

SEGMENTACIÓN Y SEGUIMIENTO DE MUONES UTILIZANDO TÉCNICAS DE DEEP LEARNING PARA EL EXPERIMENTO CONNIE

Karina Aquino Cantero¹, Carlos Silva Gines², Ing. PhD. Diego Stalder³, Ing. PhD. Jorge Molina³, Ing. Gustavo Román³

¹Alumno, edithaquinoc@fiuna.edu.py, Facultad de Ingeniería. Universidad Nacional de Asunción. San Lorenzo, Paraguay

²Alumno, csilva@fiuna.edu.py, Facultad de Ingeniería. Universidad Nacional de Asunción. San Lorenzo, Paraguay

³Orientador, Facultad de Ingeniería. Universidad Nacional de Asunción. San Lorenzo, Paraguay

Palabras clave:

- Muón
- Aprendizaje profundo
- YoloV8
- Yolo
- CONNIE
- Neutrino

Resumen

El experimento CONNIE utilizó 12 sensores CCD para detectar antineutrinos y explorar interacciones de neutrinos cerca del reactor Angra II en Brasil. Los sensores registraron imágenes de partículas, mayormente muones, generando un catálogo extenso. Se propuso la segmentación de instancias y se desarrollaron modelos para clasificar muones y detectar eventos superpuestos. Los resultados mostraron 99 % de exactitud en seguimiento y una precisión media de 48.5 %, con el modelo logrando un 61.1 % de precisión en la detección de objetos. Estos hallazgos resaltan el potencial de los modelos y su relevancia en entender las variaciones en la tasa de muones en CONNIE.

Keywords

- Muon
- Deep Learning
- YOLO
- YoloV8
- CONNIE
- Neutrine

Abstract

CONNIE experiment used 12 CCD sensors near Angra II reactor, in Brasil, aimed to detect low-energy antineutrinos and study non-standard neutrino interactions. Sensors captured particle images, mainly muons, forming a catalog of events. Proposed instance segmentation continued prior work, developing models for muon classification and tracking, and detecting overlapping events. Evaluation showed 99 % accuracy in tracking, a mean precision of 48.5 % in segmenting, and 61.1 % precision in locating objects, shedding light on muon rate variations in CONNIE's experimental context.

1. INTRODUCCIÓN

El aprendizaje y la ciencia de datos emerge como un pilar fundamental en la exploración de los vastos conjuntos de datos generados en la física de partículas en un análisis complejo para extraer información significativa [1, 2]. Los métodos tradicionales de análisis para tales conjuntos de datos pueden resultar lentos, demandantes en recursos humanos y susceptibles a errores. Por ello, el uso de algoritmos automáticos, como el aprendizaje profundo (Deep Learning) y el aprendizaje automático (Machine Learning), se ha vuelto cada vez más importante en los últimos años. Estos métodos permiten analizar eficientemente grandes conjuntos de datos, identificar patrones y clasificar eventos con precisión [3, 4]. La aplicación de estas técnicas puede potenciar considerablemente la velocidad y precisión de la investigación en física de partículas, facilitando la exploración de preguntas cada vez más complejas sobre la naturaleza del universo.

El experimento CONNIE (Coherent Neutrino-Nucleus Interaction Experiment) fue diseñado para detectar antineutrinos generados en reactores y para probar interacciones no estándar de neutrinos mediante la medición de los retrocesos de baja energía (CEVNS) [5]. En este experimento, se pueden emplear algoritmos automáticos para identificar muones entre otras partículas y estimar la contaminación de fondo del experimento. Este enfoque permite que los investigadores se centren en los fenómenos más significativos y raros, reduciendo significativamente el tiempo necesario para el análisis. Por lo tanto, el uso de algoritmos automáticos, como los utilizados en el experimento CONNIE, resulta crucial para mejorar nuestra comprensión de los bloques fundamentales del universo.

En relación con los muones, estos no son directamente detectados por CONNIE, pero desempeñan un papel crucial en la investigación. Los muones son partí-

culas cargadas que se generan en abundancia a partir de la interacción de neutrinos y antineutrinos con la materia, especialmente en interacciones cósmicas de alta energía. El detector CONNIE, ubicado cerca del reactor nuclear Angra II en Angra dos Reis, Brasil, cuenta con 12 sensores CCD y ha estado registrando imágenes desde agosto de 2016. Aunque el software de extracción de datos desarrollado por la colaboración CONNIE cuenta con sus propios algoritmos de detección de eventos, carece de la capacidad para clasificar eventos individuales y superpuestos con precisión. Es importante señalar que, en física, cuando se habla de evento, se hace referencia a la interacción de una partícula estudiada con el material del detector, que luego, tras el proceso de reconstrucción, creará diferentes trazos en las imágenes. En este contexto este trabajo final de grado propone técnicas para separar eventos superpuestos y realizar el seguimiento de muones, aprovechando técnicas de aprendizaje automático y profundo para mejorar la precisión y eficiencia del proceso.

Revisando la literatura, se evidencia que estudios previos, como el de Winter et al [6], han demostrado formas de clasificar partículas basadas en su morfología. Mientras que [7] logró la clasificación utilizando técnicas de aprendizaje profundo. Por ello, proponemos la segmentación por instancia y clasificación de redes neuronales convolucionales (CNNs) [8], conocidas por su capacidad para capturar dependencias espaciales en imágenes. Específicamente, el modelo CNN propuesto se inspira en VGG16 [9], una arquitectura ampliamente utilizada en tareas de clasificación de imágenes.

Cuando múltiples partículas se superponen en una única imagen, esta no puede ser clasificada correctamente por la CNN. Para abordar este problema, el trabajo utiliza una estrategia de aprendizaje profundo para la segmentación por instancia, que implica identificar objetos individuales dentro de una imagen, utilizando la arquitectura YOLO (You Only Look Once) [10]. Esta arquitectura es conocida por su rapidez en procesamiento y entrenamiento, lo que la hace ideal para analizar grandes cantidades de datos. Al implementar esta técnica, los investigadores lograron identificar y separar eventos superpuestos, mejorando la precisión de sus resultados y permitiéndoles comprender mejor el comportamiento de las partículas en el experimento CONNIE.

En adición, se destaca el seguimiento de muones, que tiene por objetivo estimar el ángulo de incidencia de los muones en el CCD mediante el uso de técnicas de regresión del aprendizaje automático. Esta metodología, basada en la correlación entre las trayectorias detectadas de los muones y los ángulos observados en el CCD, constituye un componente esencial para comprender con precisión el comportamiento de estas partículas en el contexto del experimento lleva-

do a cabo en CONNIE.

1.1. Objetivos Específicos

- Realizar simulaciones con el software Geant para la generación de imágenes en el laboratorio.
- Generar un dataset de datos sintéticos obtenidos mediante las simulaciones.
- Comparar e implementar las arquitecturas que según la literatura han demostrado mejor desempeño.
- Desarrollar e implementar una arquitectura de deep learning capaz de clasificar las diferentes partículas que existen en el experimento.
- Evaluar el desempeño aplicando las métricas adecuadas para la segmentación de instancia.
- Desarrollar e implementar un método de seguimiento de muones en las diferentes capas del detector del experimento CONNIE.
- Disponibilizar el desarrollo y publicar resultados para futuras investigaciones.

1.2. Alcance y Limitaciones

La Facultad de Ingeniería de la Universidad Nacional de Asunción participa en el experimento CONNIE como miembro fundador, busca actualmente detectar los antineutrinos generados en el reactor nuclear de Angra 2 situado en la planta Almirante Álvaro Alberto en Angra dos Reis, Brasil.

Existe una dificultad en cuanto a la detección que son los ruidos de fondo generados por los rayos cósmicos, para evitarlo se construyen blindajes con diferentes materiales como plomo, polietileno y cobre para ayudar a bloquear las distintas partículas del fondo radiactivo terrestre, pero todos éstos no frenan a los muones que atraviesan la Tierra constantemente a un flujo de un muón por minuto por cada centímetro cuadrado.

Los muones dejan trazas en el detector que son de fácil distinción pero que a su vez su participación se complica cuando este también genera otras partículas de menor energía que se confunden con las señales generadas por los neutrinos.

Este trabajo propone eliminar dicho problema en el experimento mencionado con técnicas de deep learning, ya que la cantidad de información que se puede obtener de un detector así son enormes, los cuales nos benefician a la hora de aplicar estas técnicas.

2. MARCO TEÓRICO

2.1. El experimento CONNIE y el muón

El reto experimental en la detección de CEvNS (Dispersión Coherente de Neutrinos Electrónicos) radica

en la baja energía de los neutrinos de reactor (alrededor de 1 MeV), los retrocesos nucleares resultantes de sus interacciones aún más bajos en energía (del orden de keV), y la necesidad de detectar la ionización sub-keV causada por estos retrocesos en el material del detector [11].

El Experimento de Interacción Neutrino-Núcleo Coherente (CONNIE) busca detectar la dispersión coherente de antineutrinos de reactor en núcleos de silicio utilizando CCDs de silicio de alta resistividad totalmente agotados. Para lograr la sensibilidad requerida para medir la ionización sub-keV debida a los retrocesos en los núcleos de silicio, es esencial reducir al máximo el umbral de energía de detección y minimizar el ruido de lectura de los sensores.

Los sensores CCD utilizados en CONNIE fueron desarrollados en colaboración con LBNL Micro Systems Labs y constan de una matriz cuadrada con 16 millones de píxeles de $15 \times 15 \mu\text{m}^2$ cada uno, con un grosor de $657 \mu\text{m}$. Estos CCD se encuentran dentro de una caja de cobre y están instalados en un recipiente de cobre al vacío para reducir interferencias [12].

El detector está rodeado de un blindaje que incluye 15 cm de plomo para absorber fotones y dos capas de 30 cm de polietileno de alta densidad para detener neutrones cosmogénicos. CONNIE está ubicado cerca del reactor nuclear de Angra 2 en Río de Janeiro, Brasil, donde se estima una densidad de flujo de neutrinos de $7,8 \times 10^{12} \bar{\nu} s^{-1} cm^{-2}$ a nivel del mar. El experimento ha estado operando de manera continua desde la instalación de los CCD científicos en 2016, con interrupciones solo para mantenimiento.

Además de los neutrinos, los muones también desempeñan un papel importante en CONNIE. Estas partículas se generan cuando los rayos cósmicos interactúan con la atmósfera terrestre y, en su mayoría, se originan a partir de piones que resultan de estas interacciones. Los muones, partículas más masivas que los electrones, tienen una vida relativamente corta y pierden energía al ionizar átomos durante su trayectoria antes de desintegrarse en electrones y neutrinos.

2.2. Software geant4

En el contexto de la física de partículas y nuclear moderna, la necesidad de desarrollar software robusto y complejo es esencial, especialmente para la simulación precisa de detectores de partículas cada vez más grandes y sofisticados. Geant4 es una herramienta desarrollada para abordar este desafío, permitiendo la simulación detallada de cómo las partículas interactúan con la materia a través de una amplia gama de funcionalidades, como seguimiento, geometría, modelos de física e impactos. Este conjunto de herramientas, creado en colaboración global de físicos e ingenieros de software, se basa en tecnología orientada a objetos y está implementado en C++. Ha

encontrado aplicaciones en campos como la física de partículas, física nuclear, diseño de aceleradores, ingeniería espacial y física médica [13].

Además, mediante la integración de capacidades de inteligencia artificial, como la segmentación de datos, en combinación con herramientas como Geant4, es posible mejorar la precisión y eficiencia en la identificación de eventos y en la calidad de las simulaciones. Esto resulta en un avance significativo en la investigación de física de partículas y en la comprensión de cómo las partículas interactúan con la materia.

2.3. Inteligencia artificial

En términos simples, se puede definir como la capacidad de las máquinas para realizar actividades que normalmente requieren inteligencia humana. La IA implica la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones, tal y como lo haría un ser humano [14].

Durante los últimos años, los enfoques basados en algoritmos de inteligencia artificial han emergido como herramientas sumamente eficaces en la tarea de clasificación de imágenes, especialmente cuando se dispone de amplios conjuntos de datos. Estos métodos buscan automatizar una serie de procesos con el objetivo de minimizar la dependencia de la intervención humana en el proceso [15]. El propósito principal es lograr una clasificación precisa y eficiente de imágenes, aprovechando el poder de los algoritmos para reconocer patrones y características relevantes en las imágenes, lo que a su vez ha demostrado tener un impacto significativo en diversas aplicaciones prácticas.

2.4. Algoritmos de regresión

Los algoritmos de regresión en Machine Learning son métodos estadísticos utilizados para modelar la relación entre una o varias variables independientes (conocidas como variables predictoras o variables explicativas) y una variable dependiente (conocida como variable de respuesta o variable objetivo) continua. El objetivo principal de estos algoritmos es predecir valores numéricos continuos para la variable de respuesta a partir de los valores de las variables predictoras [16].

A continuación, se presentan algunos de los algoritmos de regresión más comunes en Machine Learning:

- Regresión Lineal: es un método de regresión clásico que modela la relación entre una variable dependiente y una o varias variables predictoras a través de una función lineal. Este algoritmo asume que la relación entre las variables es lineal y utiliza técnicas de optimización para encontrar los coeficientes que mejor ajustan los datos.
- Regresión logística: es un método de regresión

utilizado para modelar la relación entre una variable dependiente binaria y una o varias variables predictoras. En lugar de predecir valores continuos, la regresión logística predice la probabilidad de que la variable dependiente sea igual a 1. Este algoritmo utiliza una función logística para modelar la relación entre las variables.

- **Regresión Polinómica:** se utiliza comúnmente en problemas en los que la relación entre las variables predictoras y la variable de respuesta no es lineal, como en el caso de curvas de crecimiento, fenómenos físicos, y en general, cualquier situación donde se espera que la relación entre las variables tenga una forma curva.

2.5. Aprendizaje profundo

El aprendizaje profundo, también conocido como Deep Learning, es una rama del aprendizaje automático (Machine Learning) que se enfoca en desarrollar algoritmos capaces de aprender representaciones cada vez más abstractas y complejas de los datos a través de múltiples capas de procesamiento. Este enfoque ha demostrado ser eficaz en tareas de reconocimiento de patrones como visión por computadora, procesamiento del lenguaje natural, identificación de objetos y detección de fraudes [17]. La principal ventaja del aprendizaje profundo es su capacidad para generar automáticamente características de alto nivel a partir de datos crudos, eliminando la necesidad de definir manualmente estas características [8].

2.6. Redes neuronales artificiales y neuronas biológicas

El sistema nervioso está compuesto por redes de neuronas biológicas interconectadas. Una neurona biológica consta de tres partes principales: el cuerpo celular, las dendritas y el axón. Las dendritas transmiten información al cuerpo celular, que luego decide si enviar un mensaje nervioso a través del axón. Este proceso depende de si la suma de la entrada supera un umbral de excitación.

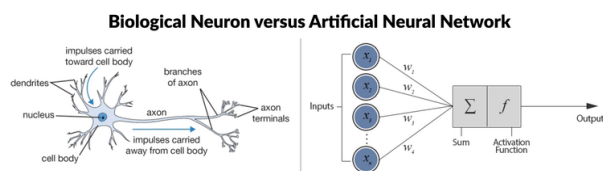


Figura 1: Comparación entre una neurona biológica y una red neuronal artificial **Fuente:** Datacamp

Las redes neuronales artificiales, como los perceptrones, se asemejan a las neuronas biológicas en su funcionamiento, ver Figura 1. En un perceptrón, las entradas se ponderan y se aplica una función de activación

para determinar la salida. Los términos clave son el vector de pesos (ajustado durante el entrenamiento), el vector de entrada y la función de activación. Los perceptrones se utilizan para modelar relaciones en datos, siguiendo principios similares a los de las neuronas biológicas.

2.7. El algoritmo del perceptrón

El algoritmo simple del Perceptrón ya no se utiliza en la práctica, ya que otros algoritmos son mucho más eficaces. Además, las neuronas biológicas no se utilizan individualmente, sino que suelen estar vinculadas a otras neuronas.

El interés por el algoritmo del perceptrón proviene de una técnica demostrada en 1989 por George Cybenko [18] que consiste en enlazar y apilar capas de perceptrón para proporcionar una mayor complejidad. Un algoritmo de este tipo se llama Perceptrón Multicapa, a menudo abreviado como MLP.

2.8. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (Convolutional Neural Networks o CNN) son un tipo de red neuronal que ha demostrado ser muy efectivo en el análisis de imágenes, permitiendo la clasificación, detección de objetos, segmentación y generación de imágenes.

La CNN procesa sus capas imitando al cortex visual del cerebro humano para identificar distintas características en las entradas. Para ello, la CNN contiene varias capas ocultas especializadas y con una jerarquía: esto significa que las primeras capas detectan propiedades o formas básicas y se van especializando hasta llegar a capas más profundas capaces de reconocer formas complejas como un rostro o una silueta.[19]

2.9. Arquitectura de una CNN

La arquitectura de una CNN incluye diferentes tipos de capas, como la capa convolucional, la capa de agrupación y la capa completamente conectada. Estas capas pueden complementarse con capas de normalización y abandono según las necesidades del modelo [20].

La capa convolucional tiene como objetivo detectar características visuales en las imágenes, permitiendo que la red aprenda y reconozca características en diferentes partes de la imagen. Operando sobre un tensor 3D (w, h, d) y una matriz de pesos (m, n) , conecta las neuronas de una capa anterior con las de una capa de convolución utilizando un bloque de pesos de (w, h, d) . Esto reduce la complejidad en comparación con (m, n) debido a las dimensiones más pequeñas de (w, h) [21].

La capa de agrupación, o pooling, disminuye las dimensiones espaciales (ancho y alto) de los datos de entrada, pero no afecta la profundidad del volumen

(número de canales). Esto reduce la cantidad de parámetros y el cálculo en las capas posteriores, además de mitigar el sobreajuste. Utiliza dos hiperparámetros, extensión espacial (F) y Salto (S), para producir un nuevo volumen de tamaño $(W_2 \times H_2 \times D_2)$ [?].

La capa completamente conectada, o FC, se encuentra al final de la arquitectura y se conecta a todas las neuronas de salida. Aplica una combinación lineal y una función de activación para clasificar la entrada de la imagen, devolviendo un vector de tamaño correspondiente al número de clases, donde cada componente representa la probabilidad de pertenecer a una clase [22].

2.10. Arquitectura YOLOv8

La serie de modelos YOLO (You Only Look Once) es conocida por su precisión y tamaño de modelo pequeño. YOLOv8, desarrollado por Ultralytics en enero de 2023, se destaca como uno de los modelos YOLO más avanzados hasta la fecha.

YOLOv8 emplea una arquitectura de red neuronal convolucional profunda, compuesta por capas convolucionales, capas de agrupación y capas completamente conectadas. Utiliza bloques CSP (Cross Stage Partial connections) repetitivos para mejorar la eficiencia y precisión de detección.

Una característica distintiva de YOLOv8 es la detección en múltiples escalas. Divide la imagen en cuadrículas y realiza detecciones a diferentes escalas, lo que permite capturar objetos de varios tamaños en una sola pasada.

Aunque aún no se ha publicado oficialmente ningún artículo sobre YOLOv8, la arquitectura ha sido aprobada por los creadores y se encuentra disponible en el repositorio de Ultralytics [23].

2.11. Segmentación de Instancias en YOLOv8

La segmentación de instancias va un paso más allá de la detección de objetos y consiste en identificar objetos individuales en una imagen y segmentarlos del resto de la imagen. El resultado de un modelo de segmentación de instancias es un conjunto de máscaras o contornos que delimitan cada objeto de la imagen, junto con etiquetas de clase y puntuaciones de confianza para cada objeto. Reentrenamos los cinco modelos preentrenados que ofrece Ultralytics con nuestro conjunto de datos etiquetados.

2.12. Funciones de activación y pérdidas en YOLOv8

Las funciones de activación son un componente esencial en las redes neuronales, ya que permiten que el modelo capture relaciones no lineales entre las variables de entrada y salida. Sin estas funciones, las redes neuronales solo podrían realizar transformaciones lineales, lo que limitaría su capacidad para modelar patrones complejos en los datos. Un ejemplo de

función de activación utilizada en YOLOv8 es la función SiLU [24], que se define como $SiLU(x) = x \cdot \sigma(x)$, donde $\sigma(x)$ es la función sigmoide. Esta función juega un papel importante en el procesamiento de información en esta arquitectura.

En el campo de las redes neuronales, las funciones de pérdida se emplean para medir la discrepancia entre las predicciones del modelo y las salidas reales. El objetivo principal de una red neuronal es minimizar esta discrepancia, también conocida como pérdida o costo. En el caso de YOLOv8, se utilizan diferentes funciones de pérdida para la clasificación y la delimitación de las cajas. Para la pérdida de clasificación, se aplica la pérdida de entropía cruzada binaria (BCE), que mide la discrepancia entre las etiquetas reales y las probabilidades predichas. Para la delimitación de cajas, se emplea la métrica de intersección sobre la unión (IoU) [25] y una función de costo denominada CloU, que tiene en cuenta factores como la distancia euclidiana y la consistencia de la relación de aspecto para mejorar la precisión.

Además, se menciona la pérdida focal de distribución (DFL) [26], una técnica que se utiliza para enfocar rápidamente los valores cercanos a las etiquetas objetivo y mejorar la precisión en el aprendizaje de las cajas delimitadoras.

2.13. Bloques residuales y CSP

Los bloques principales en YOLOv8 son los bloques residuales y los bloques CSP (Cross-Stage Partial connections) [27]. Estos bloques se utilizan en la arquitectura de backbone de YOLOv8 para extraer características de las imágenes de entrada de manera eficiente.

Los bloques residuales utilizados en YOLOv8 son similares a los utilizados en la arquitectura de red neuronal convolucional ResNet. Estos bloques permiten que la red aprenda a saltar capas y agregar información de características de múltiples niveles de profundidad en la red.

Por otro lado, los bloques CSP utilizados en YOLOv8 son una variante de los bloques residuales que tienen conexiones parciales entre capas. Estas conexiones parciales mejoran la eficiencia computacional y reducen el costo computacional de la red.

2.14. Salida

En cuanto a las salidas, YOLOv8 utiliza una serie de mapas de características en diferentes escalas para detectar objetos de diferentes tamaños en la imagen de entrada. Cada mapa de características se divide en una cuadrícula de celdas y cada celda se encarga de detectar objetos dentro de su área.

La predicción de cada celda se compone de varias salidas, incluyendo la confianza de la detección, las coordenadas de la caja delimitadora del objeto y las

probabilidades de las clases del objeto. Estas salidas se utilizan para determinar si un objeto está presente en una celda dada y, de ser así, cuál es el tipo de objeto y dónde se encuentra en la imagen.

3. FLUJO DE PROCESOS DEL TRABAJO

En este estudio, se exploró la generación y procesamiento de datos sintéticos para la predicción de ángulos de incidencia, segmentación de partículas solapadas y seguimiento en el contexto del experimento CONNIE. Se utilizó un conjunto diverso de imágenes sintéticas y del CCD, con enfoque en partículas solapadas y muones. Se aplicaron técnicas de etiquetado y preprocesamiento para mejorar la calidad de los datos. Se desarrollaron modelos de regresión lineal y polinómica para predecir ángulos de incidencia y se implementaron modelos de clasificación utilizando YOLOv8 preentrenado.

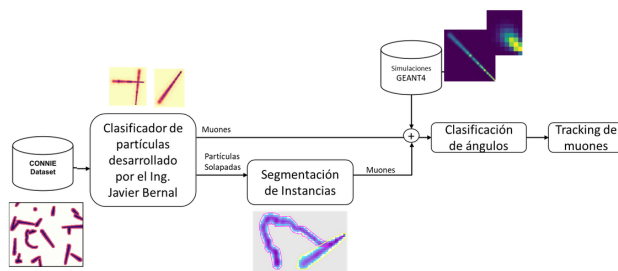


Figura 2: Flujo de procesos del presente trabajo. **Fuente:** Elaboración propia

El flujo de procesos en el presente estudio se presenta gráficamente en la Figura 2. El primer paso implica la utilización del trabajo desarrollado por el Ingeniero Javier Bernal [28], cuyo objetivo principal radica en la clasificación de las partículas detectadas en el CCD. Este proceso de clasificación arroja varias categorías de salida, que son muones, gusanos, partículas solapadas, spots, muones inseguros, fallos de píxeles e indefinidos. Para el propósito de este estudio, nos enfocamos en las partículas clasificadas como muones y partículas solapadas.

Las partículas identificadas como solapadas son sometidas a un proceso de segmentación de instancias, con el fin de separarlas en la categoría de muones. Posteriormente, a cada muón individual se le determina la longitud de su trazo, y se calcula el ángulo de incidencia mediante un enfoque de clasificación basado en regresión lineal. Finalmente, una vez obtenido el ángulo de incidencia de cada muón, se tiene la capacidad de predecir el punto de impacto en cada CCD, lo que permite la construcción de la trayectoria (seguimiento).

4. METODOLOGÍA DEL TRABAJO

4.1. SEGMENTACIÓN DE PARTÍCULAS

Para entrenar la segmentación de instancia se utilizó un conjunto de datos del Ingeniero Javier Bernal [28], que consta de 3171 imágenes de eventos aleatorios en el experimento CONNIE. Estas imágenes se almacenan en formato .fits y se utilizan para clasificar diferentes partículas. El conjunto de datos se preprocesó para equilibrar las clases y se enfocó en las imágenes de partículas solapadas.

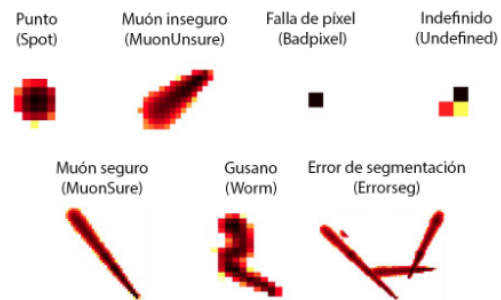


Figura 3: Clases observadas en el dataset. **Fuente:** Ing. Javier Bernal

4.1.1. Segmentación de instancias en YOLOv8

En este trabajo se ha re-entrenado y evaluado el rendimiento de varios modelos de segmentación de YOLOv8, a saber: YOLOv8n-seg, YOLOv8s-seg, YOLOv8m-seg, YOLOv8l-seg y YOLOv8x-seg. Estos modelos fueron pre-entrenados en el conjunto de datos COCO [29]. Cada modelo tiene diferentes tamaños y, como resultado, diferentes velocidades de inferencia y puntuaciones de precisión (mAP).

Tabla 1: Comparación de los distintos modelos YOLOv8 en nuestro conjunto de datos.

Model	Size (px)	mAP box 50-95	mAP mask 50-95	Velocidad CPU ONNX (ms)	Velocidad A100 TensorRT (ms)	Params (M)	Flops (B)
YOLOv8n-seg	640	36.7	30.5	96.1	1.21	3.4	12
YOLOv8s-seg	640	44.6	36.8	155.7	1.47	11.8	42.6
YOLOv8m-seg	640	49.9	40.8	317.0	2.18	27.3	110.2
YOLOv8l-seg	640	52.3	42.6	572.4	2.79	46.0	220.5
YOLOv8x-seg	640	53.4	43.4	712.1	4.02	71.8	344.1

Fuente: Documentación Ultralytics YOLOv8. [30]

Se observó que el tamaño del modelo está directamente relacionado con el mAP (precisiones medias promedio) y, de manera inversa, con el tiempo de inferencia. Los modelos más grandes tienden a tener mAP más altos, lo que indica una mayor precisión en la detección y segmentación de objetos, pero también requieren más tiempo de inferencia. Los modelos más pequeños, en cambio, son más rápidos en términos de tiempo de inferencia, pero pueden tener un mAP

comparativamente menor. La elección del modelo depende de las necesidades específicas del proyecto: los modelos más grandes son adecuados cuando se dispone de más datos, mientras que los modelos más pequeños son más eficientes cuando el espacio es limitado.

En la segmentación de instancias, se busca no solo detectar objetos en una imagen, sino también segmentar cada objeto individualmente, asignando píxeles específicos a cada instancia. El mAP en la segmentación de instancias se utiliza para evaluar la precisión de la detección y la segmentación de objetos en una imagen. Este mAP se calcula considerando la precisión y la exhaustividad de las detecciones y segmentaciones en diferentes umbrales de similitud entre las máscaras predichas y las máscaras reales de las instancias.

4.2. SEGUIMIENTO DE MUONES

El conjunto de datos se generó utilizando la herramienta llamada Geant 4 para simular el paso de partículas a través de la materia, incluyendo una amplia gama de funciones y procesos físicos [31]. Se simuló el paso de partículas a través de 10 CCDs (dispositivo de carga acoplada) para estudiar cómo interactúan las partículas con los detectores. Las simulaciones se dividieron en dos partes: simulaciones con ángulos de incidencia aleatorios y simulaciones con ángulos de incidencia identificados.

4.2.1. Dataset para predicción de ángulo de incidencia y seguimiento

Se generó un conjunto de datos tomando como base las simulaciones de Geant en archivos ROOT, donde se lanzaron muones en ángulos de 0° a 85°. Se realizaron cálculos para medir la longitud de los trazos en píxeles, y se generó una tabla con todas las características del evento, incluyendo la longitud del muón, como input para el modelo de regresión lineal.

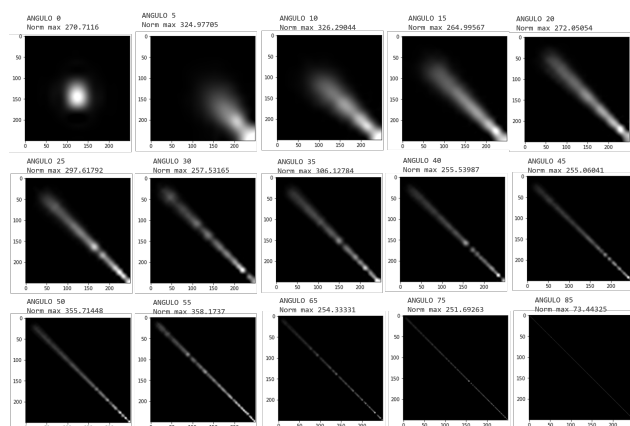


Figura 4: Muones con varios ángulos de incidencia.
Fuente: Elaboración propia

4.2.2. Regresión lineal y polinómica

Además, en este trabajo se ha realizado un análisis de regresión lineal y regresión polinómica. La regresión lineal es una técnica que modela relaciones entre una variable dependiente y una o más variables independientes. Se evaluó el modelo utilizando el coeficiente de determinación (R^2), el error cuadrático medio (MSE), la raíz del error cuadrático medio (RMSE), y el error absoluto medio (MAE).

En la regresión polinómica, se ajustaron curvas polinómicas a los datos para capturar relaciones no lineales. Se utilizó el método de Mínimos Cuadrados Ordinarios (OLS), y se evaluaron los modelos utilizando el coeficiente de determinación (R^2), el R^2 ajustado, la estadística F y el log-likelihood.

Estas técnicas estadísticas y de aprendizaje automático son útiles para comprender y modelar relaciones entre variables, así como para evaluar la calidad de los modelos en la tarea de segmentación y regresión.

5. ANÁLISIS DE RESULTADOS

En este capítulo, se aborda la red YOLOv8, que detecta, clasifica y segmenta objetos en imágenes, evaluando su precisión y velocidad. Se comparan los 5 modelos de YOLOv8 entrenados en el conjunto de datos del CONNIE, se exponen las métricas de evaluación tomadas en cuenta. Luego, se muestran los resultados de dos modelos: un regresor lineal y polinómico. El regresor lineal y polinómico predice valores basados en variables. Se discuten las métricas de rendimiento del regresor, como el error cuadrático medio y el R2R2, y cómo las variables afectan las predicciones.

5.1. Ambiente Experimental

A continuación, se describen detalladamente las herramientas software y el hardware que se han utilizado para llevar a cabo los experimentos en el entrenamiento del modelo YOLO (You Only Look Once) durante esta investigación.

5.1.1. Software

Sistema Operativo Fue utilizado Ubuntu 20.04 LTS como sistema operativo base. Esta versión de Linux brinda una compatibilidad robusta con herramientas de aprendizaje automático y facilita la administración de paquetes y librerías.

Lenguaje de Programación Python 3.8 fue el lenguaje principal de programación. Es ampliamente reconocido en la comunidad de la visión por computadora y ofrece una variedad de bibliotecas útiles para el procesamiento de imágenes y el entrenamiento de modelos.

5.1.2. Librerías Principales

- **OpenCV:** fue utilizada para la manipulación y procesamiento previo de las imágenes.
- **NumPy:** Esta librería ha facilitado las operaciones matriciales y otras operaciones matemáticas de alto nivel.
- **Darknet:** Es el marco de referencia para el entrenamiento de modelos YOLO. Ofrece implementaciones optimizadas y es compatible con GPU para acelerar el entrenamiento.
- **TensorFlow y Keras:** Aunque Darknet es el marco principal para YOLO, TensorFlow y Keras se usaron para algunos experimentos adicionales, análisis y comparaciones.

5.1.3. Hardware

Computadora Principal

- **Procesador:** Intel i9-10900K de 10 núcleos a 3.7 GHz.
- **RAM:** 64 GB DDR4.
- **Almacenamiento:** SSD NVMe de 1 TB.

Tarjeta Gráfica

Fue utilizada una NVIDIA GeForce RTX 2080 con 24 GB de memoria GDDR6X. Esta tarjeta gráfica no solo es compatible con CUDA y cuDNN, sino que también proporciona una aceleración significativa en el entrenamiento del modelo YOLO.

5.2. Segmentación de instancias

Para evaluar el rendimiento del modelo entrenado en nuevos datos no vistos, se dividió en conjuntos de entrenamiento, validación y prueba, en la Figura 5 se observa la cantidad de cada división. Esta partición permitió una evaluación precisa de la capacidad de generalización del modelo y su capacidad para manejar eficazmente instancias novedosas.

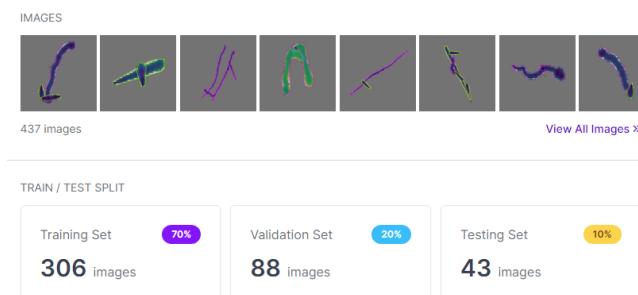


Figura 5: División del dataset en Roboflow. **Fuente:** Elaboración propia

El desbalance en los datos puede producir que la red entre en overfitting, alterando así la capacidad de clasificación de las imágenes. Ya que los eventos pueden ser tomados en cualquier dirección (porque se está analizando la naturaleza de la partícula solo por su forma e intensidad), los dos métodos utilizados para el aumento de datos fueron:

- **Rotaciones:** Las imágenes fueron giradas de manera aleatoria respecto a un eje perpendicular al plano de la imagen.
- **Espejado:** Las imágenes fueron invertidas respecto a una línea del plano.

La cantidad de parámetros de entrada a la red debe ser fija, por lo tanto el tamaño de las imágenes de entrada a la red debe ser el mismo para todos. Las imágenes del dataset ya eran cuadradas pero se ajustaron a el tamaño recomendado en Roboflow y quedaron imágenes de 640x640.

5.3. Métricas de Evaluación

En este trabajo se analizó las métricas mAP para la máscara y para la caja, tanto el mAP de máscara como el mAP de caja son medidas de precisión en la detección de objetos, pero se enfocan en aspectos ligeramente diferentes de la tarea. El mAP de máscara evalúa la precisión en la generación de máscaras de segmentación detalladas para objetos, mientras que el mAP de caja se centra en la precisión en la predicción de las cajas delimitadoras que rodean los objetos.

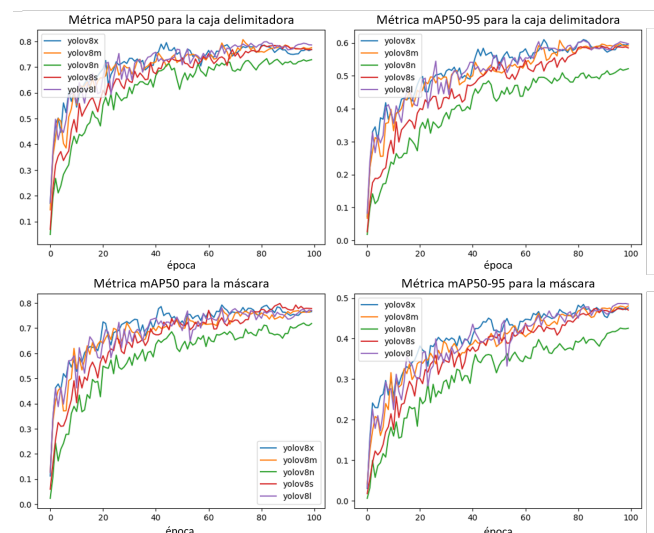


Figura 6: Métricas mAP. **Fuente:** Elaboración propia

En la Figura 6 se puede apreciar como varían las precisiones promedio con los cinco modelos, la métrica mAP50 evalúa la precisión promedio considerando solo un valor de umbral IoU (generalmente 0.50),

mientras que mAP50-95 evalúa la precisión promedio en un rango más amplio de umbrales IoU, desde 0.50 hasta 0.95. La métrica mAP50-95 proporciona una visión más completa del rendimiento de detección del modelo en una variedad de condiciones.

Tabla 2: Comparación de los distintos modelos YOLOv8 en nuestro conjunto de datos.

Model	Size (px)	mAP box 50-95	mAP mask 50-95	Speed GPU Tesla T4 (ms)	Params (M)	Flops (B)
yolov8n-seg	640	0.521	0.425	20.9	3.26	12
yolov8s-seg	640	0.588	0.475	26.4	11.78	42.4
yolov8m-seg	640	0.597	0.477	37.3	27.22	110
yolov8l-seg	640	0.604	0.484	53.5	45.91	220.1
yolov8x-seg	640	0.611	0.485	84.5	71.72	343.7

Fuente: Elaboración propia.

Entonces, obteniendo las métricas expuestas en la Tabla 2 para nuestro conjunto de datos aseguramos un mejor funcionamiento en la segmentación para el modelo yolov8x-seg.

5.3.1. Función de pérdidas

YoloV8 utiliza diversas funciones de pérdida para mejorar la predicción, para el entrenamiento podemos observarlo en la Figura 7 las cuatro utilizadas en nuestro conjunto de imágenes.

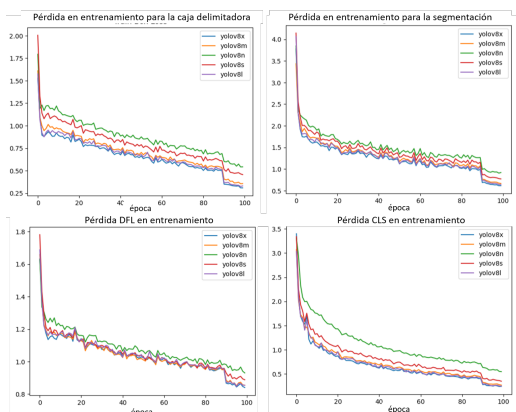


Figura 7: Funciones de pérdida para los cinco modelos en el entrenamiento. Fuente: Elaboración propia.

Box Loss se refiere a la función de costo CIOU, Segmentation Loss a la pérdida de entropía cruzada binaria con Logits, que se diferencia del BCE explicado en que esta función de pérdida combina una capa Sigmoid y la función de pérdida BCE en una sola clase, DFL Loss se refiere a la pérdida de distribución focal y, por último, CLS Loss que es la pérdida de entropía cruzada binaria (BCE).

En general, se espera que todas las funciones de pérdida disminuyan a medida que el modelo se entrena y aprende. Un descenso abrupto seguido de una estabilización en un valor bajo para cada función de pérdida

sugiere un entrenamiento exitoso del modelo. Sin embargo, es importante tener en cuenta que la interpretación exacta puede variar según las características específicas del conjunto de datos y la arquitectura del modelo.

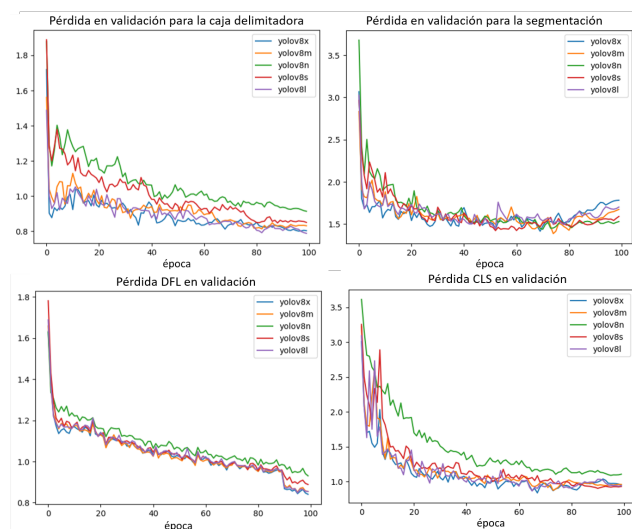


Figura 8: Funciones de pérdida para los cinco modelos en la validación. Fuente: Elaboración propia.

Para nuestro conjunto de validación observamos los comportamientos de la Figura 8. Al comparar todos los modelos vemos que el yolov8x-seg es el que mejores resultados arroja para nuestro conjunto de datos, a pesar de ser el modelo más pesado y el que más tiempo de inferencia requiere, fue el elegido ya que lo que interesa en este trabajo es la precisión en las predicciones de muones y otras partículas solapadas.

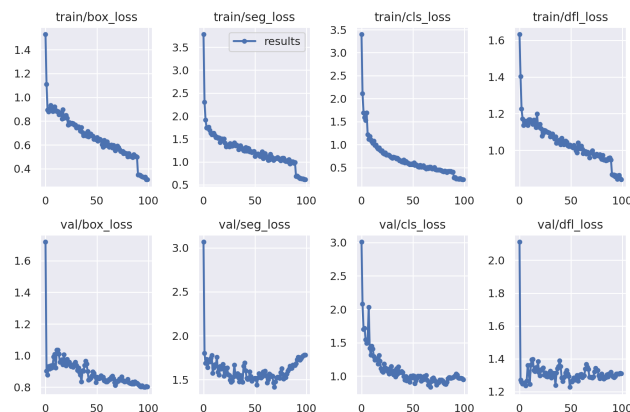


Figura 9: Función de pérdida Fuente: Elaboración propia.

5.4. Resultados del entrenamiento para segmentación de YOLOv8

La figura 10 ilustra una colección de predicciones positivas verdaderas, mostrando casos en los que el mo-

delo identifica y clasifica con precisión los objetos de acuerdo con la verdad sobre el terreno. Por el contrario, la figura 11 ofrece una representación visual de los casos de clasificación errónea, en los que las predicciones del modelo divergen de las etiquetas reales. Estas figuras contrastadas proporcionan una visión visual del rendimiento del modelo, destacando tanto sus puntos fuertes en la detección precisa como sus limitaciones en los casos de clasificación errónea.



Figura 10: Ejemplo de una instancia correcta de la segmentación con el modelo seleccionado.

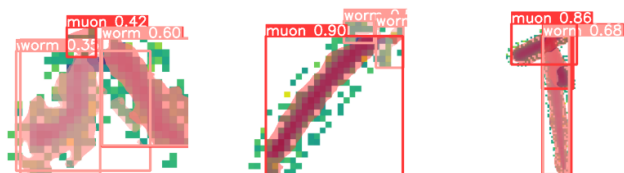


Figura 11: Ejemplo de una instancia incorrecta de la segmentación con el modelo seleccionado.

Por último, para evaluar de forma exhaustiva el rendimiento secuencial de los modelos de clasificación (CNN) y YOLOv8, se ha centrado en un escenario real que incluía un único fotograma CCD con 3.965 eventos distintos. Aprovechando el modelo de clasificación de forma secuencial, se ha observado que entre estos eventos, 2465 se clasificaron correctamente como muones, 599 se identificaron como manchas y 226 se categorizaron como gusanos. Además, 481 eventos se han reconocido adecuadamente como instancias solapadas, mientras que un número más limitado de 194 se han clasificado como indefinidos. Cabe destacar que estos resultados se han obtenido aprovechando el modelo con la mayor recuperación, lo que subraya aún más la competencia del modelo a la hora de capturar instancias de varias clases.

5.5. Seguimiento de muones

Regresión lineal

En la regresión lineal se obtuvieron los siguientes resultados con respecto a los datos de prueba.

En la siguiente figura se puede observar los puntos de los datos reales y la correlación lineal ajustada a los datos.

Se espera una falla importante de la aproximación de los datos para valores bajos y altos.

Tabla 3: Tabla de las métricas de la regresión lineal.

Metrica	Resultado
Error cuadrático medio	93.53
RMSE - Raíz del error cuadrático medio	9.67
MAE - Error medio absoluto	7.54

Fuente: Elaboración propia.

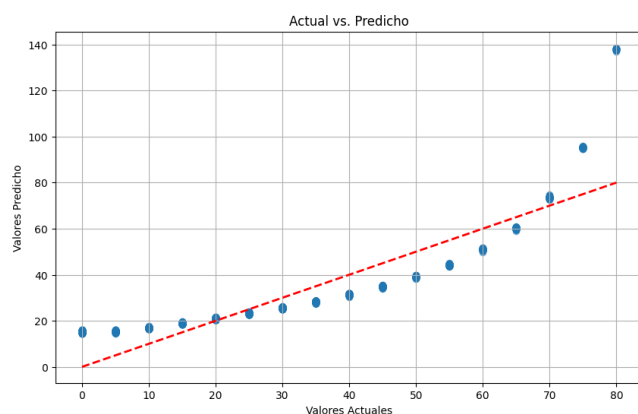


Figura 12: Regresión Lineal **Fuente:** Elaboración propia.

Regresión Polinómica En la Figura 13, en color rojo se muestran los ángulos predichos y en azul los ángulos reales, no se nota mucha diferencia en cuanto a las predicciones entre los de grado 4 y 5.

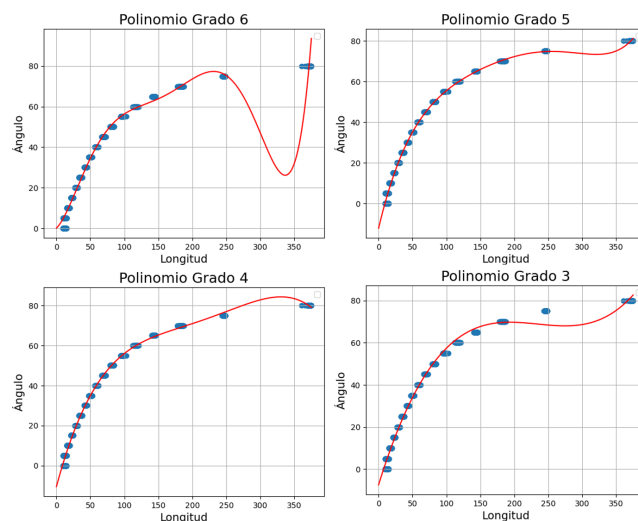


Figura 13: Comparación de resultados entre los modelos de regresión polinómica de distintos grados. **Fuente:** Elaboración propia

Ya que los resultados obtenidos entre los polinomios 4 y 5 no son muy diferentes, se las compara a continuación, para finalmente tomar la decisión de cual se utilizará.

Tabla 4: Comparación de métricas entre modelos de polinomios grado 4 y 5.

Grado de polinomio	4	5
R-cuadrado	0.995	0.996
R-cuadrado Ajustado	0.995	0.996
Estadística F	6.655e+05	5.902e+05
Logaritmo de la razón de verosimilitud (Log-Likelihood)	-20710	-20088
AIC	4.143e+04	4.019e+04
BIC	4.147e+04	4.023e+04

Fuente: Elaboración propia.

Con estos resultados, el mejor R^2 y R^2 ajustado es el del grado 5 pero es casi insignificante por sobre el grado 4, en cuanto a la estadística F y el Log-likelihood (valor absoluto) se prefiere el valor mayor comparando con el AIC y BIC que es mejor cuanto menor es el valor, concluimos en la elección del polinomio de grado 4 teniendo en cuenta estas consideraciones. Utilizando el polinomio de grado 4 para predecir los ángulos dependiendo de la longitud de los muones en píxeles podemos obtener algunas métricas extra, tal como:

- Error cuadrático medio o mean squared error (MSE): en promedio, las predicciones del modelo tienen un error cuadrático medio de 1.8206 al cuadrado en relación con los valores reales. En general, un MSE de 1.8206 sugiere un nivel aceptable de precisión.
- Raíz del error cuadrático medio o root mean squared error (RMSE): Las predicciones tienen un error de 1.3493 en relación con los valores reales. Un RMSE de 1.3493 indica un nivel razonable de precisión.
- Error absoluto medio o mean absolute error (MAE): las predicciones tienen un error absoluto medio de 0.9098 en relación con los valores reales. Un MAE de 0.9098 sugiere que las predicciones están cerca de los valores reales.

Tabla 5: Tabla de métricas de la regresión polinómica.

Metrica	Resultado
Error cuadrático medio	1.8206
RMSE - Raíz del error cuadrático medio	1.3493
MAE - Error medio absoluto	0.9098

Fuente: Elaboración propia.

5.6. Predicción de ángulo de incidencia

En la figura 14 se muestra que la predicción para ángulos muy pequeños, cercanos a 0° y ángulos que se acercan a 90° , deja de ser tan preciso que para los

ángulos entre estos mencionados, si bien, el polinomio de grado 4 es el que mejor se ajusta a esta curva, estas imprecisiones se deben a que mientras mayor sea el ángulo de incidencia (entre 70° y 90°) la longitud de su trazo varía mucho más entre longitudes más grandes, sin embargo para ángulos entre 0° y 5° el trazo que deja en el CCD se asemeja a un punto. También, en la tabla 6 se observan los coeficientes de este polinomio donde podemos concluir que los correspondientes a las variables con exponente 3 y 4 podrían no aportar tanta información a la predicción pero son necesarios en el enfoque de este proyecto ya que hace una diferencia para los ángulos mayores de 70° , como se pudo ver en la figura 13.

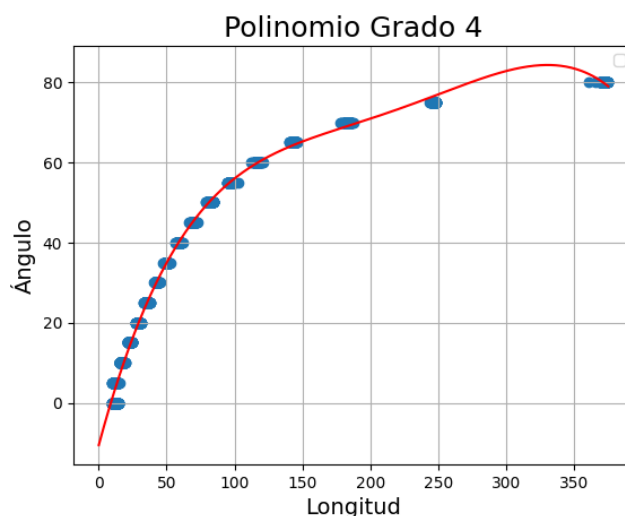


Figura 14: Gráfico de longitud y ángulos reales vs predichos. **Fuente:** Elaboración propia.

Tabla 6: Coeficientes del polinomio de grado 4.

variables	coeficiente
constante	-10.5924
x1	1.2644
x2	-0.0083
x3	2.597e-05
x4	-2.976e-08

Fuente: Elaboración propia.

6. Conclusión

Este estudio se centró en evaluar cómo las técnicas avanzadas de Deep Learning, particularmente el modelo YOLOv8, realizan la segmentación y el seguimiento de muones en el experimento CONNIE. Dentro de las métricas evaluadas, se destacó el mAP (caja delimitadora) de 61.1 % y mAP (máscara) de 48.5 % del modelo yolov8x-seg, a pesar de su mayor complejidad y tiempo de ejecución. Este modelo mostró una

disminución progresiva de pérdidas, consolidándose como la mejor opción.

La evaluación de las regresiones lineal y polinómica para el ángulo de incidencia reveló resultados interesantes. La regresión polinómica, con un 99.5% de precisión, superó a la lineal (93.53%), indicando la posibilidad de enfoques más complejos para mejorar las predicciones de ángulos.

El estudio subrayó el potencial significativo del Deep Learning en la física de partículas, específicamente en la segmentación y seguimiento de muones. A pesar de los avances logrados en precisión, se identificaron áreas que requieren más refinamiento y enfoque, destacando la constante búsqueda de calidad y precisión en este campo.

Los resultados obtenidos resaltan la relevancia del Deep Learning en la física de partículas, mostrando avances prometedores con implicaciones significativas para la investigación en este campo científico crucial.

Referencias

- [1] R. Acciarri, C. Adams, R. An, J. Asaadi, M. Auger, L. Bagby, B. Baller, G. Barr, M. Bass, F. Bay, M. Bishai, A. Blake, T. Bolton, L. Bugel, L. Camilleri, D. Caratelli, B. Carls, R. Castillo Fernandez, F. Cavanna, H. Chen, E. Church, D. Cianci, G. H. Collin, J. M. Conrad, M. Convery, J. I. Crespo-Anadón, M. Del Tutto, D. Devitt, S. Dytman, B. Eberly, A. Ereditato, L. Escudero Sanchez, J. Esquivel, B. T. Fleming, W. Foreman, A. P. Furmanski, G. T. Garvey, V. Genty, D. Goeldi, S. Gollapinni, N. Graf, E. Gramellini, H. Greenlee, R. Grosso, R. Guenette, A. Hackenburg, P. Hamilton, O. Hen, J. Hewes, C. Hill, J. Ho, G. Horton-Smith, C. James, J. J. de Vries, C. M. Jen, L. Jiang, R. A. Johnson, B. J. P. Jones, J. Joshi, H. Jostlein, D. Kaleko, G. Karagiorgi, W. Ketchum, B. Kirby, M. Kirby, T. Kobilarcik, I. Kreslo, A. Laube, Y. Li, A. Lister, B. R. Littlejohn, S. Lockwitz, D. Lorca, W. C. Louis, M. Luethi, B. Lundberg, X. Luo, A. Marchionni, C. Mariani, J. Marshall, D. A. Martinez Caicedo, V. Meddage, T. Miceli, G. B. Mills, J. Moon, M. Mooney, C. D. Moore, J. Mousseau, R. Murrells, D. Naples, P. Nienaber, J. Nowak, O. Palamara, V. Paolone, V. Papavassiliou, S. F. Pate, Z. Pavlovic, D. Porzio, G. Pulliam, X. Qian, J. L. Raaf, A. Rafique, L. Rochester, C. R. von Rohr, B. Russell, D. W. Schmitz, A. Schukraft, W. Seligman, M. H. Shaevitz, J. Sinclair, E. L. Snider, M. Soderberg, S. Söldner-Rembold, S. R. Soleti, P. Spentzouris, J. Spitz, J. St. John, T. Strauss, A. M. Szelc, N. Tagg, K. Terao, M. Thomson, M. Touns, Y. T. Tsai, S. Tufanli, T. Usher, R. G. Van de Water, B. Viren, M. Weber, J. Weston, D. A. Wickremasinghe, S. Wolbers, T. Wongji-rad, K. Woodruff, T. Yang, G. P. Zeller, J. Zen-namo, and C. Zhang, "Convolutional neural networks applied to neutrino events in a liquid argon time projection chamber," *Journal of Instrumentation*, vol. 12, no. 3, p. P03011, Mar. 2017.
- [2] B. Abi, R. Acciarri, M. Acero *et al.*, "Volume i. introduction to dune," *Journal of Instrumentation*, vol. 15, no. 08, p. T08008, aug 2020. [Online]. Available: <https://dx.doi.org/10.1088/1748-0221/15/08/T08008>
- [3] D. Bourilkov, "Machine and deep learning applications in particle physics," *International Journal of Modern Physics A*, vol. 34, no. 35, p. 1930019, Dec. 2019.
- [4] R. Acciarri, B. Baller, V. Basque, C. Bromberg, F. Cavanna, D. Edmunds, R. Fitzpatrick, B. Fleming, P. Green, C. James *et al.*, "A deep-learning based raw waveform region-of-interest finder for the liquid argon time projection chamber," *Journal of Instrumentation*, vol. 17, no. 01, p. P01018, 2022.
- [5] A. Aguilar-Arevalo, X. Bertou, C. Bonifazi, M. Butner, G. Cancelo, A. C. Vazquez, B. C. Vergara, C. Chavez, H. D. Motta, J. D'Olive, J. D. Anjos, J. Estrada, G. F. Moroni, R. Ford, A. Foguel, K. P. H. Torres, F. Izraelevitch, A. Kavner, B. Kilminster, K. Kuk, H. Lima, M. Makler, J. Molina, G. Moreno-Granados, J. Moro, E. Paolini, M. S. Haro, J. Tiffenberg, F. Trillaud, and S. Wagner, "The CONNIE experiment," *Journal of Physics: Conference Series*, vol. 761, p. 012057, oct 2016. [Online]. Available: <https://doi.org/10.1088/1742-6596/761/1/012057>
- [6] M. Winter, J. Bourbeau, S. Bravo, F. Campos, M. Meehan, J. Peacock, T. Ruggles, C. Schneider, A. L. Simons, and J. Vandenbroucke, "Particle identification in camera image sensors using computer vision," *Astroparticle Physics*, vol. 104, pp. 42–53, Jan. 2019.
- [7] M. Piekarczyk, O. Bar, Bibrzycki, M. Niedźwiecki, K. Rzecki, S. Stuglik, T. Andersen, N. M. Budnev, D. E. Alvarez-Castillo, K. A. Cheminant, D. Góra, A. C. Gupta, B. Hnatyk, P. Homola, R. Kamiński, M. Kasztelan, M. Knap, P. Kovács, B. Łozowski, J. Miszczyk, A. Mozgova, V. Nazari, M. Pawlik, M. Rosas, O. Sushchov, K. Smelcerz, K. Smolek, J. Stasielak, T. Wibig, K. W. Woźniak, and J. Zamora-Saa, "Cnn-based classifier as an offline trigger for the credo experiment," *Sensors*, vol. 21, no. 14, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/14/4804>

- [8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [9] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv e-prints*, p. arXiv:1409.1556, Sep. 2014.
- [10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [11] I. Nasteva, "Low-energy reactor neutrino physics with the connie experiment," in *Journal of Physics: Conference Series*, vol. 2156, no. 1. IOP Publishing, 2021, p. 012115.
- [12] C. Collaboration, A. Aguilar-Arevalo, X. Bertou, C. Bonifazi, M. Butner, G. Cancelo, A. C. Vazquez, B. C. Vergara, C. R. Chavez, H. D. Motta, J. C. D'Olivo, J. D. Anjos, J. Estrada, G. F. Moroni, R. Ford, A. Foguel, K. P. H. Torres, F. Izraelevitch, A. Kavner, B. Kilminster, K. Kuk, H. P. L. J. au2, M. Makler, J. Molina, G. Moreno-Granados, J. M. Moro, E. E. Paolini, M. S. Haro, J. Tiffenberg, F. Trillaud, and S. Wagner, "The connie experiment," 2016.
- [13] J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. A. Dubois, M. Asai, G. Barrand, R. Capra, S. Chauvie, R. Chytrcek *et al.*, "Geant4 developments and applications," *IEEE Transactions on nuclear science*, vol. 53, no. 1, pp. 270–278, 2006.
- [14] L. Rouhiainen, "Inteligencia artificial," *Madrid: Alienta Editorial*, 2018.
- [15] J. Alzubi, A. Nayyar, and A. Kumar, "Machine learning from theory to algorithms: an overview," in *Journal of physics: conference series*, vol. 1142, no. 1. IOP Publishing, 2018, p. 012012.
- [16] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.
- [17] J. Heaton, "Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618," *Genetic Programming and Evolvable Machines*, vol. 19, no. 1-2, pp. 305–307, 2018.
- [18] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [19] Á. Artola Moreno, "Clasificación de imágenes usando redes neuronales convolucionales en python," 2019.
- [20] Y. Aljure Jiménez, "Clasificación de flores con redes neuronales convolucionales," 2021.
- [21] F. Chollet, *Deep Learning with Python*. Manning Publications, 2017.
- [22] SuperDataScience, "Convolutional Neural Networks (CNN) - Step 4: Full Connection," <https://www.superdatascience.com/convolutional-neural-networks-cnn-step-4-full-connection>, 2018, accedido el 28 de abril de 2023.
- [23] G. Jocher, A. Chaurasia, and J. Qiu, "YOLO by Ultralytics," Jan. 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
- [24] S. Elfving, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *CoRR*, vol. abs/1702.03118, 2017. [Online]. Available: <http://arxiv.org/abs/1702.03118>
- [25] S. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. D. Reid, and S. Savarese, "Generalized intersection over union: A metric and A loss for bounding box regression," *CoRR*, vol. abs/1902.09630, 2019. [Online]. Available: <http://arxiv.org/abs/1902.09630>
- [26] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," 2020.
- [27] Y.-F. Lu, Q. Yu, J.-W. Gao, Y. Li, J.-C. Zou, and H. Qiao, "Cross stage partial connections based weighted bi-directional feature pyramid and enhanced spatial transformation network for robust object detection," *Neurocomputing*, vol. 513, pp. 70–82, 2022.
- [28] J. A. B. Sigala, *Deep Learning Aplicado a la Identificación de Muones del Experimento CONNIE*. Facultad de Ingeniería UNA, 2023.
- [29] Ultralytics, "ultralytics/ultralytics/cfg/datasets/coco.yaml at main · ultralytics/ultralytics — github.com," [Accessed 09-08-2023].
- [30] "Ultralytics Segmentation Documentation," <https://docs.ultralytics.com/tasks/segment/>, 2023, accedido el 7 de mayo de 2023.
- [31] S. Agostinelli, J. Allison, K. a. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Bannerjee, G. Barrand *et al.*, "Geant4—a simulation toolkit," *Nuclear instruments and methods in*

physics research section A: Accelerators, Spec-
trometers, Detectors and Associated Equipment,

vol. 506, no. 3, pp. 250–303, 2003.