
Análisis técnico de mercados de criptomoneda con redes neuronales convolucionales

Technical analysis of cryptocurrency markets with convolutional neural networks

Andres Felipe Caicedo Rueda^a
andrescaicedor@usantotomas.edu.co

Wilmer Dario Pineda Rios^b
wilmerpineda@usantotomas.edu.co

Resumen

Con el análisis técnico de los mercados financieros se estudia la interacción del mercado usando principalmente gráficos para intentar pronosticar la tendencia del precio en el futuro. Sin embargo, aunque los instantes de interacción del mercado junto con sus indicadores técnicos se representan como imágenes, los esfuerzos de automatización de las estrategias de compra y venta de activos se han dirigido especialmente hacia los métodos algorítmicos. La naturaleza visual del análisis técnico invita a buscar alternativas de solución dentro del ámbito de la visión por computador usando técnicas de clasificación de imágenes y detección de objetos. En este trabajo se construyó una base de datos de 240 imágenes a partir de la información cronológica de las variables de interacción del mercado (Precio, Volumen) y algunos de los indicadores técnicos más populares (EMA, Bollinger, MACD, RSI). En dicha base se etiquetaron los objetos correspondientes a las clases de los eventos que conforman una estrategia de compra y venta (Cruce de EMA, Cruce del MACD). Posteriormente, con las imágenes etiquetadas de la base se entrenó un modelo de aprendizaje profundo de redes neuronales convolucionales (CNN) para detectar objetos de esas mismas clases en imágenes nunca vistas y así identificar señales que permitan tomar decisiones de compra y venta en el mercado de crypto activos con monedas como Bitcoin, Ethereum o ADA entre muchas otras. Como resultado del estudio se obtuvieron dos modelos para la detección de cruces de EMA y MACD, el primero de tipo EfficientDet0 con AP50 del 76.69 % y el segundo de tipo EfficientDet4 con AP50 del 87.26 %. Esta técnica de detección de objetos para diseñar e implementar estrategias de compra y venta es fácil e intuitiva y gracias a su nivel de abstracción podría ser usada con diferentes periodicidades, activos y mercados.

Palabras clave: análisis técnico, redes neuronales convolucionales, detección de objetos, criptomonedas.

1. Introducción

El análisis técnico ayuda a identificar los momentos de compra y venta de un activo financiero en el mercado con información. El comportamiento de un activo se muestra a través de gráficos e indicadores y por medio del análisis técnico se identifican las dinámicas y tendencias de un activo teniendo en cuenta su valor histórico, con el propósito de pronosticar su comportamiento en el futuro. Para la identificación del comportamiento de los activos se usan las variables de interacción del mercado, precio y volumen, e indicadores técnicos como las velas de apertura, máximo, mínimo y cierre (OHLC), las medias móviles (SMA, EMA), las bandas de Bollinger, el indicador de fuerza relativa (RSI) o el indicador de convergencia

^aEstudiante

^bDirector

y divergencia de medias móviles (MACD). Mediante el uso de estas herramientas se plantean diferentes estrategias para identificar los momentos más adecuados para la compra y venta de activos.

La compraventa se puede hacer manualmente a través de inspección visual en donde se pretende identificar los momentos adecuados para hacer las operaciones de acuerdo con la estrategia elegida, pero tiene la desventaja de que está limitada por la capacidad y experiencia de la persona que la realiza. Por otro lado, la compraventa usando estrategias de manera algorítmica permite su automatización, pero a medida que las estrategias usadas se hacen más robustas, el software para implementarlas se hace más complejo.

Es importante entonces preguntarse: para los activos del mercado de criptomonedas ¿Es posible usar la experiencia de los analistas financieros para construir un modelo que identifique visualmente y de manera automática los eventos de una estrategia de análisis técnico en imágenes que representen el comportamiento en el tiempo de un activo?

Por esta razón, el propósito de este trabajo es construir un modelo que identifique de manera sencilla los momentos clave para realizar operaciones de compra y venta de un activo del mercado a partir de tres técnicas que se han desarrollado en la literatura mencionada en las siguientes secciones de este documento:

- La inclusión de información de análisis técnico del mercado para hacer el pronóstico.
- La representación de los momentos de un activo del mercado, sus variables e indicadores técnicos como una base de datos de imágenes
- La identificación de objetos con modelos de redes neuronales convolucionales de última generación en las imágenes que representan los momentos del mercado.

Para lograr el objetivo, se recopiló información de precio y volumen de diferentes activos del mercado de las criptomonedas. Así mismo, se implementaron las funciones para calcular algunos de los indicadores más usados en el análisis técnico de los mercados. Tanto las variables como los indicadores se representaron gráficamente en imágenes que conformaron una base de datos de momentos de mercado. La base de imágenes se utilizó para localizar y etiquetar objetos de las clases correspondientes a los eventos que componen la estrategia elegida. La base se separó en conjuntos de entrenamiento, pruebas y validación y con los cuales se entrenó un modelo de red neuronal convolucional para la detección de objetos de las clases de eventos de la estrategia. El modelo detecta los objetos en imágenes nuevas permitiendo generar alertas y señales que se pueden usar para tomar decisiones de compra y venta de activos.

En la primera parte de este documento se presenta información sobre estudios que se han realizado relacionados con la predicción de precios de los activos del mercado mediante el uso de modelos de series de tiempo univariadas, modelos de series de tiempo multivariadas, aprendizaje profundo por refuerzo y clasificación de imágenes. Acto seguido, se exponen los temas que, al cruzarse, conforman la temática de este estudio: el análisis técnico de los mercados y la visión artificial. Por último, se describe la metodología del estudio, sus resultados y conclusiones.

2. Trabajo relacionado

Los datos del comportamiento de los activos en el mercado son series de tiempo. Los modelos de series de tiempo tienen como objetivo varios propósitos de estudio, en el caso del mercado financiero su función básica es representar el valor de los activos a través del tiempo. Pero hay tareas más complejas en el análisis de series de tiempo, en especial en los mercados financieros, como el ajuste de curva, la clasificación y el pronóstico.

Para la clasificación de series de tiempo - es decir, para solucionar el problema de asignar una clase o categoría a una serie de tiempo nunca antes vista - se han propuesto varios modelos de aprendizaje

profundo. Algunos entregan buen rendimiento a pesar de que no requieren preprocesamiento o creación de características de entrada a la red como es el caso de las Redes Completamente Convolucionales (FCN) o la estructura ResNet [Wang, Yan y Oates 2016]. Usando agrupamiento promedio en el modelo convolucional se aprovecha el Mapa de Activación de Clase (CAM) para reconocer la región de los datos brutos que más contribuye a la identificación de las etiquetas específicas.

Para el pronóstico de series de tiempo multivariadas se han propuesto modelos de aprendizaje profundo con arquitecturas de redes LSTNet. La red LSTNet combina las fortalezas de las redes neuronales convolucionales (CNN), las redes neuronales recurrentes (RNN) y un componente autorregresivo para extraer patrones de dependencia entre variables a corto plazo y descubrir patrones de tendencia a largo plazo [Lai y col. 2017].

Se ha estudiado el desempeño de redes neuronales convolucionales (CNN) y Redes de memoria de Largo y Corto plazo (LSTM) para reconocer patrones de gráficos comunes en un historial de datos del mercado de activos. En el artículo «Stock Chart Pattern recognition with Deep Learning» [Velay y Daniel 2018], el modelo LSTM obtuvo tasas de detección aceptables mientras que los modelos CNN 1D y 2D no alcanzaron niveles de precisión razonables. El estudio recomienda utilizar codificadores para reducir las matrices dispersas permitiendo que el modelo pueda extrapolar información y mejore la tasa de detección de la CNN. Este enfoque también sugiere utilizar los codificadores para limpiar la serie temporal antes de que entre en la red LSTM de modo que identifique los patrones sin tener en cuenta el ruido de la volatilidad de precios.

Hay una clasificación bajo una taxonomía unificada de modelos de series de tiempo con aprendizaje profundo, que los divide en dos categorías principales: generativos y discriminatorios. Los resultados muestran que el aprendizaje profundo puede lograr rendimiento de vanguardia para clasificación de series de tiempo con arquitecturas como redes neuronales completamente convolucionales (FCN) y redes residuales profundas (RRP) [Fawaz y col. 2019]. Además, muestra cómo el efecto de caja negra de los modelos profundos que los hace imposibles de interpretar, se puede mitigar con una clase de visualización del mapa de activación que destaca qué elementos de la serie temporal de entrada contribuyeron de manera significativa para la identificación de la clase. Admiten que la clasificación de series de tiempo, a diferencia de las tareas de visión artificial y procesamiento de lenguaje natural, todavía carece de un estudio exhaustivo del aumento de datos y el aprendizaje de transferencia.

En «Multivariate LSTM-FCNs for time series classification» [Karim y col. 2019] se presentó una transformación a los modelos de clasificación de series de tiempo univariantes existentes. La innovación de los modelos se presentó con una red totalmente convolucional con memoria larga a corto plazo (LSTM-FCN) y otra red totalmente convolucional con una capa de atención y memoria (ALSTM-FCN). Ambas clasificaciones de serie de tiempo dan como resultado modelos de clasificación de series de tiempo multivariante que incluyen un bloque de compresión y excitación al bloque convolucional para mejorar aún más la precisión. Los modelos propuestos superan a la mayoría de los modelos de última generación y requieren un preprocesamiento y extracción de características mínimas.

Hay iniciativas de automatización de swing trading utilizando aprendizaje profundo por refuerzo [Azhikodan, Bhat y Jadhav 2019]. El modelo de red neuronal profunda basado en gradientes con reglas deterministas se entrena para elegir comprar, vender o mantener un activo para maximizar la ganancia. Este tipo de modelo reconoce la necesidad de un sistema que apoye el algoritmo de aprendizaje por refuerzo pronosticando la tendencia en el valor del activo. Para reforzar la predicción del comportamiento de los activos, este modelo utilizó una red neuronal convolucional recurrente para predecir la tendencia bursátil de las noticias financieras.

Otros enfoques se basan en el éxito del reconocimiento de imágenes y examinan el valor de transformar el análisis tradicional de series de tiempo en un problema de clasificación de imágenes. Estos estudios sugieren que la transformación de un problema de clasificación de series de tiempo de carácter numérico continuo puede dar una perspectiva útil en el análisis técnico de las series de tiempo del valor de los activos. La metodología parte de la creación de una base de datos de imágenes de series temporales financieras codificadas como gráficos de velas (caja y bigotes), en estas imágenes se etiquetan tres estra-

teguas comerciales. Utilizando las imágenes, se entrenó más de una docena de modelos de clasificación de aprendizaje automático y se descubrió que los algoritmos logran abstraer las complicadas reglas de generación de etiquetas multiescala cuando los datos se representan visualmente [Cohen, Balch y Veloso 2019].

Una popular herramienta de aprendizaje profundo llamada memoria larga a corto plazo (Long Short Term Memory - LSTM) ha demostrado ser muy eficaz en muchos problemas de predicción de series de tiempo. Este modelo utiliza datos macroeconómicos para el análisis fundamental y datos de indicadores técnicos para el análisis técnico [Yildirim, Toroslu y Fiore 2021]. Esta estrategia mixta, que combina dos LSTM separados correspondientes a estos dos conjuntos de datos, resultó ser bastante exitosa en los experimentos utilizando series de tiempo reales de activos comerciales.

En 2019 se propuso un modelo de predicción del precio de los activos del mercado basado en red neuronal convolucional (CNN). Con el objetivo de valorar el rango de aplicación de nuevos métodos de aprendizaje en los mercados de valores se eligieron 9 indicadores como valores técnicos que modelo de pronóstico, y los indicadores técnicos se convirtieron en imágenes en los gráficos de series de tiempo. Para verificar la utilidad del aprendizaje y su impacto sobre el reconocimiento de imágenes en los mercados de activos, los valores técnicos del modelo propuesto se compararon con el modelo típico de red neuronal artificial (ANN) y el modelo de máquina de vectores de soporte (SVM). El resultado mostró que la herramienta CNN es una buena opción para construir modelos de predicción de existencias [Sim, Kim y Ahn 2019].

Por último, en 2020 se planteó un modelo que permite predecir la tendencia de las series de tiempo del valor de un activo y, como resultado, arroja una decisión de compra o venta sobre el mismo, lo que representa una ayuda para los inversores. El sistema utiliza un modelo de detección de objetos en tiempo real YOLO (You Only Look Once: Sólo se mira una vez) y reconoce los movimientos de un activo del mercado financiero sobre el gráfico de velas en 2D. Su objetivo final es proporcionar un mecanismo de previsión que refuerce las determinaciones de 'compraventa' que deben hacerse en el proceso de toma de decisiones en conjunto con los demás análisis técnicos básicos. Se encontró que el modelo tuvo un éxito de un 85 % de acierto en la predicción del valor de los activos en los que se enfocó y adicionalmente se generó un beneficio del 100 %. Este modelo se puede utilizar en cualquier serie que se pueda adaptar a gráficos de velas [Birogul, Temür y Köse 2020].

Como se puede observar, algunos de los estudios realizados hasta el momento sobre el pronóstico del precio de un activo en el mercado, utilizan series de tiempo univariadas para la predicción del precio del activo en el mercado. Otros tienen como propósito el estudio de series de tiempo multivariadas incluyendo además del precio, información adicional como indicadores de análisis técnico, variables macroeconómicas o noticias. Hay algunos estudios que además del pronóstico del precio intentan identificar los instantes ideales para realizar compra y venta de activos y automatizar estas operaciones. Por último, algunos de los estudios citados anteriormente plantean la posibilidad de analizar la información no como conjuntos de datos temporales sino como imágenes.

3. Análisis técnico de los mercados

El análisis técnico, está enfocado en predecir el comportamiento futuro del mercado teniendo en cuenta los movimientos básicos del precio y datos sobre volúmenes previos. El análisis técnico se usa comúnmente en valores y otros activos de los mercados financieros tradicionales, pero también es un componente importante de la compraventa de monedas digitales en el mercado de criptomonedas.

A diferencia del análisis fundamental, que para detectar oportunidades de inversión tiene en cuenta factores como estados financieros (Balance General, Estado de Resultados, Estado de Flujo de Efectivo, etc) o el valor intrínseco del activo; el análisis técnico se centra únicamente en el histórico del precio y el volumen de los activos. Por eso es usado como herramienta para estudiar los movimientos del precio y el volumen de un activo, y muchos analistas lo utilizan tratando de identificar tendencias y oportunidades de compraventa favorables.

A pesar de que hay registros del análisis técnico que se remontan al siglo XVII en Ámsterdam y siglo XVIII en Japón, el análisis técnico moderno frecuentemente está relacionado al trabajo de Charles Dow. Periodista financiero y fundador de The Wall Street Journal, Dow fue una de las primeras personas en identificar que los activos y mercados, a menudo, se mueven por tendencias que pueden ser segmentadas y examinadas. Su estudio daría a luz posteriormente, a la Teoría Dow que alentaría nuevos desarrollos en el campo del análisis técnico.

En las etapas iniciales, el enfoque rudimentario del análisis técnico se basaba en fichas hechas a mano y cálculos manuales; pero con el avance de la tecnología y la informática moderna, el análisis técnico se generalizó, hasta el punto de convertirse en la actualidad en una importante herramienta para muchos inversores y analistas.

Existen varias estrategias de comercio de activos financieros con diferentes periodos de tiempo en los que la posición en el mercado permanece abierta, es decir, distintos plazos entre la compra y la venta:

- Buy and hold (Meses, años)
- Swing (Semanas, días)
- Day (Apertura y cierre)
- Intraday (Horas, minutos)

A medida que aumenta la duración de una operación, también crece su riesgo asociado. En este estudio nos centramos en operaciones con duraciones cortas para manejar un riesgo menor con respecto a las operaciones de mayor duración. Se manejan posiciones abiertas con duraciones de minutos y horas, con la meta de capitalizar las rápidas fluctuaciones en el precio aprovechando la volatilidad del mercado de los crypto activos. La mayoría de los sistemas de negociación intradía se basan en el análisis técnico.

El análisis técnico estudia la interacción del mercado entendida como las tres fuentes principales de información disponibles: precio, volumen e interés abierto, primordialmente a través del uso de gráficos, con el propósito de predecir futuras tendencias del precio y tomar decisiones en periodos de tiempo pequeños. (John J Murphy - Technical Analysis Of The Financial Markets)

Entrar y salir del mercado de manera eficiente es la clave: la liquidez insuficiente aumenta los costos relacionados con el comercio asociados con los diferenciales y el deslizamiento. Las estrategias intradiarias dependen de obtener pequeñas ganancias mientras se asume un riesgo limitado repetidamente para generar rentabilidad.

Los indicadores se clasifican en cuatro tipos de acuerdo a la información con los que se construyen:

- Indicadores de tendencia
- Indicadores de momento
- Indicadores de volatilidad
- Indicadores de volumen

También se categorizan de acuerdo al momento en el tiempo que representan:

- Hacia adelante (Leading)
- Hacia atrás (Lagging)

3.1. Indicadores técnicos

3.1.1. Velas OHLC

Los gráficos de velas (Figura 1) son un indicador técnico que usa representaciones de barras para mostrar cuatro datos durante un período: precio de apertura, máximo, mínimo y de cierre (OHLC). Generalmente el precio de cierre se considera como el más importante por ser cronológicamente el último dato disponible. Estos gráficos pueden mostrar un impulso creciente o decreciente del precio. Cuando la apertura y el cierre están muy separados, muestra un fuerte impulso, y cuando la apertura y el cierre están muy juntos, muestra indecisión o un impulso débil. El máximo y el mínimo muestran el rango de precios completo del período, indicador de volatilidad. Las velas sirven para representar cualquier período de tiempo. Los extremos de la línea vertical del centro de la barra representan el precio máximo y mínimo del período. Los lados superior e inferior del rectángulo representan el precio de apertura y cierre del periodo. Cuando la barra es de color verde, el precio de cierre es el lado de arriba de la barra y el precio de apertura es el lado de abajo de la barra. Cuando la barra es de color rojo, el precio de cierre es el lado de abajo de la barra y el precio de apertura es el lado de arriba de la barra.



Figura 1: Representación cronológica del precio como velas OHLC. Fuente: elaboración propia

3.1.2. Volumen

En los mercados financieros, el volumen se refiere a la cantidad de dinero que se negoció en torno al activo durante un período de tiempo determinado. En el mercado de cripto activos, el volumen indica el valor total de las transacciones -compra o venta- ocurridas durante un periodo de tiempo específico. El volumen es en sí mismo un indicador técnico de volatilidad que se representa como un histograma (Figura 2).

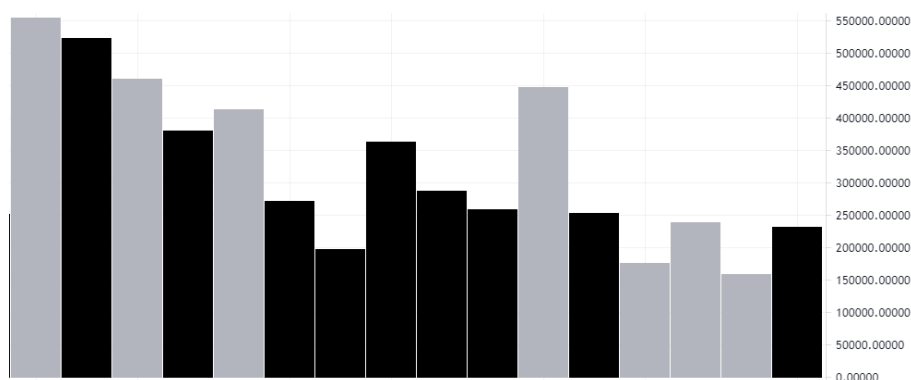


Figura 2: Representación cronológica del volumen del mercado. Fuente: elaboración propia

3.1.3. Medias móviles

Una media móvil es el promedio aritmético de los n datos anteriores. Cuanto mayor sea el valor de n , más datos históricos afectan su tendencia. El cruce de medias móviles con diferentes valores de n es una estrategia muy común para entrar y salir del mercado. No son un indicador predictivo per se, sino que marcan la tendencia de acuerdo a los datos históricos y simplemente muestran el precio promedio durante un período de tiempo determinado.

Cuando una media móvil de corto plazo (rápida) cruza por encima de una media móvil de largo plazo (lenta), se podría considerar como una señal de compra, ya que indica que la tendencia del precio puede estar cambiando al alza. Cuando una media móvil rápida cruza por debajo de una media móvil lenta, se podría considerar como una señal de venta, ya que indica que la tendencia del precio puede estar cambiando a la baja, como se observa en el cruce del ejemplo de la Figura 3, en donde la línea oscura es la media móvil lenta y la más clara es la rápida.

Existen varios métodos de cálculo de medias móviles, la simple es el promedio de los n datos anteriores, la ponderada en donde se asigna más peso de manera lineal a los datos más recientes y la exponencial en donde se asigna más importancia a los datos más recientes de manera exponencial. La fórmula para calcular la media móvil simple es:

$$SMA_n = \frac{1}{n} \sum_{i=0}^{n-1} x_{n-i}$$

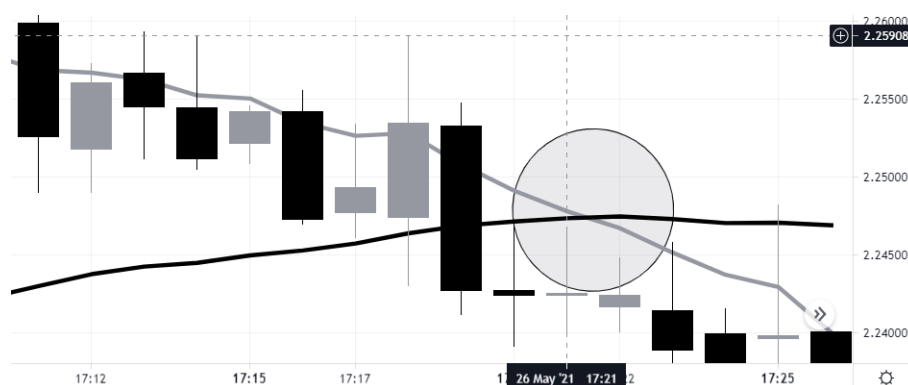


Figura 3: Medias móviles y sus cruces. Fuente: elaboración propia

3.1.4. MACD

La convergencia divergencia de media móvil (MACD) es un indicador de referencia que proporciona el precio promedio al que se ha negociado un valor a lo largo del día, según el volumen y el precio. El indicador fue creado por Gerald Appel a finales de la década de 1970.

Los gráficos MACD generalmente incluyen tres componentes: la línea MACD, la línea de señal y un histograma de la diferencia entre los dos. En su forma más simple: una señal alcista ocurre cuando la línea MACD cruza por encima de la línea de Señal, y una señal bajista ocurre cuando la línea MACD cruza por debajo de la línea de Señal (Figura 4).

El valor de la línea MACD se calcula restando el promedio móvil exponencial (EMA) de 26 períodos del EMA de 12 períodos. Una vez que se completa ese cálculo, se puede determinar la línea de señal y un histograma de la diferencia de los dos.

La línea de señal es simplemente un promedio móvil de la línea MACD, típicamente una EMA de 9 períodos, que "suaviza" los valores de la línea MACD, produciendo un conjunto de datos "más lento".

$$MACD = EMA_{12} - EMA_{26}$$

$$Signal = EMA_9 \cdot MACD$$

$$MACD_{Hist} = MACD - Signal$$

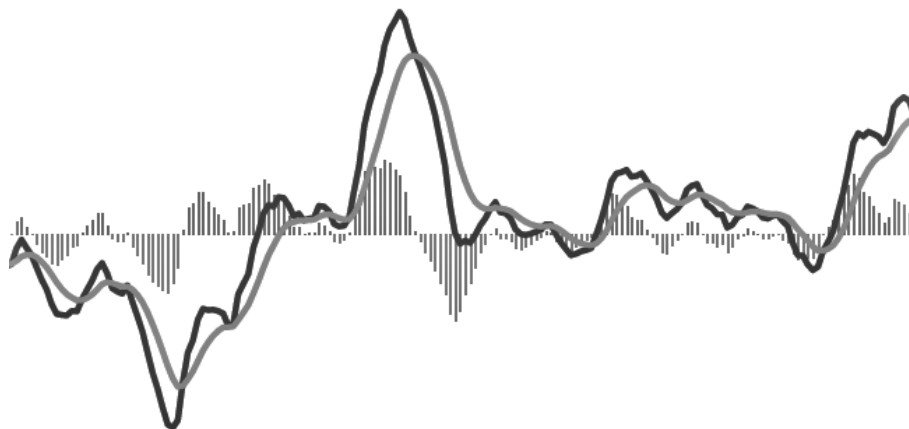


Figura 4: Convergencia y divergencia de medias móviles (MACD). Fuente: elaboración propia

3.1.5. RSI

El índice de fuerza relativa (RSI) es un indicador de impulso utilizado en el análisis técnico que mide la magnitud de los cambios de precio recientes para evaluar las condiciones de sobrecompra o sobreventa en el precio de un activo. El RSI se muestra como un oscilador (un gráfico de líneas que se mueve entre dos extremos) y puede tener una lectura de 0 a 100.

La interpretación y el uso tradicionales del RSI son que los valores de 70 o más indican que un valor se está sobrecomprado o sobrevalorado y puede estar preparado para un cambio de tendencia o un retroceso correctivo en el precio. Una lectura de RSI de 30 o menos indica una condición de sobreventa o infravalorada (Figura 5).

$$RSI_{pasouno} = 100 - \left[\frac{100}{1 + \frac{GananciaPromedio}{PerdidaPromedio}} \right]$$

3.1.6. Bandas de Bollinger

Las bandas de Bollinger son una herramienta de análisis técnico definida por un conjunto de líneas de tendencia que trazan dos desviaciones estándar (positiva y negativamente) de una media móvil simple (SMA) del precio de un activo (Figura 6).

El primer paso para construir las Bandas de Bollinger es calcular el promedio móvil simple del valor en cuestión, generalmente utilizando una SMA de 20 días. Posteriormente, se obtienen las bandas envolventes sumando y restando la desviación estándar a la media móvil simple del paso anterior.

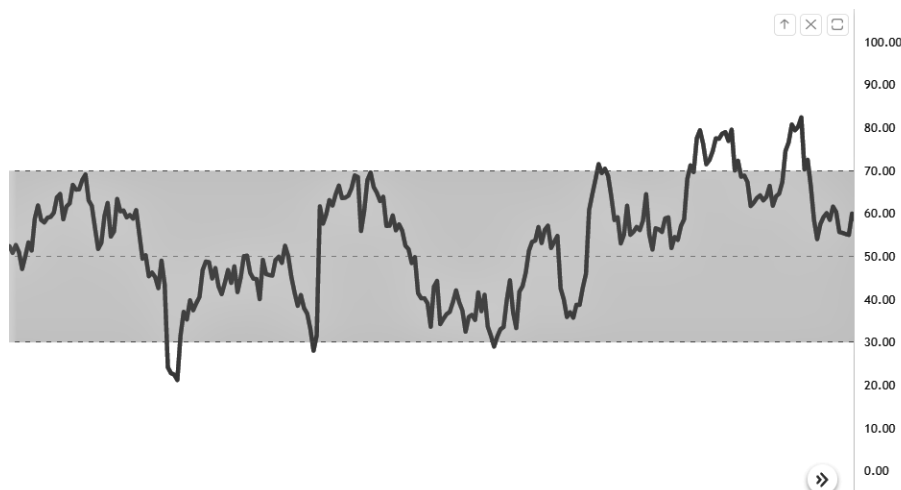


Figura 5: Índice de fuerza relativa (RSI). Fuente: elaboración propia

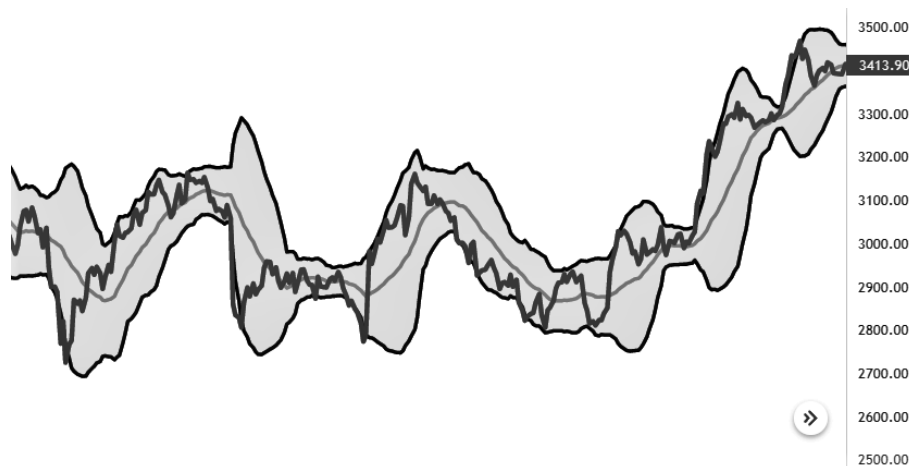


Figura 6: Bandas de Bollinger. Fuente: elaboración propia

3.2. Estrategias de compraventa

Una estrategia de compraventa es una serie de pasos que responden a un objetivo de retornos financieros en un plan de inversión. Las estrategias de compraventa analizan los patrones de las series de tiempo de los activos y con este análisis intentan predecir su comportamiento futuro para entregar un plan de compra o venta para el inversionista. Hay muchas estrategias dependiendo de ese objetivo y los métodos que los inversionistas prefieran. Tienen como objetivo brindar algún grado de certeza sobre los rendimientos financieros que se obtendrán en las operaciones, pero su eficiencia depende de la habilidad de un analista para reconocer patrones sistemáticamente.

Las estrategias de compraventa expuestas a continuación se conocen como estrategias de búsqueda alfa, ya que su objetivo principal es generar rendimientos positivos por encima del mercado independientemente de la dirección del mercado. Están basadas en herramientas financieras simples que analizan el comportamiento de un solo activo o criptomoneda. Las señales de estas estrategias se derivan de datos estructurados de series de tiempo financieras. Estas estrategias no usan otras fuentes de datos no estructuradas como noticias o publicaciones de redes sociales.

3.2.1. Estrategia de Medias Móviles

Una de las estrategias más populares se basa en medias móviles simples (SMA) para generar señales de compraventa en el mercado. Estas estrategias se han popularizado por los analistas técnicos. La idea básica es que una SMA a corto plazo que tiene un valor más alto que una SMA a más largo plazo indica una posición de mercado de compra y el escenario opuesto indica una posición de mercado neutral o de venta.

Cuando la SMA de corto plazo de un valor cruza por encima de la SMA de largo plazo se configura una posible señal de compra. Por ejemplo, la configuración clásica aquí es cuando la SMA de 20 días cruza por encima de la SMA de 100 días. Esta es una señal alcista e indica que el precio del valor puede seguir subiendo.

Cuando la SMA de corto plazo de un valor cruza por debajo de su SMA de largo plazo se configura una posible señal de venta. Por ejemplo, cuando la SMA de 20 días cruza por debajo de la SMA de 100 días. Esta es una señal bajista e indica que el precio del valor puede seguir cayendo.

Esta es una estrategia que se ha usado durante décadas e inició con el análisis técnico de los activos del mercado [Brock, Lakonishok y LeBaron 1992], que estudió empíricamente estas estrategias de manera sistemática: 'En este documento, exploramos dos de las reglas técnicas más simples y populares: oscilador de promedio móvil y ruptura del rango comercial (resistencia y soporte niveles). En el primer método, las señales de compra y venta se generan mediante dos promedios móviles, un período largo y un período corto. Nuestro estudio revela que el análisis técnico ayuda a predecir cambios en las acciones'.

3.2.2. Estrategia de Momento

La idea básica detrás de las estrategias de momento es la suposición de que un activo financiero se comporta de acuerdo con su desempeño reciente durante algún tiempo adicional. Por ejemplo, cuando un índice bursátil ha tenido un rendimiento negativo en promedio durante los últimos cinco días, se asume que su rendimiento también será negativo mañana.

Hay dos tipos básicos de estrategias de momento. El primer tipo son las estrategias de momento transversales. Al seleccionar de un grupo más grande de activos, estas estrategias compran aquellos activos que recientemente han tenido un rendimiento superior en relación con sus pares (o un índice de referencia) y venden aquellos activos que han tenido un rendimiento menor. En «Momentum Strategies» [Chan, Jegadeesh y Lakonishok 1996] y «Profitability of Momentum Strategies: An Evaluation of Alternative Explanations» [Jegadeesh y Titman 2001] se estudia este tipo de estrategias concluyendo que los activos continúan obteniendo resultados mayores y menores, respectivamente, al menos durante un cierto período de tiempo.

El segundo tipo son las estrategias de momento de series de tiempo. Estas estrategias compran aquellos activos que recientemente han tenido un buen desempeño y venden aquellos activos que recientemente han tenido un desempeño pobre. En este caso, el índice de referencia son los rendimientos pasados del propio activo. En «Time series momentum» [Moskowitz, Ooi y Pedersen 2012] se analizó este tipo de estrategia en múltiples mercados. Escriben: "... en lugar de centrarse en los rendimientos relativos de los valores en la sección transversal, el momento de las series de tiempo se centra exclusivamente en el rendimiento pasado del propio valor ... nuestro hallazgo del impulso de las series de tiempo en prácticamente todos los activos que examinamos parece desafiar el 'paseo aleatorio', que en su forma más básica implica que saber si un precio subió o bajó en el pasado no debería ser informativo sobre si subirá o bajará en el futuro".

3.2.3. Estrategia de reversión de media

En las estrategias de reversión de media, se supone que un activo financiero vuelve a algún nivel medio o de tendencia si actualmente está lo suficientemente lejos de dicho nivel. Por ejemplo, suponga que el

precio de una criptomoneda está 10 USD por debajo de su nivel de SMA de 200 días de 100. Entonces se espera que el precio vuelva a su nivel de SMA pronto.

En términos generales, las estrategias de reversión a la media se basan en un razonamiento opuesto a las estrategias de impulso. Si un activo financiero se ha desempeñado 'demasiado bien' en relación con su tendencia, su valor posiblemente va a bajar y viceversa. En otras palabras, mientras que las estrategias de impulso (series de tiempo) suponen una correlación positiva entre los rendimientos, las estrategias de reversión a la media suponen una correlación negativa. Balvers [Wu, Balvers y Gilliland 2000] escribe: "La reversión a la media se refiere a una tendencia de los precios de los activos a volver a una trayectoria de tendencia".

Trabajando con un promedio móvil simple (SMA) como un proxy para una 'trayectoria de tendencia', una estrategia de reversión a la media se puede probar de manera similar a las estrategias de SMA e impulso. La idea es definir un umbral para la distancia entre el precio actual de los activos y la SMA, que indica una posición de compra o venta.

4. Visión artificial

La visión artificial es un campo de la inteligencia artificial (IA) que permite a programas de computador identificar información a partir del análisis de imágenes. La visión artificial entrena a los computadores para realizar tareas de clasificación, identificación de objetos y análisis de escenas de acuerdo al contexto.

La visión artificial requiere de mucha información para aprender. Ejecuta análisis de datos repetidamente hasta que identifica características en las imágenes que le permite diferenciarlas. Para lograr este objetivo se usa el aprendizaje profundo con las redes neuronales convolucionales (CNN).

El aprendizaje profundo permite que un programa de computador identifique información sobre datos visuales. Los algoritmos de aprendizaje profundo permiten aprender al computador por sí solo, sin que nadie lo programe. Una CNN realiza operaciones matemáticas para hacer predicciones sobre las imágenes. La red neuronal ejecuta convoluciones y verifica la precisión de sus predicciones en una serie de iteraciones hasta que las predicciones adquieren la precisión deseada.

4.1. Redes Neuronales

Una red neuronal calcula una función de las entradas propagando los valores calculados de las neuronas de entrada a las neuronas de salida y utilizando los pesos como parámetros intermedios. El aprendizaje ocurre cambiando los pesos que conectan las neuronas. Así como se necesitan estímulos externos para aprender en redes neuronales biológicas, el estímulo externo en las redes neuronales artificiales lo proporcionan los datos de entrenamiento que contienen ejemplos de pares de entrada-salida de la función que se va a aprender. Ajustando sucesivamente los pesos entre neuronas sobre muchos pares de entrada-salida, la función calculada por la red neuronal se perfecciona con el tiempo para proporcionar predicciones más precisas. La habilidad de calcular con precisión funciones de entradas nunca vistas mediante el entrenamiento sobre un conjunto finito de pares entrada-salida se conoce como generalización del modelo. La utilidad del aprendizaje automático se debe a la capacidad de los modelos para aplicar lo aprendido a partir de los datos de entrenamiento en nuevos datos nunca antes vistos.

La red neuronal más simple se conoce como perceptrón, contiene una sola capa de entrada y un nodo de salida. Las redes neuronales multicapa contienen múltiples capas computacionales; las capas intermedias adicionales (entre la entrada y la salida) son denominadas capas ocultas porque los cálculos realizados no son visibles para el usuario. La arquitectura específica de las redes neuronales multicapa se conoce como redes de alimentación hacia adelante, porque las capas sucesivas se alimentan entre sí en la dirección de avance desde la entrada hasta la salida.

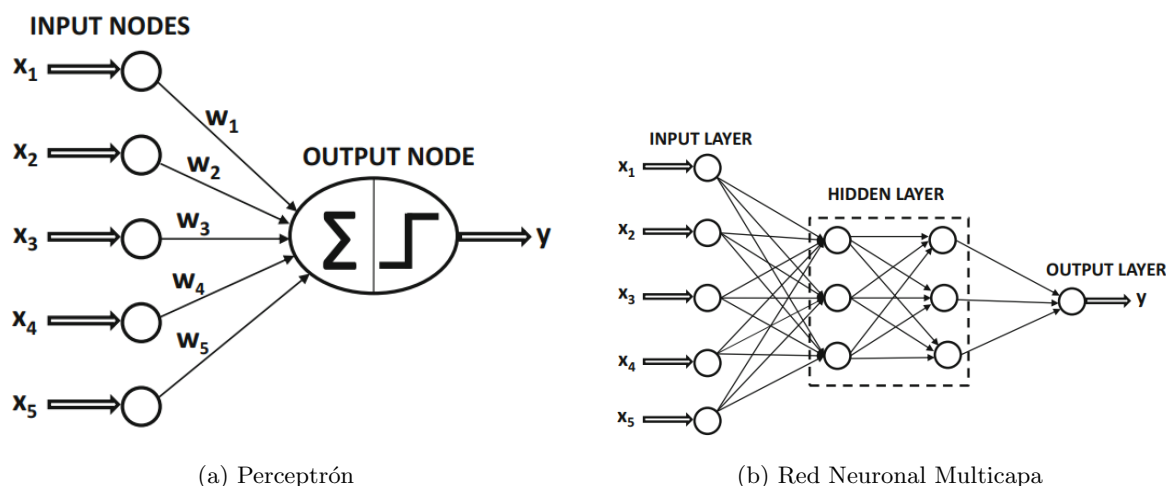


Figura 7: Red Neuronal Unicapa y Multicapa. Fuente: Aggarwal, 2018

4.2. Redes Neuronales Convolucionales

Las redes neuronales convolucionales fueron las primeras redes profundas exitosas. Están inspiradas en el comportamiento de la corteza visual de los animales, que tienen pequeños grupos de células sensibles a regiones específicas del campo visual. La sensibilidad de las células también depende de la forma, tamaño y orientación de los objetos en el campo visual. Por ejemplo hay grupos de células neuronales que responden a los bordes verticales y, de igual modo, otros grupos de células que se activan con los bordes horizontales. Estos grupos de células están conectados usando una arquitectura de capas.

Éste modelo biológico se reprodujo en modelos neuronales que han evolucionado con el tiempo iniciando con la arquitectura Neocognitron, que luego dio paso a la primera arquitectura completamente convolucional: LeNet. A partir de ese momento, la red neuronal convolucional se ha mantenido básicamente igual. Las arquitecturas que han surgido luego se diferencian en que utilizan más capas y agregan más funciones estables de activación como la ReLU [Aggarwal 2018].

Las redes neuronales convolucionales (CNN) fueron concebidas para trabajar con matrices con dependencias espaciales en regiones locales como las imágenes. Las características de la entrada en una red neuronal convolucional tienen dependencia entre sí basadas en distancias espaciales.

Las redes neuronales convolucionales agrupan características similares a partir de regiones locales con patrones parecidos. La propiedad principal de las redes neuronales convolucionales es que usan convoluciones. Una convolución es un producto escalar entre un conjunto de ponderaciones organizado en una matriz y entradas similares organizadas en matrices extraídas de diferentes partes del volumen de entrada. La multiplicación de matrices entre el filtro y la entrada es conocida como retorsión, que da el nombre a las redes neuronales convolucionales. Esta operación es fundamental para trabajar datos con un alto nivel de localidad espacial como por ejemplo las imágenes. Las redes neuronales convolucionales son redes que utilizan convoluciones en al menos una de sus capas, aunque la mayoría de las redes neuronales convolucionales utilizan varias capas de convoluciones.

Sin importar que tan sofisticada sea un CNN, tiene tres elementos fundamentales: los filtros de convolución (Convolutions), la aplicación de una función de activación a la salida de los filtros (ReLU) y la agrupación de la salida de la función de activación (Pooling).

Las redes neuronales convolucionales cambian tres aspectos de las redes neuronales convencionales para facilitar la clasificación de imágenes: Reduce el número de nodos de entrada, tolera desplazamientos pequeños en la ubicación de los píxeles de la imagen y toma ventaja de las correlaciones que se observan en las imágenes complejas.

4.2.1. Convolución

En una red neuronal convolucional, los parámetros se organizan en conjuntos de estructuras tridimensionales, conocidas como filtros o núcleos. El filtro suele ser cuadrado y más pequeño que las de la capa a la que se aplica, y su profundidad siempre es la misma de la capa a la que se aplica.

Suponga que las dimensiones del filtro en la \bar{q} ésima capa son $Fq \times Fq \times dq$, por ejemplo, $5 \times 5 \times 3$. La convolución coloca el filtro en cada posición posible de la capa calculando el producto escalar entre los parámetros $Fq \times Fq \times dq$ en el filtro y la cuadrícula correspondiente en el volumen de entrada del mismo tamaño. Al realizar convoluciones en la capa \bar{q} ésima, se puede alinear el filtro en las posiciones $L_{q+1} = (Lq - Fq + 1)$ a lo largo de la imagen y $A_{q+1} = (Aq - Fq + 1)$ a lo ancho de la imagen. Esto da como resultado un total de $L_{q+1} \times A_{q+1}$ productos escalares posibles, lo que define el tamaño de la siguiente capa oculta.

El filtro \bar{p} ésimo en la capa \bar{q} ésima tiene parámetros indicados por el tensor tridimensional $W^{(p,q)} = [w_{ijk}^{(p,q)}]$. Los índices i, j, k indican las posiciones a lo largo de la altura, el ancho y la profundidad del filtro. Los mapas de características en la \bar{q} ésima capa están representados por el tensor tridimensional $H^{(q)} = [h_{ijk}^{(q)}]$. Cuando el valor de q es 1, el caso especial correspondiente a la notación $H^{(1)}$ simplemente representa la capa de entrada. Entonces, las operaciones convolucionales desde la capa \bar{q} ésima hasta la capa $(q + 1)$ ésima se definen como sigue:

$$h_{ijp}^{(q+1)} = \sum_{r=1}^{F_q} \sum_{s=1}^{F_q} \sum_{k=1}^{d_q} w_{rsk}^{(p,q)} h_{i+r-1, j+s-1, k}^{(q)}$$

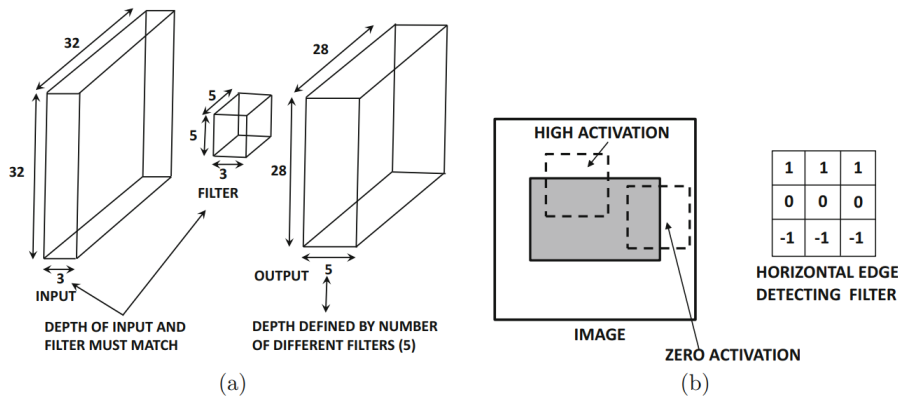


Figura 8: Convolución. Fuente: Aggarwal, 2018

En la figura 8(a) vemos que la convolución entre una capa de entrada de tamaño $32 \times 32 \times 3$ y un filtro de tamaño $5 \times 5 \times 3$ produce una capa de salida con dimensiones espaciales de 28×28 . La profundidad de la salida resultante depende del número de filtros distintos y no de las dimensiones de la capa de entrada o el filtro. Al deslizar un filtro alrededor de la imagen en la figura 8(b), se intenta buscar una característica particular en varias zonas de la imagen.

La convolución evoca los experimentos de Hubel y Wiesel que usaban excitaciones en regiones del campo visual para activar neuronas particulares. En el caso de las redes neuronales convolucionales, este campo visual está definido por el filtro, que recorre toda la imagen para identificar en ella la presencia de patrones. Los filtros de las capas iniciales tienden a detectar formas básicas, mientras que los filtros de las capas posteriores reconocen formas más complejas.

4.2.2. Activación (ReLU)

La capa de activación generalmente sigue después de una capa de convolución. El uso de la función ReLU como función de activación es reciente, y en sus inicios, la capa de activación de las redes convolucionales utilizaban las funciones sigmoide y tanh. El uso de la ReLU tiene enormes ventajas sobre otras funciones de activación tanto en términos de velocidad como de precisión. El aumento de la velocidad también está relacionado con la precisión porque permite utilizar modelos más profundos y entrenarlos durante más tiempo. En los últimos años, el uso de la función de activación de ReLU ha reemplazado casi por completo a las otras funciones de activación en el diseño de redes neuronales convolucionales.

Para cada uno de los valores de $Lq \times Bq \times dq$ en una capa, se le aplica la función de activación ReLU para crear valores de umbral de $Lq \times Bq \times dq$. Luego, estos valores se pasan a la siguiente capa. Por lo tanto, la aplicación de ReLU no cambia las dimensiones de la capa anterior porque es un simple mapeo uno a uno de los valores de activación.

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

4.2.3. Agrupamiento (Pooling)

La agrupación máxima simplemente selecciona el máximo valor de la región de tamaño $Pq \times Pq$ del mapa de características de salida de la capa de activación dq moviéndose una zancada a la vez. La zancada es el número de posiciones que se mueve el filtro cada vez en el mapa de características. Si se usa una zancada de 1, esto producirá una nueva capa de tamaño $(Lq - Pq + 1) \times (Bq - Pq + 1) \times dq$. Sin embargo, es más común usar una zancada $Sq > 1$ en la agrupación. En tales casos, la longitud de la nueva capa será $(Lq - Pq)/Sq + 1$ y la amplitud será $(Bq - Pq)/Sq + 1$. Por lo tanto, la agrupación reduce drásticamente las dimensiones espaciales de cada mapa de activación. La agrupación máxima destaca las regiones donde el filtro de convolución coincide de la mejor manera con la imagen de entrada.

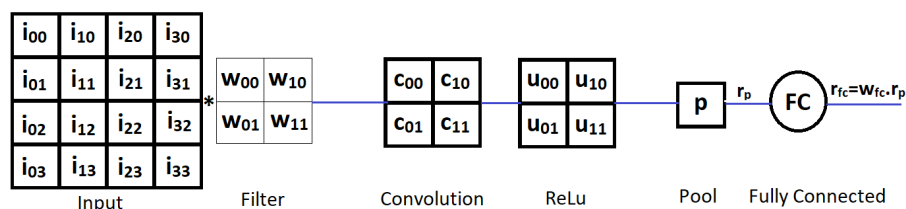


Figura 9: Convolución Activación y Agrupación. Fuente: Aggarwal, 2018

4.2.4. Entrelazado entre capas

Las capas de convolución, activación y agrupación se intercalan en la red neuronal convolucional para aumentar la eficiencia de la red. Las capas ReLU generalmente van después de las capas de convolución y suelen estar pegadas una tras la otra. Algunos diagramas de arquitectura de redes convolucionales como AlexNet no muestran explícitamente las capas ReLU porque se supone que siempre están pegadas al final de las capas convolucionales. Después de dos o tres conjuntos de combinaciones convolucional-ReLU, uno podría tener una capa de agrupación máxima. Ejemplos de este patrón básico son los siguientes:

$$CV - AC - CV - AC - AM$$

$$CV - AC - CV - AV - CV - AV - AM$$

La capa convolucional se expresa mediante el término CV, la capa de activación ReLU se expresa como AC y la capa de agrupación máxima se expresa como AM. Al repetir esta estructura varias veces, se conforma una red neuronal profunda.

El diagrama se puede complementar indicando el número de filtros, su tamaño, su relleno, y capas de agrupación. La capa de agrupación es muy importante porque reduce el tamaño de los mapas de activación. Las CNN suelen ser bastante profundas y a menudo encontramos redes convolucionales con 12 capas o más.

4.2.5. Entrenamiento de las redes neuronales convolucionales

El proceso de entrenamiento de una red neuronal convolucional utiliza el algoritmo de retropropagación en cada capa. En la capa de activación se utiliza el procedimiento común de retropropagación. Para la capa de agrupación máxima sin superposición entre grupos, solo se necesita identificar qué unidad es el valor máximo en un grupo. La derivada parcial de la pérdida con respecto al estado agrupado se devuelve a la unidad con máximo valor. A todas las entradas que no sean la entrada máxima en la cuadrícula se les asignará un valor de 0. El algoritmo de retropropagación a través de la agrupación máxima y las operaciones ReLU no son muy diferentes a las de las redes neuronales tradicionales [Ertel 2017].

Dada una red neuronal artificial prealimentada con $L > 2$ capas, w_{jk}^l representa el peso del enlace que conecta la neurona k -ésima en la capa $l - 1$ con la neurona j -ésima en la capa l , b_j^l representa el bias de la neurona j -ésima en la capa l y a_j^l representa la salida (activación) de la neurona j -ésima en la capa l .

1. **Se recibe un conjunto de ejemplos.**

2. **Para cada ejemplo de entrenamiento x :** Activar las neuronas de la primera capa $a^{x,1}$ con la entrada x .

a) **Propagación:** Para cada $l = 2, 3, \dots, L$ calcular $z^{x,l} = w^l a^{x,l-1} + b^l$ y $a^{x,l} = \sigma(z^{x,l})$

b) **Cálculo de errores en la capa de salida:** Calcular el vector $\delta^{x,L} = \nabla_a C_x \cdot \sigma'(z^{x,L})$

c) **Retropropagación del error:** Para cada $l = L-1, L-2, \dots, 2$ calcular $\delta^{x,l} = ((w^{l+1})^T \delta^{x,l+1}) \cdot \sigma'(z^{x,l})$

3. **Gradiente descendiente:** Para cada $l = L - 1, \dots, 2$ actualizar los pesos de acuerdo a la regla $w^l \rightarrow w^l - \frac{\eta}{m} \sum_x \delta^{x,l} (a^{x,l-1})^T$

4.3. Clasificación de imágenes

El desafío de reconocimiento visual a gran escala de ImageNet (ILSVRC) evalúa algoritmos para la detección de objetos y la clasificación de imágenes a gran escala. ImageNet es un conjunto de datos de más de 15 millones de imágenes etiquetadas de alta resolución que pertenecen a aproximadamente 22.000 categorías. El ILSVRC utiliza un subconjunto de ImageNet con aproximadamente 1000 imágenes en cada una de las 1000 categorías. En total, hay aproximadamente 1,2 millones de imágenes de formación, 50.000 imágenes de validación y 150.000 imágenes de prueba. Los grupos de investigación que han participado en el desafío han presentado arquitecturas que se han convertido en referente para la industria y demuestran el avance en este campo. Recientemente se han visto cambios en el diseño arquitectónico (como la activación con ReLU), pero sorprende que las arquitecturas modernas aún se basan en el diseño de LeNet-5. Los cambios más significativos de las nuevas arquitecturas con respecto a LeNet-5 son la inclusión de más capas para hacer redes más profundas, el uso de la función ReLU en la capa de activación y la optimización del entrenamiento de la red mediante el uso de hardware más potente y nuevas técnicas en software.

4.3.1. LeNet-5 (1998)

Fue una de las primeras redes neuronales convolucionales propuestas e impulsó el desarrollo del aprendizaje profundo. Tiene las unidades básicas de una CNN: capa de convoluciones, capa de agrupación y capa de conexión completa. Tiene 7 capas en total además de su entrada como se ve en la figura. La salida permite clasificar 10 categorías [Lecun y col. 1998].

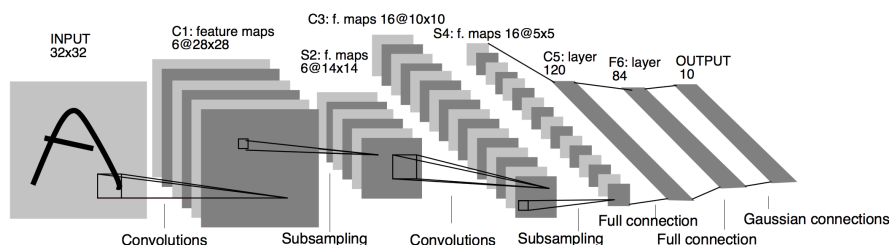


Figura 10: Arquitectura Lenet-5. Lecun y col. 1998

4.3.2. AlexNet (2012)

AlexNet se presentó en la competencia ImageNet en 2012. La red ganó con un error del 15,3%. En el segundo lugar la red ISI presentó un error de 26.1%. El aporte más importante que hizo AlexNet fue la adición de más capas, profundizando la arquitectura del modelo para mejorar el rendimiento [Krizhevsky, Sutskever e Hinton 2012]. Agregar más capas al modelo tiene efectos sobre el costo computacional durante el entrenamiento, pero se logró gracias a la utilización de unidades de procesamiento de gráficos (GPU). La arquitectura de red usa 11 capas además de la entrada. La salida permite clasificar 1000 categorías.

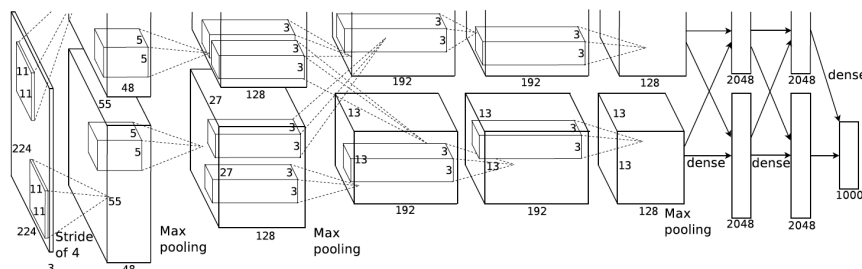


Figura 11: Arquitectura Alexnet. Krizhevsky, Sutskever e Hinton 2012

4.3.3. ZFNet (2013)

Este diseño es una sintonización de AlexNet. Reemplaza los filtros de 11x11 por unos de tamaño 7x7. Incluyó el uso de deconvoluciones para identificar el mapa de activación de clases. También ayudó a depurar visualmente cada una de las capas de la red, en lugar de utilizar únicamente el mapa de activación de clases de la última capa. Estas modificaciones mejoraron significativamente la precisión en la clasificación de imágenes. Tiene 5 capas convolucionales, capas de agrupación máxima y 3 capas completamente conectadas y una última capa softmax [Zeiler y Fergus 2014].

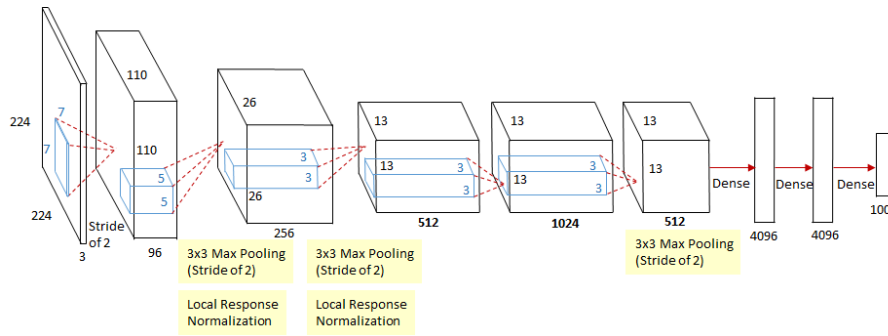


Figura 12: Arquitectura ZFNet. Zeiler y Fergus 2014

4.3.4. VGG (2014)

El aporte más importante de la arquitectura ganadora de la edición 2014 del ILSVRC es el uso de filtros muy pequeños para permitir aumentar la profundidad hasta 16 y 19 capas [Simonyan y Zisserman 2015]. Por lo demás, la red se caracteriza por su simplicidad: los únicos componentes adicionales son capas de agrupación y una capa completamente conectada. El modelo alcanza el 92,7% de precisión con el conjunto de datos de ImageNet. Supera a AlexNet al reemplazar los filtros de gran tamaño 11x11 en la primera y 5x5 segunda capa convolucional con múltiples filtros de tamaño 3×3 .

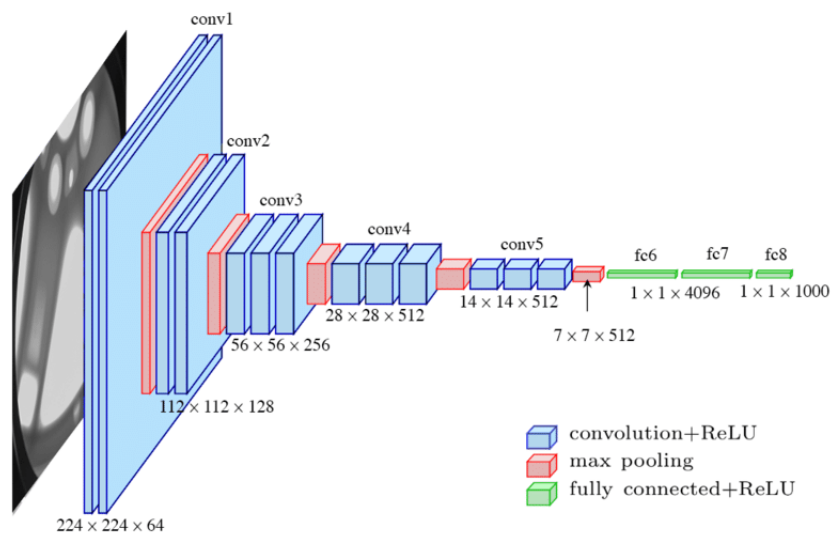


Figura 13: Arquitectura VGG. Simonyan y Zisserman 2015

4.3.5. GoogLeNet (2014)

GoogLeNet es una red neuronal convolucional profunda de 22 capas derivada de Inception Network. Se presentó en el ILSVRC de 2014. Redujo el número de parámetros de AlexNet de 60 millones a solo 4 y presentó un error de 6.67% [Szegedy y col. 2015].

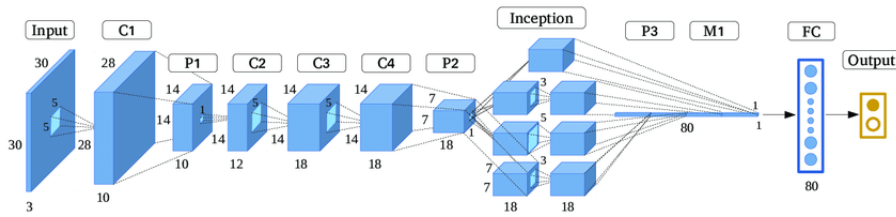


Figura 14: Arquitectura GoogLeNet. Szegedy y col. 2015

4.3.6. ResNet (2015)

Las arquitecturas propuestas hasta este momento incrementan el número de capas en el diseño y optimizan el rendimiento del entrenamiento. Pero con el aumento de la profundidad de la red, la precisión se satura y luego se degrada rápidamente. El equipo de Microsoft Research propuso una solución a este problema con la red residual ResNet [He y col. 2016]. En lugar de ajustar las capas apiladas directamente a un mapeo subyacente, las ajustaron explícitamente a un mapeo residual. Las conexiones de acceso directo son aquellas que omiten una o más capas. Esta mejora permite diseñar redes con 34, 50, 101 y hasta 152 capas de profundidad.

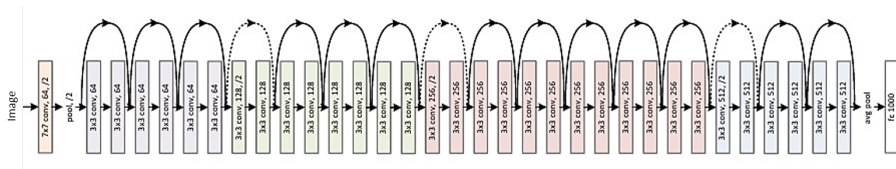


Figura 15: Arquitectura ResNet. He y col. 2016

4.3.7. EfficientNet (2019)

Las CNN normalmente se diseñan con un presupuesto computacional fijo y luego se amplían para una mayor precisión si hay más recursos disponibles. Con la arquitectura EfficientNet [Tan y Le 2019] se estudió el escalamiento del modelo y se identificó que equilibrar cuidadosamente la profundidad, el ancho y la resolución de la red puede conducir a un mejor rendimiento. Con base en esta observación, se propuso un nuevo método de escalamiento uniforme de todas las dimensiones utilizando un coeficiente compuesto simple pero altamente efectivo.

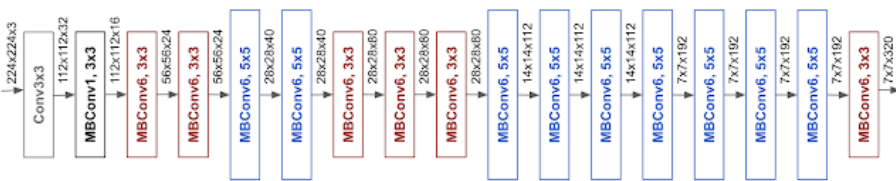


Figura 16: EfficientNet. Tan y Le 2019

4.3.8. Modelos preentrenados

Para el estudio propuesto en este trabajo, se requiere realizar la clasificación de un conjunto de imágenes con etiquetas diferentes y mucho más pequeño que ImageNet. Esta situación puede presentar problemas

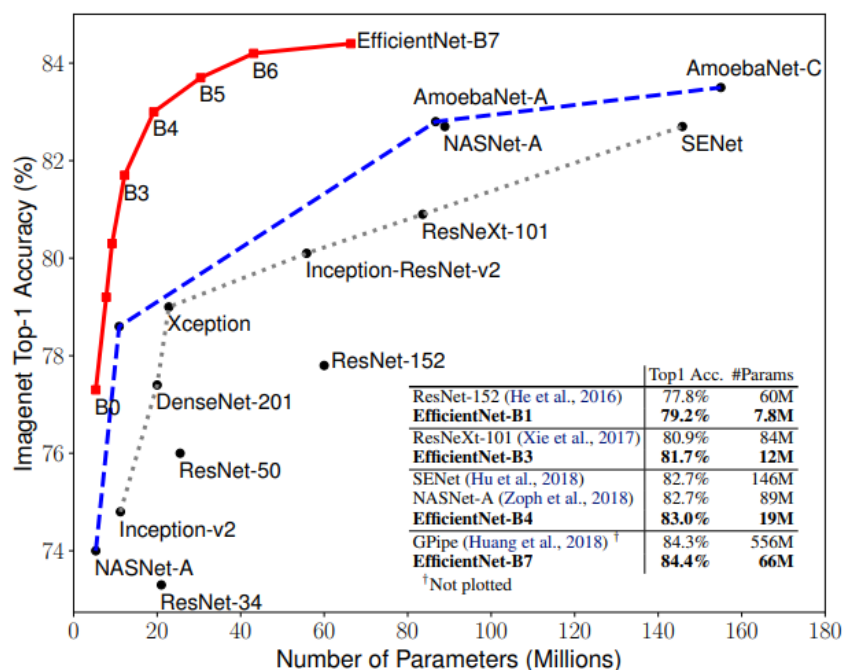


Figura 17: Comparación de tamaño vs precisión de modelos de CNNs. Tan y Le 2019

porque las redes neuronales requieren una gran cantidad de datos de entrenamiento cuando se construyen desde cero. Sin embargo, un aspecto fundamental de los datos de imágenes es que las características extraídas de un conjunto de datos en particular son altamente reutilizables en otras bases de datos.

Las características extraídas de los datos de ImageNet se pueden usar para representar un conjunto de datos de imágenes completamente diferente pasándolo a través de una red neuronal convolucional previamente entrenada extrayendo las características multidimensionales de las capas completamente conectadas. Esta nueva representación se puede utilizar para una aplicación completamente diferente, como la agrupación en clústeres o la recuperación.

La extracción de características estándar puede entenderse como aprendizaje por transferencia, porque estamos usando herramientas de dominio público como ImageNet para extraer características que permiten solucionar problemas en donde no hay suficientes datos de entrenamiento disponibles.

Si hay algunos datos de entrenamiento adicionales disponibles, se pueden usar para ajustar solo las capas más profundas (es decir, capas más cercanas a la capa de salida). Los pesos de las primeras capas (más cercanas a la entrada) son fijos. La razón para entrenar solo las capas más profundas, mientras se mantienen fijas las primeras capas, es que las capas anteriores capturan solo características primitivas como bordes, mientras que las capas más profundas identifican características más complejas. Las características básicas no cambian demasiado en diferentes situaciones, mientras que las características más específicas pueden ser sensibles dependiendo de la aplicación. Las primeras capas tienen la tarea de generalizar mientras que las capas posteriores particularizan.

4.4. Detección de objetos

La detección de objetos permite localizar e identificar objetos en una imagen usando programas de computador. Ha sido ampliamente usada en aplicaciones de la industria y del consumo masivo. Actualmente es común encontrar la detección de objetos en redes sociales como Facebook o servicios de almacenamiento

de imágenes como Google Photos para la organización y búsqueda de las mismas. Dada una imagen de entrada, se trata de predecir las ubicaciones de una cierta clase de objeto en la imagen ya sea como caja o como máscara (segmentación de instancia).

4.4.1. Metodos de dos etapas

Con estos métodos primero se generan las regiones propuestas y luego se clasifica cada región. Son métodos precisos, pero computacionalmente intensivos. Los tres métodos más representativos de esta clase son:

- **R-CNN (2013):** Extrae las regiones propuestas a través de búsqueda selectiva luego extrae las características de las regiones con una CNN y por último clasifica las regiones usando una máquina de vectores de soporte (SVM) [R. B. Girshick y col. 2013].
- **Fast R-CNN (2015):** Extrae las regiones propuestas a través de búsqueda selectiva y luego extrae las características de las regiones y las clasifica con una CNN [R. Girshick 2015].
- **Faster R-CNN (2015):** Usa una red de identificación de regiones propuestas (RPN), este modelo ya no usa búsqueda selectiva. Extrae las regiones propuestas así como sus características y por último las clasifica, todo lo anterior con la misma CNN [Ren y col. 2015].

4.4.2. Metodos de un solo disparo

Estos métodos usan una única etapa para la detección, se pueden pensar como si la RPN de la Faster R-CNN localizara y clasificara el objeto en un solo paso. Los métodos de un solo disparo son rápidos generalmente, pero tienen precisión moderada. Los métodos más representativos de esta clase son:

- **YOLO (2015):** Primero, la imagen se divide en varias cuadrículas. Cada cuadrícula tiene una dimensión de $S \times S$. Un cuadro delimitador es un contorno que resalta un objeto en una imagen. YOLO utiliza una regresión de un solo cuadro delimitador para predecir la altura, el ancho, el centro y la clase de objetos. Por último representa la probabilidad de que un objeto aparezca en el cuadro delimitador [Redmon y col. 2015].
- **SSD (2016):** Las tareas de localización y clasificación de objetos se realizan en un solo paso hacia adelante de la red usando regresión del cuadro delimitador. La red es un detector de objetos que también clasifica esos objetos detectados. La arquitectura de SSD se basa en la arquitectura VGG-16, pero descarta las capas completamente conectadas. La razón por la que se utilizó VGG-16 como red base de la arquitectura SSD fue su gran rendimiento en tareas de clasificación de imágenes de alta calidad y su popularidad para problemas en los que el aprendizaje por transferencia ayuda a mejorar los resultados [Liu y col. 2015].

4.4.3. Métodos libres de ancla

Recientemente los investigadores han tratado de remover las anclas de los métodos de detección de objetos obteniendo más flexibilidad para apalancar datos de gran escala. Las anclas se pueden remover gracias al progreso en detección profunda de puntos clave y la formulación de pérdida de detección, como la pérdida focal. A diferencia de los métodos anteriores, se parece más a la segmentación semántica. Se recomienda su uso para hacer inferencia rápida y precisa, sin embargo, los métodos de dos etapas suelen ser más precisos.

- **CornerNet (2018):** CornerNet es un modelo de detección de objetos que detecta un cuadro delimitador de objetos como un par de puntos clave, la esquina superior izquierda y la esquina

inferior derecha, utilizando una red neuronal de convolución única. Al detectar objetos como puntos clave emparejados, se elimina la necesidad de diseñar un conjunto de cajas de anclaje comúnmente utilizadas en detectores de una sola etapa anteriores. También utiliza la agrupación de esquinas, un nuevo tipo de capa de agrupación que ayuda a la red a localizar mejor las esquinas [Law y Deng 2018].

- **CenterNet (2019):** Los enfoques basados en puntos clave a menudo tienen una gran cantidad de cuadros delimitadores de objetos incorrectos, posiblemente debido a la falta de una mirada adicional en las regiones recortadas. Centernet presenta una solución que explora los patrones visuales dentro de cada región recortada con costo computacional mínimo. Construye un marco sobre un detector CornerNet y detecta cada objeto como un triplete de puntos clave, en lugar de un par, lo que mejora tanto la precisión como la recuperación. Usa dos módulos denominados agrupación de esquinas en cascada y agrupación central, que enriquecen la información recopilada por las esquinas superior izquierda e inferior derecha y proporcionan información más reconocible en las regiones centrales, respectivamente [Duan y col. 2019].

4.4.4. EfficientDet (2020)

EfficientDet es un modelo de detección de objetos creado por el equipo de Google Brain, publicado el 27 de julio de 2020. Supera a arquitecturas como YOLOv3 en tareas de detección para conjuntos de imágenes individualizados en aspectos como tiempo de entrenamiento del modelo, tamaño del modelo, velocidad de inferencia y precisión. La arquitectura de EfficientDet usa redes EfficientNet previamente entrenadas con ImageNet como columna vertebral de la arquitectura, redes de características bidireccionales ponderadas (BiFPN) para facilitar y agilizar el manejo de características multiescala, y redes compartidas para la predicción de clases y marcos delimitadores de los objetos (Figura 18). Tanto las capas BiFPN como las capas de las redes de clases y marcos se repiten varias veces a través de la arquitectura con base en diferentes restricciones de recursos como el tamaño de las imágenes de entrada o la arquitectura de EfficientNet utilizada [Tan, Pang y Le 2019].

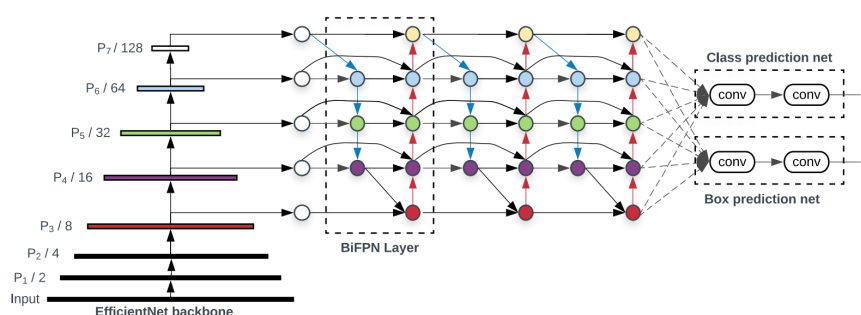


Figura 18: EfficientDet Architecture [Tan, Pang y Le 2019]

4.4.5. COCO (2017)

La base de datos de imágenes de objetos comunes en contexto (Microsoft COCO) [Lin y col. 2014] es un conjunto de datos de fotos de 91 tipos. Son en total 2,5 millones de instancias de objetos etiquetadas en más de 328000 fotos. COCO se ha convertido en uno de los estándares para verificar y comparar el rendimiento de los modelos de detección de objetos.

Una métrica fundamental para evaluar la precisión de los detectores de objetos en un conjunto de imágenes es la relación del área de intersección sobre el área de unión (IoU) de los marcos delimitadores verdaderos y pronosticados de los objetos detectados.

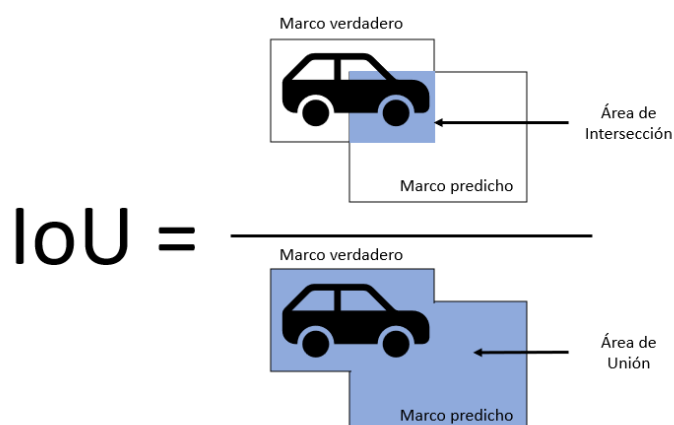


Figura 19: Intersección sobre Unión. Fuente: Elaboración propia

Los marcos delimitadores predichos de los objetos detectados por el modelo se superponen sobre los marcos delimitadores verdaderos y se calcula el cociente del área de intersección entre los dos marcos sobre el área de unión de los mismos.

El IoU se usa para determinar si la caja delimitadora de un objeto es un Verdadero Positivo, Falso Positivo o un Falso Negativo.

- Un Verdadero Positivo tiene un $IoU > 0.5$
- Un Falso Positivo tiene un $IoU < 0.5$ o una caja delimitadora duplicada
- Un Falso Negativo ocurre cuando no hay detección o cuando la caja delimitadora tiene un $IoU > 0.5$ pero la categoría de la imagen detectada es incorrecta

A partir del número de Verdaderos Positivos, Falsos Positivos y Falsos Negativos en las predicciones del modelo, se establecen su precisión (Precision) y su recuperación (Recall).

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} = \frac{TruePositives}{AllDetections}$$

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} = \frac{TruePositives}{AllGroundTruths}$$

Con base en las métricas fundamentales anteriormente mencionadas, se fijan las 12 métricas que se utilizan para caracterizar el rendimiento de un detector de objetos en COCO que se muestran a continuación.

- **Precisión Promedio (Average Precision - AP):**
 - AP:** Precisión promedio con IoU desde 0.50 hasta 0.95 con incrementos de 0.05
 - AP50:** Precisión promedio con $IoU = 0.50$
 - AP75:** Precisión promedio con $IoU = 0.75$
- **Recuperación promedio en diferentes tamaños de objetos (AP Across Scales):**
 - APs:** Precisión promedio de los objetos pequeños: $\text{área} < 32^2$
 - APm:** Precisión promedio de los objetos medianos: $32^2 < \text{área} < 96^2$
 - APl:** Precisión promedio de los objetos grandes: $\text{área} > 96^2$

- **Recuperación promedio (Average Recall - AR):**
 - ARmax1:** Recuperación promedio dada 1 detección por imagen
 - ARmax10:** Recuperación promedio dadas 10 detecciones por imagen
 - ARmax100:** Recuperación promedio dadas 100 detecciones por imagen
- **Recuperación promedio en diferentes tamaños de objetos (AR Across Scales):**
 - ARs:** Recuperación promedio para objetos pequeños: $\text{área} < 32^2$
 - ARm:** Recuperación promedio para objetos medianos: $32^2 < \text{área} < 96^2$
 - ARl:** Recuperación promedio para objetos grandes: $\text{área} > 96^2$
- **Media de la precisión promedio del modelo (Mean Average Precision - mAP):** Es la métrica más importante del reto COCO, primero se calcula el AP para el umbral de IoU de 0,5 de cada categoría de objetos, es decir, calculamos la precisión en cada valor de recuperación (0 a 1 con un tamaño de paso de 0.01), luego se repite para los umbrales de IoU desde 0.55 hasta 0.95 con incrementos de 0.05 y finalmente se toma el promedio de las diferentes categorías de objetos y los diferentes umbrales de IoU como se muestra en la Figura 20.

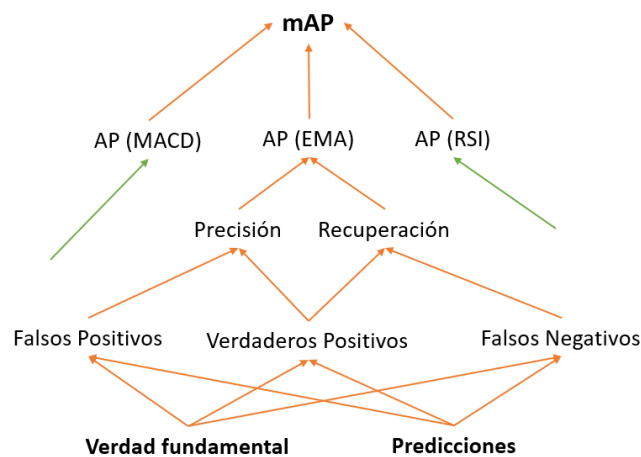


Figura 20: Pasos para calcular el mAP del modelo. Fuente: Elaboración propia

5. Metodología

De acuerdo con la literatura estudiada, consideramos que el avance en la precisión de los modelos de detección de objetos en imágenes de los últimos años, sumada a la posibilidad de implementar estrategias de compraventa de activos mediante la inspección visual de la representación gráfica de su comportamiento en el tiempo incluyendo indicadores de análisis técnico de los mercados, permitiría automatizar la detección de eventos de las estrategias mediante el uso de una red neuronal previamente entrenada con imágenes etiquetadas por un analista financiero experto.

En la metodología que se utilizó para el estudio se recolectaron los datos de precio y volumen de varios activos del mercado de las criptomonedas mediante la consulta de información en Binance. Se implementaron los cálculos de algunos de los indicadores más populares para realizar análisis técnico. A partir de la información básica del activo y sus indicadores, se generaron las imágenes de los momentos del mercado. Estas imágenes fueron etiquetadas con los objetos de cada categoría a identificar localizados mediante coordenadas en un archivo de texto plano de valores separados por coma. Por último se entrenó la red

neuronal con la base de datos de imágenes obtenida separada en conjuntos de entrenamiento, pruebas y validación.

Se pretendía construir un modelo para detectar eventos independientemente de la frecuencia de observación de los datos en la serie de tiempo y del activo. Es importante resaltar que los datos no presentan brechas en el tiempo debido a que el mercado opera de manera continua las 24 horas del día, los 7 días de la semana. En el análisis técnico en general, la identificación de los momentos relevantes para la toma de decisiones de compra y venta de criptomonedas son independientes del plazo elegido, situación que precisamente da origen a los diferentes tipos de estrategias de compra y venta como Buy and hold, swing, day e intraday. Si bien se hubiera podido realizar el mismo análisis con periodos de horas o días, la elección de periodicidad de un minuto y de estrategias de tipo intraday se debe a la facilidad para recolectar la información y a la minimización del riesgo.

En la búsqueda de un modelo que detectara eventos independientemente del activo, se estableció que los eventos de las diferentes monedas son muy similares (por ejemplo un cruce de medias móviles), y que simplemente cambia su escala de acuerdo a su valor. De hecho, popularmente se habla de que el comportamiento de las monedas alternativas (monedas diferentes al Bitcoin) está relacionado de manera estrecha con el comportamiento del Bitcoin. A pesar de lo anterior, evidentemente hay diferencias, especialmente en la volatilidad y la escala que hacen atractivas a unas monedas más que a otras en determinados momentos, pero el objetivo no es comparar los comportamientos de las monedas sino construir un modelo que ayude a detectar eventos de una estrategia de compra y venta indistintamente de la moneda en cuestión.

Consideramos que un modelo de detección de objetos sirve como herramienta de apoyo a un analista financiero para identificar en tiempo real los eventos que componen una estrategia particular de compra y venta de activos. Aunque la comparación de estrategias es un tema interesante, este análisis tampoco hace parte del alcance.

El uso práctico de la red debe contemplar dos aspectos: la periodicidad de la serie de tiempo y la versión del modelo a usar. Por ejemplo, si una red con un modelo y una versión particular analiza cada imagen en un tiempo de, digamos 90 segundos, pero se intenta detectar eventos en una serie de tiempo con periodicidad de 1 minuto, que actualiza las imágenes cada 60 segundos, la red acumulará una cola de imágenes con retraso para analizar. Por esta razón es importante elegir una periodicidad que supere ampliamente el tiempo que le toma al modelo analizar cada imagen o por el contrario, elegir un modelo que analice cada imagen en un tiempo inferior al tiempo de actualización del flujo de imágenes de acuerdo a la periodicidad.

Aunque se contempló la posibilidad de crear una arquitectura de red propia para el estudio, hubo dos aspectos que se tuvieron en cuenta para decidir usar una arquitectura existente. En primer lugar, se puede aprovechar el avance y los resultados de los modelos de última generación, producto de años de trabajo y constante evolución de las redes CNN. Por otro lado, se buscaba evaluar la viabilidad de realizar análisis técnico mediante detección de objetos, por lo tanto, era crucial usar un modelo de detección maduro y de funcionamiento comprobado para valorar esa posibilidad.

Al estudiar los diferentes modelos disponibles actualmente, y luego de comparar sus resultados reportados y publicados se decidió utilizar la arquitectura EfficientDet para el modelo de detección, basado en la red neuronal convolucional EfficientNet, la red de menor tamaño y mayor precisión de acuerdo a la comparación de la figura 17.

El modelo original es un modelo pre entrenado que detecta objetos genéricos como personas, animales, plantas o vehículos. Por el contrario, el modelo del estudio se entrenó desde cero con imágenes del comportamiento de un activo y sus correspondientes indicadores de análisis técnico con el objetivo específico de detectar eventos de compra y venta de activos en dichas representaciones gráficas. Aunque existía la posibilidad de usar los coeficientes de la red entrenada previamente y de ajustar únicamente los pesos de la capa de salida, en realidad los pesos de las entradas de todas y cada una de las capas iniciaron con valores arbitrarios y se obtuvieron los valores definitivos al finalizar el entrenamiento completo para lograr una mejor adaptación del modelo a la aplicación en cuestión.

Los detectores EfficientDet-Lite[0-4] son una familia de modelos de detección de objetos derivados de la arquitectura EfficientDet pero diseñados para ser usados en aplicaciones móviles y de internet de las cosas (IoT) de modo que se obtenga un rendimiento adecuado en dispositivos con capacidad de cómputo restringida como CPUs, GPUs y TPUs móviles. Como varios de estos dispositivos soportan de forma limitada los datos de punto flotante, generalmente se utiliza un proceso de cuantificación.

Durante el proceso de cuantificación, a las variables del modelo se les atribuye un valor discreto que ha sido seleccionado por aproximación dentro de un margen de niveles fijado previamente.

Usando variables de tipo entero en lugar de variables de punto flotante en la cuantificación post entrenamiento, es cierto que el modelo pierde un poco de precisión, pero a la vez su tamaño en bytes disminuye y su velocidad de la inferencia mejora.

Aunque la precisión del modelo es importante, la velocidad de inferencia es fundamental en este estudio porque se debe asegurar que el modelo pueda analizar las imágenes en tiempo real en series de tiempo con periodicidad de un minuto.

Tabla 1: Desempeño de cada modelo EfficientDet-Lite. [Tensorflow.org 2022]

Arquitectura del modelo	Tamaño(MB)*	Latencia(ms)**	Precisión promedio***
EfficientDet-Lite0	4.4	37	25.69 %
EfficientDet-Lite1	5.8	49	30.55 %
EfficientDet-Lite2	7.2	69	33.97 %
EfficientDet-Lite3	11.4	116	37.70 %
EfficientDet-Lite4	19.9	260	41.96 %

* Tamaño del modelo cuantificado entero

** Latencia medida en un Pixel 4 usando 4 hilos en CPU.

*** La precisión promedio es la mAP (mean Average Precision) en el conjunto de validación COCO 2017

Se usaron únicamente las versiones Lite0 y Lite4 por restricciones en el tiempo de entrenamiento de los modelos, además se presentaron algunas dificultades técnicas con las versiones más precisas por limitaciones en los recursos computacionales disponibles, así que se decidió comparar únicamente la versión más precisa con la versión más ligera y rápida.

Por último, el modelo fue exportado usando Tensorflow y Keras para realizar pruebas de inferencia con imágenes nunca vistas y se guardó el registro de la configuración de los parámetros al finalizar las sesiones de entrenamiento para permitir evaluar y comparar posteriormente los valores obtenidos o continuar el entrenamiento desde el punto de control obtenido de ser necesario.

6. Pruebas y resultados

6.1. Recolección de datos

Para este estudio se utilizaron conjuntos de datos en orden cronológico de apertura, máximo, mínimo y cierre de la variable precio y el cierre de la variable volumen de las criptomonedas Bitcoin (BTC), Ethereum (ETH), y Ada (ADA) consultados a través del API (Application Programming Interface) de la casa de cambio Binance, líder del mercado de crypto activos.

Mediante una conexión establecida con el API del exchange usando el SDK (Software Development Kit) python-binance, una librería no oficial en el lenguaje de programación Python de la versión 3 del API REST de Binance, se recopiló información con intervalos de 1 minuto de las últimas 250 horas (24 días) de cada una de las monedas para crear conjuntos de 15.000 registros como se muestra en la Figura 21.

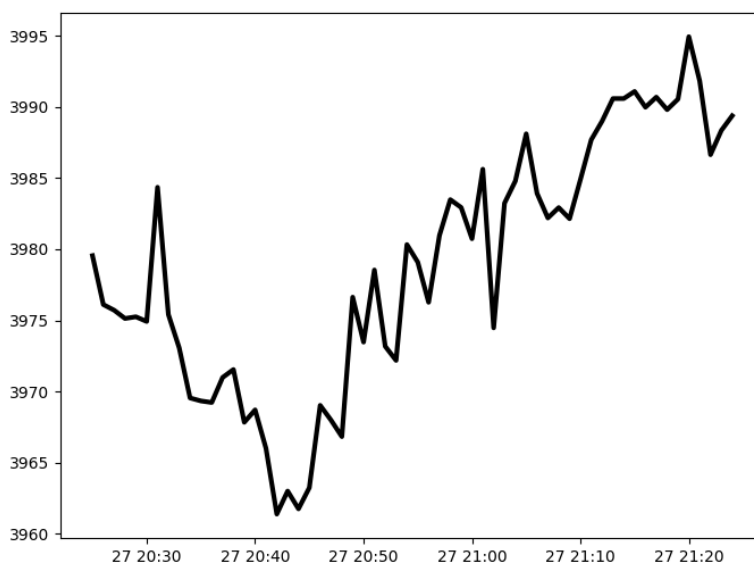


Figura 21: Serie de tiempo con datos del precio del Ethereum. Fuente: elaboración propia

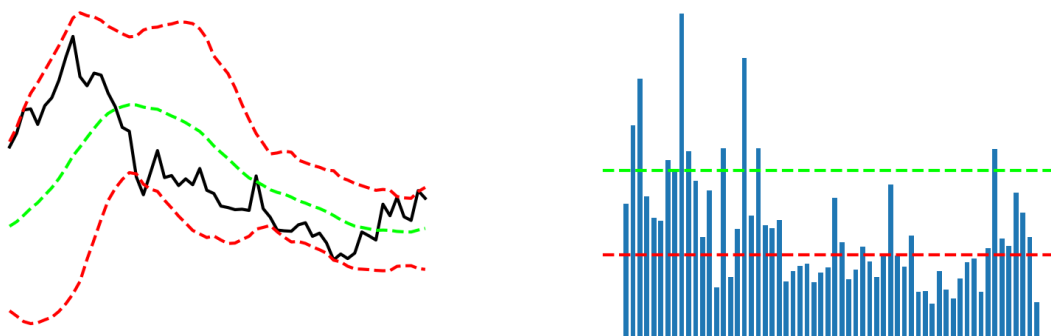
6.2. Preprocesamiento de datos

Las imágenes de los indicadores técnicos se generaron con un programa desarrollado específicamente para este estudio con lenguaje de programación Python. La implementación de los indicadores fue verificada comparando las imágenes generadas con su respectiva visualización en TradingView y Binance como se muestra en la Figura 22. Las imágenes resultantes confirmaron además la calidad de los datos obtenidos a través de la consulta del API de Binance. El código fuente del programa se publicó en un repositorio en GitHub como referencia para su revisión [Caicedo 2022].



Figura 22: Visualización de indicadores en TradingView vs Indicadores generados programáticamente con código en Python. Fuente: Elaboración propia

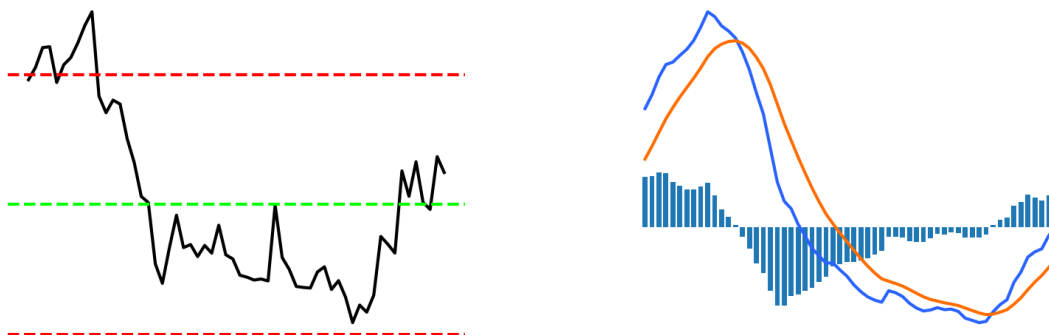
En Python se implementaron los indicadores técnicos de la media móvil simple (SMA), la media móvil exponencial (EMA), el índice de fuerza relativa (RSI), la convergencia divergencia de media móvil (MACD) y las bandas de Bollinger utilizando pandas y numpy para realizar los cálculos y la librería PyPlot para generar su representación gráfica como se muestra en las Figuras 23 y 24.



(a) Bandas de Bollinger

(b) Volumen con desviación estándar

Figura 23: Representación gráfica de indicadores. Fuente: elaboración propia



(a) RSI

(b) MACD

Figura 24: Representación gráfica de indicadores. Fuente: elaboración propia

Con la base de datos de registros de precio y volumen, se procedió a generar imágenes de las variables y los indicadores técnicos en intervalos con corrimiento de una hora para generar ahora una base de datos de 240 imágenes, como la que se muestra en la Figura 25, para la implementación de la estrategia de cruce de medias móviles con confirmación del MACD mediante el etiquetado de los cruces de la EMA y el MACD en la base de datos de imágenes (Figura 26).

Todas las imágenes de la base se generaron sin títulos y sin las etiquetas líneas de referencia o valores del eje y para hacer una abstracción de las condiciones particulares del momento y de cada uno de los cripto activos a analizar. Para efectos prácticos de la implementación de la estrategia EMA+MACD se requiere conocer únicamente la información de los cruces de las medias móviles clasificados en dos categorías: cruce de EMA y cruce de MACD. Se mantuvo la información del eje x únicamente para confirmar que las imágenes contienen intervalos de una hora de información de mercado.

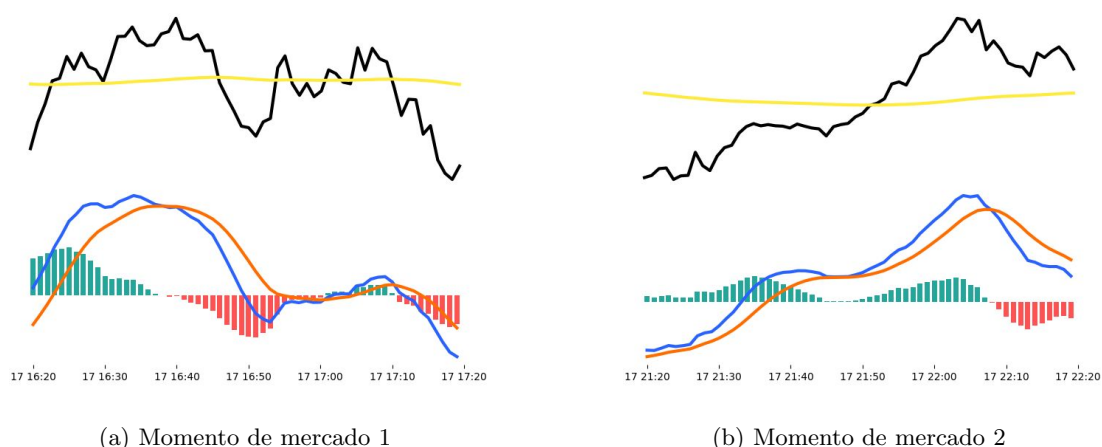


Figura 25: Representación Estrategia EMA+MACD. Fuente: elaboración propia

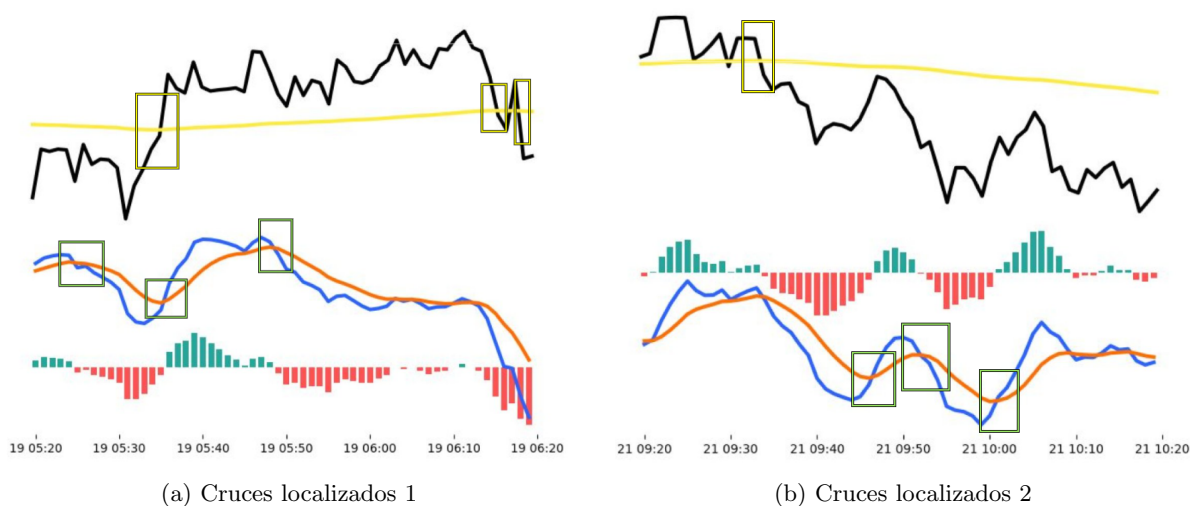


Figura 26: Etiquetado de cruces EMA+MACD. Fuente: elaboración propia

Es importante mencionar que, a la hora de etiquetar los cruces en las imágenes mediante la inspección visual de la persona experta, solamente se tuvieron en cuenta los cruces contundentes que identifican un claro cambio en la tendencia y se ignoraron los cruces falsos en donde las medias móviles no presentan un cambio de tendencia decisivo. Se observa una ventaja en este sentido con respecto al paradigma programático para la identificación de cruces en la estrategia debido a que en el segundo caso es más difícil identificar la contundencia de los cruces.

Las etiquetas se guardan como un archivo CSV en donde cada fila corresponde a un objeto localizado dentro de la imagen que lo contiene, con cada objeto asignado específicamente al conjunto de datos de prueba, entrenamiento o validación.

Cada fila tiene una etiqueta diferente: `CruceEMA` y `CruceMACD`. Los cuadros delimitadores se especifican para cada imagen utilizando los vértices superior izquierdo e inferior derecho (Listing 1).

Listing 1: CSV de objetos localizados

TRAIN, Tesis05/244.jpg, CruceEMA, 0.34104627, 0.2680965, 0.38832998, 0.2680965, 0.38832998, 0.35120642, 0.34104627, 0.35120642
TRAIN, Tesis05/244.jpg, CruceMACD, 0.34607646, 0.5375335, 0.4024145, 0.5375335, 0.4024145, 0.63672924, 0.34607646, 0.63672924
VALIDATION, Tesis05/125.jpg, CruceMACD, 0.38933602, 0.51742625, 0.4748491, 0.51742625, 0.4748491, 0.63538873, 0.38933602, 0.63538873
VALIDATION, Tesis05/119.jpg, CruceEMA, 0.28772637, 0.25201073, 0.3279678, 0.25201073, 0.3279678, 0.32707775, 0.28772637, 0.32707775
TEST, Tesis05/212.jpg, CruceMACD, 0.68712276, 0.78686327, 0.7635815, 0.78686327, 0.7635815, 0.89008045, 0.68712276, 0.89008045
TEST, Tesis05/81.jpg, CruceEMA, 0.3993964, 0.19034852, 0.45171025, 0.19034852, 0.45171025, 0.2761394, 0.3993964, 0.2761394

6.3. Modelo propuesto

Se supone que un modelo de detección de objetos permitirá identificar de manera automática y en tiempo real los cruces de las diferentes categorías que conforman una estrategia de compraventa en una imagen, como por ejemplo el cruce del precio con una EMA o cruce del MACD con la línea de la señal.

Mediante la localización de los objetos en la representación gráfica de los momentos de mercado, se pueden reconocer los eventos que ayuden a tomar decisiones de compra y venta de los diferentes crypto activos del mercado de criptomonedas.

El modelo debe ser entrenado para identificar los objetos que constituyen eventos relevantes para la toma de decisión de acuerdo con la estrategia a implementar, para este caso, la estrategia de medias móviles con confirmación de MACD.

Se identificaron dos clases de objetos: cruce del precio con una EMA de 20 periodos y el cruce del MACD con la señal como se observa en la figura 26.

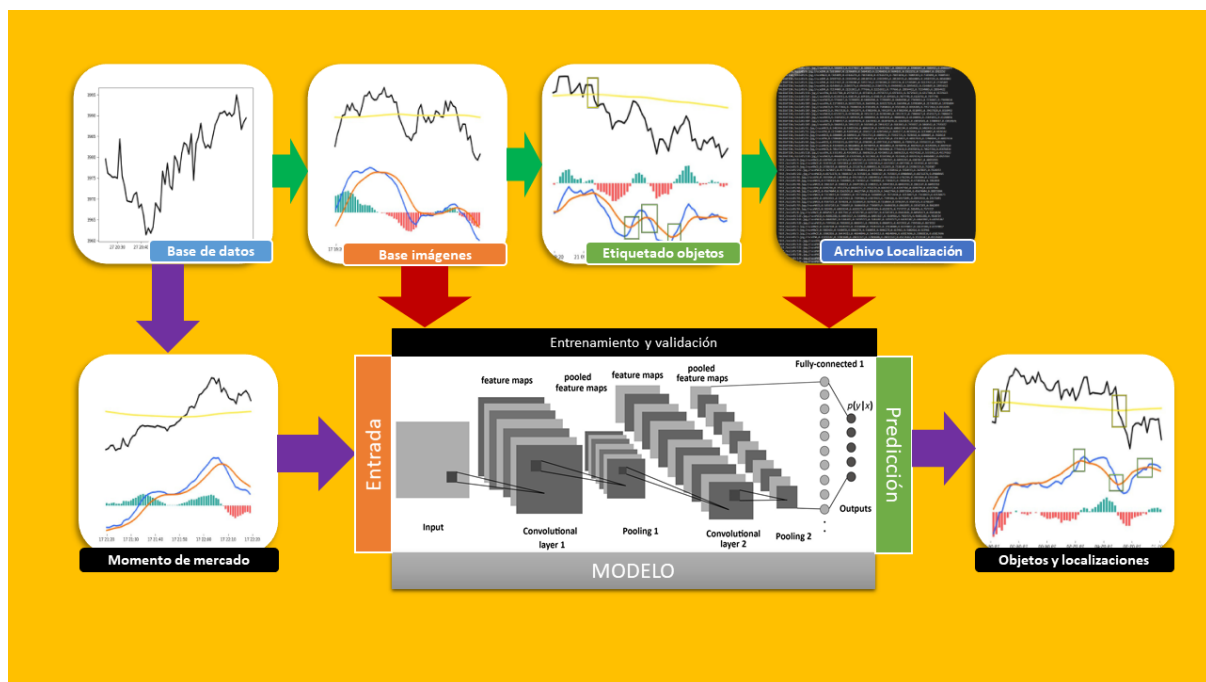


Figura 27: Diagrama del modelo. Fuente: elaboración propia

6.4. Entrenamiento

Para el entrenamiento del modelo se utilizó una base de datos de 240 imágenes que representan momentos de mercado con localización de dos categorías de eventos en indicadores técnicos: Cruce de precio con una

media móvil exponencial de 20 periodos y cruce de señal con el MACD. Los objetos de las dos categorías en las imágenes se distribuyeron como se observa en la tabla 2.

Tabla 2: Distribución de objetos por conjunto

	TRAIN	VALIDATION	TEST
Cruce MACD	493	45	68
Cruce EMA	242	22	20

Se realizaron experimentos entrenando un modelo EfficientDet-Lite0 y un modelo EfficientDet-Lite4 con TensorFlow y Keras en un notebook de Google Colaboratory conectado a un ambiente hospedado y a un ambiente local con las características de la tabla 3.

Tabla 3: Características de los ambientes para los experimentos

	Ambiente local	Ambiente hospedado
Referencia	Lenovo Legion 5	n1-highmem-2 instance
CPU	AMD Ryzen 7 4800H 2.90 GHz	2vCPU @ 2.2GHz
RAM	15.4 GB	13 GB
GPU	NVIDIA GTX 1660Ti 6GB GPU	NA
DISCO	150GB Free Space	100GB Free Space

Durante los entrenamientos se usaron 50 épocas, es decir que se revisa el conjunto de datos de entrenamiento 50 veces mientras se ajustan las variables del modelo. Se monitorea la precisión de la validación durante el entrenamiento para evitar el sobreajuste. Estableciendo el tamaño del conjunto de imágenes simultáneas en 8 observamos que se necesitan 21 pasos para revisar las 175 imágenes en el conjunto de datos de entrenamiento.

Aunque existía la posibilidad de entrenar únicamente la capa de salida, se decidió entrenar todo el modelo para mejorar la precisión. Sin embargo, hay que tener en cuenta que entrenar el modelo completo tomó más tiempo que entrenar solamente la salida.

Tabla 4: Tiempos de entrenamiento promedio

	Ambiente local	Ambiente hospedado
EfficientDet-Lite0	374.977 s	4746.227 s
EfficientDet-Lite4	3212.491 s	NA

6.5. Evaluación

Aunque en las métricas de COCO se hace énfasis en el indicador AP, para efectos prácticos, el umbral de $IoU > 50$ del AP50 es representativo, suficiente y adecuado para evaluar la precisión de los modelos en este estudio. En los dos casos se obtuvieron indicadores AP50 de notable desempeño: 76.69 % para el modelo EfficientDet-Lite0 y 87.26 % para el modelo EfficientDet-Lite4.

Es importante resaltar que el AP del modelo EfficientDet-Lite0 entrenado tiene una precisión promedio superior a la del modelo original de la tabla de desempeño (32.7 % > 25.69 %) y que, por el contrario, el AP del modelo EfficientDet-Lite4 entrenado bajó su precisión con respecto al modelo original de la tabla de desempeño (35.82 % < 41.96 %).

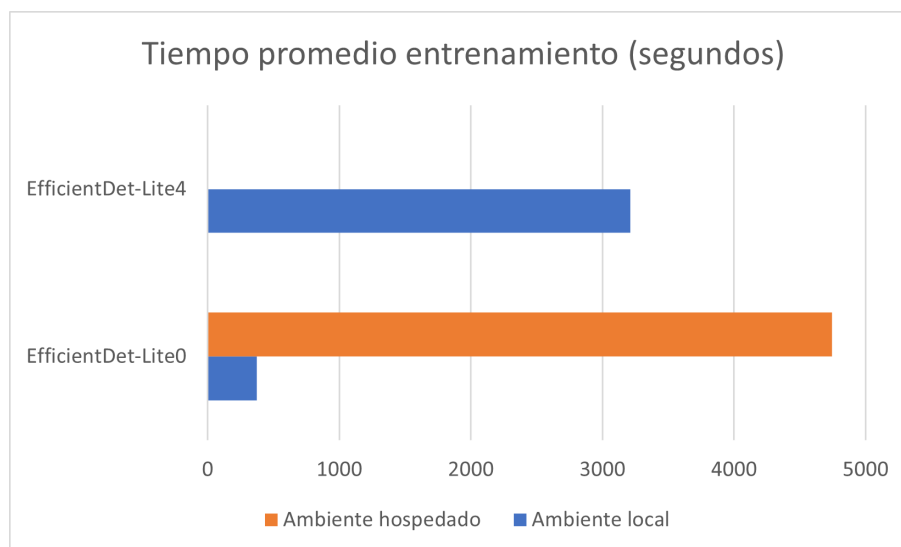


Figura 28: Comparación Tiempos entrenamiento. Fuente: elaboración propia

Tabla 5: Resultados de evaluación de los modelos con métricas de COCO

	EfficientDet-Lite0	EfficientDet-Lite4
AP	0.3270212	0.3582548
AP50	0.7669112	0.8726974
AP75	0.1366837	0.1591944
AP/CruceEMA	0.2835029	0.34331086
AP/CruceMACD	0.37053955	0.37319875
APl	-1.0	-1.0
APm	0.34065965	0.36039445
APs	0.0125	0.25
ARl	-1.0	-1.0
ARm	0.48829538	0.5410055
ARmax1	0.17985295	0.19117647
ARmax10	0.4427941	0.5032353
ARmax100	0.48294118	0.5448529
ARs	0.2	0.55

A pesar de que la AP del modelo EfficientDet-Lite0 es menor que la del modelo EfficientDet-Lite4 tanto en el AP ($32.7\% < 35.82\%$) como en el AP50 ($76.69\% < 87.26\%$), para la predicción en tiempo real es fundamental tener en cuenta los tiempos de detección del modelo de acuerdo con la periodicidad que se va a analizar. Mientras que el modelo EfficientDet-Lite4 que tiene mayor precisión AP hace la detección en 79.86 segundos en promedio en la máquina local, el modelo EfficientDet-Lite0 la hace en alrededor de 3.63 segundos. Esta duración en la detección evita que el modelo EfficientDet-Lite4 se pueda usar para análisis técnico en periodicidad de un minuto. El modelo EfficientDet-Lite0 puede trabajar en cualquier periodicidad (minuto, hora, día) con una pérdida de menos de un 9% en la precisión con respecto al modelo EfficientDet-Lite4 pero haciendo la detección hasta un 22 veces más rápido.

7. Conclusión

La implementación de estrategias de compraventa de activos de mercado se ha orientado hacia paradigmas programáticos y algorítmicos, al punto de que se han creado lenguajes de programación específicos para tal fin, como por ejemplo PineScript de TradingView, un lenguaje de script liviano enfocado a la tarea específica de desarrollar indicadores y estrategias que se utilizan para análisis técnico de los activos en diferentes mercados.

Aquí se plantea una alternativa más amigable e intuitiva orientada a usuarios sin conocimientos técnicos de programación para que a través de la localización, etiquetado y posterior identificación automática de objetos en imágenes que representan diferentes momentos del mercado y con el entrenamiento y uso de un modelo que detecte y clasifique eventos relevantes en esas imágenes se podrían programar notificaciones que permitan realizar compras o ventas de los activos que el usuario desee monitorear.

Mediante la abstracción y generalización de la representación de los momentos de mercado como imágenes, la detección de los eventos del modelo puede ser aplicada a diferentes activos de mercado (BTC, ETH, ADA... e incluso otro tipo de activos de mercado como acciones de bolsa), en diferentes periodicidades (minutos, horas, días) y en distintos instantes de tiempo.

La pérdida de un evento determinado no resulta crítico teniendo en cuenta que se presentan múltiples eventos en distintos momentos y que el objetivo del inversor no es necesariamente entrar todas y cada una de las posibles operaciones sino por el contrario, lograr la rentabilidad que se ha trazado como objetivo con la menor cantidad de transacciones posibles.

Si bien el modelo obtenido cuenta con una precisión AP50 competente que permite identificar eventos y generar señales de compraventa en tiempo real para cruces de SMA y MACD, en estudios futuros se podría comparar el desempeño del modelo de detección de objetos frente a la perspectiva algorítmica, así como también comparar el desempeño de diferentes estrategias entre sí usando este mismo enfoque.

Referencias

- Aggarwal, Charu C. (2018). *Neural Networks and Deep Learning. A Textbook*. Cham: Springer, pág. 497. ISBN: 978-3-319-94462-3. DOI: 10.1007/978-3-319-94463-0.
- Azhikodan, Akhil, Anvitha Bhat y Mamatha Jadhav (mayo de 2019). «Stock Trading Bot Using Deep Reinforcement Learning». En: *Lecture Notes in Networks and Systems*, págs. 41-49. DOI: https://doi.org/10.1007/978-981-10-8201-6_5.
- Birogul, Serdar, Günay Temür y Utku Köse (mayo de 2020). «YOLO Object Recognition Algorithm and "Buy-Sell Decision" Model over 2D Candlestick Charts». En: *IEEE Access PP*, págs. 1-1. DOI: 10.1109/ACCESS.2020.2994282.
- Brock, William, Josef Lakonishok y Blake LeBaron (1992). «Simple Technical Trading Rules and the Stochastic Properties of Stock Returns». En: *The Journal of Finance* 47.5, págs. 1731-1764. ISSN: 00221082, 15406261. URL: <http://www.jstor.org/stable/2328994>.
- Caicedo, Andres (2022). *Python Technical Analysis Indicators*. [Online; accessed 28-April-2022]. URL: <https://github.com/a3caicedo/PythonTechnicalAnalysisIndicators>.
- Chan, Louis K., Narasimhan Jegadeesh y Josef Lakonishok (1996). «Momentum Strategies». En: *The Journal of Finance* 51.5, págs. 1681-1713. DOI: <https://doi.org/10.1111/j.1540-6261.1996.tb05222.x>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1996.tb05222.x>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1996.tb05222.x>.
- Cohen, Naftali, Tucker Balch y Manuela Veloso (jul. de 2019). «Trading via Image Classification». En: *CoRR* abs/1907.10046. arXiv: 1907.10046. URL: <http://arxiv.org/abs/1907.10046>.
- Duan, Kaiwen y col. (2019). «CenterNet: Keypoint Triplets for Object Detection». En: *CoRR* abs/1904.08189. arXiv: 1904.08189. URL: <http://arxiv.org/abs/1904.08189>.

- Ertel, Wolfgang (2017). *Introduction to Artificial Intelligence*. 2.^a ed. Springer. ISBN: 978-3-319-58486-7.
- Fawaz, Hassan Ismail y col. (jul. de 2019). «Deep learning for time series classification: a review». En: *Data Mining and Knowledge Discovery* 33. DOI: 10.1007/s10618-019-00619-1.
- Girshick, Ross (abr. de 2015). «Fast r-cnn». En: DOI: 10.1109/ICCV.2015.169.
- Girshick, Ross B. y col. (2013). «Rich feature hierarchies for accurate object detection and semantic segmentation». En: *CoRR* abs/1311.2524. arXiv: 1311.2524. URL: <http://arxiv.org/abs/1311.2524>.
- He, Kaiming y col. (2016). «Deep Residual Learning for Image Recognition». En: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, págs. 770-778. DOI: 10.1109/CVPR.2016.90.
- Jegadeesh, Narasimhan y Sheridan Titman (2001). «Profitability of Momentum Strategies: An Evaluation of Alternative Explanations». En: *The Journal of Finance* 56.2, págs. 699-720. ISSN: 00221082, 15406261. URL: <http://www.jstor.org/stable/222579>.
- Karim, Fazle y col. (2019). «Multivariate LSTM-FCNs for time series classification». En: *Neural Networks* 116, págs. 237-245. ISSN: 0893-6080. DOI: <https://doi.org/10.1016/j.neunet.2019.04.014>. URL: <https://www.sciencedirect.com/science/article/pii/S0893608019301200>.
- Krizhevsky, Alex, Ilya Sutskever y Geoffrey E. Hinton (2012). «ImageNet Classification with Deep Convolutional Neural Networks». En: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., págs. 1097-1105.
- Lai, Guokun y col. (2017). «Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks». En: *CoRR* abs/1703.07015. arXiv: 1703.07015. URL: <http://arxiv.org/abs/1703.07015>.
- Law, Hei y Jia Deng (2018). «CornerNet: Detecting Objects as Paired Keypoints». En: *CoRR* abs/1808.01244. arXiv: 1808.01244. URL: <http://arxiv.org/abs/1808.01244>.
- Lecun, Y. y col. (1998). «Gradient-based learning applied to document recognition». En: *Proceedings of the IEEE* 86.11, págs. 2278-2324. DOI: 10.1109/5.726791.
- Lin, Tsung-Yi y col. (2014). «Microsoft COCO: Common Objects in Context». En: *CoRR* abs/1405.0312. arXiv: 1405.0312. URL: <http://arxiv.org/abs/1405.0312>.
- Liu, Wei y col. (2015). «SSD: Single Shot MultiBox Detector». En: *CoRR* abs/1512.02325. arXiv: 1512.02325. URL: <http://arxiv.org/abs/1512.02325>.
- Moskowitz, Tobias J., Yao Hua Ooi y Lasse Heje Pedersen (2012). «Time series momentum». En: *Journal of Financial Economics* 104.2. Special Issue on Investor Sentiment, págs. 228-250. ISSN: 0304-405X. DOI: <https://doi.org/10.1016/j.jfineco.2011.11.003>. URL: <https://www.sciencedirect.com/science/article/pii/S0304405X11002613>.
- Redmon, Joseph y col. (2015). «You Only Look Once: Unified, Real-Time Object Detection». En: *CoRR* abs/1506.02640. arXiv: 1506.02640. URL: <http://arxiv.org/abs/1506.02640>.
- Ren, Shaoqing y col. (2015). «Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks». En: *CoRR* abs/1506.01497. arXiv: 1506.01497. URL: <http://arxiv.org/abs/1506.01497>.
- Sim, Hyun, Hae Kim y Jae Ahn (feb. de 2019). «Is Deep Learning for Image Recognition Applicable to Stock Market Prediction?» En: *Complexity* 2019, págs. 1-10. DOI: 10.1155/2019/4324878.
- Simonyan, Karen y Andrew Zisserman (2015). «Very Deep Convolutional Networks for Large-Scale Image Recognition». En: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. por Yoshua Bengio y Yann LeCun. URL: <http://arxiv.org/abs/1409.1556>.
- Szegedy, Christian y col. (2015). «Going deeper with convolutions». En: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, págs. 1-9. DOI: 10.1109/CVPR.2015.7298594.
- Tan, Mingxing y Quoc V. Le (2019). «EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks». En: *CoRR* abs/1905.11946. arXiv: 1905.11946. URL: <http://arxiv.org/abs/1905.11946>.
- Tan, Mingxing, Ruoming Pang y Quoc V. Le (2019). «EfficientDet: Scalable and Efficient Object Detection». En: *CoRR* abs/1911.09070. arXiv: 1911.09070. URL: <http://arxiv.org/abs/1911.09070>.
- Tensorflow.org (2022). *Object Detection with TensorFlow Lite Model Maker*. [Online; accessed 07-March-2022]. URL: https://www.tensorflow.org/lite/tutorials/model_maker_object_detection.

- Velay, Marc y Fabrice Daniel (2018). «Stock Chart Pattern recognition with Deep Learning». En: *CoRR* abs/1808.00418. arXiv: 1808.00418. URL: <http://arxiv.org/abs/1808.00418>.
- Wang, Zhiguang, Weizhong Yan y Tim Oates (2016). «Time Series Classification from Scratch with Deep Neural Networks: A Strong Baseline». En: *CoRR* abs/1611.06455. arXiv: 1611.06455. URL: <http://arxiv.org/abs/1611.06455>.
- Wu, Yangru, Ronald Balvers y Erik Gilliland (feb. de 2000). «Mean Reversion across National Stock Markets and Parametric Contrarian Investment Strategies». En: *Journal of Finance* 55, págs. 745-772. DOI: 10.1111/0022-1082.00225.
- Yildirim, Deniz Can, Ismail Hakkı Toroslu y Ugo Fiore (2021). «Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators». En: *Financial Innovation* 7.
- Zeiler, Matthew D. y Rob Fergus (2014). «Visualizing and Understanding Convolutional Networks». En: *Computer Vision – ECCV 2014*. Ed. por David Fleet y col. Cham: Springer International Publishing, págs. 818-833. ISBN: 978-3-319-10590-1.