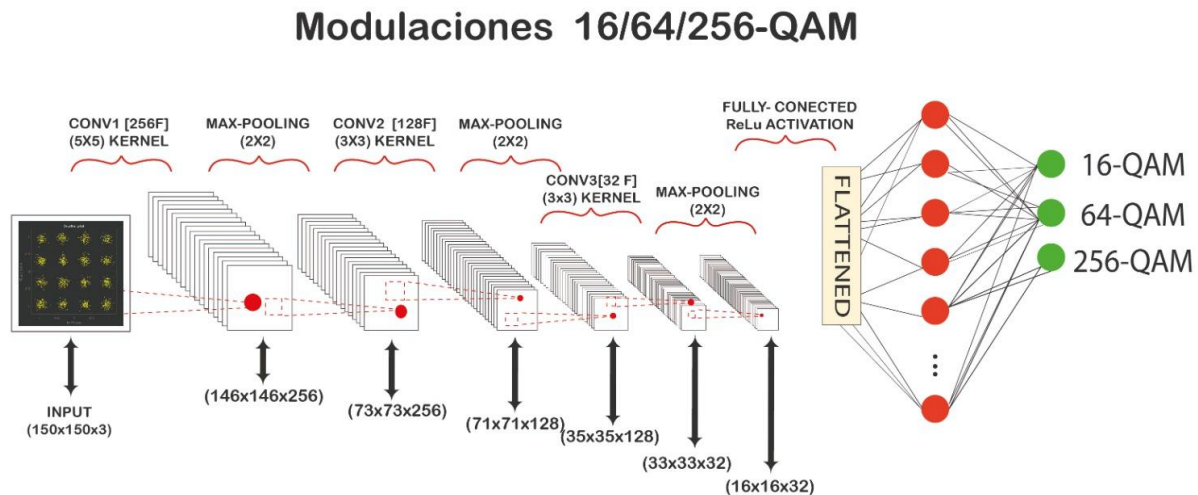


Fig. 19 Diseño de CNN multiclase para modulaciones 16/64/256-QAM [34].

Modulación 16-64-256-PSK:

La estructura del código y modelo es bastante similar al anterior algoritmo de clasificación multiclase de QAM salvo las siguientes diferencias:

Se utilizaron 9.000 imágenes de entrenamiento para los formatos de modulación, 6.000 imágenes de validación para los formatos de modulación y 6.000 imágenes de test para los formatos de modulación.

Se define el siguiente modelo:

1era etapa: Capa convolucional de 256 filtros, con un kernel de 5x5, una capa MaxPooling 2D de 2x2 y una normalización del batch.

2da etapa: Capa convolucional de 128 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

3era etapa: Capa convolucional de 64 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch. En la Fig. 20 se puede observar un resumen esquemático del diseño de la CNN multiclase para formatos de modulación PSK.

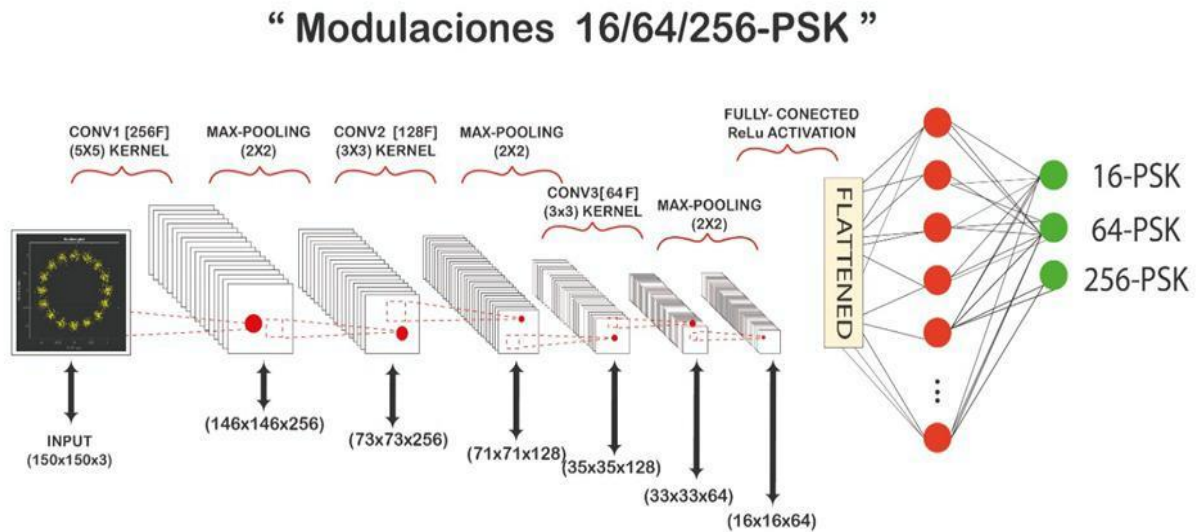


Fig. 20 Diseño de CNN multiclase para modulaciones 16/64/256-PSK [34]

Ambos modelos y los siguientes necesitan de muchos recursos computacionales, principalmente tarjeta gráfica con notable ancho de banda y gran memoria para múltiples operaciones matriciales. Por este motivo se utilizaron los recursos de Google, conocidos como los aceleradores por hardware “T4 GPU” el cual es considerablemente mejor en rendimiento que el utilizado en el clasificador binario.

Es importante destacar que al requerir mucho trabajo computacional para ambos modelos se precisa de al menos entre 3 a 4 horas para completar el entrenamiento, en el caso de que no suceda algún error del tipo overfitting, problemas con el batch size, desconexión de internet o expulsión de sesión debido a los recursos que ofrece Google de manera gratuita en el mejor de los casos. Esto es esperable en modelos complejos de aprendizaje supervisado, por lo que no es tan expedito en términos de tiempo, pero si muy útil en cuanto al transfer learning.

Como se mencionó los resultados se mostrarán en una matriz de confusión para los algoritmos, donde se podrá observar la cantidad de imágenes clasificadas correcta e incorrectamente de cada orden de modulación y entre paréntesis los porcentajes de la cantidad de imágenes clasificadas relativas al total de imágenes de entrada en cada bloque de la matriz.

5.5 Creación y ejecución de algoritmos clasificadores multiclase CNN que estime la calidad óptica de la señal en base al SNR.

En esta parte se crearon 3 algoritmos para cada tipo de modulación (QAM y PSK). 3 algoritmos para los tipos de formatos de modulación 16-64-256-QAM y otros 3 para los tipos de formatos de modulación 16-64-256-PSK, en total 6 algoritmos. Todos con 16 salidas correspondientes a el rango de SNR incluyendo los extremos (5-20).

1) Modulación 16-QAM SNRs:

Se utilizaron 6.400 imágenes de entrenamiento, 4.800 imágenes de validación en y 4.800 imágenes de test. De la misma forma que en los clasificadores anteriores se agregan las librerías, se montaron las bases de datos en Google drive y se unieron las carpetas.

En todos los algoritmos se redimensionan las imágenes a 150x150 pixeles, se define un batch size de 32. Se aplica augmentation con un “ImageDataGenerator” y “train_datagen.flow_from_directory” con especificaciones del tamaño de imágenes, el tamaño del batch, normalización de los pixeles dejándolos en rango de [0,1] y el tipo de clase “categorical”. Se etiquetan los datos predefinidos por la forma en cómo se organizaron las carpetas con los nombres de cada clase al momento de subir la base de datos, que en estos 6 algoritmos de esta sección corresponden a conjuntos de imágenes diferenciados por sus SNRs.

Se crea el modelo, utilizando 3 módulos principales, la convolución, función de activación ReLu y capa MaxPooling. La arquitectura queda de la siguiente manera:

1era etapa: Capa convolucional de 256 filtros, con un kernel de 5x5, una capa MaxPooling 2D de 2x2 y una normalización del batch.

2da etapa: Capa convolucional de 128 filtro, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

3era etapa: Capa convolucional de 64 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

4ta etapa: Capa convolucional de 32 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

Se aplana con la función “Flatten”, luego se utiliza la función de activación ReLu, con un “Dropout” de 0,5 y se termina el modelo especificando la cantidad de clases (16) que en este caso son con una función de activación “softmax”.

Se usa un optimizador y un “callback”, se compila el modelo y se entrena con epochs igual a 50. Se evalúa la precisión y perdida del modelo tanto para datos de entrenamiento como de validación y se crea una matriz de confusión a partir de los resultados de las clases originales vs las clases predichas por el modelo. En la Fig. 21 se puede observar un resumen esquemático del diseño de la CNN.

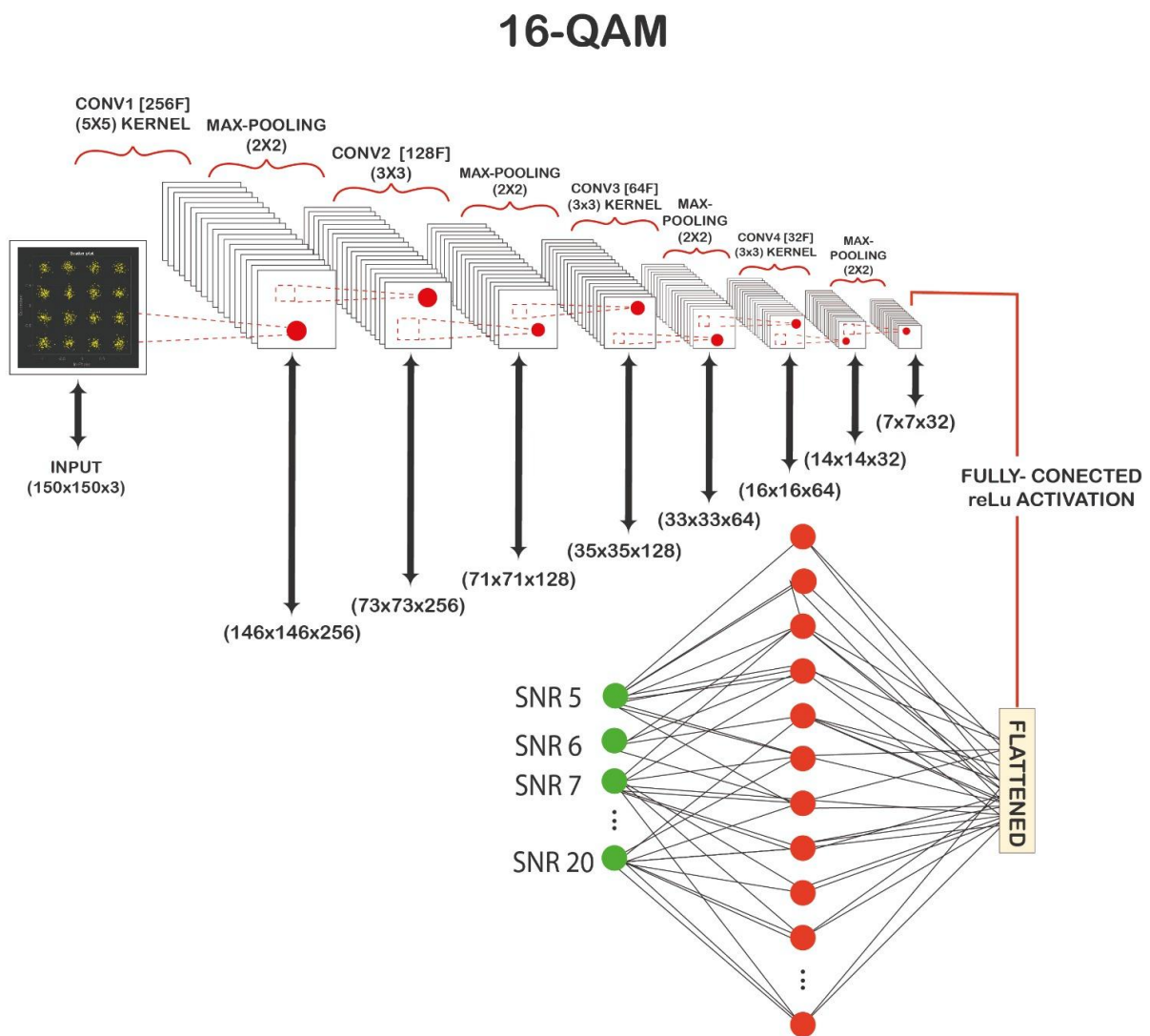


Fig. 21 Diseño de CNN Multiclase estimador calidad de señal para 16-QAM [34].

En toda esta sección (5.5) se utilizó el acelerador “T4 GPU”, además se evalúa la precisión y pérdida del modelo junto con los resultados que se mostrarán en matrices de confusión para todos los algoritmos revelando detalles acerca de la clasificación en el capítulo de 6 de resultados.

2) Modulación 64-QAM SNRs:

Se utilizaron 16.000 imágenes de entrenamiento, 6.400 imágenes de validación y 6.400 imágenes de test. De la misma forma que en los clasificadores anteriores se agregan las librerías, se montaron las bases de datos en Google drive y se unieron las carpetas.

Se redimensionan las imágenes, definiendo un batch size de 12. Se aplica “augmentation” utilizando “ImageDataGenerator” y “train_datagen.flow_from_directory” con especificaciones del tamaño de imágenes, el tamaño del batch, normalización de los pixeles dejándolos en rango de [0,1] y el tipo de clase “categorical”. Se etiquetan los datos predefinidos de conjuntos de imágenes diferenciados por sus SNRs.

Se crea el modelo, utilizando 3 módulos principales, la convolución, función de activación ReLu y capa MaxPooling. La arquitectura queda de la siguiente manera:

1era etapa: Capa convolucional de 512 filtros, con un kernel de 5x5, una capa MaxPooling 2D de 2x2 y una normalización del batch.

2da etapa: Capa convolucional de 256 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

3era etapa: Capa convolucional de 128 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

4ta etapa: Capa convolucional de 64 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

Se aplana con la función “Flatten”, luego se utiliza la función de activación ReLu, con un “Dropout” de 0,6 y se termina el modelo especificando la cantidad de clases (16) que en este caso son con una función de activación “softmax”.

Se usa un optimizador y un “callback”, se compila el modelo y se entrena con epochs igual a 40.

64-QAM

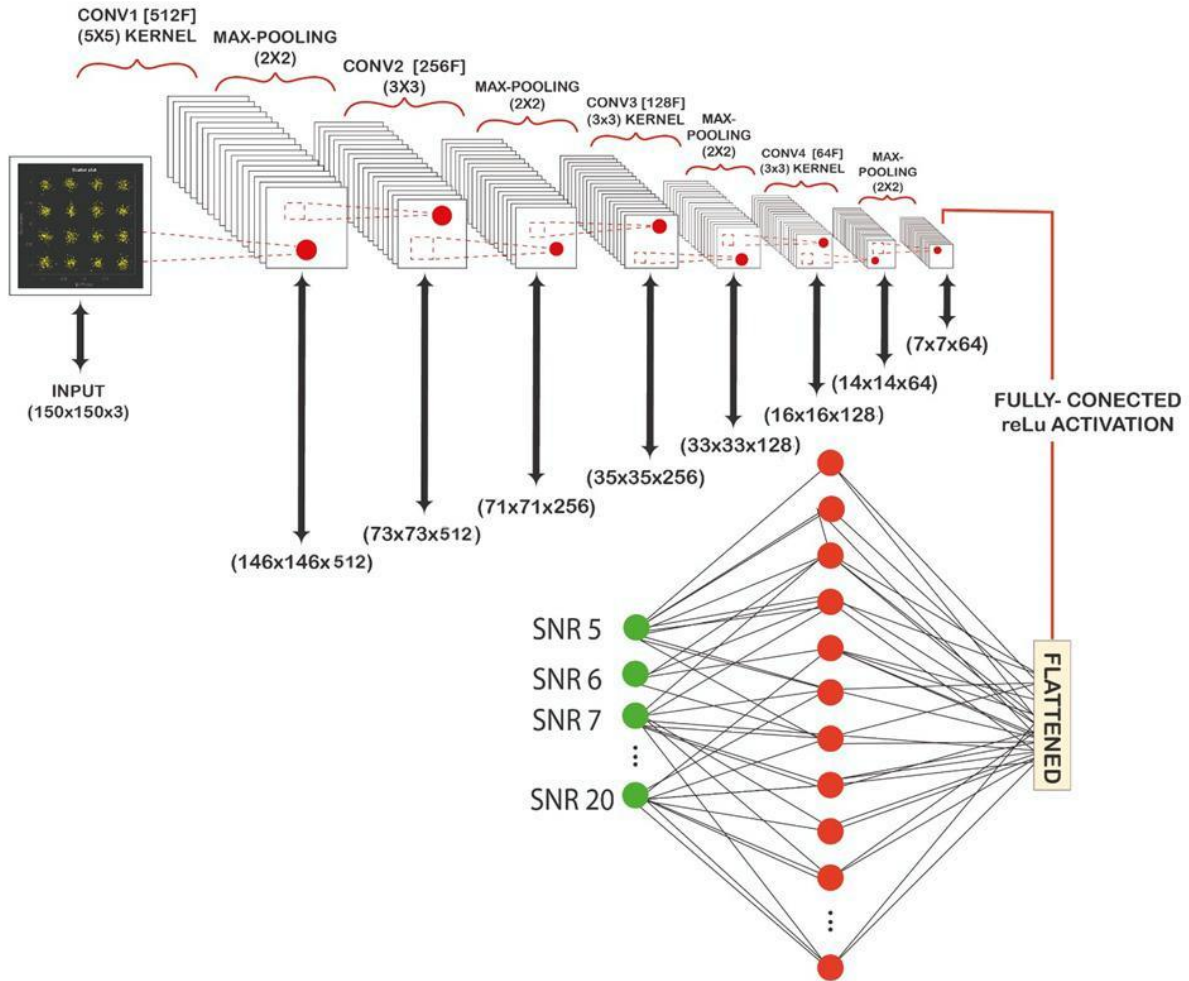


Fig. 22 Diseño de CNN Multiclase estimador calidad de señal para 64-QAM [34].

3) Modulación 256-QAM SNRs:

Se utilizaron 16.000 imágenes de entrenamiento, 8.000 imágenes de validación y 8.000 imágenes de test. De la misma forma que en los clasificadores anteriores se agregan las librerías, se montaron las bases de datos en Google drive y se unieron las carpetas.

Se redimensionan las imágenes, definiendo un batch size de 12. Se aplica augmentation utilizando “ImageDataGenerator” y “train_datage.flow_from_directory” con especificaciones del tamaño de imágenes, el tamaño del batch, normalización de los pixeles dejándolos en rango de $[0,1]$ y el tipo de

clase “categorical”. Se etiquetan los datos predefinidos de conjuntos de imágenes diferenciados por sus SNRs.

Se crea el modelo, utilizando 3 módulos principales, la convolución, función de activación ReLu y capa MaxPooling. La arquitectura queda de la siguiente manera:

1era etapa: Capa convolucional de 512 filtros, con un kernel de 5x5, una capa MaxPooling 2D de 2x2 y una normalización del batch.

2da etapa: Capa convolucional de 256 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

3era etapa: Capa convolucional de 128 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

4ta etapa: Capa convolucional de 64 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

Se aplana con la función “Flatten”, luego se utiliza la función de activación ReLu, con un “Dropout” de 0,6 y se termina el modelo especificando la cantidad de clases (16) que en este caso son con una función de activación “softmax”.

Se usa un optimizador y un “callback”, se compila el modelo y se entrena con epochs igual a 50.

256-QAM

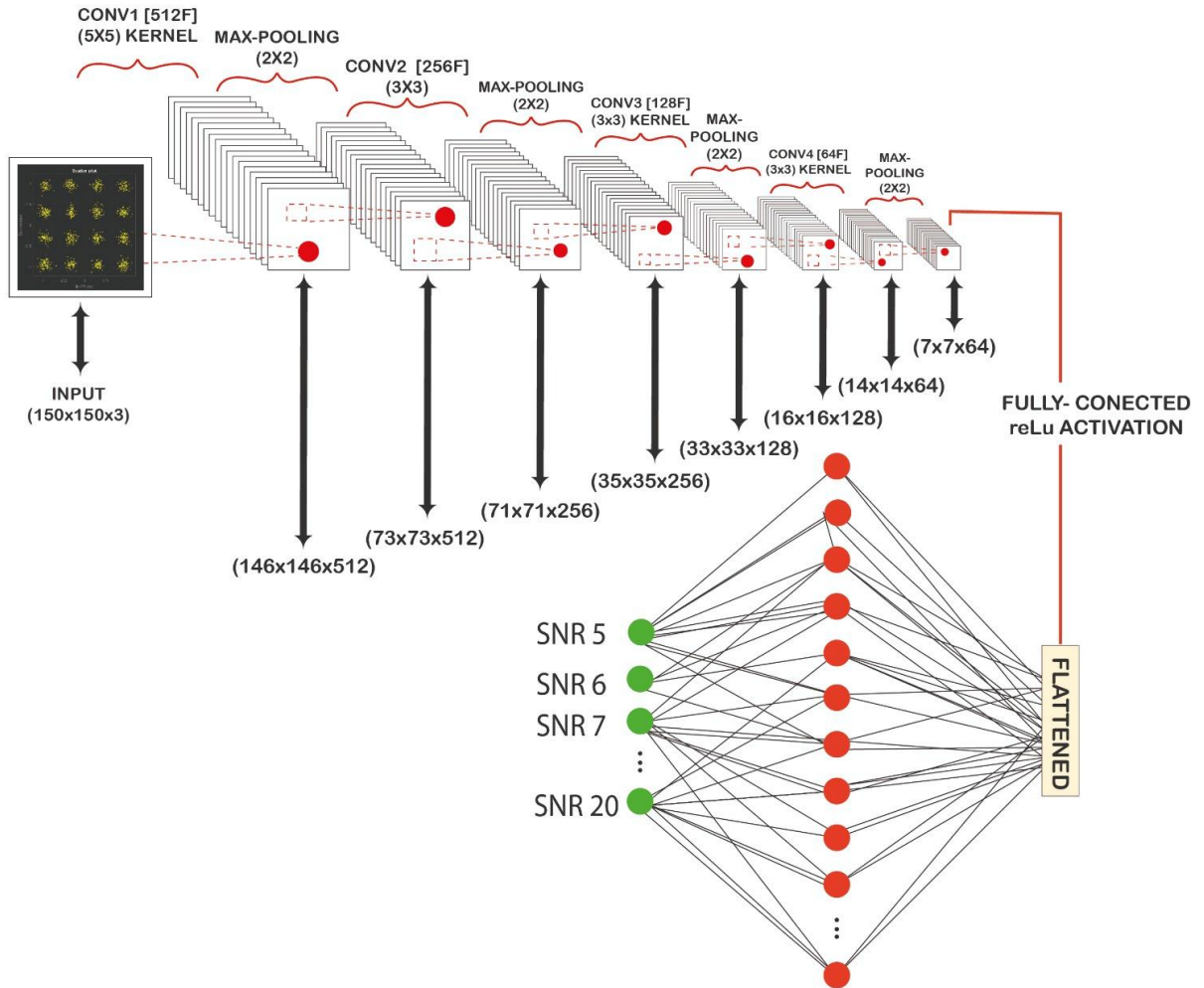


Fig. 23 Diseño de CNN Multiclase estimador calidad de señal para 256-QAM [34].

4) Modulacion 16-PSK SNRs:

Se utilizaron 16.000 imágenes de entrenamiento, 8.000 imágenes de validación y 8.000 imágenes de test. De la misma forma que en los clasificadores anteriores se agregan las librerías, se montaron las bases de datos en Google drive y se unieron las carpetas.

Se redimensionan las imágenes, definiendo un batch size de 10. Se aplica augmentation utilizando “ImageDataGenerator” y “train_datagen.flow_from_directory” con especificaciones del tamaño de imágenes, el tamaño del batch, normalización de los pixeles dejándolos en rango de [0,1] y el tipo de clase “categorical”. Se etiquetan los datos predefinidos de conjuntos de imágenes diferenciados por sus SNRs.

Se crea el modelo, utilizando 3 módulos principales, la convolución, función de activación ReLu y capa MaxPooling. La arquitectura queda de la siguiente manera:

1era etapa: Capa convolucional de 512 filtros, con un kernel de 5x5, una capa MaxPooling 2D de 2x2 y una normalización del batch.

2da etapa: Capa convolucional de 256 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

3era etapa: Capa convolucional de 128 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

4ta etapa: Capa convolucional de 32 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

Se aplana con la función “Flatten”, luego se utiliza la función de activación ReLu, con un “Dropout” de 0,6 y se termina el modelo especificando la cantidad de clases (16) que en este caso son con una función de activación “softmax”.

Se usa un optimizador y un “callback”, se compila el modelo y se entrena con epochs igual a 39.

16-PSK

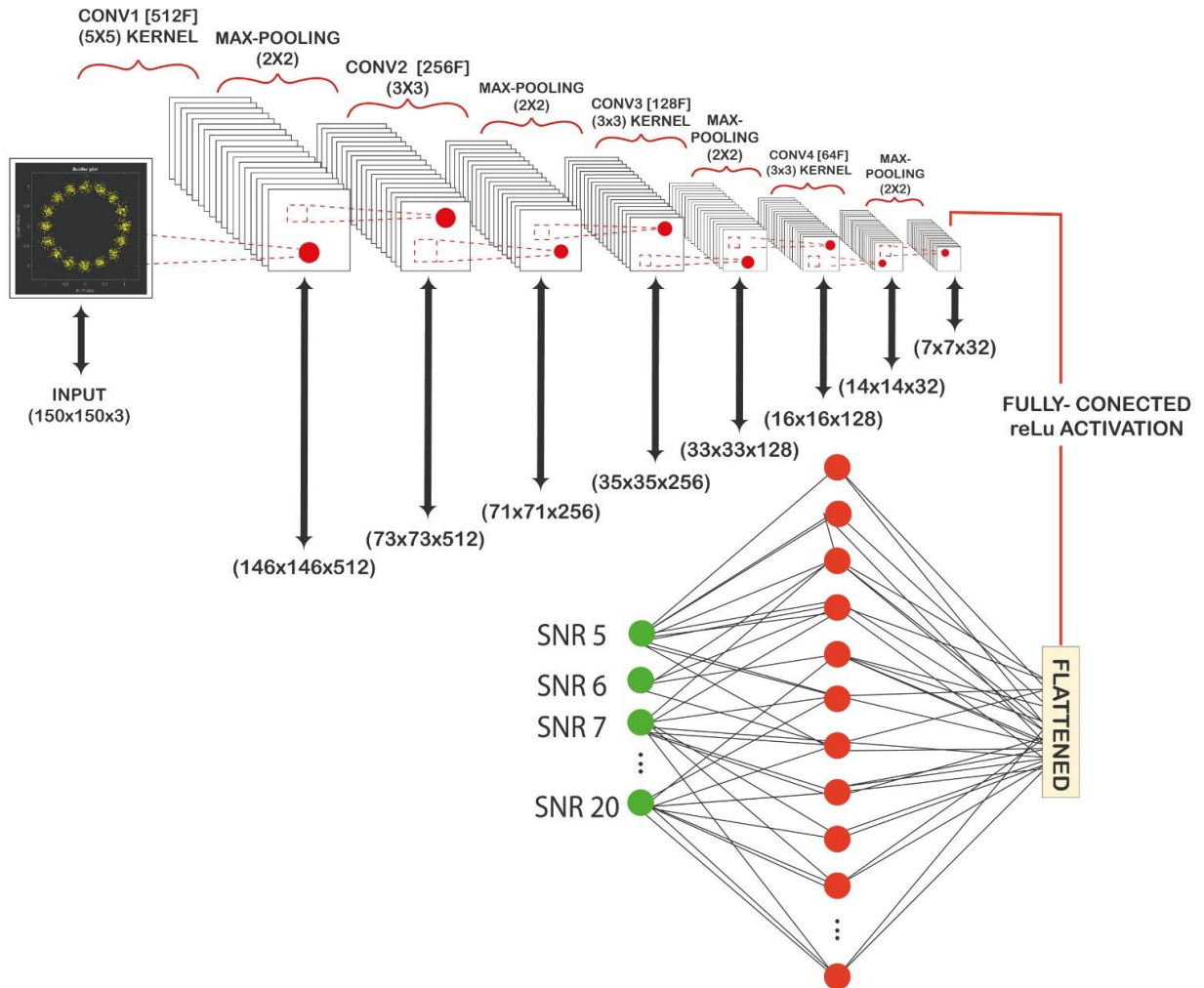


Fig. 24 Diseño de CNN Multiclase estimador calidad de señal para 16-PSK [34].

5) Modulacion 64-PSK SNRs:

Se utilizaron 16.000 imágenes de entrenamiento, 8.000 imágenes de validación y 8.000 imágenes de test. De la misma forma que en los clasificadores anteriores se agregan las librerías, se montaron las bases de datos en Google drive y se unieron las carpetas.

Todo el preprocesado de las imágenes como la redimensión, batch size, augmentation, etc.... es idéntico a la modulación 16-PSK SNRs.

Se crea el modelo, utilizando 3 módulos principales, la convolución, función de activación ReLu y capa MaxPooling. La arquitectura queda de la siguiente manera:

1era etapa: Capa convolucional de 512 filtros, con un kernel de 5x5, una capa MaxPooling 2D de 2x2 y una normalización del batch.

2da etapa: Capa convolucional de 256 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

3era etapa: Capa convolucional de 128 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

4ta etapa: Capa convolucional de 32 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

Se aplana con la función “Flatten”, luego se utiliza la función de activación ReLu, con un “Dropout” de 0,6 y se termina el modelo especificando la cantidad de clases (16) que en este caso son con una función de activación “softmax”.

Se usa un optimizador y un “callback”, se compila el modelo y se entrena con epochs igual a 39.

64-PSK

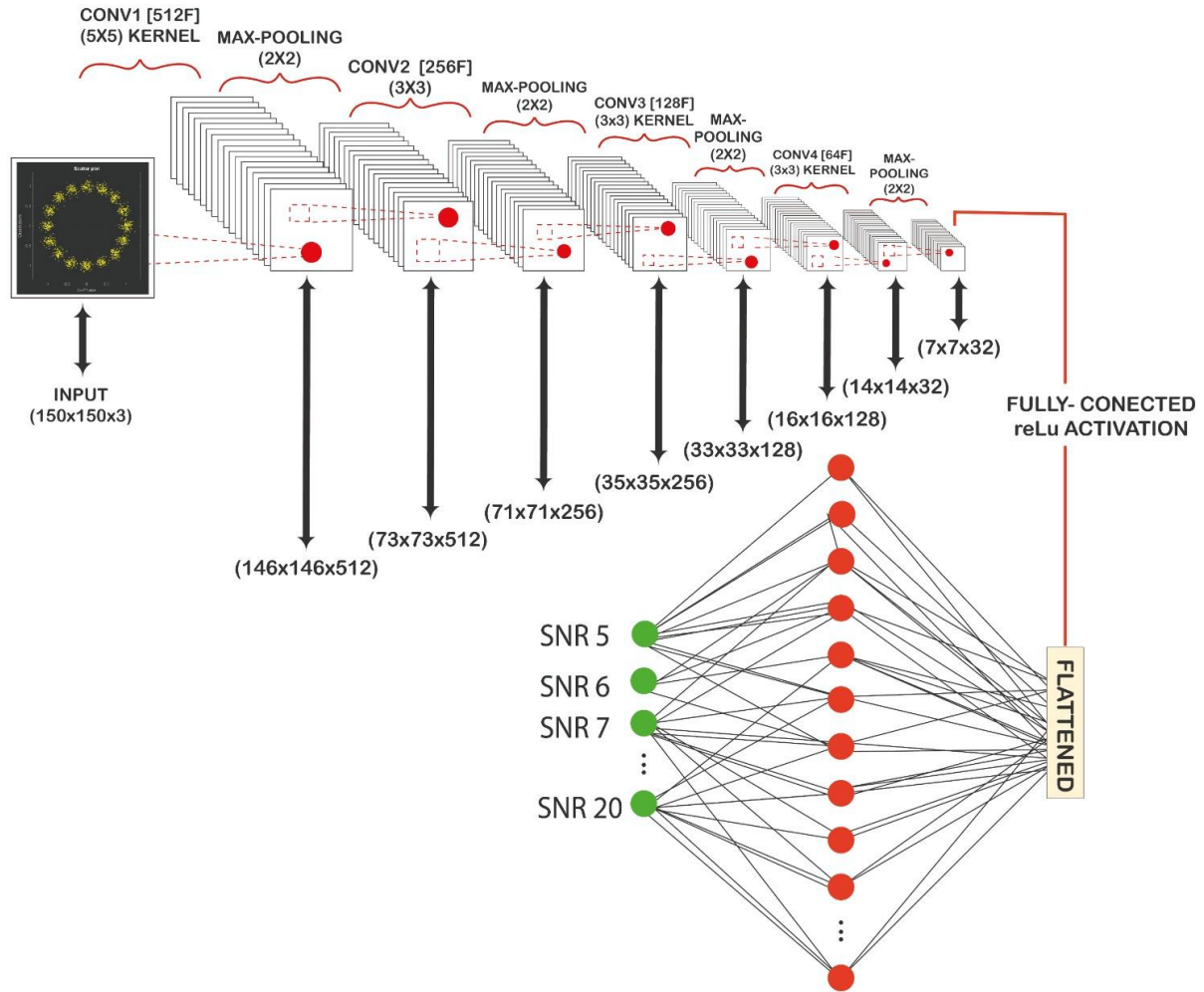


Fig. 25 Diseño de CNN Multiclase estimador calidad de señal para 64-PSK [34].

6) Modulación 256-PSK SNRs:

Se utilizaron 16.000 imágenes de entrenamiento, 8.000 imágenes de validación y 8.000 imágenes de test. De la misma forma que en los clasificadores anteriores se agregan las librerías, se montaron las bases de datos en Google drive y se unieron las carpetas.

Todo el preprocesado de las imágenes como la redimensión, batch size, augmentation, etc.... es idéntico a la modulación 16-PSK SNRs.

Se crea el modelo, utilizando 3 módulos principales, la convolución, función de activación ReLu y capa MaxPooling. La arquitectura queda de la siguiente manera:

1era etapa: Capa convolucional de 512 filtros, con un kernel de 5x5, una capa MaxPooling 2D de 2x2 y una normalización del batch.

2da etapa: Capa convolucional de 256 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch

3era etapa: Capa convolucional de 128 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

4ta etapa: Capa convolucional de 32 filtros, con un kernel de 3x3, una capa MaxPooling 2D de 2x2 y una normalización del batch.

Se aplana con la función “Flatten”, luego se utiliza la función de activación ReLu, con un “Dropout” de 0,6 y se termina el modelo especificando la cantidad de clases (16) que en este caso son con una función de activación “softmax”.

Se usa un optimizador y un “callback”, se compila el modelo y se entrena con epochs igual a 39.

256-PSK

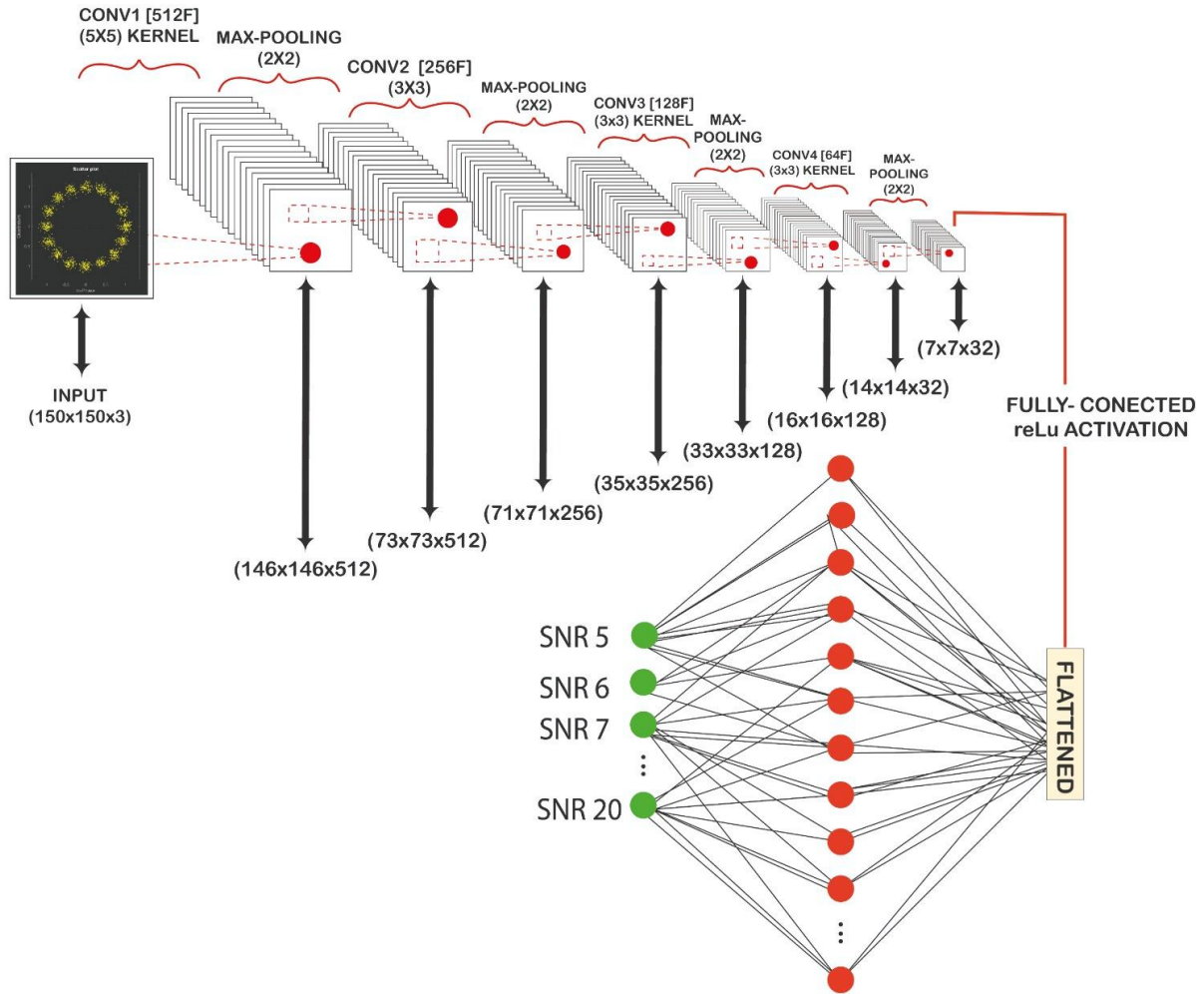


Fig. 26 Diseño de CNN Multiclase estimador calidad de señal para 256-PSK [34]

6. Resultados

6.1 Clasificación binaria QAM y PSK

La base de datos para el 1er algoritmo de clasificación binaria es más de 10.000 imágenes, por esta razón se mostrarán solo 6 por cada tipo de modulación con sus respectivos parámetros, a continuación, las 16/64/256M - QAM:

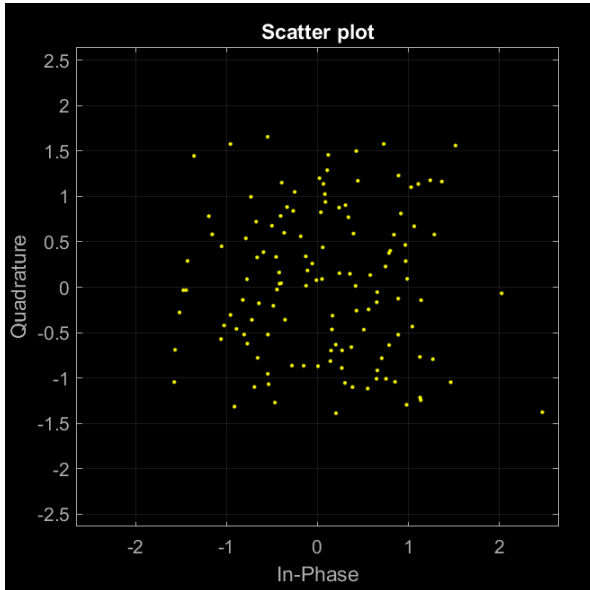


Fig. 27. Modulación 16-QAM, 128 bits y SNR = 5 [dB]

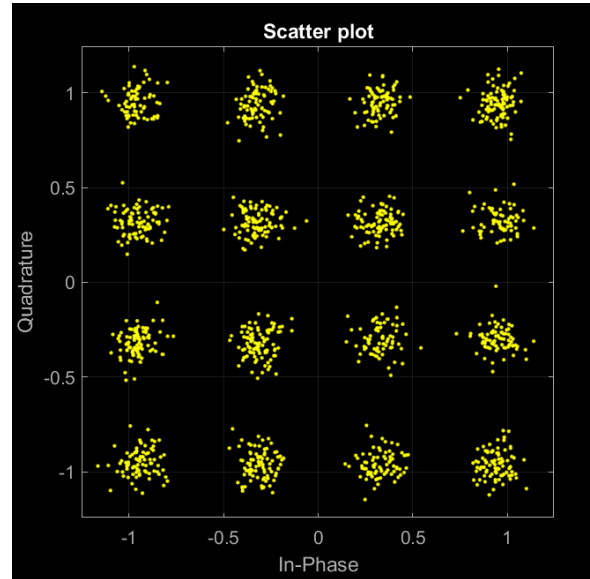


Fig. 28. Modulación 16-QAM, 1296 bits y SNR = 20 [dB]

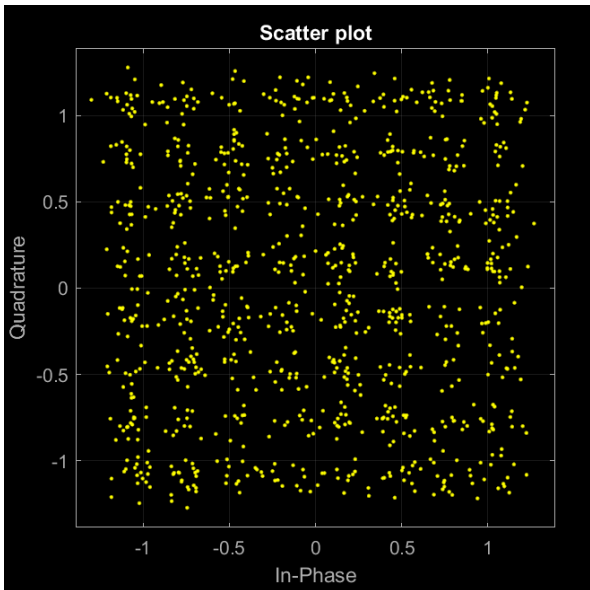


Fig. 29 Modulación 64-QAM, 960 bits y SNR = 20 [dB]

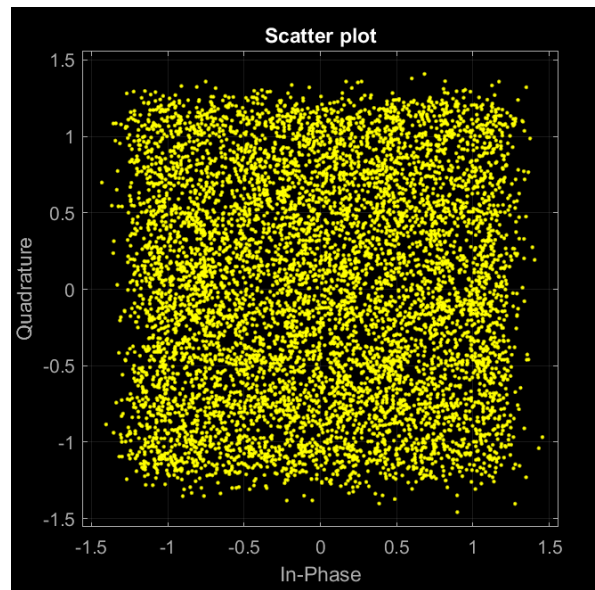


Fig. 30 Modulación 64-QAM, 7488 bits y SNR = 16 [dB]

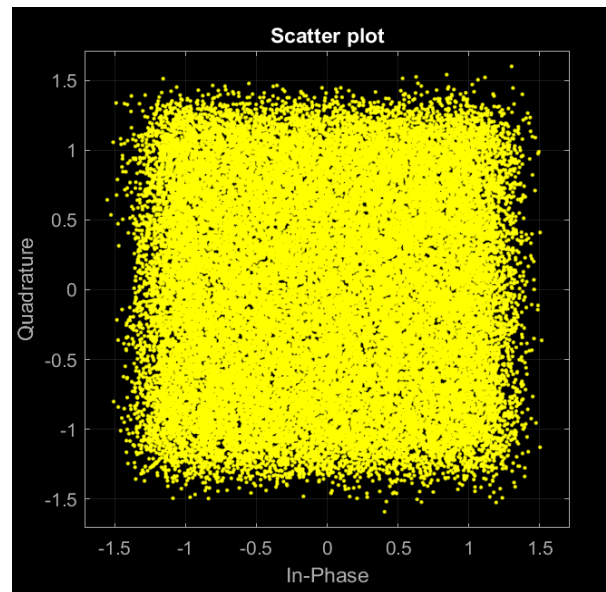
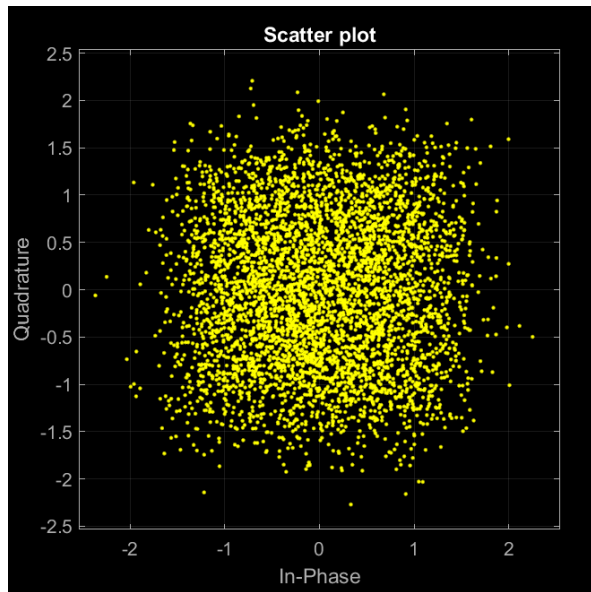


Fig. 31 Modulación 256-QAM, 4608 bits y SNR = 6 [dB] **Fig. 32** Modulación 256-QAM, 4608 bits y SNR = 6 [dB]

Imágenes de 16/64/256 - PSK:

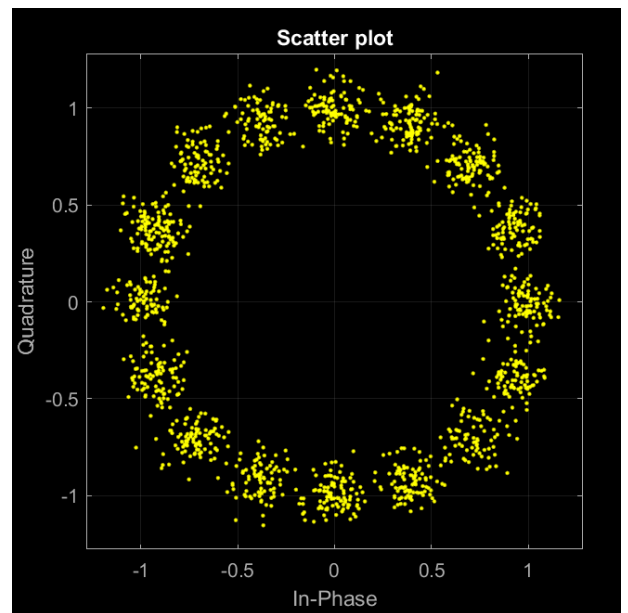
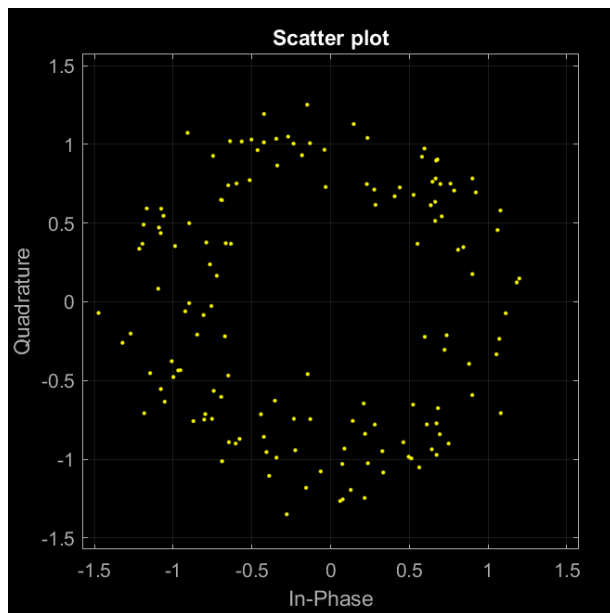


Fig. 33 Modulación 16-PSK, 144 bits y SNR = 11 [dB]

Fig. 34 Modulación 16-PSK, 1456 bits y SNR = 19 [dB]

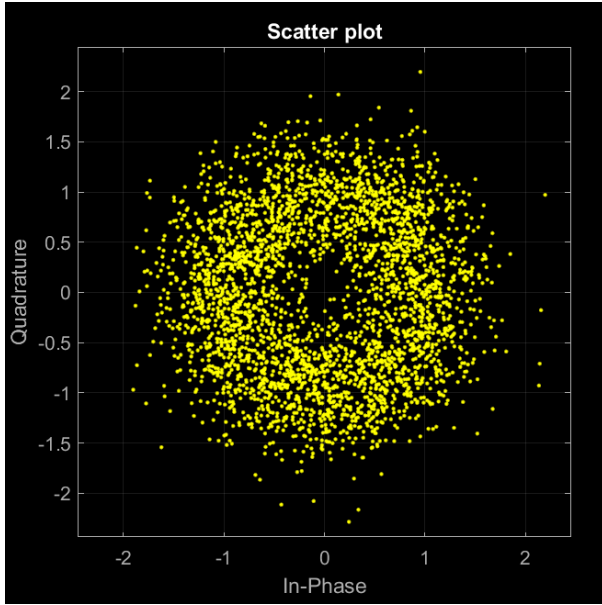


Fig. 35 Modulación 64-PSK, 3264 bits y SNR = 6 [dB]

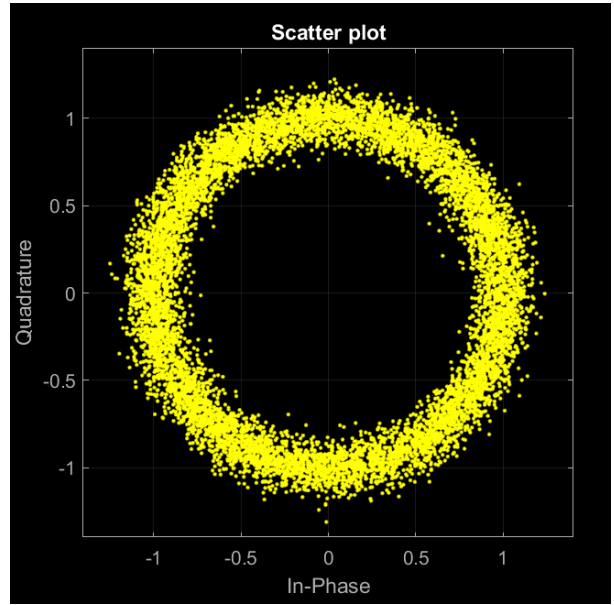


Fig. 36 Modulación 64-PSK, 9120 bits y SNR = 19 [dB]

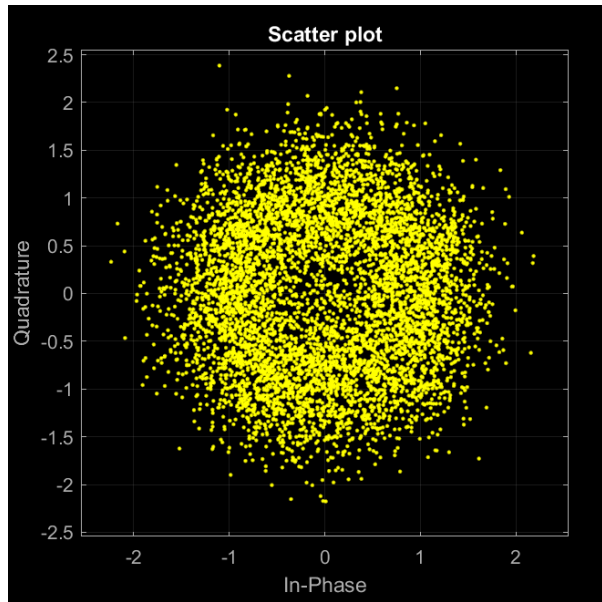


Fig. 37 Modulación 256-PSK, 6144 bits y SNR = 5 [dB]

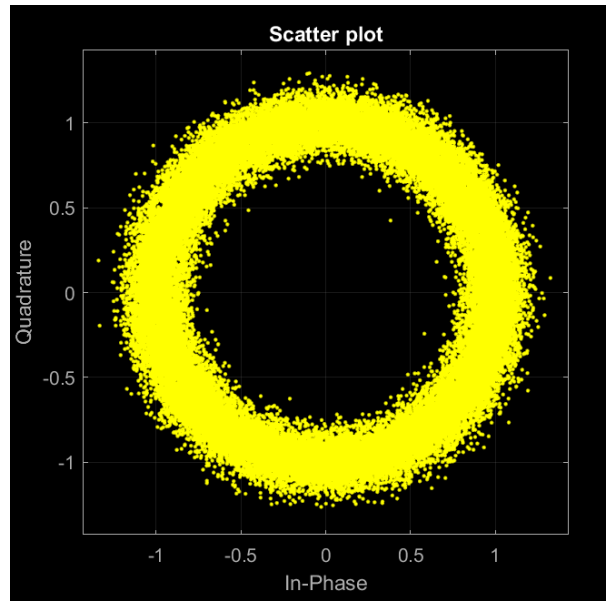


Fig. 38 Modulación 256-PSK, 41472 bits y SNR = 18 [dB]

El algoritmo creado se entrenó durante 20 épocas (epoch) obteniéndose los siguientes resultados de precisión y pérdida de entrenamiento y validación:

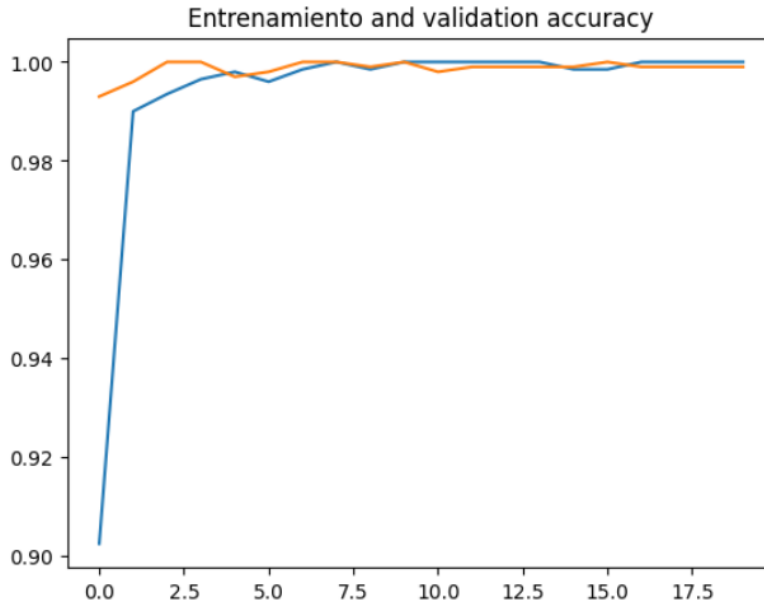


Fig. 39 Gráfico de cantidad de epoch vs precisión en porcentaje de entrenamiento y validación CNN binario [32].

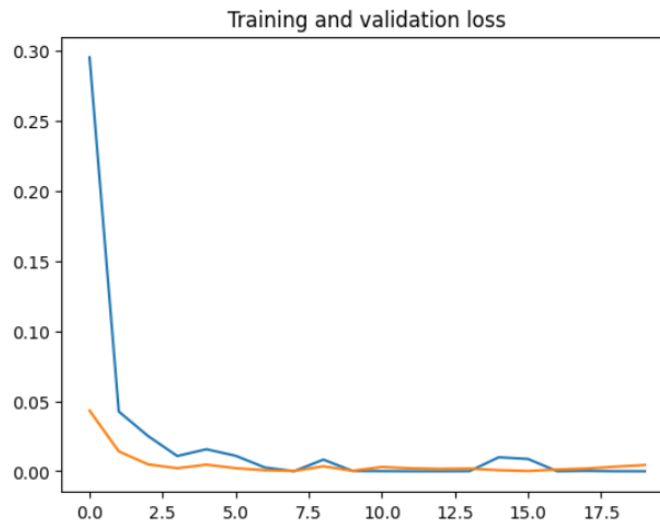


Fig. 40 Gráfico de cantidad de epoch vs perdida en porcentaje de entrenamiento y validación CNN binario [34].

Para cada gráfico la línea azul corresponde al entrenamiento y la línea naranja a la validación.

Como se puede observar en la Fig. 39 tanto el entrenamiento como la validación alcanzaron el 100%, mientras que en la Fig. 40 el train loss llego a cero y el validation loss muy cercano a cero. Cabe destacar que el tiempo de ejecución del modelo entrenado fue de aproximadamente 30 minutos.

6.2 Clasificador multiclase CNN para diferentes formatos de modulación

6.2.1 Clasificador multiclase CNN para diferentes formatos de modulación de QAM.

El algoritmo creado se entrenó durante 50 épocas (epoch) obteniéndose los siguientes resultados de precisión y pérdida de entrenamiento y validación:

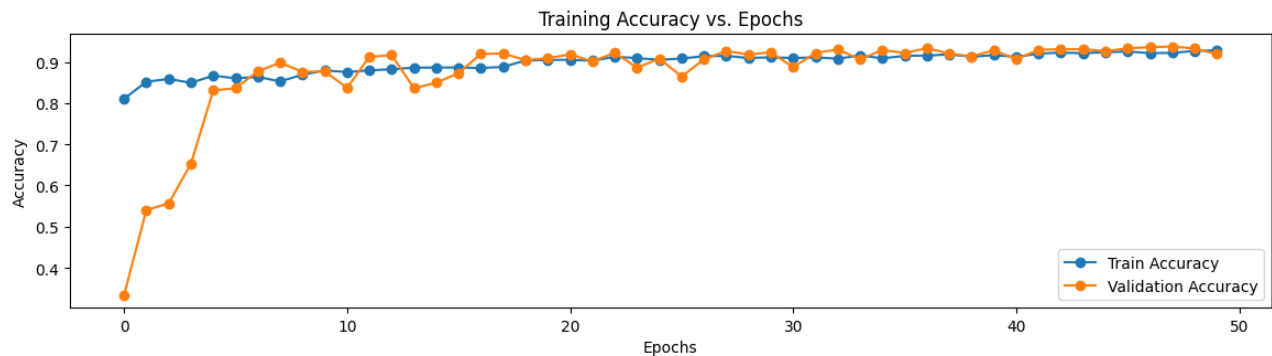


Fig. 41 Gráfico de cantidad de epoch vs precisión en porcentaje de entrenamiento y validación CNN multiclase QAM [34].

Final Train Accuracy: 0,9167

Final validation Accuracy: 0.9067

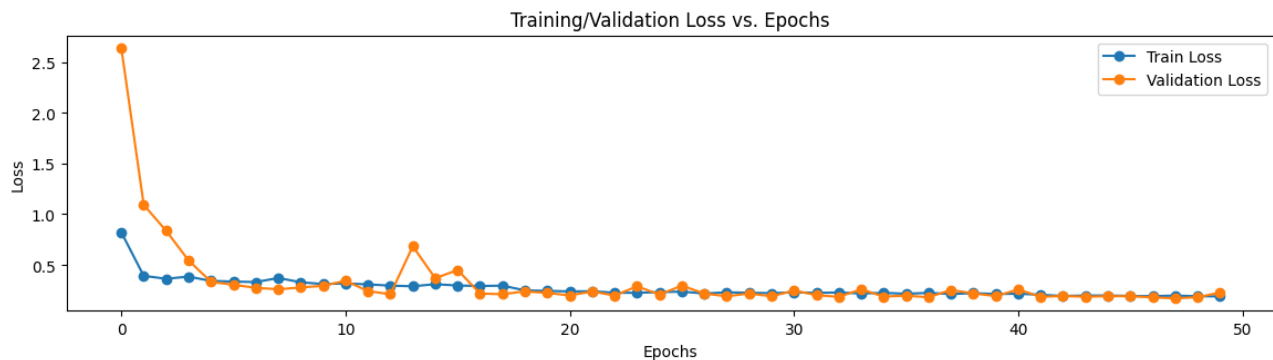


Fig. 42 Gráfico de cantidad de epoch vs pérdida en porcentaje de entrenamiento y validación CNN multiclase QAM [34].

Final Train Loss: 0,2294

Final validation loss: 0,2328

En la Fig. 43 se tiene una imagen de los resultados de la red CNN con los datos de test, la que representa visualmente más datos, conocida como la matriz de confusión. El eje vertical o columnas se observan las clases originales, mientras que en el horizontal o filas las clases predichas por el CNN. Para todas las matrices de confusión la barra vertical de la derecha posee un rango de números que parte de cero, donde en el caso de la Fig. 43 va de 0 a 600. Esta representa en unidades el número de imágenes de constelación, por lo que dependiendo de la cantidad de imágenes que posea una clase se le asocia el color correspondiente desde el más claro blanco/amarillo al azul más intenso/oscuero.

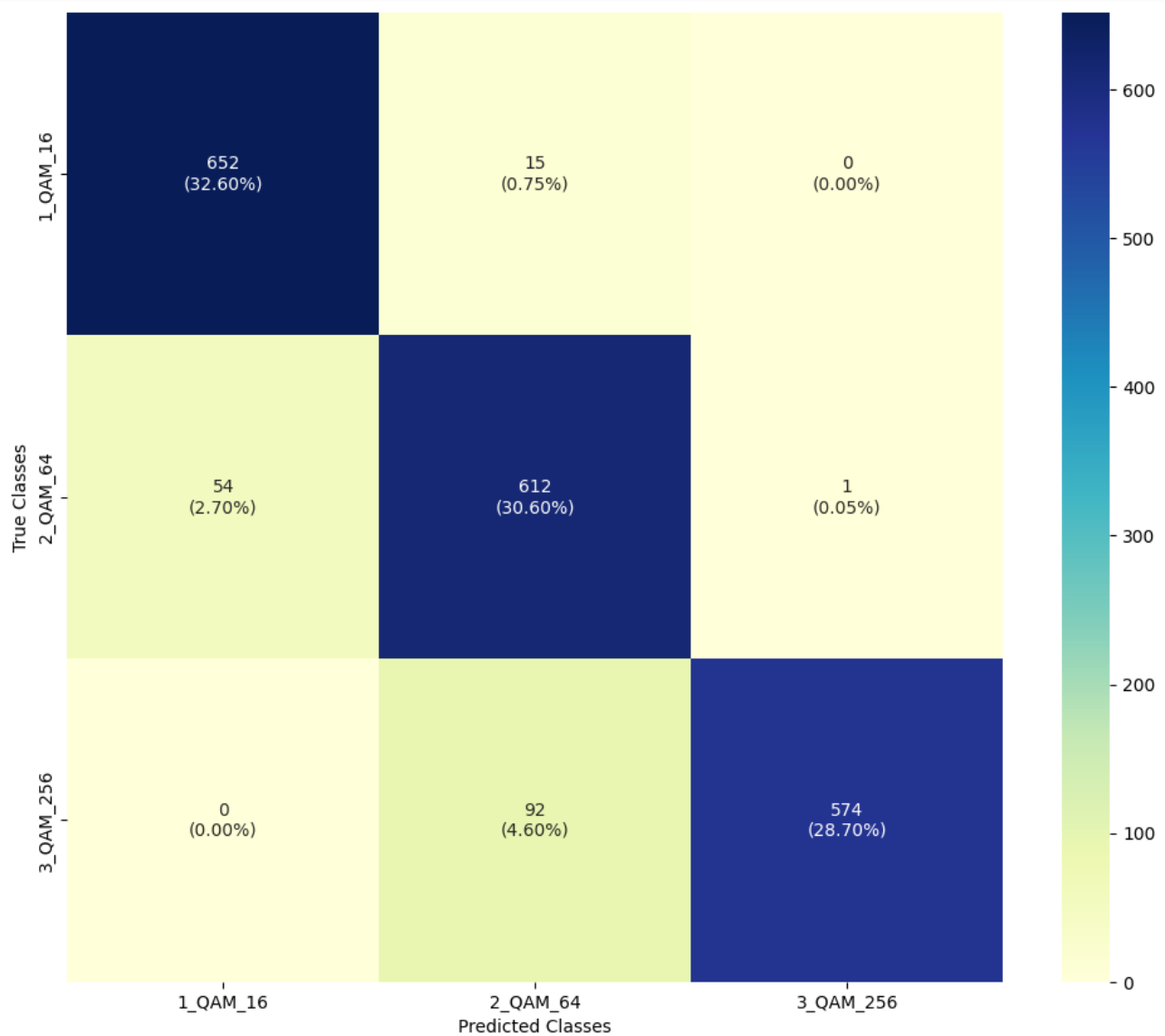


Fig. 43 Matriz de confusión de CNN multiclase 16/64/256-QAM [34].

En la Fig. 44 se observan los resultados de la precisión de la clasificación por cada clase, en este tipo de gráficos se aprecia el espectro de la peor a la mejor calidad de señal, es decir, de la que tiene menor SNR (SNR de 5) hasta la con mayor SNR (SNR de 20).

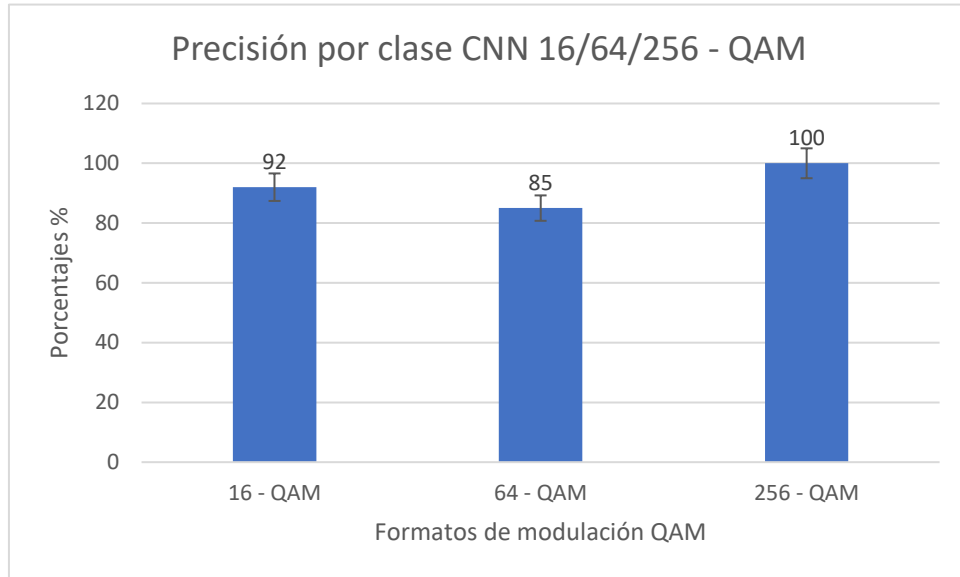


Fig. 44 Precisión por clase CNN multiclase 16/64/256-QAM [34].

Con estos datos se obtiene una precisión general en este modelo del 92%.

6.2.2 Clasificador multiclase CNN para diferentes formatos de modulación de PSK.

El algoritmo creado se entrenó durante 50 épocas (epoch) obteniéndose los siguientes resultados de precisión y pérdida de entrenamiento y validación:

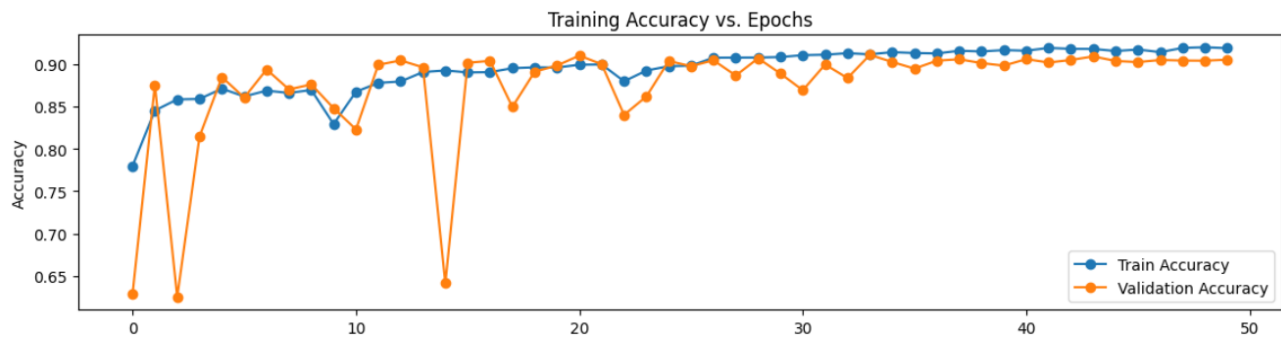


Fig. 45 Gráfico de cantidad de epoch vs precisión en porcentaje de entrenamiento y validación CNN multiclase PSK [34].

Final Train Accuracy: 0,9186

Final validation Accuracy: 0.9050

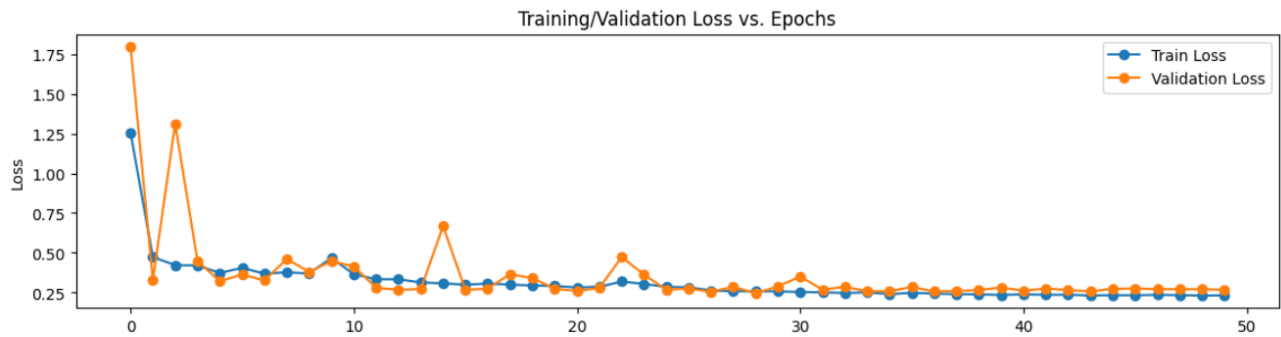


Fig. 46 Gráfico de cantidad de epoch vs pérdida en porcentaje de entrenamiento y validación CNN multiclase PSK [34].

Final Train Loss: 0,2324

Final validation loss: 0,2669

En la Fig. 47 se tiene la imagen de la matriz de confusión. El eje vertical se observan las clases originales, mientras que en el horizontal las clases predichas por el CNN.

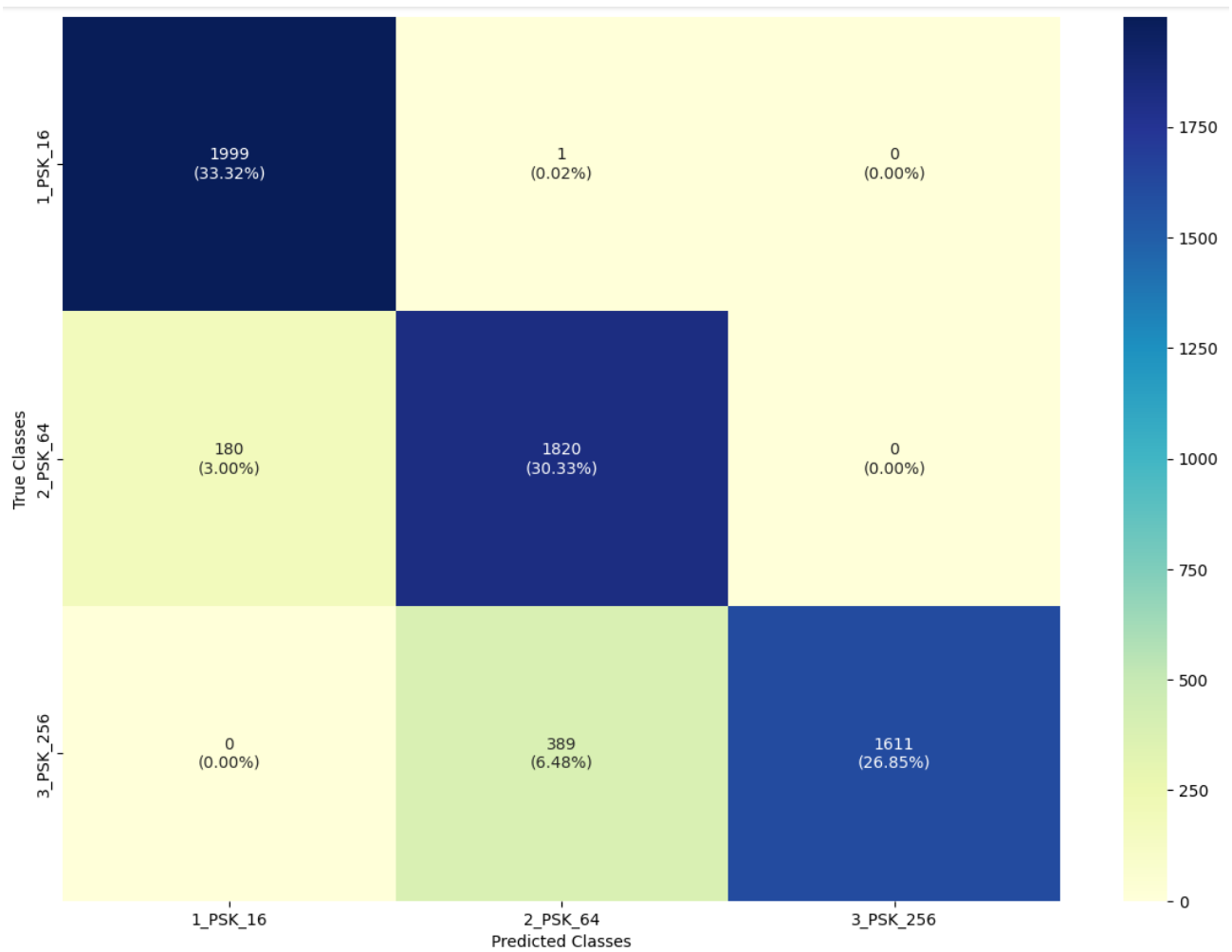


Fig. 47 Matriz de confusión de CNN multiclase 16/64/256-PSK [34].

En la Fig. 48 se observan los resultados de la precisión de la clasificación por cada clase, nuevamente y en todos los siguientes resultados de CNN en estos tipos de gráficos se aprecia el espectro de la peor a la mejor calidad de señal, es decir, de la que tiene menor SNR (SNR de 5) hasta la con mayor SNR (SNR de 20).

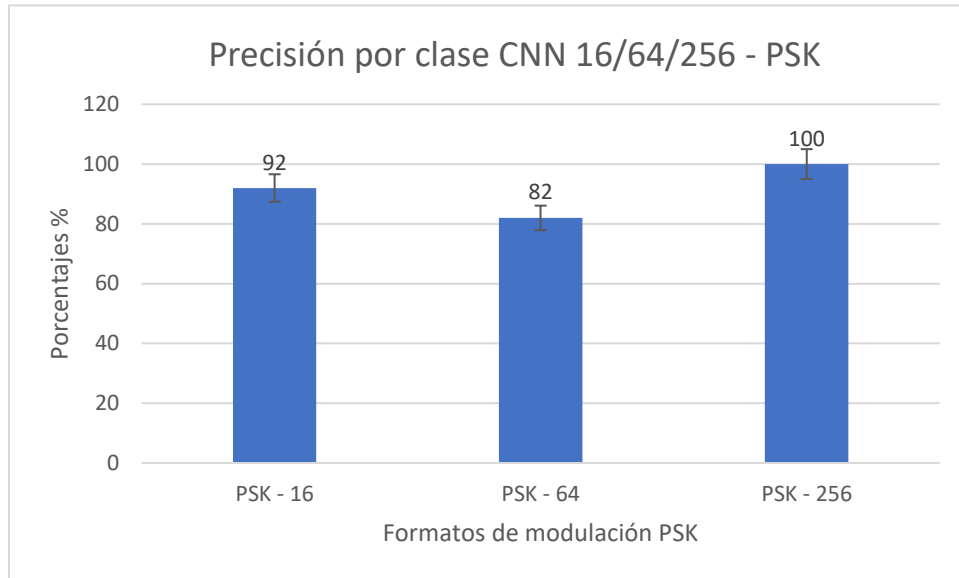


Fig. 48 Precisión por clase CNN multiclase 16-64-256-PSK [34].

Con estos datos se obtiene una precisión general en este modelo del 91%.

6.3 Clasificador multiclase CNN estimador de la calidad óptica de la señal en base al SNR.

Como se describió en la sección de metodología este clasificador es más complejo que los anteriores, utilizando en general más filtros, mayor cantidad de etapas y atención especial en regular el proceso de entrenamiento de la red neuronal.

6.3.1 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 16-QAM.

La red creada se entrenó durante 50 épocas (epoch) obteniéndose los siguientes resultados de precisión y pérdida de entrenamiento y validación:

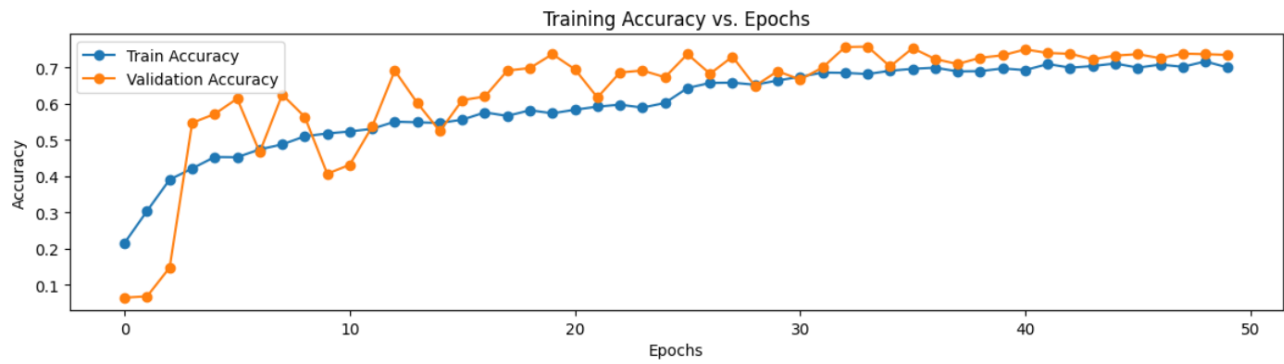


Fig. 49. Gráfico de cantidad de epoch vs precisión en porcentaje de entrenamiento y validación CNN multiclase 16-QAM [34].

Final Train Accuracy: 0,6998

Test validation Accuracy: 0.7346

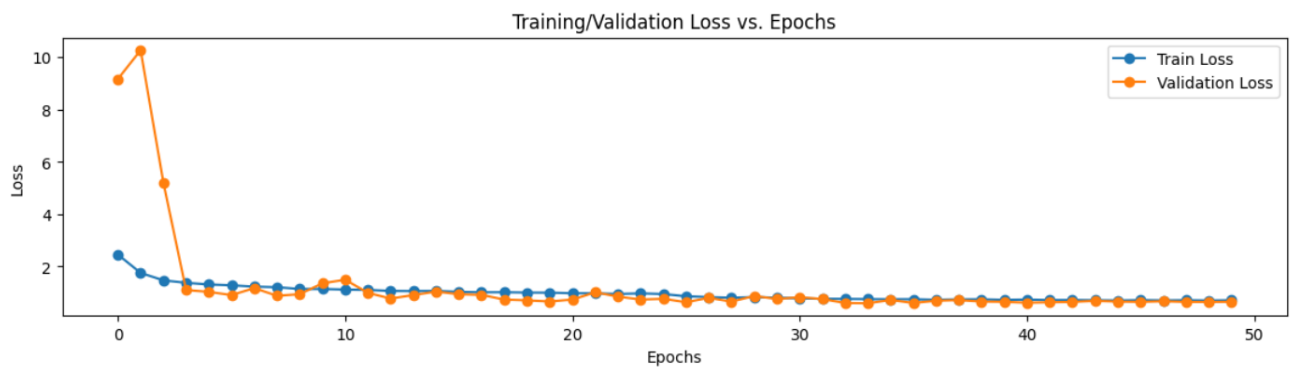


Fig. 50 Gráfico de cantidad de epoch vs pérdida en porcentaje de entrenamiento y validación CNN multiclase 16-QAM [34].

Final Train Loss: 0,7152

Final validation loss: 0,6556

En la Fig. 51 se tiene la imagen de una matriz de confusión que muestra resultados de la modulación 16-QAM, que es más grande y con más datos que las anteriores, debido al aumento de clases. En el eje vertical se observan las clases originales, mientras que en el horizontal las clases predichas por el CNN.

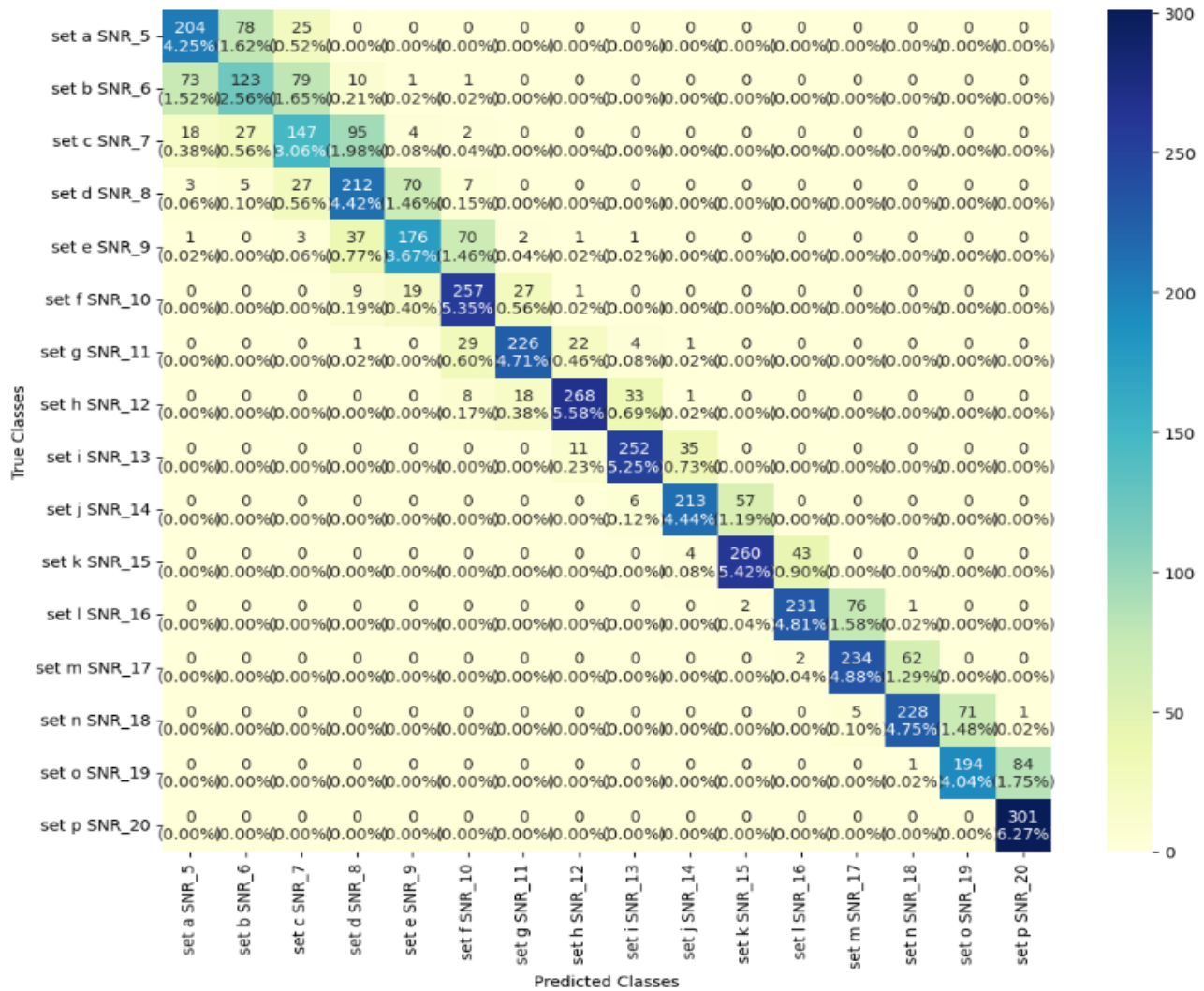


Fig. 51 Matriz de confusión de clases o rangos de SNR (5-20) predichas vs clases reales para red multiclasa 16-QAM [34].

En la Fig. 52 se observan los resultados de la precisión de la clasificación por cada clase, mostrando de la peor a la mejor calidad de señal. Además, para los gráficos de precisión por clase desde esta red en adelante se agrega una línea (puntos) de tendencia lineal de color verde.

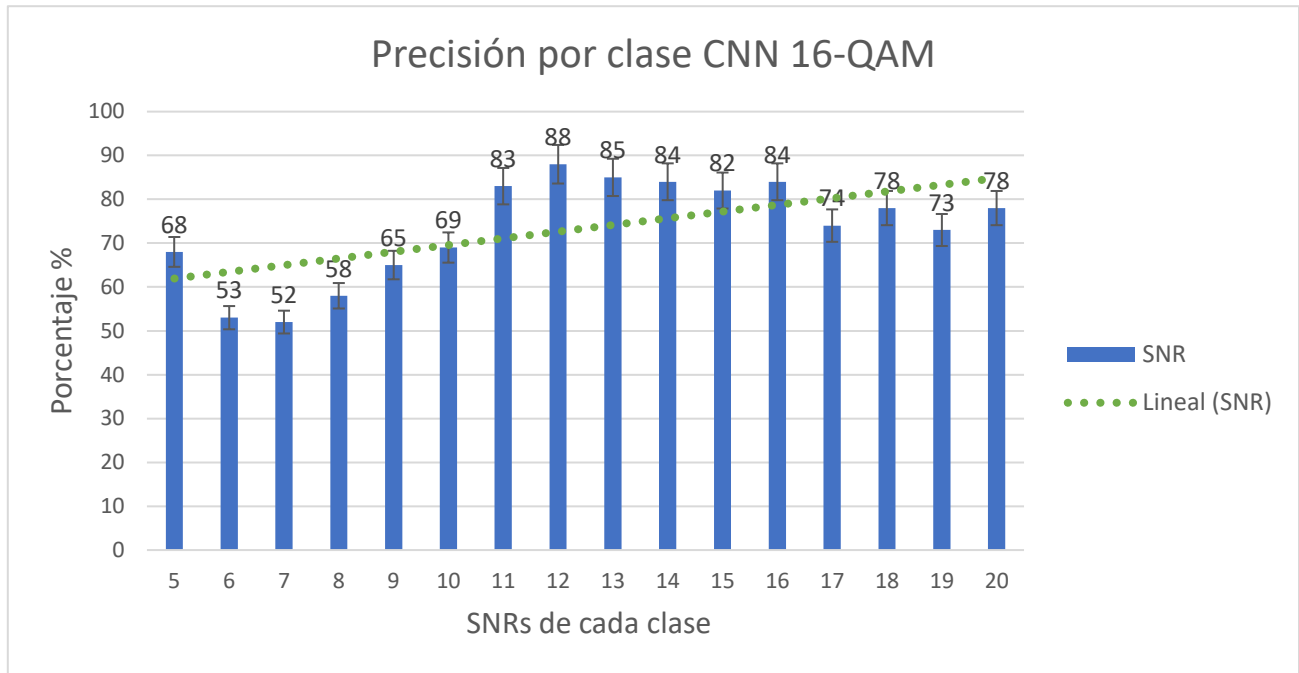


Fig. 52 Precisión por clase CNN multiclase 16-QAM [34].

Con estos datos se obtiene una precisión general en este modelo del 73%.

6.3.2 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 64-QAM.

La red se entrenó durante 39 épocas (epoch) obteniéndose los siguientes resultados de precisión y pérdida de entrenamiento y validación:

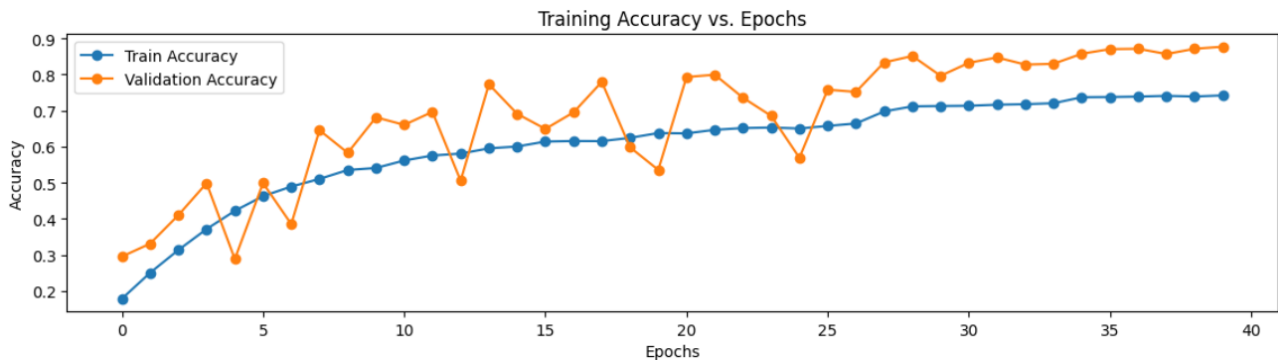


Fig. 53. Gráfico de cantidad de epoch vs precisión en porcentaje de entrenamiento y validación CNN multiclase 64-QAM [34].

Final Training Accuracy: 0,7417

Final validation Accuracy: 0,8766

Test Accuracy: 0,8765

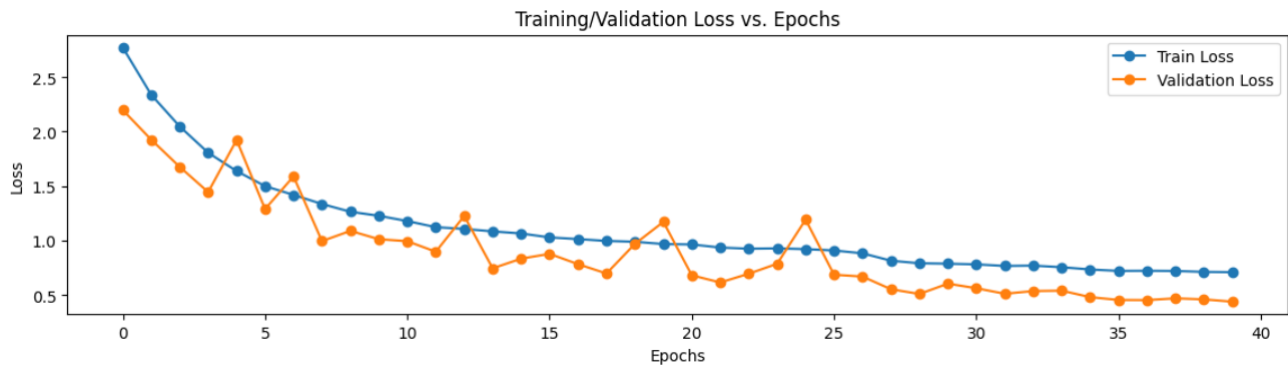


Fig. 54. Gráfico de cantidad de epoch vs pérdida en porcentaje de entrenamiento y validación CNN multiclase 64-QAM [34].

Final Train Loss: 0,7080

Final validation Loss: 0,4368

Test Loss: 0.4367

En la Fig. 55 se tiene la imagen de una matriz de confusión que muestra los resultados de la modulación 64-QAM. En el eje vertical se observan las clases originales, mientras que en el horizontal las clases predichas por el CNN.

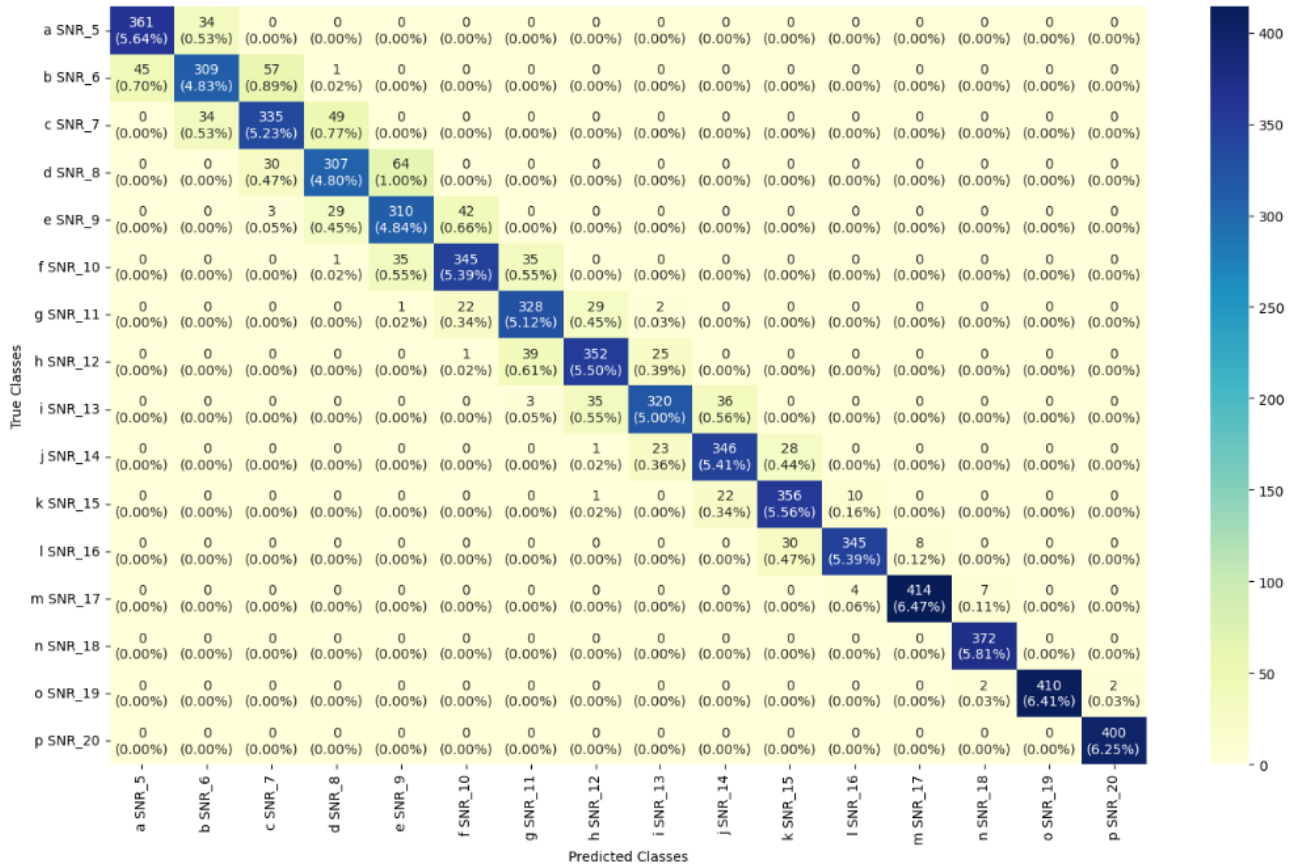


Fig. 55 Matriz de confusión de clases o rangos de SNR (5-20) predichas vs clases reales para red multiclase 64-QAM [34].

En la Fig. 56 se observan los resultados de la precisión de la clasificación por cada clase, mostrando de la peor a la mejor calidad de señal y su línea de tendencia.

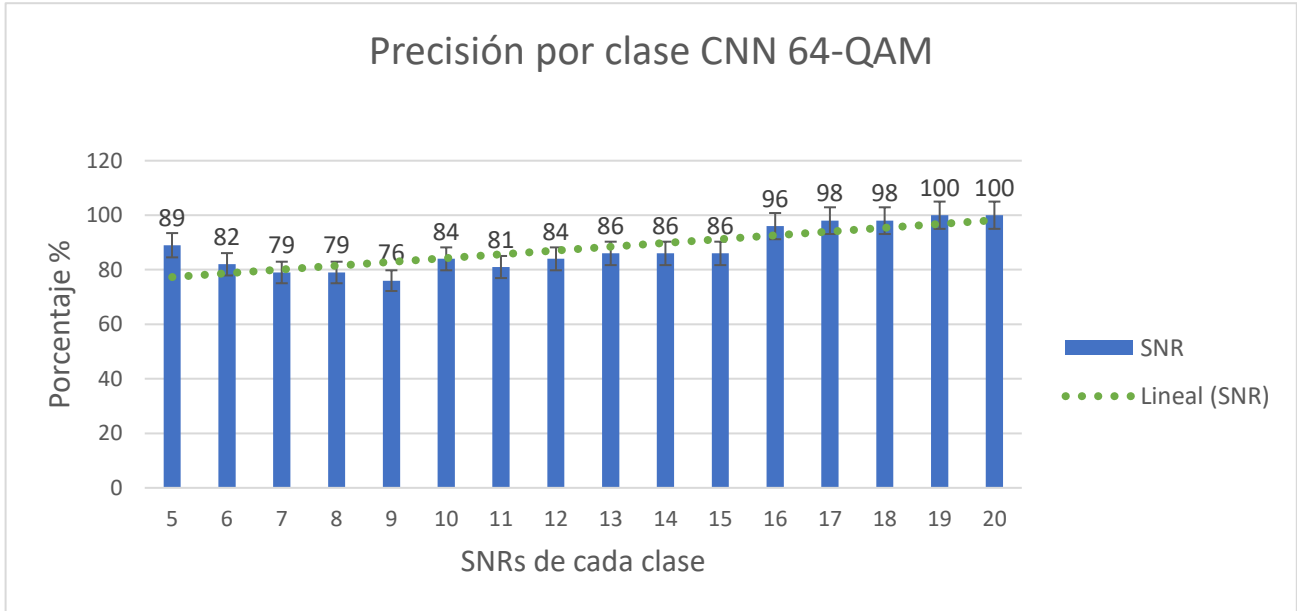


Fig. 56 Precisión por clase CNN multiclase 64-QAM [34].

Con estos datos se obtiene una precisión general en este modelo del 88%.

6.3.3 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 256-QAM.

La red se entrenó durante 39 épocas (epoch) obteniéndose los siguientes resultados de precisión y pérdida de entrenamiento y validación:

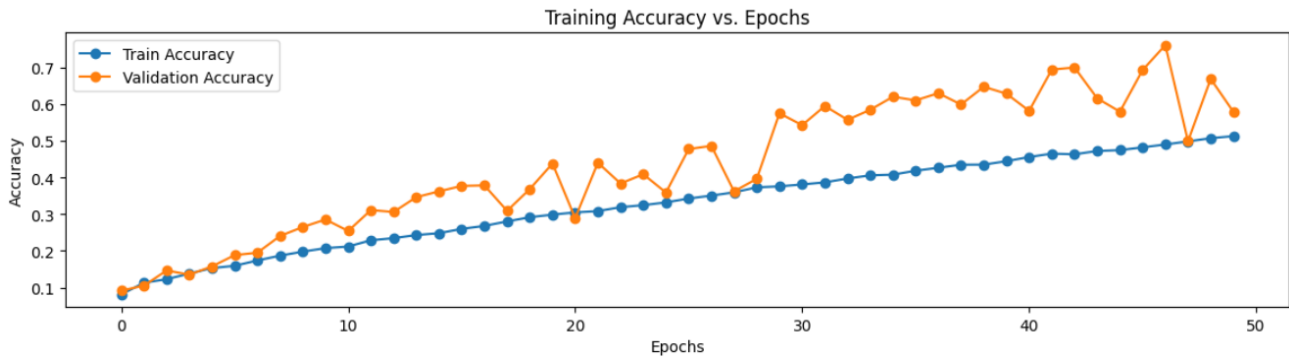


Fig. 57. Gráfico de cantidad de epoch vs precisión en porcentaje de entrenamiento y validación CNN multiclase 256-QAM [34].

Final Training Accuracy: 0,5126

Final validation Accuracy: 0,5786

Test Accuracy: 0,5784

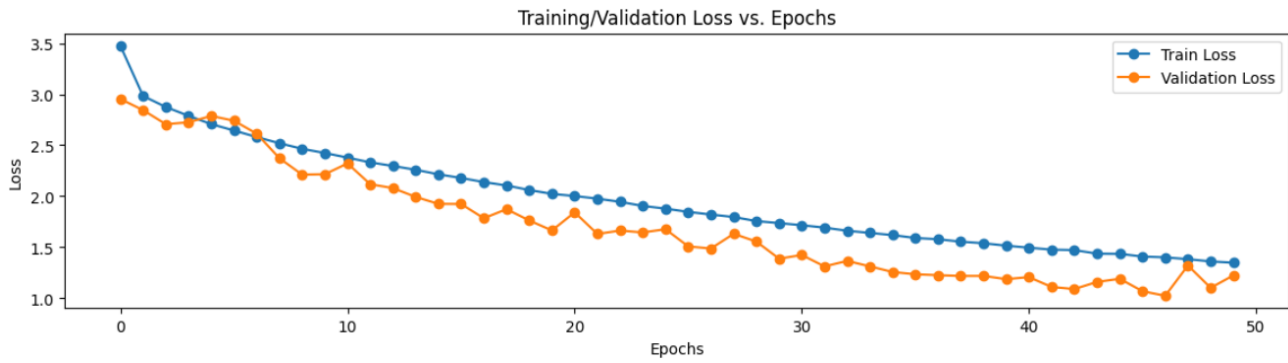


Fig. 58. Gráfico de cantidad de epoch vs pérdida en porcentaje de entrenamiento y validación CNN multiclase 256-QAM [34].

Final Train Loss: 1,3485

Final validation Loss: 1,2239

Test Loss: 1,2239

En la Fig. 59 se tiene la imagen de una matriz de confusión. En el eje vertical se observan las clases originales, mientras que en el horizontal las clases predichas por el CNN.

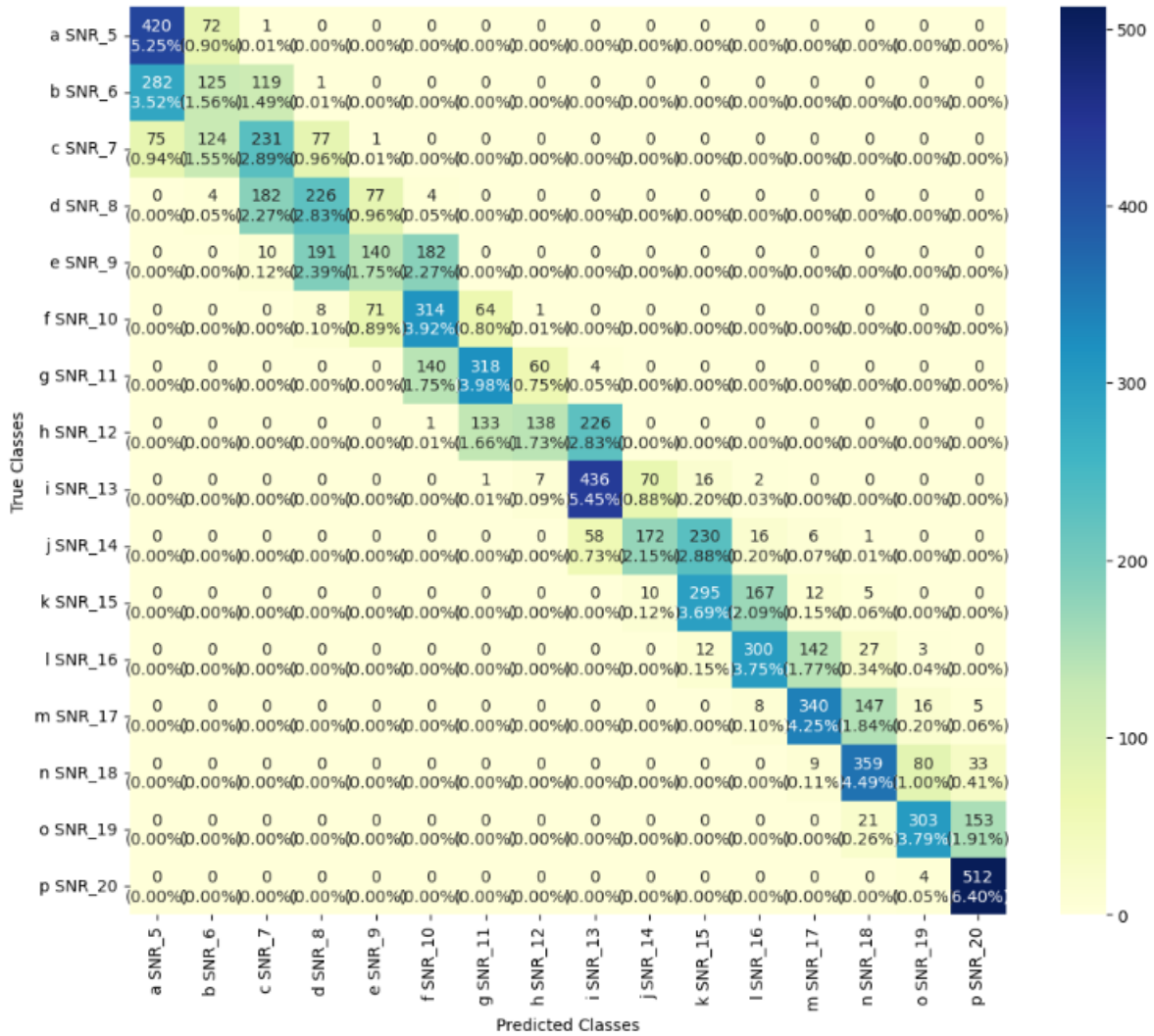


Fig. 59 Matriz de confusión de clases o rangos de SNR (5-20) predichas vs clases reales para red multiclasa 256-QAM [34].

En la Fig. 60 se observan los resultados de la precisión de la clasificación por cada clase, mostrando de la peor a la mejor calidad de señal y su línea de tendencia.

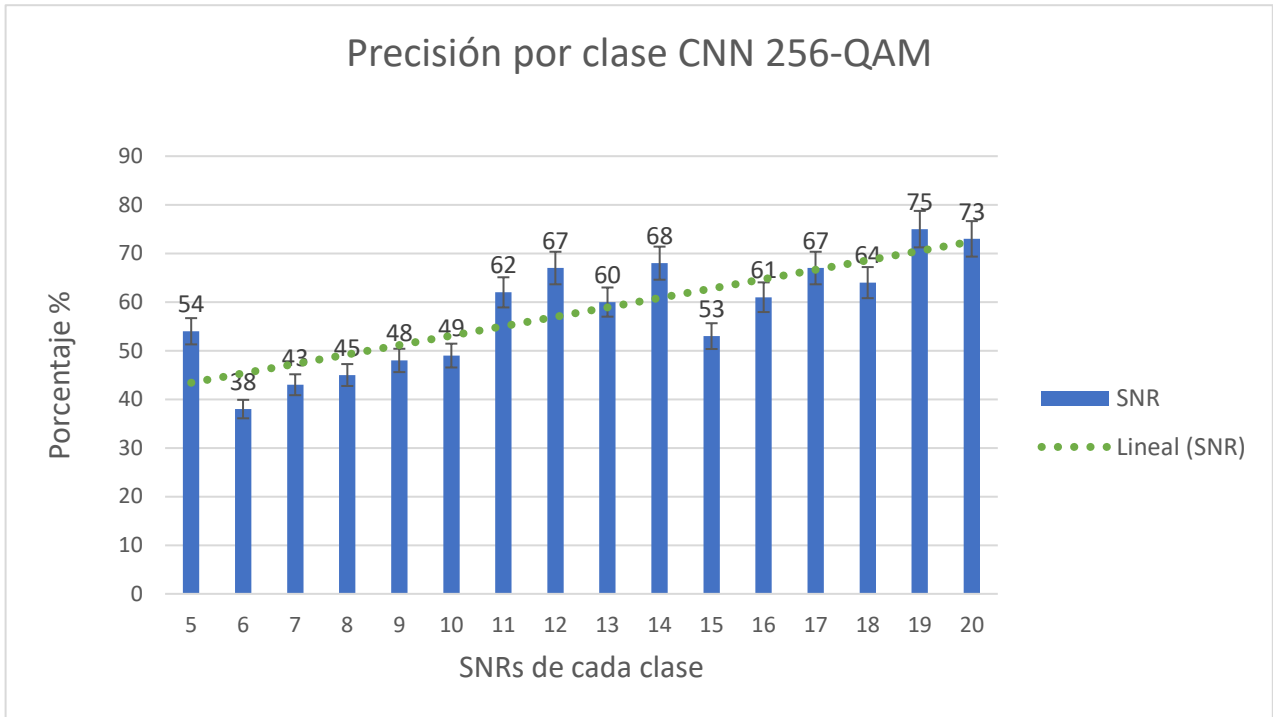


Fig. 60 Precisión por clase CNN multiclase 256-QAM [34].

Con estos datos se obtiene una precisión general en este modelo del 58%.

6.3.4 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 16-PSK.

La red se entrenó durante 39 épocas (epoch) obteniéndose los siguientes resultados de precisión y pérdida de entrenamiento y validación:

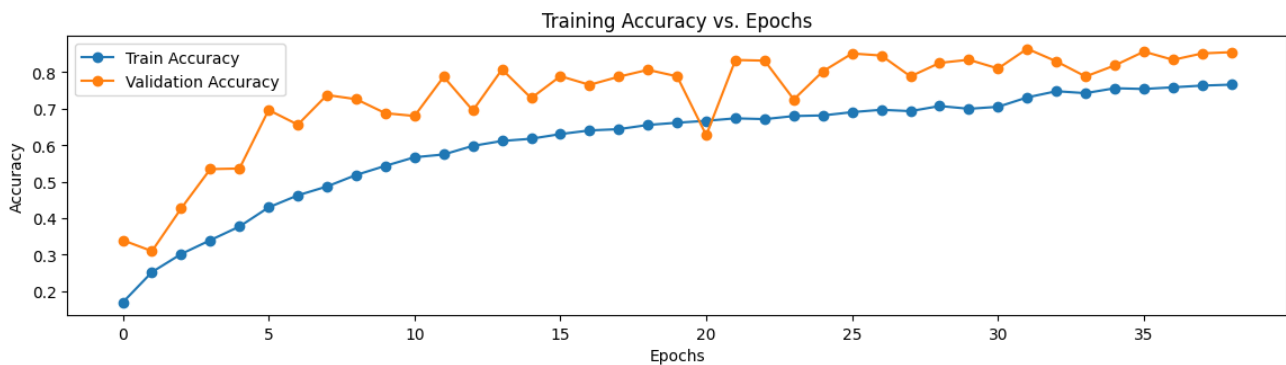


Fig. 61. Gráfico de cantidad de epoch vs precisión en porcentaje de entrenamiento y validación CNN multiclase 16-PSK [34].

Final Training Accuracy: 0,7656

Final validation Accuracy: 0,8545

Test Accuracy: 0,8544

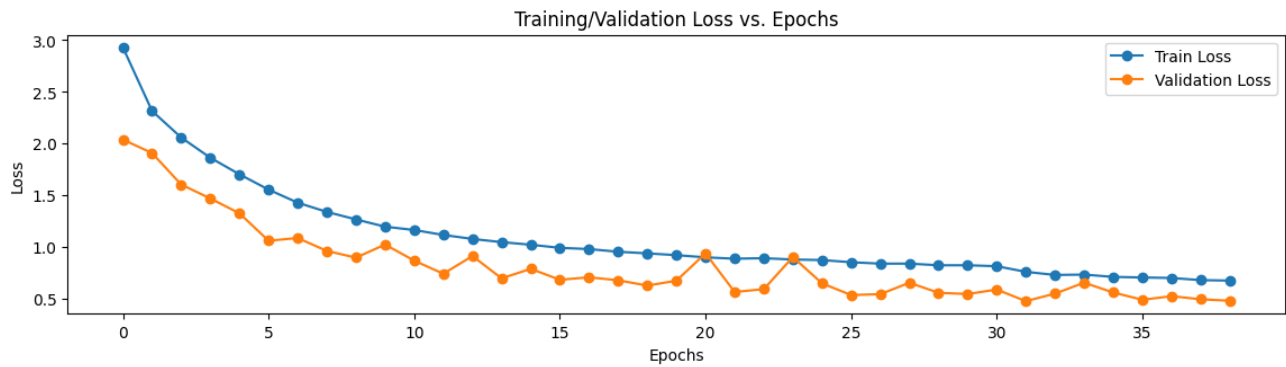


Fig. 62. Gráfico de cantidad de epoch vs pérdida en porcentaje de entrenamiento y validación CNN multiclase 16-PSK [34].

Final Train Loss: 0,6752

Final validation Loss: 0,4805

Test Loss: 0.4804

En la Fig. 63 se tiene la imagen de una matriz de confusión que muestra los resultados de la modulación 16-PSK. En el eje vertical se observan las clases originales, mientras que en el horizontal las clases predichas por el CNN.

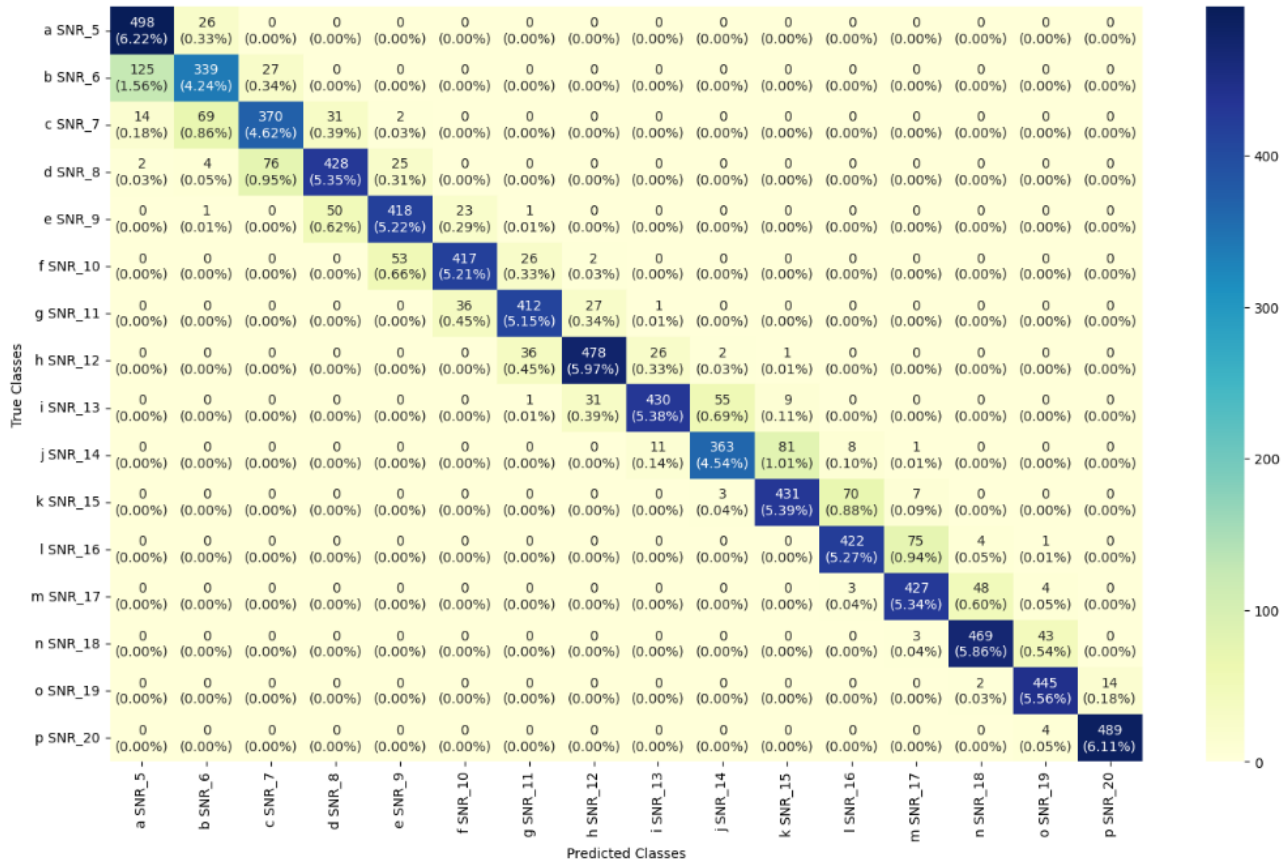


Fig. 63 Matriz de confusión de clases o rangos de SNR (5-20) predichas vs clases reales para red multiclasa 16-PSK [34].

En la Fig. 64 se observan los resultados de la precisión de la clasificación por cada clase, mostrando de la peor a la mejor calidad de señal y su línea de tendencia.

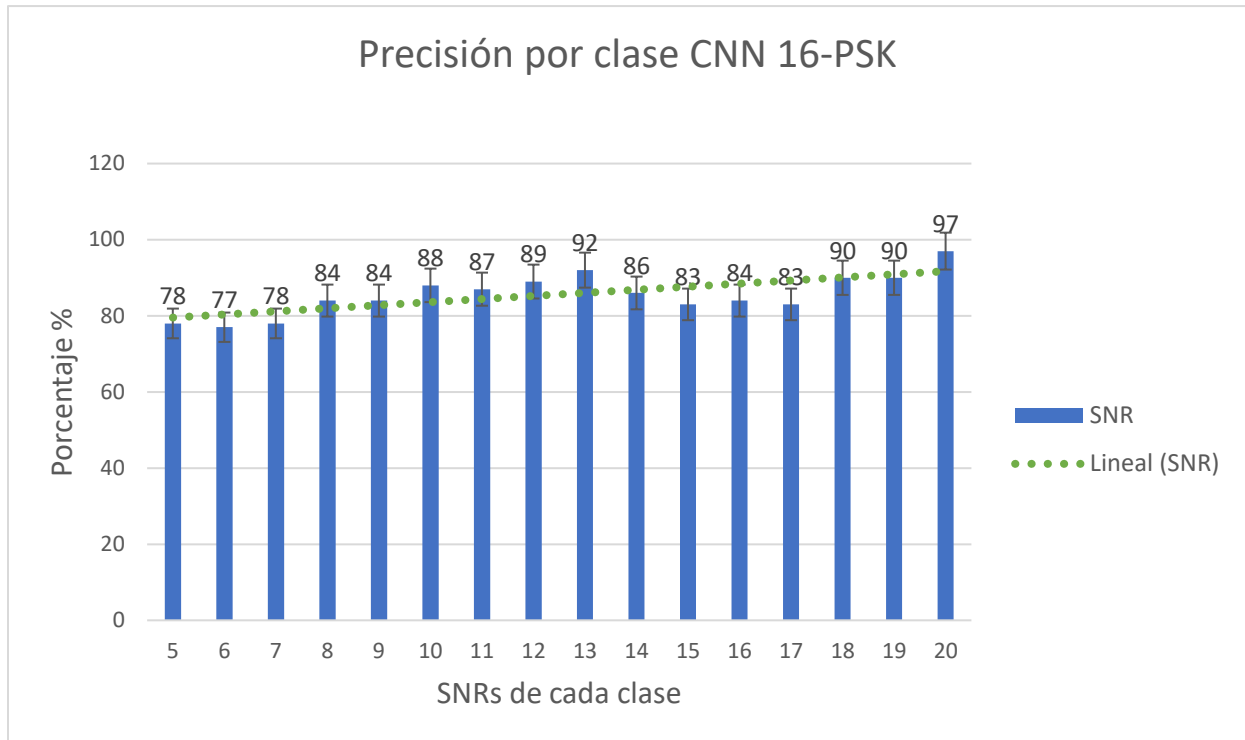


Fig. 64 Precisión por clase CNN multiclase 16-PSK [34].

Con estos datos se obtiene una precisión general en este modelo del 85%.

6.3.5 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 64-PSK.

La red se entrenó durante 39 épocas (epoch) obteniéndose los siguientes resultados de precisión y pérdida de entrenamiento y validación:

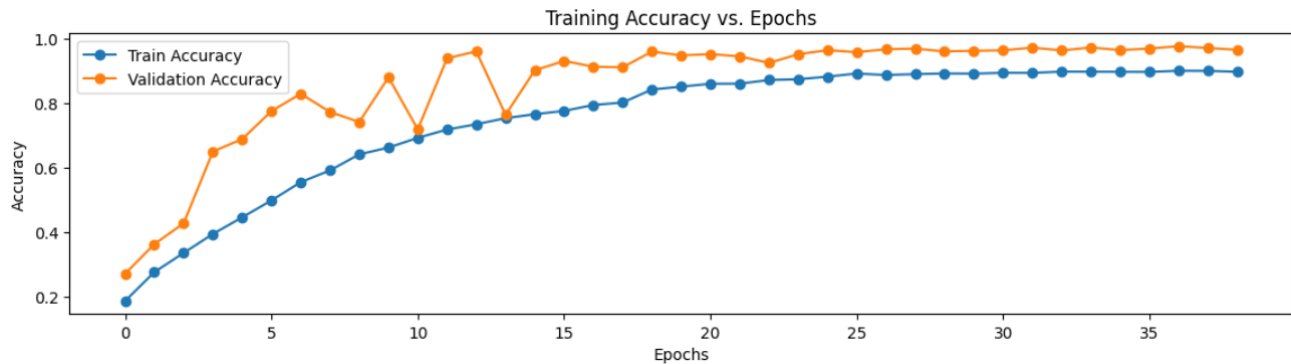


Fig. 65. Gráfico de cantidad de epoch vs precisión en porcentaje de entrenamiento y validación CNN multiclase 64-PSK [34].

Final Training Accuracy: 0,8965

Final validation Accuracy: 0,9645

Test Accuracy: 0,9644

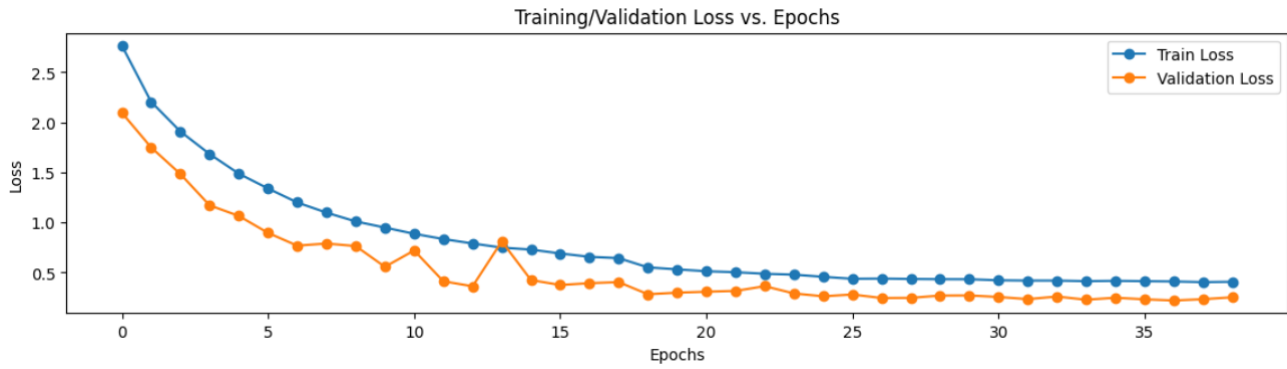


Fig. 66. Gráfico de cantidad de epoch vs pérdida en porcentaje de entrenamiento y validación CNN multiclase 64-PSK [34].

Final Train Loss: 0,4039

Final validation Loss: 0,2488

Test Loss: 0.2487

En la Fig. 67 se tiene la imagen de una matriz de confusión que muestra los resultados de la modulación 64-PSK. En el eje vertical se observan las clases originales, mientras que en el horizontal las clases predichas por el CNN.

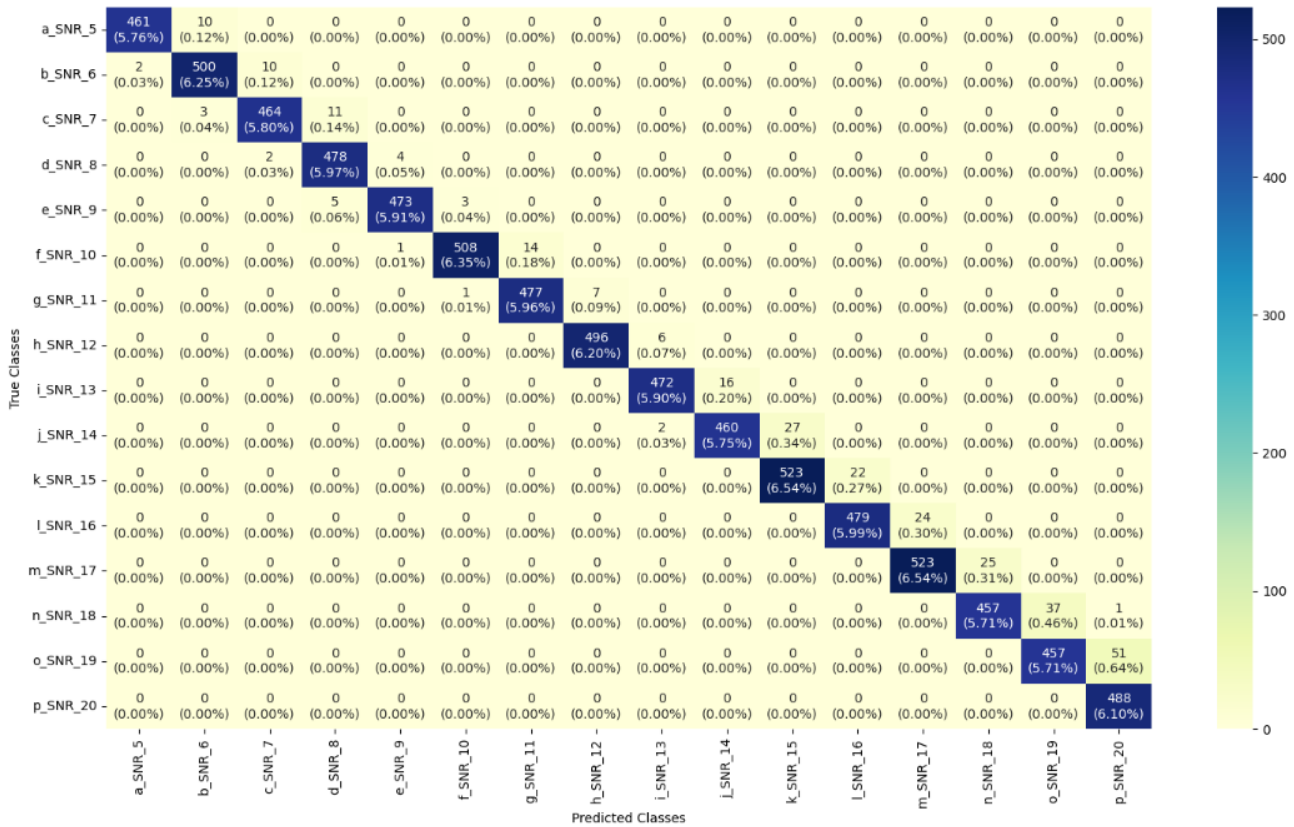


Fig. 67 Matriz de confusión de clases o rangos de SNR (5-20) predichas vs clases reales para red multiclase 64-PSK [34].

En la Fig. 68 se observan los resultados de la precisión de la clasificación por cada clase, mostrando de la peor a la mejor calidad de señal y su línea de tendencia.

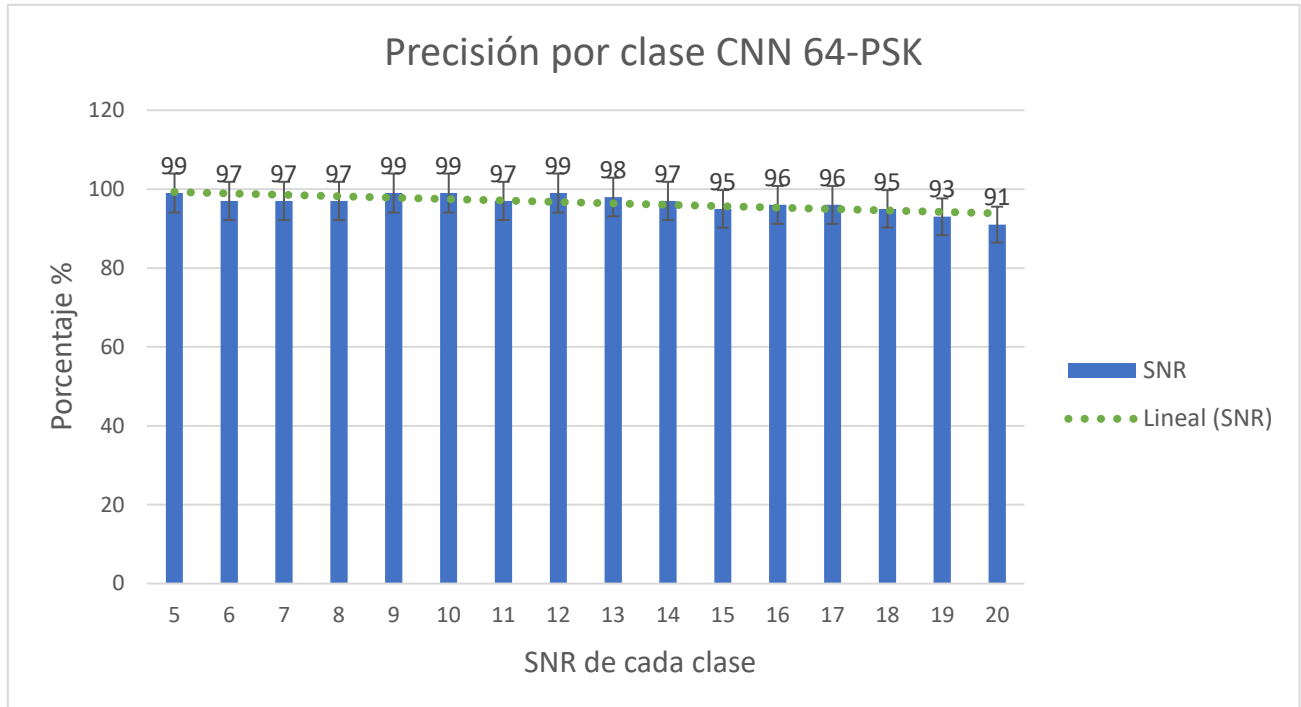


Fig. 68 Precisión por clase CNN multiclase 16-PSK [34].

Con estos datos se obtiene una precisión general en este modelo del 96%.

6.3.6 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 256-PSK.

La red se entrenó durante 39 épocas (epoch) obteniéndose los siguientes resultados de precisión y pérdida de entrenamiento y validación:

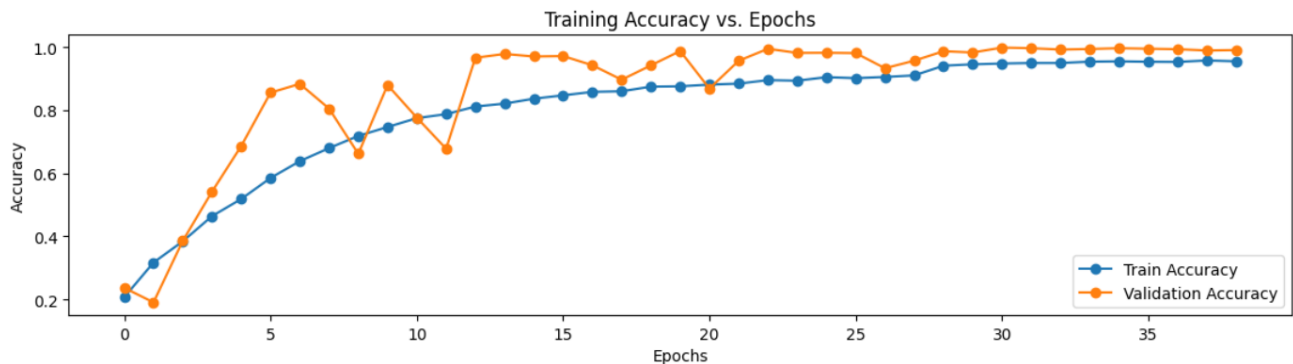


Fig. 69. Gráfico de cantidad de epoch vs precisión en porcentaje de entrenamiento y validación CNN multiclase 256-PSK [34].

Final Training Accuracy: 0,9550

Final validation Accuracy: 0,9906

Test Accuracy: 0,9907

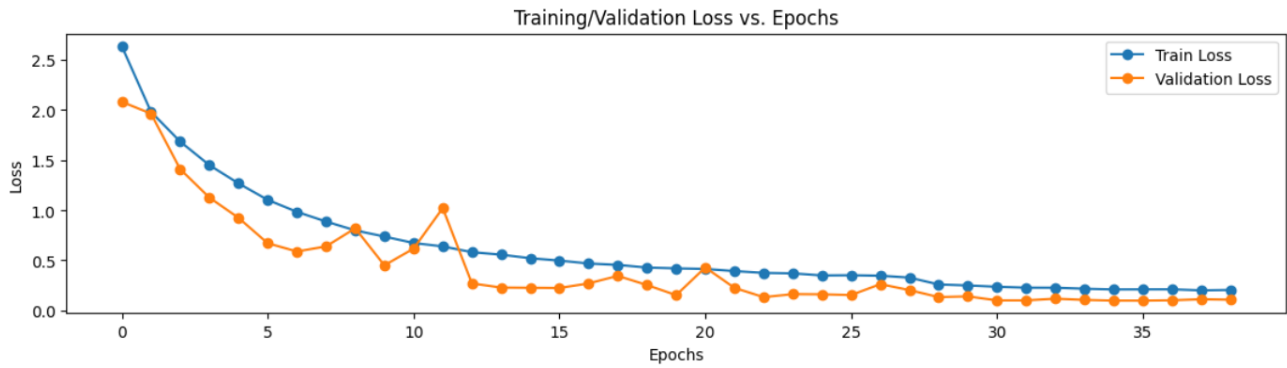


Fig. 70. Gráfico de cantidad de epoch vs pérdida en porcentaje de entrenamiento y validación CNN multiclase 256-PSK [34].

Final Train Loss: 0,2044

Final validation Loss: 0,1073

Test Loss: 0.1072

En la Fig. 71 se tiene la imagen de una matriz de confusión que muestra los resultados de la modulación 256-PSK. En el eje vertical se observan las clases originales, mientras que en el horizontal las clases predichas por el CNN.

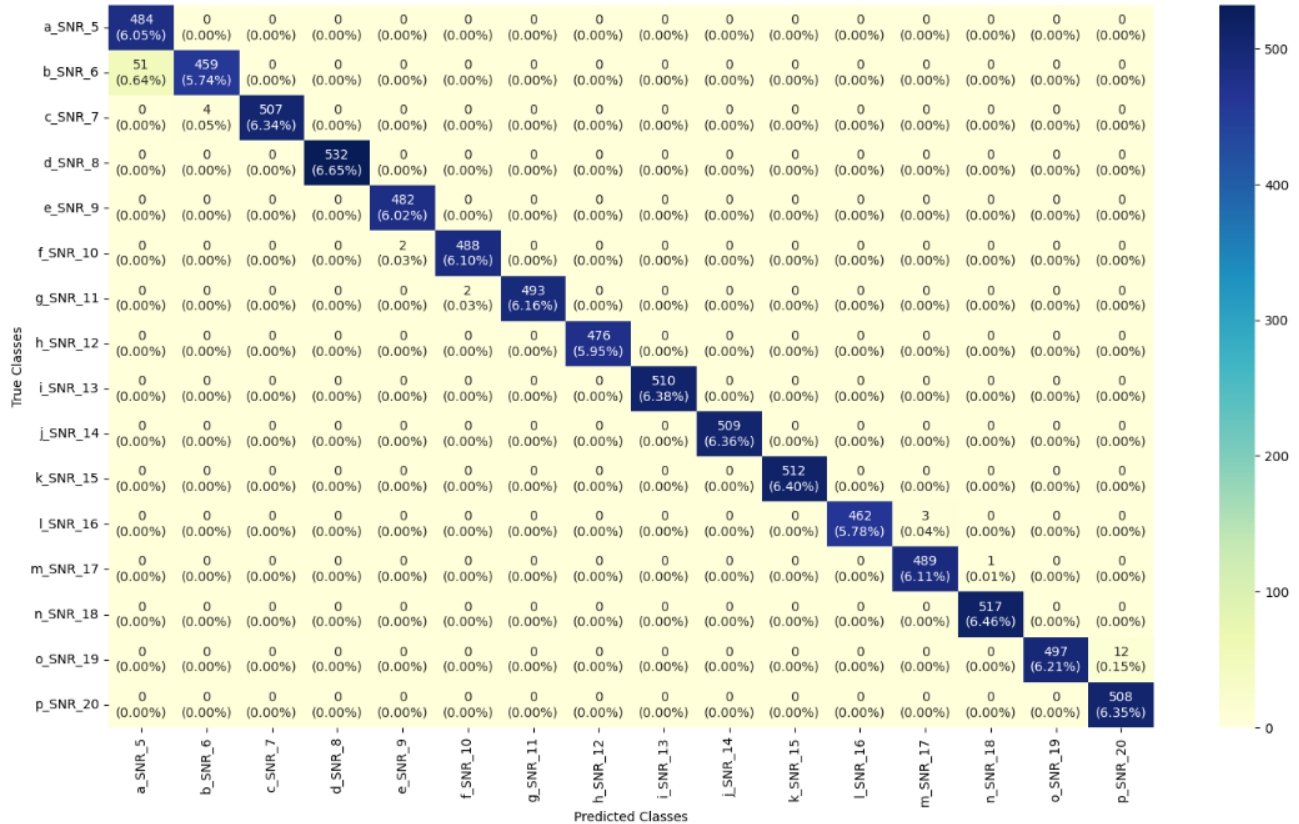


Fig. 71 Matriz de confusión de clases o rangos de SNR (5-20) predichas vs clases reales para red multiclasa 256-PSK [34].

En la Fig. 72 se observan los resultados de la precisión de la clasificación por cada clase, mostrando de la peor a la mejor calidad de señal y su línea de tendencia.

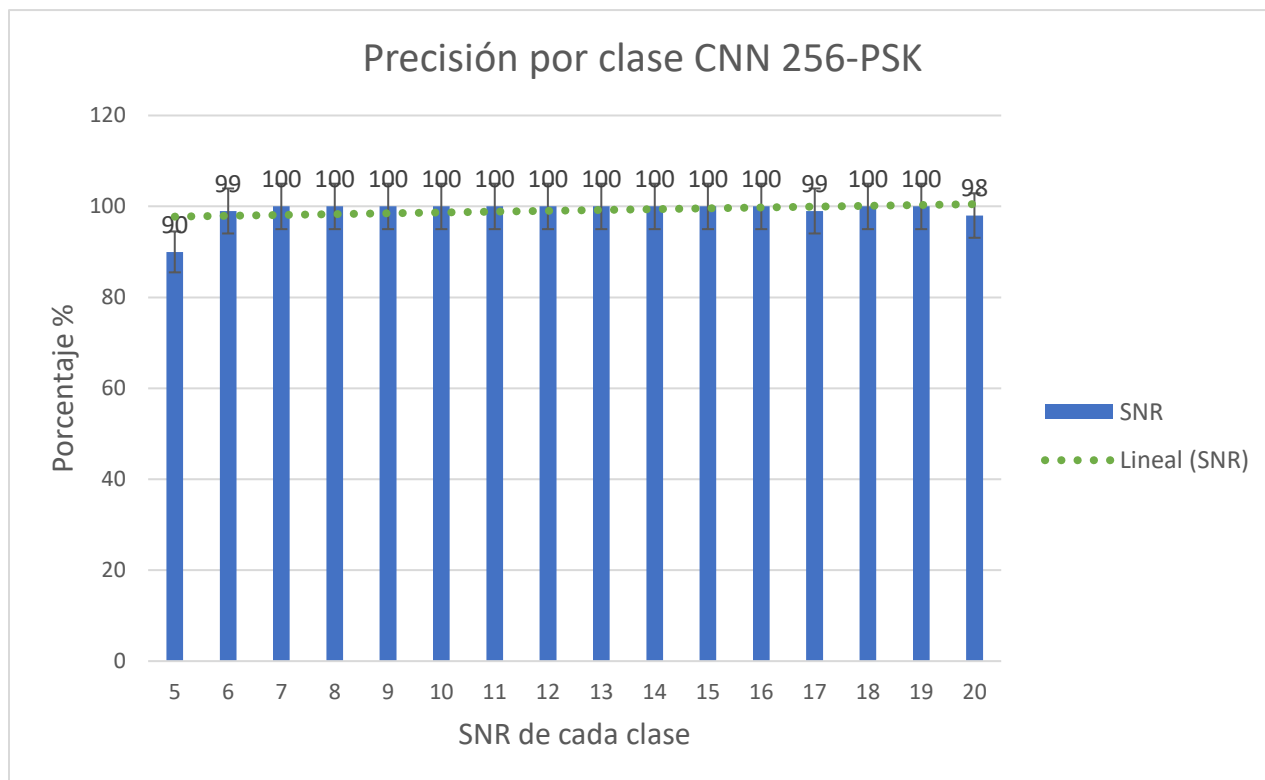


Fig. 72 Precisión por clase CNN multiclase 256-PSK [34].

Con estos datos se obtiene una precisión general en este modelo del 99%.

7. Análisis de resultados

7.1 Análisis clasificador binario QAM y PSK

En el CNN clasificador binario, se pudo observar que, en el modelo entrenado los gráficos resultantes terminaron con un 100% de precisión en entrenamiento y precisión en un breve periodo de tiempo de entrenamiento, incluso utilizando los recursos más básicos que ofrece Google Colab, lo cual demuestra que incluso para un diseño simple de una red convolucional neuronal se puede clasificar eficiente y fácilmente entre una modulación QAM y PSK.

7.2 Análisis clasificador multiclase CNN para diferentes formatos de modulación

Los análisis tanto de este sub-tema 7.2 y el 7.3 son más complejos, por lo que se explicitarán algunos datos y características que podrían no ser tan evidentes. Las clasificaciones correctas de cada clase que realiza cada red multiclase CNN viene dada por la diagonal de cada una de ellas.

La cantidad de imágenes de cada clase no es necesariamente la misma para cada una de ellas, debido a la aleatoriedad previa de la creación de las imágenes de constelación, por lo que puede ocurrir que una clase posea una precisión general relativa a la matriz de confusión más baja que otra clase que posee una mayor precisión relativa a la clasificación de su propia clase, Por ejemplo, en la matriz de confusión de la red multiclase 16-QAM asociada de la Fig. 51, la clase set b SNR_6 posee una precisión general relativa a la matriz de confusión de 2.56%, mientras que la clase set c SNR_7 posee un 3.06%, pero en cuanto a la precisión relativa a la clasificación de su propia clase, el set b SNR_6 es más eficiente que el set c SNR_7 con una precisión de 53% versus un 52% del set c SNR_7.

7.2.1 Análisis clasificador multiclase CNN para diferentes formatos de modulación de QAM.

A partir de los resultados de la matriz de confusión respectiva de la Fig. 43, se observa lo siguiente:

La clase 1_QAM_16 posee 652 predicciones correctas, representado un 32.60% del total de predicciones y una precisión de **92%** relativo a la clasificación de su propia clase. Se cometieron 15 errores clasificándolas como 2_QAM_64 lo que representa un 0.75% del total de predicciones.

La clase 2_QAM_64 posee 612 predicciones correctas, representado un 30.60% del total de predicciones y una precisión de **85%** relativo a la clasificación de su propia clase. Se cometieron 54

errores clasificándolas como 1_QAM_64 lo que representa un 2.70% y un error clasificando como 3_QAM_256 lo que representa un 0.05% del total de predicciones.

La clase 3_QAM_256 posee 574 predicciones correctas, representado un 28.70% del total de predicciones y una precisión de **100%** relativo a la clasificación de su propia clase. Se cometieron 92 errores clasificándolas como 2_QAM_64 lo que representa un 4.60% del total de predicciones.

7.2.2 Análisis clasificador multiclase CNN para diferentes formatos de modulación de PSK.

A partir de los resultados de la matriz de confusión respectiva de la Fig. 47, se observa lo siguiente:

La clase 1_PSK_16 posee 1999 predicciones correctas, representado un 33.32% del total de predicciones y una precisión de **92%** relativo a la clasificación de su propia clase. Se cometió 1 error clasificándola como 2_PSK_64 lo que representa un 0.02% del total de predicciones.

La clase 2_PSK_64 posee 1820 predicciones correctas, representado un 30.33% del total de predicciones y una precisión de **82%** relativo a la clasificación de su propia clase. Se cometieron 180 errores clasificándolas como 1_PSK_16 lo que representa un 3.00% del total de predicciones.

La clase 3_PSK_256 posee 1611 predicciones correctas, representado un 26.85% del total de predicciones y una precisión de **100%** relativo a la clasificación de su propia clase. Se cometieron 389 errores clasificándolas como 2_PSK_64 lo que representa un 6.48% del total de predicciones.

Un análisis comparativo de la precisión de cada clase para cada tipo de modulación se puede observar en la Fig. 73

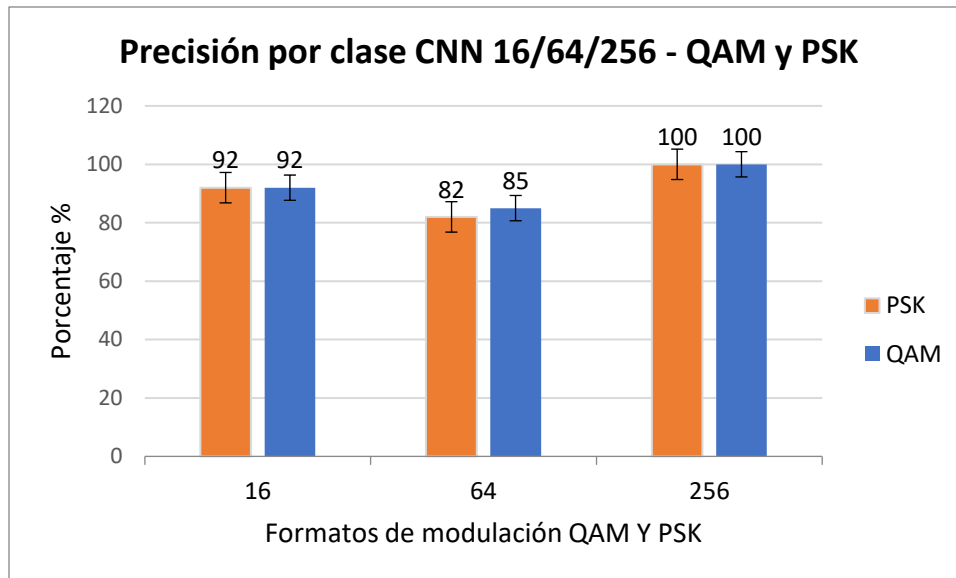


Fig. 73 Comparación de precisión por clase CNN multiclase 16/64/256-QAM y PSK [34].

7.3 Análisis clasificador multiclase CNN estimador de la calidad óptica de la señal en base al SNR.

7.3.1 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 16-QAM.

A partir de los resultados de la matriz de confusión respectiva de la Fig. 51, se tienen las siguientes observaciones generales:

La clase con la precisión más deficientemente es el set c SNR_7 que posee 147 predicciones correctas, representado un 3.06% del total de predicciones y una precisión de **52%** relativo a la clasificación de su propia clase. Se clasificaron erróneamente 146 imágenes entre las clases set a SNR_5, set b SNR_6, set d SNR_8, set e SNR_9 y set f SNR_10 lo que representa un ponderado de 3.04% del total de predicciones.

La clase con la precisión más eficiente es el set h SNR_12 que posee 268 predicciones correctas, representado un 5.58% del total de predicciones y una precisión de **88%** relativo a la clasificación de su propia clase. Se clasificaron erróneamente 60 imágenes entre las clases set f SNR_10, set g SNR_11, set i SNR_13 y set j SNR_14 lo que representa un ponderado de 1.26% del total de predicciones.

7.3.2 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 64-QAM.

A partir de los resultados de la matriz de confusión respectiva de la Fig. 55, se tienen las siguientes observaciones generales:

La clase con la precisión más deficientemente es el set e SNR_9 que posee 310 predicciones correctas, representado un 4.84% del total de predicciones y una precisión de **76%** relativo a la clasificación de su propia clase. Se clasificaron erróneamente 74 imágenes entre las clases set c SNR_7, set d SNR_8 y set f SNR_10 lo que representa un ponderado de 1.16% del total de predicciones.

Las clases con las precisiones más eficiente son el set o SNR_19 y el p SNR_20 que poseen 410 y 400 predicciones correctas, representado un 6.41 y 6.25% del total de predicciones respectivamente y una precisión de **100%** para ambas relativo a la clasificación cada uno de su propia clase. Para el set o SNR_19 se clasificaron erróneamente 4 imágenes entre las clases set n SNR_18 y set p SNR_20 lo que representa un ponderado de 0.06% del total de predicciones.

7.3.3 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 256-QAM.

A partir de los resultados de la matriz de confusión respectiva de la Fig. 59, se tienen las siguientes observaciones generales:

La clase con la precisión más deficientemente es el set b SNR_6 que posee 125 predicciones correctas, representado un 1.56% del total de predicciones y una precisión de **38%** relativo a la clasificación de su propia clase. Se clasificaron erróneamente 402 imágenes entre las clases set a SNR_5, set c SNR_7 y set d SNR_8 lo que representa un ponderado de 5.02% del total de predicciones.

La clase con la precisión más eficiente es el set o SNR_19 que posee 303 predicciones correctas, representado un 3.79% del total de predicciones y una precisión de **75%** relativo a la clasificación de su propia clase. Se clasificaron erróneamente 174 imágenes entre las clases set n SNR_18 y set p SNR_20 lo que representa un ponderado de 2.17% del total de predicciones.

7.3.4 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 16-PSK.

A partir de los resultados de la matriz de confusión respectiva de la Fig. 63, se tienen las siguientes observaciones generales:

La clase con la precisión más deficientemente es el set b SNR_6 que posee 339 predicciones correctas, representado un 4.24% del total de predicciones y una precisión de **77%** relativo a la clasificación de su propia clase. Se clasificaron erróneamente 152 imágenes entre las clases set a SNR_5 y set c SNR_7, lo que representa un ponderado de 1.90% del total de predicciones.

La clase con la precisión más eficiente es el set p SNR_20 que posee 489 predicciones correctas, representado un 6.11% del total de predicciones y una precisión de **97%** relativo a la clasificación de su propia clase. Se clasificaron erróneamente 4 imágenes de la clase set o SNR_19, lo que representa un 0.05% del total de predicciones.

7.3.5 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 64-PSK.

A partir de los resultados de la matriz de confusión respectiva de la Fig. 67, se tienen las siguientes observaciones generales:

La clase con la precisión más deficientemente es el set p SNR_20 que posee 488 predicciones correctas, representado un 6.10% del total de predicciones y una precisión de **91%** relativo a la clasificación de su propia clase.

Las clases con las precisiones más eficientes son el set a SNR_5, e SNR_9, set f SNR_10 y set h SNR_12 que poseen 461, 473, 508 y 496 predicciones correctas, representado un 5.76%, 5.91%, 6.35% y 6.20% del total de predicciones respectivamente y una precisión de **99%** para todas relativo a la clasificación de cada uno de su propia clase. De estas últimas 4 clases se clasificaron erróneamente entre 6 y 15 imágenes, lo que representa entre un 0.07% y 0.19% del total de predicciones.

7.3.6 Clasificador CNN multiclase estimador de calidad óptica de la señal en base al SNR 256-PSK.

A partir de los resultados de la matriz de confusión respectiva de la Fig. 71, se tienen las siguientes observaciones generales:

La clase con la precisión más deficientemente es el set a SNR_5 que posee 484 predicciones correctas, representado un 6.05% del total de predicciones y una precisión de **90%** relativo a la clasificación de su propia clase.

Las clases con las precisiones más eficientes son entre el set c SNR_7, y set o SNR_19 que poseen 507, 532, 482, 488, 493, 476, 510, 509, 512, 462, 489, 517 y 497 predicciones correctas, representado un 6.34%, 6.65%, 6.02%, 6.10%, 6.16%, 5.95%, 6.38%, 6.36%, 6.40%, 5.78%, 6.11%, 6.46% y 6.21% del total de predicciones respectivamente y una precisión de **100%** para todas relativo a la clasificación de cada uno de su propia clase.

En todas las redes CNN de modulaciones QAM se consigue una línea de tendencia de precisión de clases con una orientación de abajo hacia arriba, es decir, pendiente positiva, lo cual concuerda con la teoría en cuanto a que a medida que la calidad de señal es mejor, ósea, un mayor SNR, los algoritmos podrán clasificar de mejor manera o tendrán un mayor porcentaje de precisión en las clases con más altos valores de SNR. Este análisis es visible en las Fig. 52, 56 y 60.

En cuanto a las precisiones generales la CNN 16-QAM consiguió una precisión general de 73%, la que es mayor que la CNN 256-QAM que logró una precisión general de 58%. Esto concuerda con lo que se espera ya que el modelo 16-QAM es más simple, posee una base de datos más pequeña y su clasificación en teoría debería ser más fácil, debido a que la cantidad de puntos involucrados en el diagrama de constelación son menores por lo que es más fácil clasificarlos. Comparando la red CNN 16-QAM con la 64-QAM, pese a que la última logró una mejor precisión de 88% versus la otra que obtuvo una precisión menor de 73%, la CNN 64-QAM tuvo un mejor rendimiento porque la estructura de su modelo es más sofisticada y posee una base de datos más de 2 veces más grande enriqueciendo su entrenamiento. En la Fig. 74 se comparan los resultados de la clasificación por clases de todas estas modulaciones QAM, mientras que en la Fig. 75 se comparan los resultados generales de las modulaciones QAM.

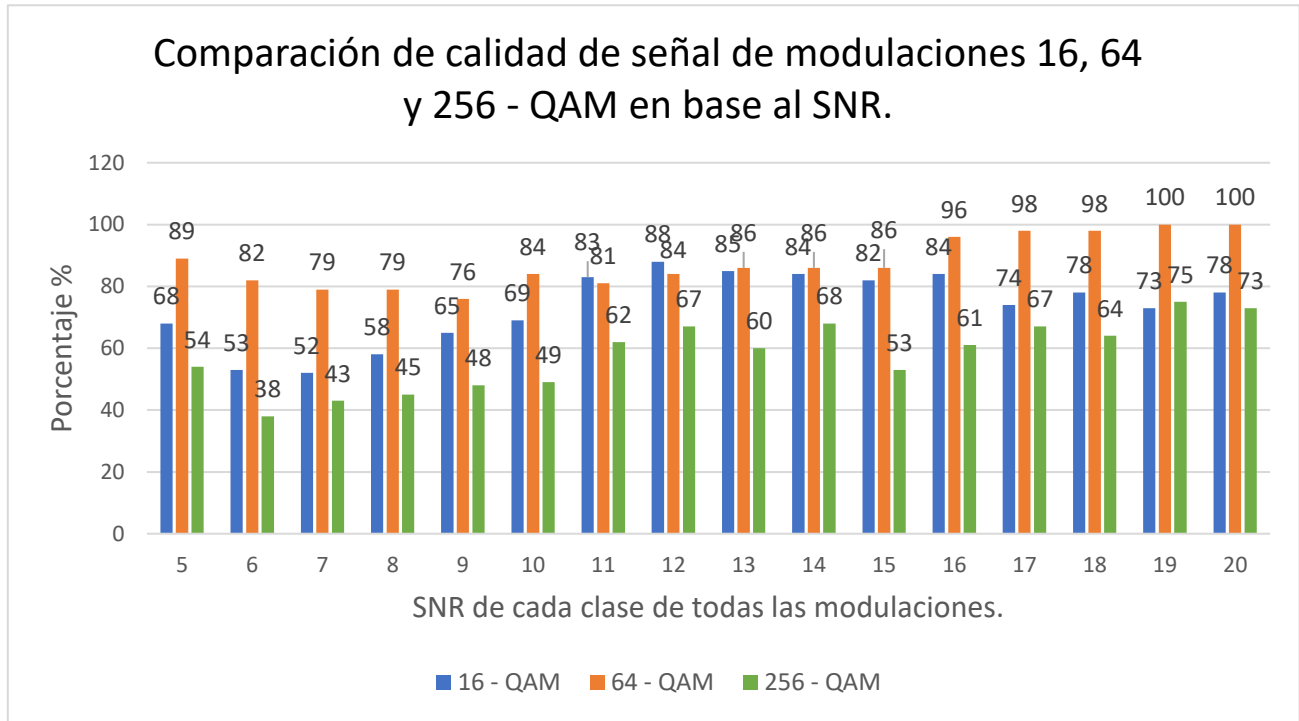


Fig. 74 Comparación de calidad de señal de CNN de modulaciones 16, 64 y 256-QAM en base al SNR. [34].

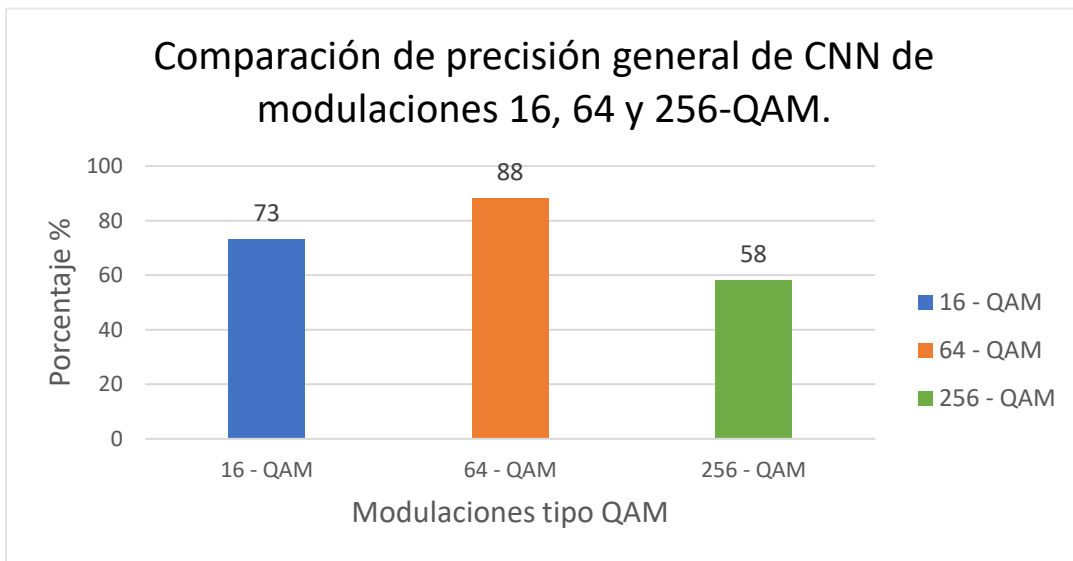


Fig. 75 Comparación de precisión general de CNN de modulaciones 16, 64 y 256-QAM [34].

En las cuanto a las redes CNN de modulaciones PSK, en 2/3 de ellas se consigue una línea de tendencia de precisión de clases con una orientación de abajo hacia arriba, es decir, pendiente positiva, lo cual concuerda con la teoría en cuanto a que a medida que la calidad de señal es mejor, debido a los niveles más altos de SNR, el único caso en el que ocurrió lo opuesto fue el de la modulación 64-PSK, debido principalmente al rendimiento de precisión de la clase con SNR de 20. Este algoritmo particularmente se ejecutó en más de 10 ocasiones dando resultados similares, lo que no es concordante con lo esperado en la teoría, pese a su buen rendimiento general. Este análisis es visible en las Fig. 64, 68 y 72.

En cuanto a las precisiones generales de las modulaciones PSK los resultados son anormales para lo que se espera, debido a que aunque las 3 CNN poseían la misma arquitectura de modelo de aprendizaje y la misma cantidad de imágenes de la base de datos, los resultados difieren de lo que a priori se puede esperar, debido a que a medida que el orden de formato de modulación es mayor, implica que son más puntos en el diagrama de constelación, por lo que debería ser más complejo y difícil para un algoritmo lograr clasificar en base al SNR. En los resultados se puede observar como la precisión general va aumentando a medida que se aumenta el orden de formato de modulación, es decir, para la CNN 16-PSK se tiene una precisión de 85%, para la CNN 64-PSK un 96% y para la CNN 256-PSK un 99%.

En la Fig. 76 se comparan los resultados de la clasificación por clases de todas estas modulaciones PSK. Mientras que en la Fig. 77 se comparan los resultados generales de las modulaciones PSK.

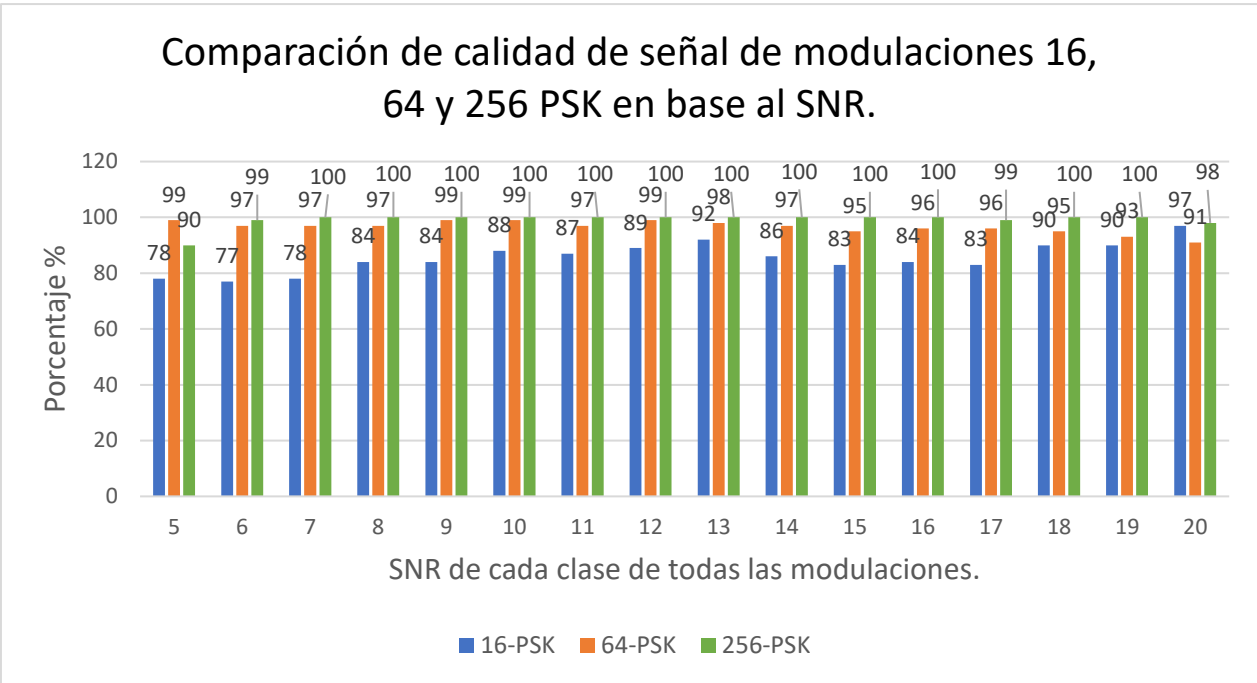


Fig. 76 Comparación de calidad de señal de CNN de modulaciones 16, 64 y 256-PSK en base al SNR. [34].

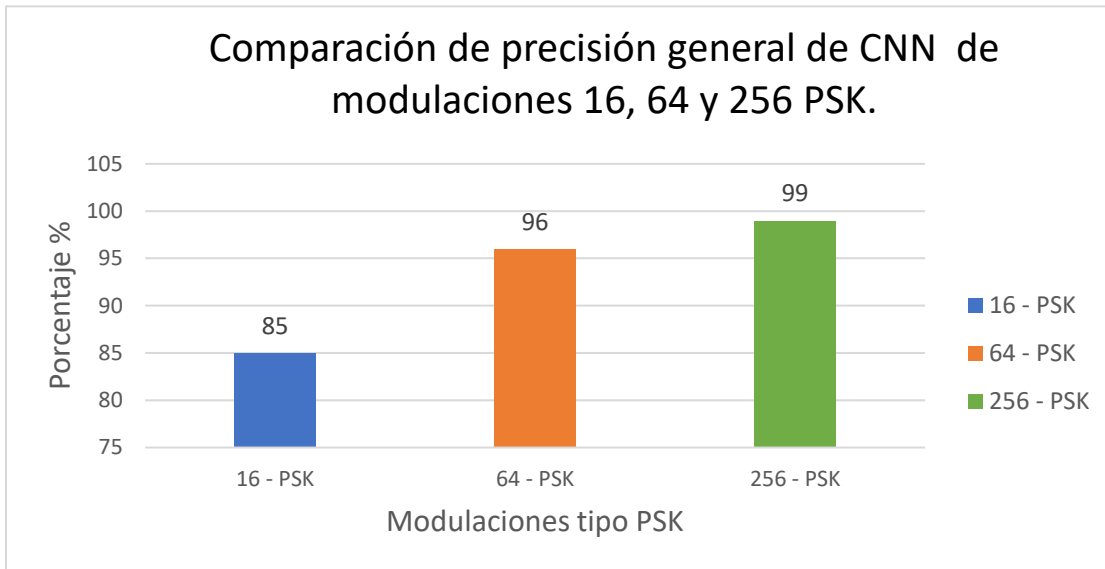


Fig. 77 Comparación de precisión general de CNN de modulaciones 16, 64 y 256-PSK [34].

Si se compara por orden de modulación, las modulaciones QAM poseen menor precisión general que las modulaciones PSK, como se puede ver en la Fig. 78. Aun así, los resultados son buenos para la mayoría, puesto que pueden clasificar con márgenes de error pequeños para todo el espectro de SNR, donde es razonable que se vean afectados mayoritariamente el anterior y posterior nivel de SNR al cual se clasificó erróneamente.

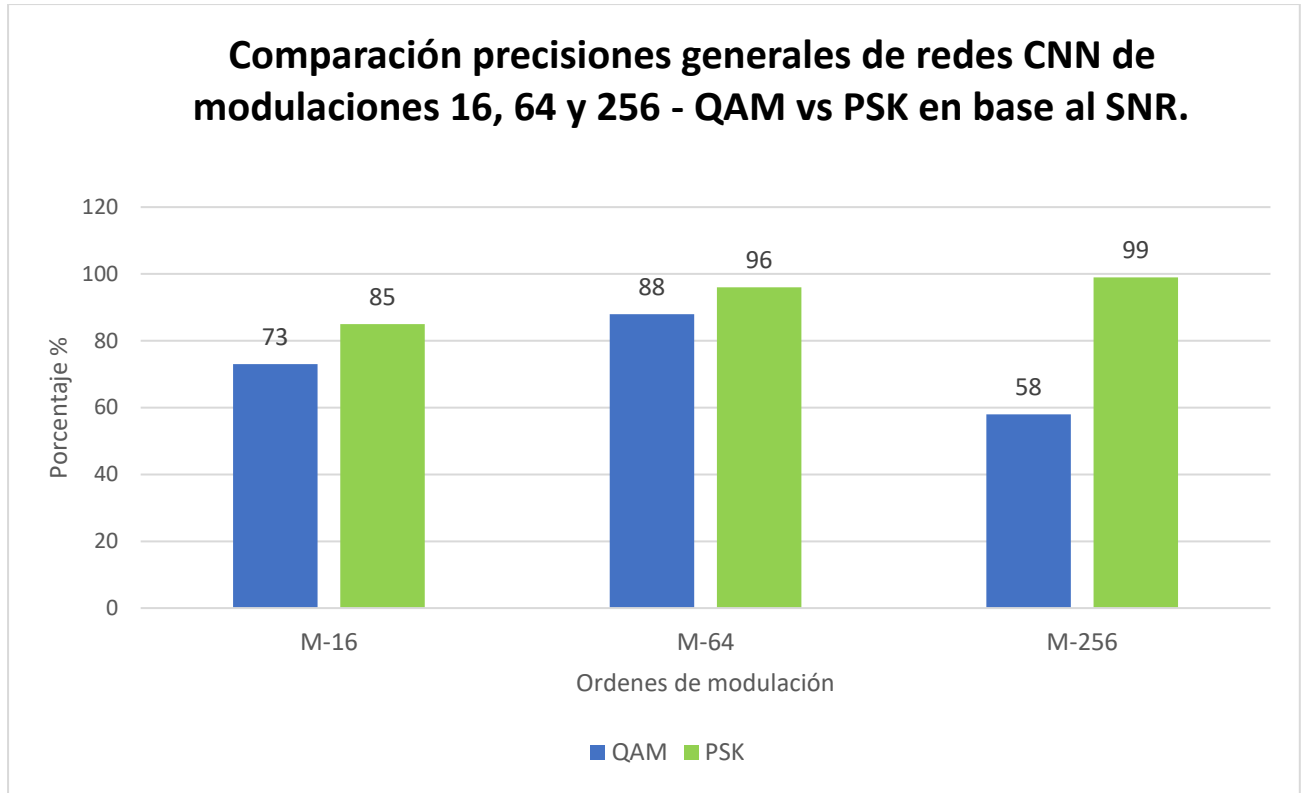


Fig. 78 Comparación de precisión general de redes CNN de modulaciones 16, 64 y 256- QAM vs PSK [34].

8. Conclusiones

Resumen

Se creó el canal AWGN en el entorno de programación Matlab, para diferentes formatos de modulación, enfocándose en las modulaciones QAM y PSK, en las que se generó diferentes imágenes de constelaciones asociadas a estas con parámetros de número de símbolos diferentes, cantidad máxima de símbolos a transmitir y SNR. Se modificó el código para generar diferentes constelaciones y así crear una base de datos de estas imágenes. Además, se consiguió entrenar modelos de clasificación binaria y multiclase para los 2 tipos de modulación, con sus parámetros asociados y resultados de evaluaciones de precisión y pérdidas de datos de entrenamiento, validación y testeo. Para los casos de modulación multiclase de ordenes de modulación diferentes se generaron métricas visuales como gráficos de barras y matrices de confusión para poder analizar en mayor detalle cómo se desarrollaron las redes CNNs entrenadas para cada caso entregando las precisiones generales de cada red CNN.

Conclusiones

El modelo de red CNN binaria se entrenó alcanzando un 100% de precisión en su entrenamiento, lo cual demuestra que incluso para un diseño simple de una red convolucional neuronal se puede clasificar eficiente, fácil y en corto periodo de tiempo entre una modulación QAM y PSK.

Las redes CNN entrenadas con modulación multiclase 16/64/256-QAM superaron en precisión a las entrenadas con modulación multiclase 16/64/256-PSK. A pesar de que el primer modelo contaba con una base de datos más pequeña y una arquitectura de CNN ligeramente más simple, logró una precisión general del 92%, mientras que el segundo modelo alcanzó una precisión general de 91%. Esto implica que estas arquitecturas de redes CNN son ligeramente más eficientes en la clasificación de formatos de modulación QAM en comparación con los formatos PSK.

Las redes CNN que estiman la calidad de señal en base al SNR poseen las siguientes precisiones generales: 16-QAM 73%, 64-QAM 88%, 256-QAM 58%, 16-PSK 85%, 64-PSK 96% Y 256-PSK 99%. Si se compara por orden de modulación, las modulaciones QAM poseen menor precisión general que las modulaciones PSK, como se puede ver en la Fig. 78. Aun así, los resultados son buenos para la mayoría, puesto que pueden clasificar con márgenes de error pequeños para todo el espectro de SNR, donde es razonable que se vean afectados mayoritariamente el anterior y posterior nivel de SNR

al cual se clasificó erróneamente. En la Tabla 3 se realiza una ilustración de las precisiones generales de todas las redes CNN entrenadas de esta investigación, en la que los tipos de redes CNN se encuentran el modelo binario, el multiclase para diferentes formatos de modulación y el multiclase que estima la señal óptica de la señal en base al SNR para las filas y los diferentes formatos de modulación para las columnas.

Tabla 3. Precisiones generales de todas las redes CNN entrenadas [34]

		Modulación								
		QAM vs PSK	16/64/256 QAM	16/64/256 PSK	16-QAM	64-QAM	256-QAM	16-PSK	64-PSK	256-PSK
Tipo de Red CNN	Binaria	100%	-	-	-	-	-	-	-	-
	Multiclase F. Mod	-	91%	92%	-	-	-	-	-	-
	Multiclase Base SNR	-	-	-	73%	88%	58%	85%	96%	99%

Las arquitecturas de todos estos modelos comparadas con otros modelos como los AlexNet, InceptionV1 y VGG16 los cuales consumen una cantidad enorme de recursos informáticos [11], son mucho más simples, además en todo el proceso de ejecución de las redes CNN en Google colab no se tuvo que realizar algún pago, lo cual es notable para la complejidad de operaciones matriciales que se debían realizar para que los modelos pudieran completar su aprendizaje, por lo que derivó en muchos momentos de espera y tiempos de ejecución, pero que se vieron traducidos en muy buenos resultados de precisión generales considerando que en cuanto a herramientas informáticas, tanto hardware como software, basta con tener un computador básico con conexión a internet para poder realizar todo este estudio.

Trabajo Futuro

Lo realizado en esta memoria fue clasificar imágenes de diagramas de constelaciones en base a sus SNR utilizando redes CNN. En el mundo de las redes ópticas, específicamente en los detectores coherentes lo realizado fue un MFI, debido a que identifica los formatos de modulación por medio de su clasificación, pero además de agrega la clasificación en cuanto a los SNR que posea cada constelación, por lo que podría implementarse en el DSP de un detector coherente para incrementar la confiabilidad y disponibilidad de redes ópticas que hasta hace unos pocos años no existía de manera comercial [3]. Además, se puede ampliar y proyectar a una automatización mucho más grande como por ejemplo utilizando más tipos de modulación como Frequency Shift Keying (FSK) o Amplitude-shift keying (ASK), diferentes ordenes de modulación e incluso realizar una clasificación en tiempo real, lo cual se acercaría a un desarrollo completo de la inteligencia artificial para la clasificación de formatos de modulación.

Otro factor importante es que si bien, con este estudio se clasifican las imágenes con el fin de tomar mejores decisiones en pos de una mejora en los servicios de la red óptica, no se clasifica el símbolo recibido, por lo que en ese aspecto se podría combinar con otras herramientas de machine learning para realizar ambas tareas e incluso otras en el proceso de la transmisión y recepción de señales.

Finalmente debido a la eficiencia de este tipo de estudios con las condiciones y alcances que se definieron. Puede reutilizarse fácilmente gracias al transfer learning, también desarrollarse otros algoritmos de herramientas de machine learning para mejorar las diferentes etapas que posea una red óptica, añadiéndole inteligencia de la cual se puede ir realimentando, sin la obligatoria interacción del ser humano, siendo más precisos y en constante aprendizaje, por lo que si bien actualmente existen muchos sistemas de redes ópticas actualizados con implementaciones en sus detectores que utilizan inteligencia artificial, todo apunta a que en los próximos años las investigaciones y ejecuciones de los diversos sistemas de redes ópticas empezarán una transición continua y persistente del hardware por software.

9. Bibliografía

- [1] Rentao Gu, Zeyuan Yang, Yuefeng Ji, “Machine learning for intelligent optical networks: A comprehensive survey”, Beijing Laboratory of Advanced information Network, Beijing University of Post and Telecommunications (BUPT), Beijing, China, 2020.
- [2] Página web de Cisco, <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>.
- [3] Waddah S. Saif, Maged A. Esmail, Amr M. Ragheb, Tariq A. Alshawi, and Saleh A. Alshebeili, “Machine Learning Techniques for Optical Performance Monitoring and Modulation Format Identification A Survey”, 2020.
- [4] Reda Ayassi, Ahmed Triki, Noel Crespi, Roberto Minerva and Maxime Laye, “Survey on the Use of Machine Learning for Quality of Transmission Estimation in Optical Transport”, 2022.
- [5] Página web <https://www.m2optics.com/blog/what-is-the-difference-between-otdr-and-olts#:~:text=The%20OTDR%20is%20a%20more,to%20varying%20degrees%20and%20capabilities>
- [6] Md. Saifuddin Faruk *OSA* and Seb J. Savory, Digital Signal Processing for Coherent Transceivers Employing Multilevel Formats, 2017.
- [7] Pedro J. Vidal-Moreno, Carlos Becerril, Maria R. Fernandez-Ruiz, Hugo Martins, Sonia Martin-Lopez and Miguel Gonzalez-Herraez, Noise analysis in direct detection and coherent detection phase-sensitive optical time-domain reflectometry systems, Agosto 2023.
- [8] Alan E. Willner, Optic Fiber Telecommunications VII, 2019
- [9] Md. Saifuddin Faruk *OSA* and Seb J. Savory, Digital Signal Processing for Coherent Transceivers Employing Multilevel Formats, 2017
- [10] David R. Pauluzzi and Norman C. Beaulieu, A Comparison of SNR Estimation Techniques for the AWGN Channel, 2000
- [11] Xiaojuan Xie, Shengliang Peng y Xi Yang, Deep Learning-Based Signal-To-Noise Ratio Estimation Using Constellation Diagrams, 2020
- [12] Página web de Google, <https://developers.google.com/machine-learning/practica/image-classification?hl=es-419> accedido a lo largo del semestre.

- [13] Stuart J Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach (AIMA)* · 3rd edition, Prentice Hall, 2009.
- [14] Página web de Iberdrola, <https://www.iberdrola.com/innovacion/que-es-inteligencia-artificial>, 2023.
- [15] Página web de IBM, <https://www.ibm.com/topics>
- [16] Página web de codigofuente, [https://www.codigofuente.org/redes-neuronales-profundas-tipos-caracteristicas/#:~:text=Una%20red%20neuronal%20profunda%20\(DNN,modelar%20relaciones%200no%20lineales%20complejas](https://www.codigofuente.org/redes-neuronales-profundas-tipos-caracteristicas/#:~:text=Una%20red%20neuronal%20profunda%20(DNN,modelar%20relaciones%200no%20lineales%20complejas).
- [17] Dan C. Ciresan, Ueli Meier, Jonathan Masci, Luca M. Gambardella y Jurgen Schmidhuber, *Flexible, High Performance Convolutional Neural Networks for Image Classification*, 2011.
- [18] Página web de Intel, <https://www.intel.com/content/www/us/en/products/docs/processors/what-is-a-gpu.html>
- [19] Página web de Intel, <https://www.intel.la/content/www/xl/es/products/docs/processors/cpu-vs-gpu.html>
- [20] Página web <https://en-academic.com/dic.nsf/enwiki/977082>
- [21] Página web <https://academic-accelerator.com/encyclopedia/signal-to-noise-ratio>
- [22] Página web <https://www.analog.com/en/design-center/landing-pages/001/beginners-guide-to-dsp.html>
- [23] Tianhua Xu, *Digital Signal Processing for Optical Communications and Networks*, 2017
- [24] Yahya Amira, *Optical Communication Systems*, Enero 2023
- [25] CHAPTER1 Elements of a digital communication system, apuntes de clases del departamento de School of Electrical and Electronic Engineering, Universiti Sains Malaysia.
- [26] Md. Saifuddin Faruk *OSA* and Seb J. Savory, *Digital Signal Processing for Coherent Transceivers Employing Multilevel Formats*, 2017
- [27] Josh W. Nevin, Sam Nallaperuma, Nikita A. Shevchenko, Xiang Li, Md. Saifuddin Faruk, Seb J. Savory, “Machine learning for optical fiber communication systems: An introduction and overview”, Fiber Optic Communication System Laboratory, Electrical Engineering Division, Department of Engineering, University of Cambridge, United Kingdom, 2021.
- [28] Jordi Torres, *Deep Learning Introducción práctica con Keras*, 2018
- [29] Pedro Ponce Cruz, *Inteligencia artificial con aplicaciones a la ingeniería*. Editorial Alfaomega, 2011.

[30] Página web <https://albertotb.com/curso-ml-R/Rmd/12-nn/12-nn.html#25>

[31] Anirudha Ghosh, Abu Sufian, Farhana Sultana, Amlan Chakrabarti, Debashis, De Fundamental Concepts of Convolutional Neural Network, 2020.

[32] Página de Google, <https://developers.google.com/machine-learning/glossary?hl=es-419>,
accedido a lo largo del semestre.

[33] Página web <https://keras.io/api/layers/activations/>

[34] Fuente: Elaboración propia.