



Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO FIN DE GRADO

Grado en Física

**Detección de rayos cósmicos mediante deep learning.
Diseño y desarrollo del dispositivo experimental**

Autor: Miguel Franco Hernando

**Tutores: Manuel Luis González Hernández y
Luis Miguel Molina Martín**

Año 2023/2024

Resumen

El objetivo de este trabajo es diseñar un dispositivo experimental capaz de detectar rayos cósmicos mediante deep learning. De esta manera, en primer lugar, se presenta una introducción a los rayos cósmicos y las cascadas atmosféricas, esenciales para la caracterización del objeto de estudio. Por otro lado, se introducen los sensores CCD (charged-coupled device) como medio para realizar detecciones de partículas cargadas. Posteriormente se muestra el dataset de entrenamiento y el modelo de IA empleado en el dispositivo, detallando todo el proceso seguido durante su desarrollo. Para finalizar, se presenta el dispositivo físico con el que se ha desplegado el modelo y recogido las detecciones. Por último, se analizan los resultados obtenidos durante la experimentación, analizando la validez del dispositivo experimental.

Abstract

The aim of this work is to design an experimental device capable of detecting cosmic rays using deep learning. Firstly, an introduction to cosmic rays and atmospheric cascades is presented, essential for characterizing the object of study. On the other hand, CCD (charged-coupled device) sensors are introduced as a means to detect charged particles. Subsequently, the training dataset and the AI model used in the device are shown, detailing the entire process followed during its development. Finally, the physical device with which the model has been deployed and detections have been collected is presented. Lastly, the results obtained during experimentation are analyzed, assessing the validity of the experimental device.

Índice general

Introducción	6
1. Rayos Cósmicos	8
1.1. Caracterización	8
1.1.1. Origen	9
1.1.2. Cascadas atmosféricas de rayos cósmicos	11
1.2. Detección	13
1.2.1. CREDO	13
1.2.2. DECO	14
2. Proceso de Detección y Dataset	16
2.1. Detección con CCD	16
2.1.1. Fotodiodos	16
2.1.2. CCD	17
2.2. Dataset	18
2.2.1. Obtención	18
2.2.2. Análisis	21
3. Modelo y entrenamiento	23
3.1. Redes neuronales	23
3.1.1. Convolución	26
3.1.2. Pooling	26
3.2. Modelo YOLO	27
3.3. Generación del dataset	28
3.4. Entrenamiento	30
4. Dispositivo Experimental	34
4.1. Jetson Orin NX	34
4.2. Cámara	35
4.3. Manejo	36
4.4. Dispositivo alternativo	37

5. Resultados y conclusiones	38
5.1. Resultados obtenidos	38
5.2. Conclusiones	39
5.3. Líneas de trabajo futuras	40
A. Código	41
B. Detecciones	44
Bilbiografía	48

Introducción

Los rayos cósmicos son partículas de alta energía que alcanzan la atmósfera terrestre. Su descubrimiento a principios del siglo XX supuso una revolución total en la física de partículas. En ellos se encontraron partículas elementales como los positrones y los muones o incluso la primera partícula con extrañeza, los kaones.

La detección de los rayos cósmicos es una tarea compleja. La medida directa de estos eventos es muy complicada, por lo que se emplean medidas indirectas. Cuando un rayo cósmico alcanza la atmósfera, se inicia una cascada de partículas secundarias. Estudiando estas cascadas es posible caracterizar el rayo primario.

Por su parte, la informática, y más recientemente la inteligencia artificial, han transformado la sociedad actual. Estas herramientas pueden ser usadas para multitud de procesos: clasificación de imágenes, generación de texto... e incluso para detectar partículas. Además, los sensores CCD (*charge-coupled device*), incluidos en muchas cámaras actuales, son capaces de responder cuando una partícula los atraviesa. Con las dos herramientas anteriores es posible obtener un detector de rayos cósmicos.

Así, el objetivo de este trabajo es construir un dispositivo portable capaz de detectar rayos cósmicos con la ayuda de los sensores CCD y la inteligencia artificial. Este proyecto surge como propuesta de la empresa ITCL Centro Tecnológico (Instituto Tecnológico de Castilla y León) para la realización del trabajo de fin de grado del grado en física.

En cuanto a la organización del trabajo, el capítulo primero se centra en la caracterización de los rayos cósmicos: estudia su origen y las cascadas atmosféricas. Además, se presentan dos proyectos colaborativos sobre la detección de estos fenómenos.

El uso de sensores CCD para realizar las detecciones se analiza en el capítulo segundo. Además, se comenta la base de datos de la que se dispone para entrenar nuestros modelos de inteligencia artificial. En el capítulo tercero se detalla el funcionamiento de las redes neuronales y se introduce el modelo

específico empleado en el trabajo. Igualmente se detalla su entrenamiento y se valoran diversas métricas.

Finalmente, el capítulo cuarto introduce el dispositivo experimental empleado en la práctica. Para concluir, el último capítulo analiza los resultados obtenidos durante las experimentaciones y propone nuevas líneas de trabajo. Es necesario mencionar que se ha incluido un anexo al final del trabajo con el script utilizado para realizar las detecciones. El resto del código de generación del dataset y entrenamiento es posible encontrarlo en el siguiente repositorio de [GitHub](#). No se ha incluido en el trabajo debido a su extensión.

Para finalizar, expresar mi agradecimiento a Manuel, Erik y Adrián por su dedicación en este proyecto.

Junio 2024.
Miguel Franco Hernando.

Capítulo 1

Rayos Cósmicos

Este capítulo tiene como objetivo principal introducir el concepto de rayo cósmico, analizando su espectro energético y explorando sus posibles fuentes. También se examina el fenómeno de las cascadas atmosféricas, estrechamente relacionado con los rayos cósmicos y fundamental para caracterizar su energía. Finalmente, se abordan brevemente los desafíos asociados con la detección de estos fenómenos, presentando los proyectos colaborativos CREDO y DECO, bases del resto del trabajo, como posibles soluciones.

1.1. Caracterización

La atmósfera terrestre recibe diferentes tipos de radiación, abarcando un amplio rango de flujo y energía. El espectro energético es extenso, comprende desde los $10^{-3}eV$, como la radiación cósmica de microondas, hasta energías excepcionalmente altas de partículas cargadas que alcanzan los $10^{20}eV$.

Generalmente, el término *rayos cósmicos* (CR) se refiere a aquellas partículas provenientes de fuera del sistema solar, es decir, con energías por encima de los 10^9eV . Este tipo de radiación fue descubierta por Victor Hess en 1912, cuando midió un incremento de la radiación ionizante al aumentar la altura en una ascensión en globo durante un eclipse de Sol [1]. Esto indicó que la radiación registrada no provenía de dicha estrella. Los rayos cósmicos están compuestos principalmente por protones, aunque también hay presentes núcleos más pesados, electrones, positrones y antiprotones.

En el estudio de su espectro energético se observa que el flujo de partículas incidentes decrece rápidamente al aumentar la energía, como se puede observar en la figura 1.1. De esta forma, el flujo de rayos cósmicos parece que sigue una ley del orden de $\sim E^{-2.7}$ [2]. Sin embargo, esta tendencia se acentúa en el rango entre los $10^{15}eV - 10^{18}eV$, en la denominada región *knee-ankle*,

alcanzando una tendencia de $\sim E^{-3.0}$.

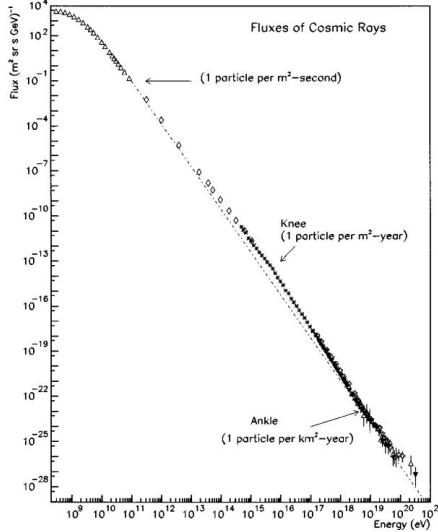


Figura 1.1: Espectro del flujo de rayos cósmicos en función de la energía. Nótese el cambio de tendencia en la región *knee-ankle*. Imagen extraída de [3].

1.1.1. Origen

Desde el descubrimiento de Hess a principios del siglo XX, todavía no existe una teoría clara sobre la fuente de origen de este tipo de radiación. En este contexto, destaca el criterio de Hillas, que permite analizar posibles candidatos a fuentes de rayos cósmicos que acelerarían las partículas gracias a la presencia de campos magnéticos.

Este criterio está basado en la aproximación de que si la partícula abandona la región donde fue acelerada, ya no es posible que sea capaz de ganar más energía. De esta manera, es posible estimar la energía máxima que podría alcanzar una partícula a través de la relación siguiente [2, pág 9]:

$$E_{max} = qBRc,$$

donde q es la carga del electrón, B el campo magnético, R el radio de la trayectoria confinada y c la velocidad de la luz. Con la expresión anterior, realizando la representación gráfica de la figura 1.2, es posible analizar posibles fuentes de rayos cósmicos en función de su energía. Esta aproximación nos da diversos candidatos como estrellas de neutrones o núcleos activos de galaxias. Estos últimos se corresponden con regiones compactas en el centro de las galaxias que emiten una cantidad reseñable de energía a lo largo de

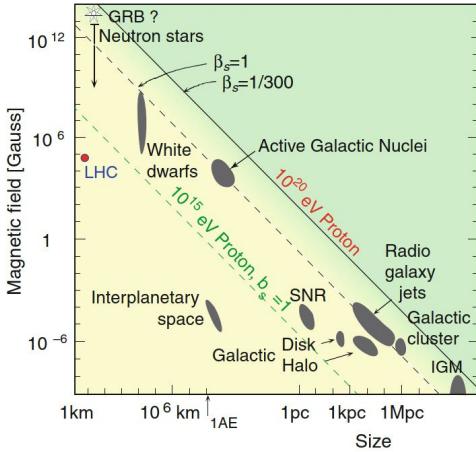


Figura 1.2: Criterio de Hillas. Imagen extraída de [4, pág 8].

todo el espectro electromagnético. Estas emisiones están causadas por agujeros negros [5, pág 2]. En este contexto, es interesante analizar el posible mecanismo de creación de rayos cósmicos en este último objeto astronómico.

Agujeros Negros

Los agujeros negros súper masivos son considerados como una de las fuentes de rayos cósmicos más plausibles [6]. En este caso, el mecanismo de aceleración estaría provocado por el proceso de Penrose. Este se basa en que en los agujeros negros existe una zona, la ergosfera, donde las partículas pueden ser capaces de tener una energía negativa respecto al infinito. De esta manera, si una partícula se divide en dos, es posible que una tenga energía negativa, por lo que por la conservación de la energía, la otra partícula tendría mayor energía que la partícula original. Este proceso combinado con un campo magnético presente en las cercanías de un agujero negro sería capaz de extraer energía de manera *ultraeficiente* [7]. De esta forma, dada la siguiente reacción nuclear:

$$n \rightarrow p^+ + e^- + \bar{\nu}_e,$$

un protón podría ser acelerado a muy altas energías. En la figura 1.3 se puede comprobar que este método es una posible fuente de rayos cósmicos, ya que las posibles energías coinciden con las de estudio. De igual forma, se puede observar el diagrama de una simulación numérica del proceso de Penrose.

Un hecho destacable, es que este modelo aplicado al agujero negro Sgr A*¹ proporciona una energía máxima para los protones similar a la de la

¹Se trata del agujero negro que se encuentra en el centro de nuestra galaxia.

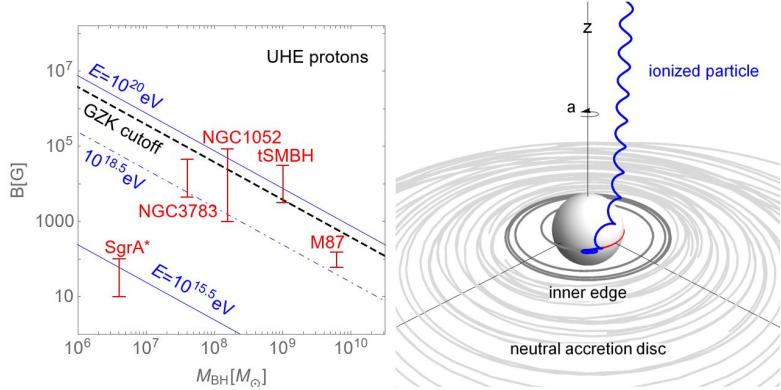


Figura 1.3: Izquierda: Representación del campo magnético en función de la masa del agujero negro (en masas solares). Refleja que hay agujeros negros cercanos a la Tierra capaces de acelerar protones en el rango de energías de los rayos cósmicos. Derecha: Simulación numérica de una partícula que cae en un agujero negro y se desintegra. La partícula positiva (azul) escapa mientras que la negativa (rojo) cae. Imágenes extraídas de [6].

knee, aunque esta teoría requiere de mayores comprobaciones [8]. De igual forma, las expresiones teóricas para estas energías se encuentran en [8], pero no las tratamos aquí debido a su complejidad.

1.1.2. Cascadas atmosféricas de rayos cósmicos

Cuando los rayos cósmicos alcanzan la atmósfera terrestre crean cascadas atmosféricas (*air showers* en inglés). Esto es debido a que la partícula incidente, al llegar a la atmósfera, interactúa con moléculas produciendo un gran número de partículas secundarias que a su vez repiten el proceso, de ahí el nombre de cascada. Una representación gráfica se puede observar en la figura 1.4. Este tipo de fenómenos se suelen estudiar con simulaciones de Monte-Carlo debido a la gran cantidad de partículas que participan en el fenómeno, que podrían exceder el número de 10^{10} [9].

En función del tipo de partícula primaria se desarrollará una componente u otra de la cascada. Por ejemplo, un electrón producirá una componente electromagnética. En un modelo simple, en este proceso, aparecen fotones que crean pares electrón-positrón, por lo que estos son los tres posibles productos que se pueden registrar. En el caso de una cascada hadrónica estas son iniciadas por núcleos o hadrones. Sus productos finales son más complicados de analizar. Sin embargo, en un modelo simple se asume que solo se producen piones, de los cuales un porcentaje decae en muones. Estos últimos son capaces de alcanzar la superficie de la Tierra y, por tanto, pueden ser detectados [4, pág 9]. El estudio de estas cascadas es importante para la ca-

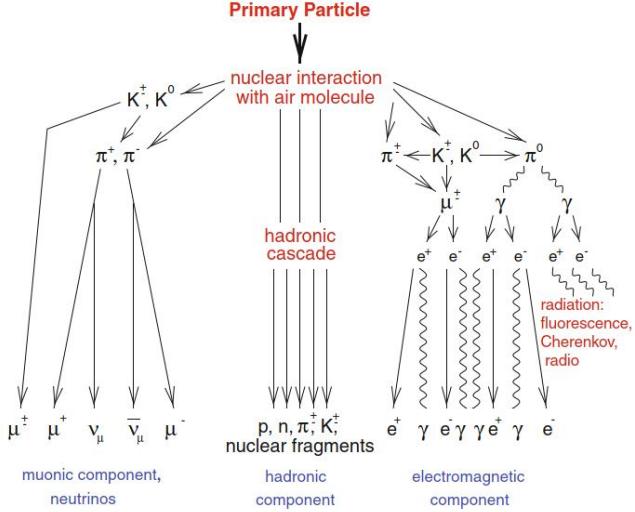


Figura 1.4: Esquema del desarrollo de una cascada atmosférica iniciada por un rayo cósmico. Se observan las componentes posibles: muónica y de neutrinos, electromagnética y hadrónica. Imagen extraída de [4].

racterización del rayo primario, ya que es posible estimar la energía original a partir del número de partículas detectadas. De esta manera, una estimación simple puede venir dada por la siguiente expresión [4, pág 10], válida para bajas altitudes:

$$E_0 \approx a(N_e + bN_\mu),$$

con $a = 0,85\text{GeV}$ y $b = 25$ a nivel del mar; donde N_e y N_μ representan el número de electrones y muones detectados respectivamente.

En el estudio de estas cascadas es relevante el papel que juega la relatividad especial, ya que algunas partículas se mueven a velocidades cercanas a la de la velocidad de la luz. Para ilustrar este hecho, vamos a tratar el caso de los muones. Estos se generan a una altura de unos 15km de altura sobre la superficie terrestre, con una velocidad aproximada de $2,98 \cdot 10^8\text{m/s}$ [10]. Los muones tienen una vida media aproximada de $2,19\mu\text{s}$, los cuales siguen la ley de decaimiento exponencial,

$$N(t) = N_0 e^{-\frac{t}{\tau}},$$

donde τ es la vida media. De esta manera, podemos tomar N_0 como el número de muones presentes a 15km de altura y queremos ver cuantos llegan a nivel del mar. Sin correcciones relativistas, el tiempo entre la generación de un muon y detección a nivel del mar es $50,3\mu\text{s}$. Con este dato, podemos calcular:

$$\frac{N_{\text{detector}}}{N_0} = \exp\left(-\frac{50,3 \cdot 10^{-6}}{2,19 \cdot 10^{-6}}\right) \approx 1,06 \cdot 10^{-10}.$$

Según esto, la fracción de muones detectados sería ínfima, lo que va en contra de los resultados experimentales [10]. Sin embargo, teniendo en cuenta la alta velocidad del muon es posible realizar correcciones relativistas. En este caso, tenemos que el factor de Lorentz es $\gamma = \frac{1}{\sqrt{1+v^2/c^2}} = 9,14$. Aplicando las expresiones usuales, es posible calcular el tiempo que el muon tarda en llegar al detector en su sistema de referencia: $t_\mu = t_{Tierra}/\gamma = 5,50\mu s$. Con esto se tiene que:

$$\frac{N_{detector}}{N_0} = \exp\left(-\frac{5,50 \cdot 10^{-6}}{2,19 \cdot 10^{-6}}\right) \approx 8,12 \cdot 10^{-2}.$$

Estos resultados sí concuerdan con las experimentaciones habituales [10], lo que reivindica la importancia de la relatividad especial en nuestro problema. Obsérvese que los resultados con correcciones y sin ellas difieren en ocho órdenes de magnitud.

1.2. Detección

Habitualmente las técnicas de medición para las cascadas atmosféricas se basan en la detección de partículas a nivel terrestre o en medidas indirectas de la energía contenida en las cascadas electromagnéticas [4, pág 11]. Este último método se basa en el estudio de la radiación de Cherenkov de electrones que viajan en la atmósfera. La detección a nivel terrestre requiere una gran cantidad de detectores, ya que es necesario que cubran un área extenso. Ante este inconveniente han surgido proyectos colaborativos como [CREDO](#) o [DECO](#), siendo estos la base del presente trabajo.

1.2.1. CREDO

El proyecto CREDO, acrónimo de *Cosmic-Ray Extremely Distributed Observatory*, consiste en una colaboración global dedicada a la observación y estudio de los rayos cósmicos. Este proyecto tiene como objetivo, además del estudio habitual de los CR y las *air showers* correspondientes, la detección de cascadas atmosféricas extensas, ver la figura 1.5. Este tipo de cascadas se caracterizan porque el rayo cósmico incidente interacciona antes de llegar a la atmósfera (e.g. interacción fotón-fotón [11, pág 2]), por lo que se forman varias cascadas.

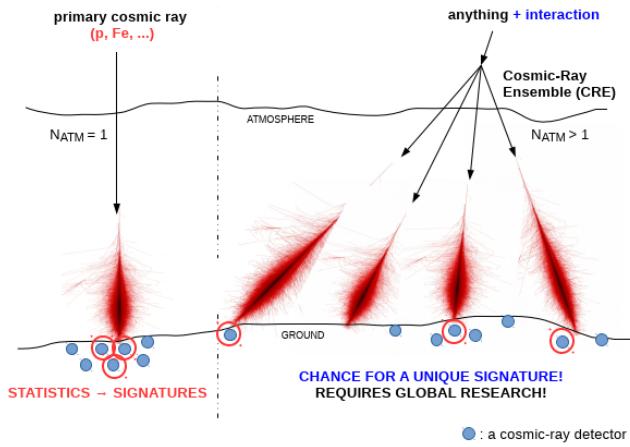


Figura 1.5: Cascada atmosférica extensa.Imagen extraída de [11].

En este contexto aparece la necesidad de un detector de carácter global. Sin embargo, no se han realizado estudios de este tipo de cascadas, debido a la infraestructura necesaria para ello, ya que sería necesario un conjunto de detectores incluso más grande que para una cascada usual.

Ante este problema, el proyecto CREDO realiza una propuesta: utilizar los teléfonos móviles y sus cámaras con sensores CMOS/CCD para realizar las detecciones. En el siguiente capítulo se desarrollará esta idea.

1.2.2. DECO

Este proyecto, acrónimo de *Distributed Electronic Cosmic-ray Observatory*, realiza una propuesta similar al caso anterior. A través del uso de los sensores CCD/CMOS de los teléfonos móviles se propone crear una red global de detectores de partículas producidos por rayos cósmicos o desintegraciones radioactivas. Han conseguido un gran número de colaboradores, consiguiendo detectores en todos los continentes como se puede observar en la figura 1.6.

A la vista de lo anterior, es posible esgrimir que este tipo de proyectos de carácter global pueden suponer una revolución en la recolección de información de los rayos cósmicos, permitiendo avanzar en el estudio de este fenómeno.



Figura 1.6: Localizaciones de detectores empleados por DECO. Imagen extraída de [12].

Capítulo 2

Proceso de Detección y Dataset

Este capítulo tiene como objetivo presentar la capacidad de los detectores CCD de funcionar como detectores de partículas, exponiendo su manera de operar. De igual forma, se introduce el dataset del que dispone el proyecto *DECO*, el cual será la base del trabajo posterior. En él es posible diferenciar tres tipos de marcas, los denominados *tracks*, *worms* y *spots*. Se analizan los posibles candidatos capaces de realizar tales marcas. Finalmente, se desarrolla un modelo sencillo para los *tracks*. A través de él, es posible realizar una estimación de la dirección original de la partícula que dejó la marca de estudio.

2.1. Detección con CCD

Como se comentó en el capítulo anterior, los proyectos que motivan este trabajo intentan crear una red global de detectores de rayos cósmicos a través de los teléfonos móviles. Por esta razón es importante caracterizar la interacción de los rayos con los sensores.

2.1.1. Fotodiodos

En primer lugar, se analizan los fotodioidos. Estos dispositivos están formados por una unión P-N. Su funcionamiento está basado en la creación de pares electrón-hueco debido a la incidencia de un fotón con suficiente energía en la zona de carga espacial (efecto fotoeléctrico). Debido al campo eléctrico presente en esta zona, se establece una corriente causada por los pares generados en esta interacción, la denominada fotocorriente. Por tanto, la corriente total que circula por el diodo será la suma de esta fotocorriente y la denominada corriente oscura, que es aquella presente cuando el diodo se

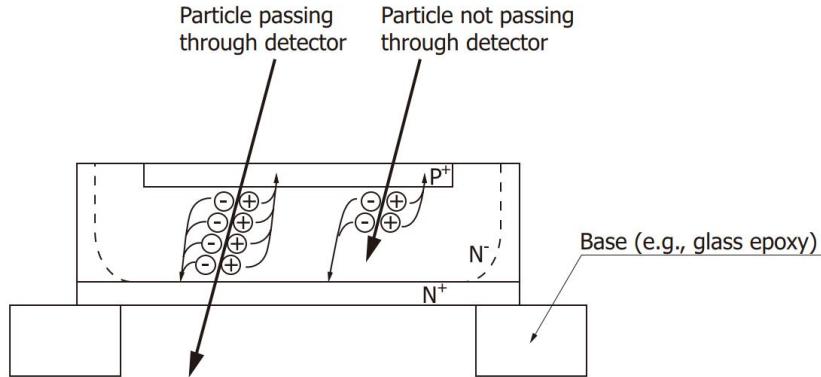


Figura 2.1: Partícula incidente en un fotodiodo. En la región P^+ y N^+ las cargas producidas se recombinan, por lo que no contribuyen a la corriente. Imagen extraída de [13].

encuentra en una oscuridad total.

Sin embargo, este fenómeno también puede ocurrir cuando una partícula cargada incide en el fotodiodo, como se puede observar en la figura 2.1. En este fenómeno, cuando la energía es adecuada, la partícula incidente la disipa mientras recorre una trayectoria rectilínea. La longitud recorrida depende del tipo de partícula y, por supuesto, de la energía de esta. Además, la partícula incidente a su paso va creando los pares electrón-hueco debido a la interacción coulombiana. El número de pares generados no depende del tipo de partícula, si no de la energía inicial de esta. A $300K$, un par electrón-hueco se genera de media por cada $3,62eV$ [13]. De esta manera, es posible estimar la pérdida de energía a través de la medición de esta carga generada.

De igual forma, es posible estimar el ángulo de incidencia de la partícula. Para este propósito es necesario disponer de varios fotodiodos. Si se consigue conocer la posición por la que pasa por varios fotodiodos en principio se podría reconstruir el ángulo de incidencia. Se incidirá en este punto más adelante.

2.1.2. CCD

Los dispositivos CCD (*charge-coupled device*) son dispositivos formados por varios arrays de fotodetectores. Estos suelen ser una modificación de los fotodiodos, los denominados *pinned photodiodes*. Sin embargo, las consideraciones anteriores se mantienen. En los CCD, cada fotodiodo reacciona a la radiación de manera independiente para después, mediante un convertidor analógico-digital, convertir la carga generada en voltaje para posteriormente manejar esta información en un ordenador, obteniendo la imagen deseada.

Un esquema de la operación se presenta en la figura 2.2. No se profundiza más en el funcionamiento de este dispositivo ya que no es objeto de este trabajo. Para más información se puede consultar [14, págs 9-14] y [15, págs 4-13].

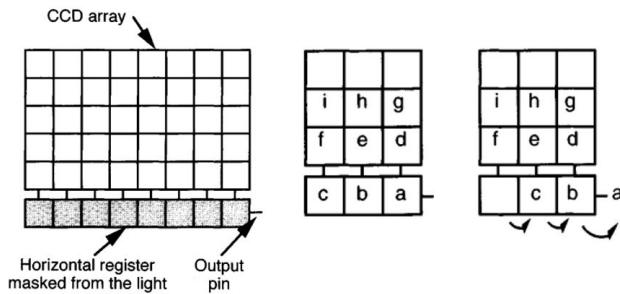


Figura 2.2: Izquierdo: Array de fotodetectores que captan la información. Derecha: Esquema de transmisión de información. Imagen extraída de [15, pág 10].

2.2. Dataset

En esta sección se presenta el dataset de imágenes que ha sido referencia en los apartados posteriores del trabajo. Este proviene del proyecto *DECO*.

2.2.1. Obtención

El proyecto *DECO* [16] recoge potenciales detecciones de rayos cósmicos usando una aplicación móvil ¹. Esta aplicación graba a través de la cámara, que debe estar cubierta para evitar ruido, imágenes de manera continua y aplica un filtro online para seleccionar los posibles candidatos a detección de partículas. Estos candidatos se almacenan, al igual que los metadatos correspondientes a la detección, con información como la fecha o las coordenadas del detector (ver figura 2.3).

TIME (UTC)	LOCATION	TYPE OF EVENT	EVENT ID	IMAGE	DEVICE MODEL
Jan 30, 2021 9:49 PM	lat: 52.34 lon: -1.57 Map It	Standard event, Spot	631482290	Image	Moto G

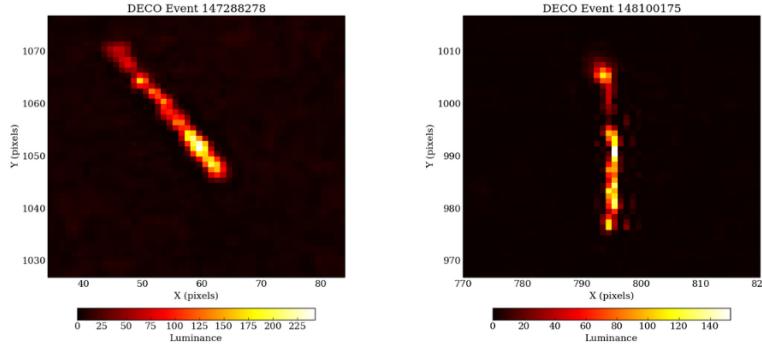
Figura 2.3: Imagen extraída de <https://www-old.wipac.wisc.edu/deco/data>.

¹Disponible en <https://www-old.wipac.wisc.edu/deco/app>.

Dentro de las detecciones, hay tres tipos de eventos producidos por partículas cargadas observados por *DECO*, los denominados *tracks*, *worms* y *spots* [16]. Los *tracks* se corresponden con líneas rectas de píxeles brillantes, generadas, probablemente, por partículas de alta energía (GeV) que no padecen prácticamente scattering. Generalmente están causados por muones a nivel del mar o incluso por rayos cósmicos primarios a altitudes superiores a varios kilómetros.

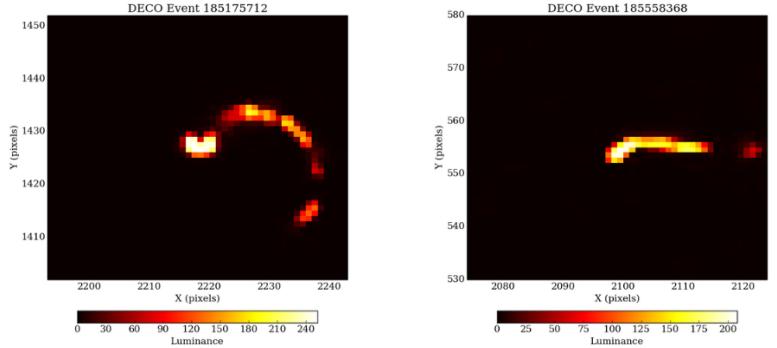
Los denominados *worms* reciben este nombre debido a sus trayectorias curvas. Este patrón se asocia a las trazas resultantes de electrones de bajas energías (MeV), producidos por decaimientos radioactivos por ejemplo, que sufren varios procesos de dispersión coulombiana dentro del sensor. Estos electrones se deben generar cerca del sensor para poder ser detectados.

Finalmente, los *spots* coinciden con pequeños grupos de píxeles brillantes. Estos pueden ser producidos por electrones de bajas energías, como por ejemplo los producidos por dispersión Compton² de rayos gamma causados por desintegraciones, que se absorben rápidamente. Los *spots* también pueden ser producidos por partículas alfa, muones o directamente por rayos cósmicos con una incidencia normal al sensor. Los tres tipos de eventos se pueden observar en las imágenes 2.4a, 2.4b y 2.4c. Es importante destacar que partículas alfa producidas por desintegraciones podrían causar también *tracks*, aunque se deberían generar muy cerca del sensor debido a su poca capacidad de penetración.

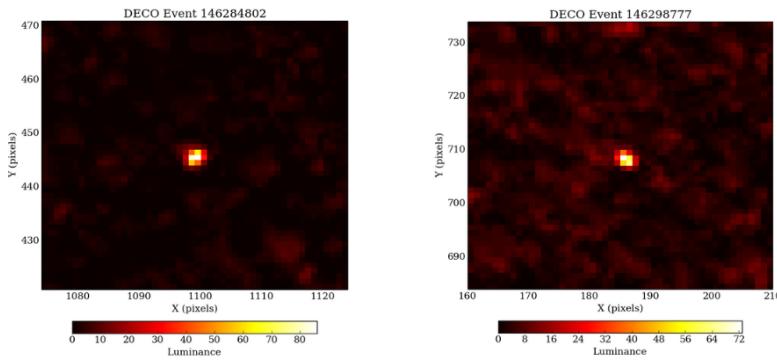


(a) *Tracks*.

²Este fenómeno consiste en la interacción entre un fotón y un electrón supuesto en reposo. Tras esta interacción tanto el electrón como el fotón se dispersan, donde se observa en este último un aumento de su longitud de onda.



(b) *Worms.*



(c) *Spots.*

Figura 2.4: Las tres imágenes anteriores han sido extraídas de <https://www-old.wipac.wisc.edu/deco/data>.

	Tracks	Worms	Spots
Muon	✓	✗	✓
Alfa	✓	✗	✓
Electrón	✗	✓	✓
CR (grandes altitudes)	✓	✗	✓

Cuadro 2.1: Tabla resumen de los posibles candidatos a *tracks*, *worms* y *spots*. El símbolo ✓ indica la posibilidad de que el candidato cause la detección de estudio. El símbolo ✗ denota la imposibilidad.

Analizando el conjunto de imágenes disponible, se observa que hay *tracks* más brillantes en un extremo que otro. Para partículas que se detienen esto podría indicar un pico de Bragg [17]. La curva de Bragg muestra cómo varía

el ratio de energía depositada en un material por una partícula en función de la profundidad. Esta curva exhibe un pico pronunciado justo antes de que la partícula se detenga por completo, el denominado pico de Bragg.

Mientras que la probabilidad de que un muon se detenga a su paso por el detector es casi ínfima, no es así en el caso de las partículas alfa [17]. Para el estudio de los *tracks* dejados por muones este hecho podría ayudar a descartar detecciones producidas por partículas alfa, eliminando aquellas en las que claramente una cola del *track* tenga mayor intensidad que la otra, aunque este proceso quedará fuera del alcance de nuestro trabajo.

2.2.2. Análisis

Como se ha observado, los detectores CCD obtienen detecciones muy variadas. El siguiente paso es tratar de dar la mejor caracterización de los eventos posibles. En este trabajo nos vamos a centrar en la detección de muones. En este contexto, se va a estudiar la distribución de longitudes, en píxeles, de los *tracks* causados por muones.

Para realizar este estudio se realiza la suposición de que la superficie de detección es un plano que reacciona a la ionización de manera uniforme con un espesor dado H . De igual forma, se denota por L la longitud de la marca dejada por un muon en la componente del plano del sensor. Ahora, la distribución del ángulo cenital de los rayos cósmicos incidentes, θ , es proporcional a $\cos^2 \theta$ y el flujo total integrado por unidad de energía y unidad de ángulo sólido es de ~ 1 partícula por minuto y por cm^2 [17]. Con lo anterior se tiene

$$dN \propto A_{\text{eff}}(\theta) \cos^2 \theta d\Omega.$$

Como $A_{\text{eff}}(\theta) = a \cos \theta$ y $d\Omega = \sin \theta d\theta d\varphi$ se llega a que

$$\begin{aligned} dN &\propto \cos^3 \theta \sin \theta d\theta, \\ dN &\propto d(\cos^4 \theta). \end{aligned}$$

Recordando que $\cos \theta = H/\sqrt{L^2 + H^2}$, debido a las definiciones de L y H , se llega a la expresión

$$d(\cos^4 \theta) = \frac{-2(L^2 + H^2)2LH^4}{(L^2 + H^2)^4} dL.$$

Y finalmente se obtiene

$$\frac{dN}{dL} = B \frac{LH^4}{(L^2 + H^2)^3}. \quad (2.1)$$

Esta última ecuación, 2.1, nos da una expresión para la distribución de longitudes, donde B es una constante de normalización. Esta ecuación da unos resultados aceptables que se asemejan a la realidad [17].

Estos resultados permiten estimar el espesor del sensor de un dispositivo de manera aceptable. Una vez conocido este dato es posible dar una predicción del ángulo azimutal de la partícula incidente conociendo la longitud en píxeles del *track*. Lo que junto con la orientación del dispositivo de detección permitiría calcular de manera completa la dirección de incidencia.

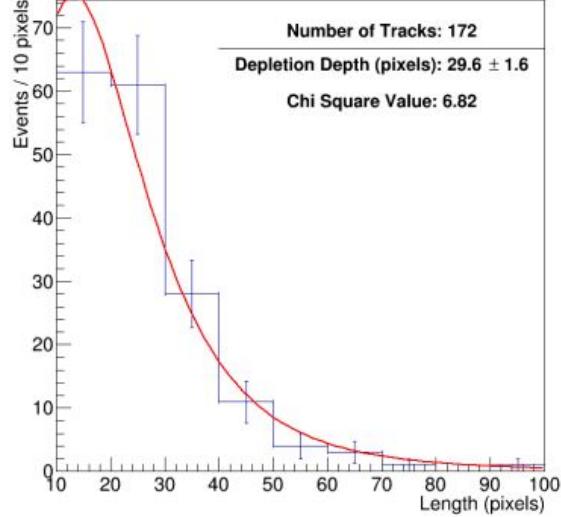


Figura 2.5: Ajuste de la distribución de longitudes considerando una distribución $\cos^2 \theta$ para el ángulo azimutal. Gráfica extraída de [17].

Capítulo 3

Modelo y entrenamiento

El siguiente capítulo se dedica a presentar los fundamentos de las redes neuronales empleadas en el desarrollo del dispositivo experimental de este trabajo. Además, introduce el dataset generado para el entrenamiento de nuestro modelo y analiza la validez de uso de este.

De esta manera, se presenta una introducción básica al concepto de red neuronal, explicando el funcionamiento básico de las redes neuronales densas. Posteriormente, se detallan las operaciones de convolución y *pooling* en el contexto del procesamiento de imágenes, bases de las redes neuronales convolucionales. Después, se presenta el modelo YOLO, un algoritmo de detección de objetos empleado en el dispositivo experimental final. Se detalla el proceso de generación del dataset así como diversos inconvenientes encontrados durante el mismo. Finalmente se comenta el entrenamiento y varias métricas que permiten valorar si el modelo empleado es adecuado para nuestro objetivo de detectar partículas.

3.1. Redes neuronales

Las redes neuronales son modelos computacionales inspirados en la estructura y funcionamiento del cerebro humano, compuestos por “neuronas” que procesan y transmiten información. Al igual que en nuestro cerebro, una neurona es relativamente inútil. Es la interconexión y el uso de un gran número de ellas las que hacen de las redes neuronales una gran herramienta para numerosas aplicaciones: detección de imágenes, procesamiento de lenguaje natural, regresiones... A nivel de clasificación, esta tecnología se puede englobar dentro del *machine learning*, incluido en el contexto de la inteligencia

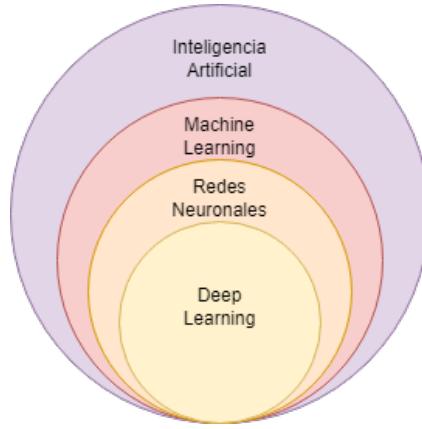


Figura 3.1: Varios campos de la inteligencia artificial y su relación. Imagen de elaboración propia.

artificial, como se puede ver en el diagrama de la figura 3.1¹. Dentro de las redes neuronales es importante destacar un subconjunto, el *Deep Learning*. Este tipo de redes neuronales destacan por tener muchas capas intermedias, no solo las de entrada y salida, por lo que son modelos más complejos.

Vamos a caracterizar ahora el funcionamiento básico de una neurona dentro de una red neuronal. En un caso general, una neurona tendrá un input dado por un vector $\vec{x} = (x_1, \dots, x_n)$. Además, cada input tiene asignado un peso, recogidos en el vector $\vec{w} = (w_1, \dots, w_n)$ y un bias global b . De esta manera, el output de una neurona es simplemente

$$\vec{x} \cdot \vec{w} + b = b + \sum_{i=1}^n x_i w_i,$$

que en esencia es una aplicación lineal. Sin embargo, este output se debe pasar por una función adicional, la denominada función de activación. Esta se encarga de modificar el output de una neurona de manera adecuada para que el valor final que se pase a la siguiente neurona sea adecuado. En este proceso se emplean una gran variedad de funciones según el objetivo que se persiga. Un esquema de este mecanismo se recoge en la imagen 3.2.

Generalmente, la estructura de una red neuronal densa tiene la forma de la imagen 3.3. Aquí se observa la interconexión entre neuronas. En general, una red puede tener decenas e incluso cientos de capas ocultas y cada una con una extensión variada.

¹Esta imagen, al igual que todas las de elaboración propia, ha sido realizada en la web draw.io.

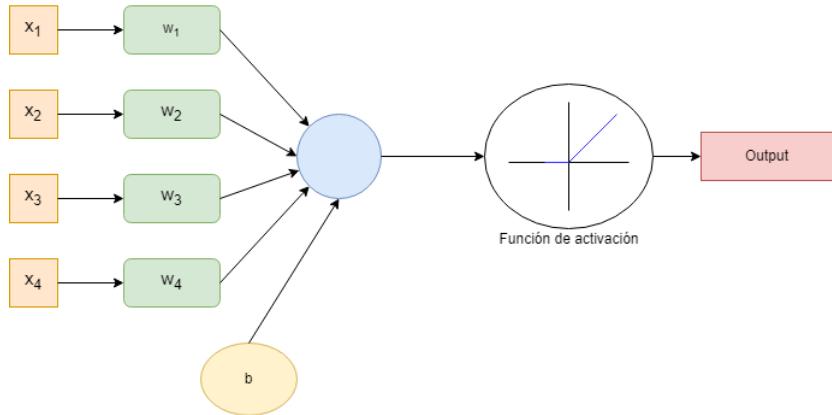


Figura 3.2: Funcionamiento de una neurona. Imagen de elaboración propia.

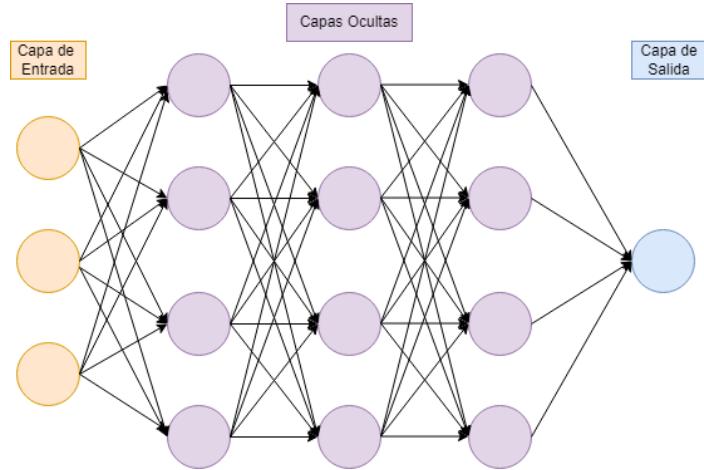


Figura 3.3: Red neuronal densa. Imagen de elaboración propia.

El uso de este tipo de modelos está justificado por el *Teorema de aproximación universal* [18], [19]. Este resultado nos dice que es posible aproximar cualquier función continua en un conjunto compacto por una red neuronal densa. Es decir, que el conjunto de las redes neuronales es denso en el conjunto de las funciones continuas en un conjunto compacto.

Para nuestro propósito de detectar partículas es necesario procesar imágenes, identificando las zonas donde se encuentran las detecciones y clasificándolas en *tracks*, *worms* o *spots*. Para estos procesos se emplea un tipo específico de redes neuronales. Es importante conocer que una imagen almacenada no es más que una matriz de píxeles, donde cada uno toma un valor diferente. Se tendrán tantas matrices como canales de colores tenga la imagen, típicamente tres. En este contexto juegan un papel fundamental dos tipos de operaciones: la convolución y el *pooling*.

3.1.1. Convolución

Esta operación permite extraer información y características relevantes de una imagen. En esta operación se hace uso de una matriz denominada *kernel*, generalmente de tamaño 3×3 o 5×5 , aunque puede tomar dimensiones arbitrarias $n \times n$. De esta manera, para cada píxel de la imagen que no pertenezca a los bordes, se multiplica componente a componente el *kernel* por la matriz de píxeles cuyo centro es el píxel seleccionado y tiene las mismas dimensiones que el *kernel*. Posteriormente, se suman todos los valores determinados y se sustituye en la posición del píxel original el valor calculado. Para los píxeles cercanos a los bordes ocurre que no es posible realizar la operación anterior, ya que no es posible considerar una matriz centrada en dicho píxel de las dimensiones del *kernel* con los datos de la imagen. Sin embargo, es posible aplicar la técnica conocida como *padding*, la cual consiste en ampliar la imagen con valores 0 o 1 en los bordes hasta que se pueda realizar la convolución. Por tanto, tras realizar este proceso, se obtiene una matriz de píxeles de dimensiones menores a la original si no se aplica el *padding* o una matriz de tamaño idéntico si se aplica esta técnica. Esta operación queda representada en la imagen [3.4](#).

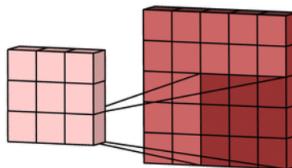


Figura 3.4: Proceso de convolución. Imagen extraída de [\[20\]](#).

Esta operación se usa en imágenes debido a que es capaz de respetar la distribución geométrica de los píxeles de la imagen, lo que implica que en última instancia se respetan los objetos. Esto no ocurre con las redes neuronales densas vistas en el apartado anterior.

3.1.2. Pooling

Esta operación, más sencilla que la anterior, simplemente divide la imagen en pequeñas regiones, generalmente celdillas 2×2 . Posteriormente, devuelve un valor por cuadrícula, típicamente el valor medio o el valor máximo. Esta operación permite reducir las dimensiones de la imagen a la mitad, reduciendo el coste computacional.

3.2. Modelo YOLO

El modelo empleado en el trabajo para predecir la presencia de muones en las detecciones es el denominado modelo YOLO, acrónimo de *You only look once* (solo miras una vez). Se trata de un algoritmo de detección de objetos en tiempo real. Este tipo de modelos son capaces de detectar la presencia de varios objetos en una imagen determinando las posiciones de estos, generalmente a través de *bounding boxes*. Estas últimas son simplemente rectángulos donde el modelo dice que se encuentra el objeto detectado con una cierta probabilidad. Las predicciones hechas por este modelo se pueden ver en la figura 3.5.

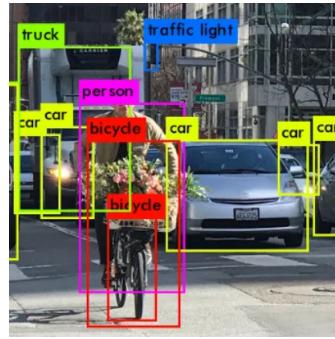


Figura 3.5: Predicción realizada con el modelo YOLO. Imagen extraída de [21].

YOLO fue publicado por primera vez en 2016 [22]. Es un modelo de *Deep Learning* sofisticado. Sin embargo, está basado en las operaciones que hemos comentado anteriormente de convolución y *pooling*. Su esquema se presenta en la figura 3.6. Cuenta con 24 capas convolucionales y 4 capas de *pooling*.

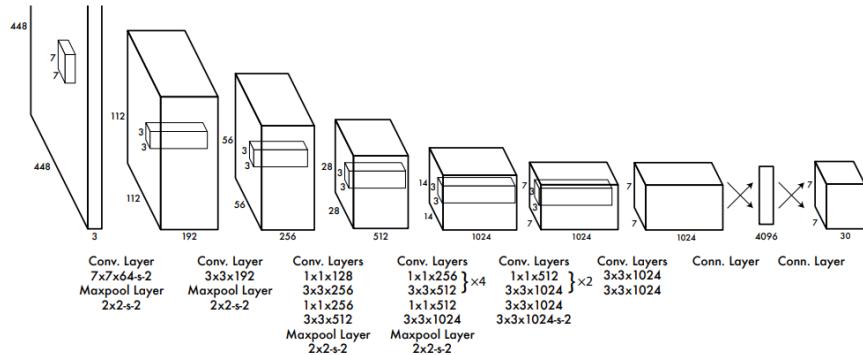


Figura 3.6: Esquema del modelo YOLO. Imagen extraída de [22].

El modelo divide la imagen original en una cuadrícula de dimensiones

$S \times S$. Para cada una, el modelo predice un número B de *bounding boxes*. Además, le asigna a cada una un centro, un ancho y un alto además de una confianza. Finalmente, a cada región se le da una probabilidad de que pertenezca a cada clase de objeto que el modelo es capaz de predecir, C . Así, la salida del modelo es una tupla de valores de dimensiones $S \times S \times (B \cdot 5 + C)$. No se profundiza más en este aspecto, ya que no es objeto del trabajo. Para más información se puede consultar la publicación original de este modelo [22].

3.3. Generación del dataset

El modelo YOLO está implementado en el lenguaje de programación *Python*. Para su entrenamiento ha sido necesario emplear un dataset personalizado. Este dataset se ha generado con las imágenes del proyecto DECO comentadas en el capítulo anterior. Sin embargo, las detecciones disponibles no se corresponden a las que uno realiza con su cámara, ya que están recortadas y con una escala de color artificial. Además, cuentan con un inconveniente aún mayor: cada imagen tiene una escala de valores independiente, como se puede comprobar en la imagen 3.7. En esta figura la primera detección alcanza una intensidad máxima de aproximadamente 150 (de un máximo de 255), mientras que el valor maximal de la segunda se sitúa en torno a los 230. Sin embargo, ambas tienen la misma escala de color.

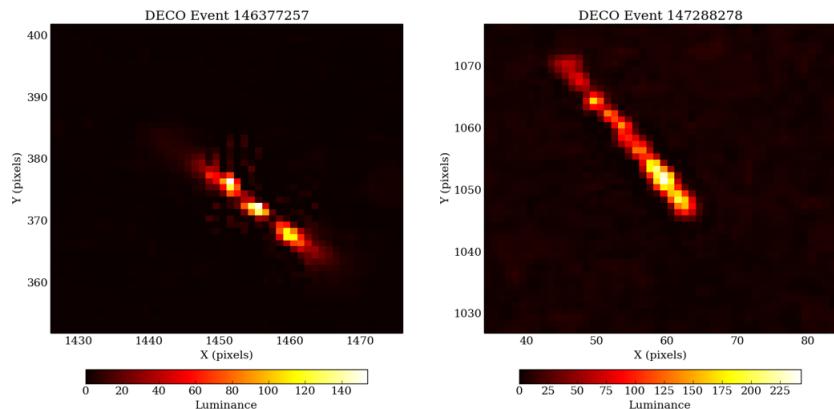


Figura 3.7: Cada detección tiene una escala diferente, alcanzando un valor máximo distinto aunque la escala de colores sea la misma.

Este problema impide utilizar las imágenes de manera directa en la creación de un dataset de entrenamiento. De esta manera, ha sido necesario, mediante visión por ordenador, calcular el valor máximo que toma cada imagen. Para

esto, se ha usado OCR (*Optical Character Recognition*), una técnica capaz de identificar los números presentes en la escala de color de cada imagen, para, posteriormente, hallar el valor máximo que aparece en dicha escala. Con esta técnica, se ha calculado un valor máximo de 154 para la primera detección de la imagen 3.7, mientras que para la segunda se ha obtenido un valor de 244.

Gracias a la técnica anterior, es posible pasar cada detección a escala de grises, pudiendo comparar, ahora sí, las intensidades de cada detección. En nuestro propósito de conseguir un dataset de entrenamiento, hay que conseguir unas imágenes similares a las que se pueden obtener en la práctica. Por tanto, las detecciones en escala de grises se han redimensionado a 50×50 píxeles, su tamaño original en la imagen de la que se han extraído. Este proceso se ilustra en la figura 3.8.

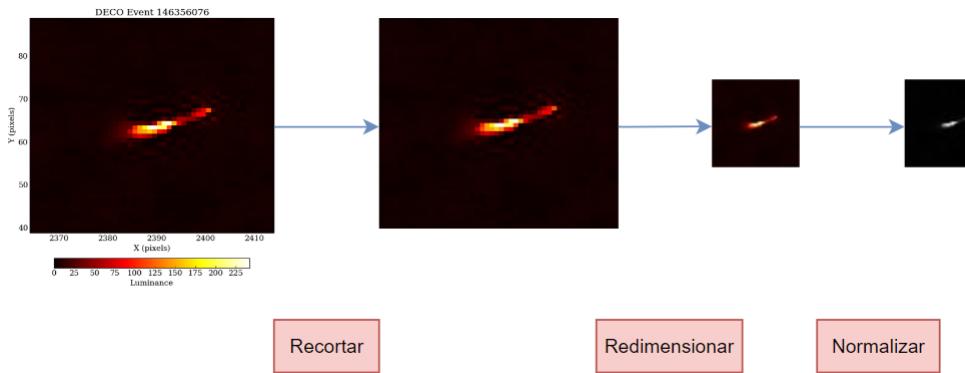


Figura 3.8: Procesamiento de la imágenes para conseguir general el dataset.

Finalmente, sobre una imagen de fondo negro con ruido, simulando lo que verá nuestra cámara que capturará las detecciones, se insertan varias en ubicaciones aleatorias. Con esta técnica, se han generado 1500 imágenes para entrenar y validar nuestro modelo. Además, todos los detalles del dataset de entrenamiento se han recogido en un archivo .csv con los parámetros recogidos en la tabla 3.1. De esta forma, la imagen 2 del dataset contiene dos detecciones, en este caso *tracks*, cada uno con las coordenadas especificadas por las coordenadas *x* e *y* de la tabla dadas en píxeles.

Es importante notar que en el contexto de las imágenes, el origen de coordenadas se suele tomar en la esquina superior izquierda, de tal manera que al moverse a la derecha se avanza en el eje *x* y hacia abajo en el *y*. Este convenio se ha empleado en este caso. Finalmente, una representación de esta

imagen número dos se encuentra en la figura 3.9, donde se ha aumentado la intensidad de las detecciones para facilitar su visualización.

Image ID	Image Path	Muon Type	x	y	w	h
1	output_1.png	spot	77	324	50	50
1	output_1.png	track	951	71	50	50
2	output_2.png	track	756	484	50	50
2	output_2.png	track	603	85	50	50
3	output_3.png	worm	361	688	50	50
4	output_4.png	worm	479	383	50	50

Cuadro 3.1: Datos recogidos en el .csv.

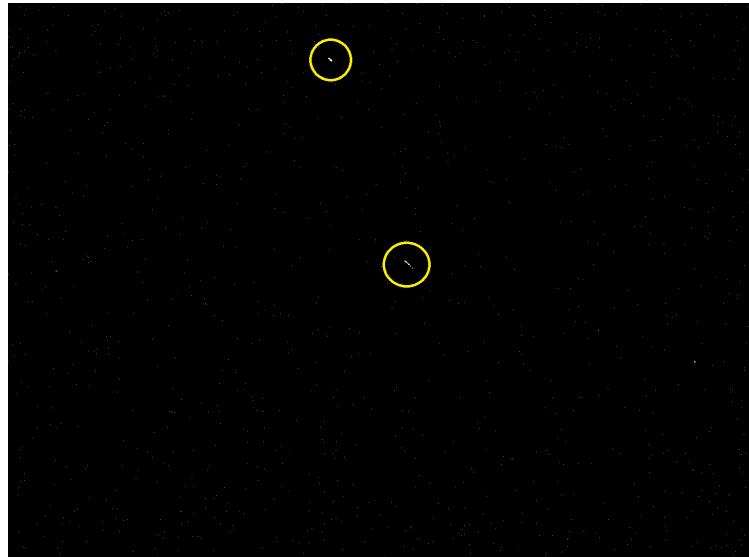


Figura 3.9: Imagen 2 del dataset.

3.4. Entrenamiento

El entrenamiento de nuestro modelo YOLO ha sido realizado en el entorno *Google Colab*. Esta herramienta permite acceder a hardware de gran potencia como GPUs o TPUs de manera gratuita por tiempo limitado. Como el modelo empleado es bastante pesado a nivel computacional, *Google Colab* es muy adecuado para nuestros propósitos. Para destacar su capacidad, el entrenamiento del modelo tardaba del orden de días en una máquina local y pasó al orden de pocas horas en este entorno de ejecución.

El dataset empleado para el entrenamiento contaba con casi mil detecciones de cada tipo. Por tanto, es un conjunto de datos equilibrado, lo que lo hace adecuado para nuestro objetivo. Para medir si nuestro entrenamiento ha sido óptimo, es posible analizar varias métricas. Para este tipo de modelos suele ser útil analizar dos: precisión y *recall*. La primera hace referencia a la capacidad del modelo para predecir objetos correctamente. Responde a la pregunta de qué proporción de detecciones positivas han sido correctas. La segunda métrica mide si el modelo es capaz de encontrar todas las cajas delimitadoras reales. Esta relacionada con la proporción de detecciones reales que el modelo es capaz de detectar. La definición matemática de ambas viene dada por las siguientes expresiones:

$$\text{Precisión} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

donde TP denota las detecciones correctas, FP las detecciones incorrectas y FN son las detecciones reales que el modelo no ha predicho. En un modelo adecuado ambos parámetros toman un valor cercano a uno. Esto indica que todas las detecciones son correctas y que a su vez no se le escapa ninguna detección.

Los valores de estas métricas durante el entrenamiento se pueden ver en la figura 3.10. Como se puede observar, al finalizar este proceso estos valores son cercanos a uno, lo que indica que nuestro modelo final realiza buenas predicciones sobre las imágenes de entrenamiento.

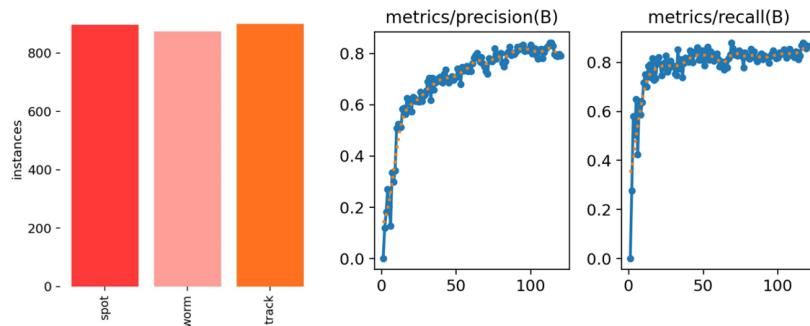


Figura 3.10: La primera gráfica refleja el número de cada tipo de detecciones incluidas en el dataset de entrenamiento. La segunda y tercera gráfica muestran el valor de la precisión y el *recall* respectivamente en cada época del entrenamiento. Al finalizar toman valores cercanos a uno.

Finalmente, para analizar la validez de nuestro modelo se presenta la matriz de confusión de la imagen 3.11. Esta herramienta permite evaluar el desempeño de un modelo sobre un conjunto de datos. Dado un dataset de

validación y fijado un tipo de detección, cada columna representa como ha clasificado el modelo esa clase. Por ejemplo, la columna asociada a *spots* nos dice que el 87% han sido clasificados en su clase, mientras que un 9% se han clasificado como *worms*. De esta manera los elementos de cada columna deben sumar uno.

Interpretando la matriz para nuestro caso es posible obtener varias conclusiones. En primer lugar, vemos que las diagonales toman un valor elevado para los *spots* y los *tracks* y algo más bajo para los *worms*. Esto nos indica que el modelo clasifica de manera adecuada los dos primeros tipos de detecciones y tiene un rendimiento inferior, aunque aún así adecuado, para los *worms*. Sin embargo, nuestro modelo puede tener alguna dificultad con este último tipo de detecciones, ya que puede confundir los *worms* con el fondo. Esto queda representado por el elemento de matriz (2, 4), el cual toma un valor ligeramente alto. Aún así, con esta métrica podemos concluir que el modelo es adecuado para nuestros propósitos.

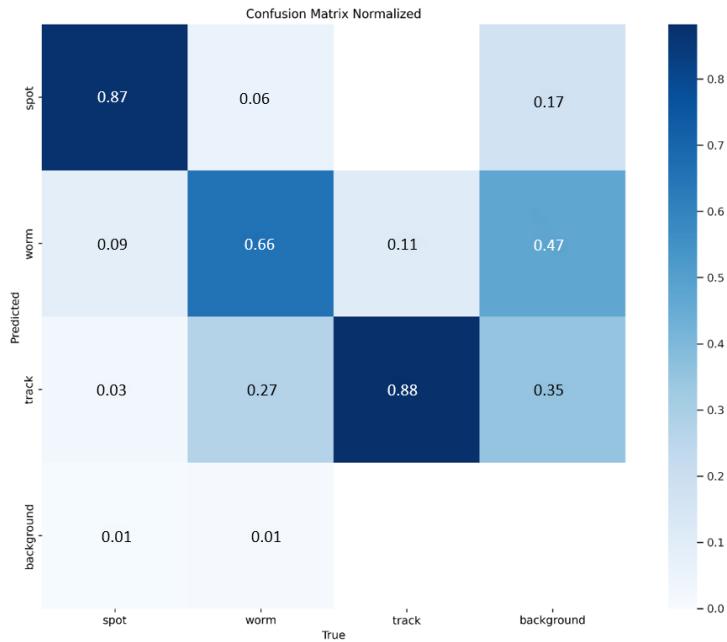


Figura 3.11: Matriz de confusión.

Para finalizar, se presenta la predicción del modelo para una imagen del conjunto de validación del entrenamiento en la figura 3.12. El número que aparece en la imagen de la derecha se corresponde a la probabilidad que asigna el modelo al tipo de detección predicha. Como se puede observar, en este caso todas las predicciones son correctas para nuestra imagen.

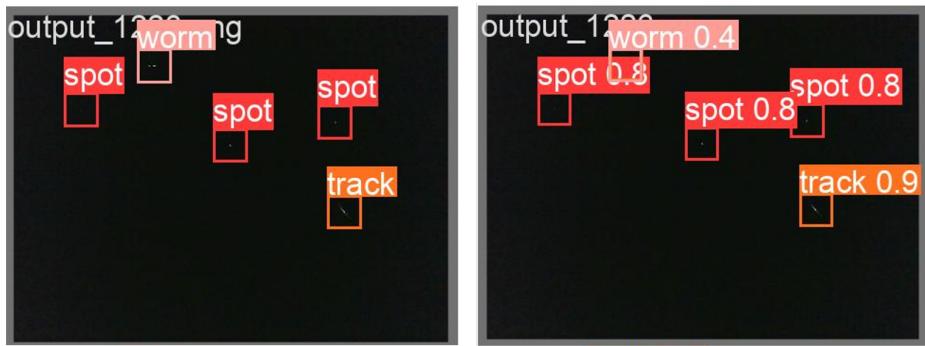


Figura 3.12: Inferencia del modelo sobre una imagen del conjunto de validación.

Capítulo 4

Dispositivo Experimental

En nuestro objetivo de crear un dispositivo capaz de captar y clasificar detecciones de rayos cósmicos de manera automática, es necesario tener un aparato donde se procesen las imágenes captadas por nuestra cámara: la Jetson Orin NX con 16 GB de RAM. Así, en este capítulo se detalla el dispositivo experimental usado y su configuración. Además, se presenta la tarjeta donde inicialmente se iba a desplegar el modelo, detallando las razones por las que fue descartada.

4.1. Jetson Orin NX

Este dispositivo, desarrollado por la compañía *Nvidia*, permite implementar el modelo YOLO en su interior. Por tanto, conectando una cámara a esta tarjeta, es posible captar y clasificar detecciones en tiempo real. Una imagen de la Jetson se puede encontrar en la imagen 4.1. Tiene unas dimensiones reducidas, lo que permite realizar experimentaciones en diferentes localizaciones.

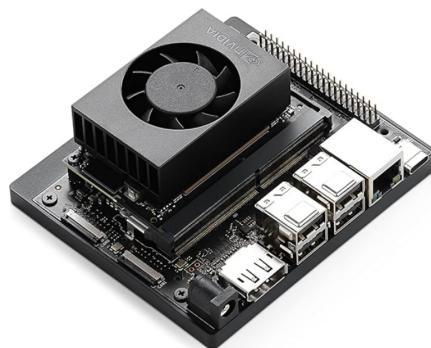


Figura 4.1: Jetson Orin NX.

Este dispositivo funciona con Ubuntu en su interior, en particular la versión 22.04.4 LTS. En cuanto a especificaciones, es importante destacar que este dispositivo cuenta con una *GPU* (unidad de procesamiento gráfico). Este elemento es fundamental en el tratamiento de información ya que permite la paralelización. Este proceso acelera en gran medida el procesamiento de datos. Para entender este método es importante comprender como funciona la computación habitual. En esencia, cuando se tiene una tarea esta se divide en unidades más pequeñas y se procesan una detrás de otra. Sin embargo, con la paralelización estas unidades se procesan a la vez, lo que permite ahorrar tiempo de manera drástica. Un paralelismo más cercano puede ser cuando se hace la cola en una caja del supermercado. Para aligerar el tiempo de espera se abren múltiples cajas para “procesar” a la gente de manera más eficiente.

4.2. Cámara

La cámara empleada para tomar las detecciones ha sido una webcam de la marca *Trust*. Se puede ver en la figura 4.2. Tiene un tamaño reducido, en línea con la idea de hacer nuestro dispositivo portable. Para realizar las posibles detecciones es necesario cubrir el sensor para que no detecte luz.



Figura 4.2: Cámara empleada para realizar las detecciones.

Este dispositivo ha generado varios problemas durante la experimentación. Principalmente, se han registrado falsas detecciones en una región de la cámara, como se puede ver en la imagen 4.3. Esto se debe a que un conjunto específico de píxeles tomaban valores anormalmente altos con relativa frecuencia, lo que causaba problemas en los resultados. Si en el resto de la imagen se obtienen detecciones cada varias horas, en esta región se obtienen cada pocos minutos, lo que justifica que no son detecciones reales. Se desconoce la causa de este inconveniente aunque se cree que es debido a un defecto en el propio sensor. La solución planteada fue dejar de considerar las últimas columnas de los fotogramas captados por la cámara. De esta manera,

cuando una imagen es analizada para comprobar si contiene una detección esta región es excluida.

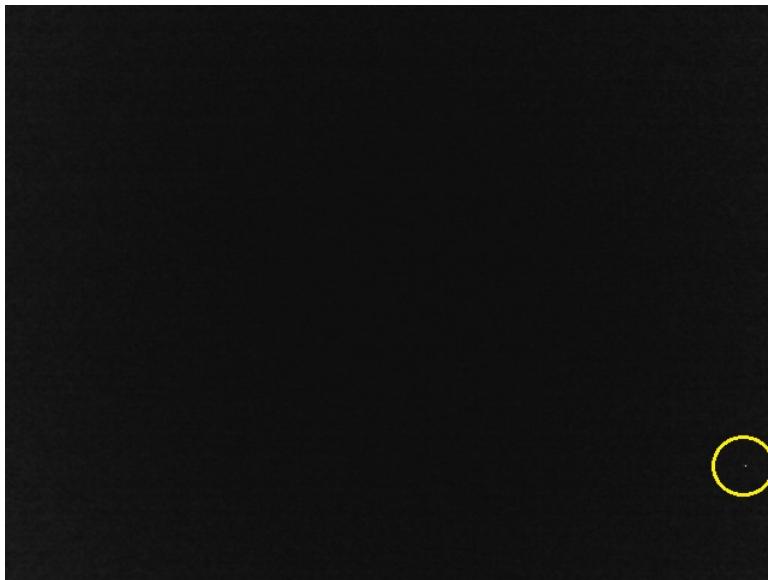


Figura 4.3: Región en la que aparecen falsas detecciones.

4.3. Manejo

Una vez se tiene el modelo entrenado, es necesario implementarlo en la Jetson. Para conectarse desde un ordenador al dispositivo se ha usado el programa *WinScp*. Para poder usarlo es necesario establecer una conexión SSH conociendo las IPs de ambos dispositivos. Esta herramienta permite la transferencia de archivos desde nuestro ordenador al dispositivo. El script para capturar detecciones se encuentra en el apéndice A al final del trabajo. Este se debe ejecutar desde la terminal con el comando

```
python detect_and_infer.py
```

El programa anterior busca detecciones durante una hora, aunque es posible modificarlo para un tiempo arbitrario. Cuando encuentra una detección, esta se almacena en una carpeta llamada *detections* y se pasa por el modelo para que clasifique la detección encontrada. Una vez se realiza la inferencia, la imagen con las *bounding boxes* correspondientes se guarda en la carpeta *detections_inferred*.

4.4. Dispositivo alternativo

Antes de emplear la *Jetson* para desplegar nuestro modelo, por requerimientos del proyecto se intentó hacer uso de otro dispositivo: la tarjeta Ultra 96. Esta se puede ver en la figura 4.4 y está diseñada por la compañía *Avnet*. Este dispositivo tiene unas dimensiones incluso más pequeñas que la *Jetson*, lo que la hacía ideal para crear un dispositivo portable. Sin embargo, su funcionamiento interno es más complejo. Para poder ejecutar el modelo, este se debía simplificar, pasando los pesos a números enteros. Esto hace que el modelo fuese más ligero y eficiente a costa de bajar ligeramente la precisión. Para esta finalidad se empleó la herramienta *Vitis-AI*. En particular se siguió el tutorial que se puede encontrar en su [GitHub](#) oficial.

Sin embargo, el uso de este dispositivo suponía un inconveniente que no se pudo solucionar: era necesario postprocesar la salida del modelo para poder escoger las *bounding boxes* finales. Este proceso no pudo ser llevado a cabo debido a su complejidad. Este contratiempo causó varias semanas de retraso en el desarrollo del dispositivo experimental. Por otro lado, usando la *Jetson* el problema desaparece, ya que es capaz de usar el modelo original sin simplificar los pesos, en el que no es necesario este postprocesamiento.



Figura 4.4: Tarjeta Ultra 96.

Capítulo 5

Resultados y conclusiones

En este capítulo se presentan los resultados obtenidos con el dispositivo experimental diseñado a lo largo del trabajo. Se discute si estos se corresponden con lo esperado y se plantean varias líneas de trabajo futuras.

5.1. Resultados obtenidos

Una vez preparado el dispositivo experimental se ha procedido a experimentar con él. En total se han realizado cerca de 100 horas de grabación en busca de detecciones, obteniendo los resultados de la tabla 5.1. Se comprueba, por tanto, que el fenómeno de estudio no es muy frecuente. El modelo ha clasificado de manera correcta todas las detecciones encontradas, asignando una gran confianza en cada resultado. De esta manera, cada *spot* registrado tiene asignada una probabilidad de serlo superior al 70 %. Por su parte, los *tracks* tienen una probabilidad siempre superior al 55 %.

	Spots	Worms	Tracks
Detecciones	22	0	8

Cuadro 5.1: Número de detecciones por categoría

Como se puede ver, solo se han obtenido *spots* y *tracks* en nuestra experimentación. Una muestra de las detecciones se presenta en las imágenes de la figura 5.1. En la parte superior se encuentran un *spot* y un *track* obtenidos durante la experimentación respectivamente. A su vez, en la parte inferior se presentan las detecciones inferidas por el modelo con la probabilidad asignada a cada una. En el apéndice B se pueden encontrar todas las detecciones obtenidas inferidas.



(a) Detecciones obtenidas.



(b) Detecciones inferidas.

Figura 5.1

5.2. Conclusiones

A la vista de los resultados presentados anteriormente podemos extraer varias conclusiones. En primer lugar, se puede comprobar que el dispositivo experimental diseñado cumple con su cometido de manera adecuada. Ha permitido registrar las detecciones objetivo del trabajo de manera satisfactoria clasificándolas con gran precisión.

Por otro lado, no se han observado *worms*. Como se vio en el capítulo dos, estos eran producidos por electrones de bajas energías (MeV). Su ausencia sugiere que el resto de detecciones pueden venir únicamente de muones, y por tanto de las cascadas de rayos cósmicos (ver tabla 2.1). Se descartan las partículas alfa debido a que no se ha estado cerca de ningún emisor, al menos de manera consciente. Además, nuestro modelo tenía ligeros problemas identificando este tipo de detecciones. Al no ser la clase predominante, esto confirma que el tipo de modelo para realizar las clasificaciones es adecuado.

Sin embargo, uno de los puntos negativos del trabajo ha sido la poca cantidad de detecciones obtenidas. Este problema viene causado por el poco tiempo del que se ha dispuesto para captar imágenes junto con la baja frecuencia del evento de estudio. Aún así, las detecciones registradas son precisas y el modelo las clasifica con alta probabilidad en su clase adecuada.

En resumen, se ha obtenido un dispositivo experimental capaz de detectar y clasificar detecciones de muones, cumpliendo así el objetivo planteado al inicio del trabajo. Gracias a su pequeño tamaño, este aparato puede instalarse en diversas ubicaciones o integrarse en una red de detectores. Esto podría ayudar a estudiar los rayos cósmicos permitiendo detectar elementos de las cascadas atmosféricas de manera eficaz en diversas localizaciones.

5.3. Líneas de trabajo futuras

Tras la realización del trabajo se proponen varias ideas para continuar con el desarrollo de este proyecto.

- (I) En primer lugar, aumentando en gran medida el número de detecciones sería posible aplicar el modelo propuesto al final del capítulo dos para caracterizar el espesor de nuestro sensor. Esto permitiría calcular la dirección del muon original.
- (II) Por otra parte, si se dispusiera de varios dispositivos ubicados en diferentes lugares cercanos se podría analizar la correlación entre eventos. Esto sería posible revisando si se obtienen detecciones en tiempos próximos en diferentes ubicaciones.

Apéndice A

Código

A continuación se presenta el código empleado para realizar las detecciones con la cámara.

```
#Importamos primero las librerias necesarias

import cv2
import os
from datetime import datetime
from ultralytics import YOLO

#Cargamos el modelo
model = YOLO("yolo/best.pt")

# Inicializa la camara 0
cap = cv2.VideoCapture(0)

# Definir el tiempo de duracion (en segundos)
duracion_segundos = 3600

# Obtener el tiempo de inicio
tiempo_inicio = datetime.now()

# Carpeta donde se guardaran las detecciones
carpeta_detecciones = "detections"
# Carpeta donde se guardaran las inferencias
carpeta_inferencia = "detections_inferred"

# Contador de imagenes almacenadas
contador_imagenes = 0
limite_imagenes = 100

try:
    while True:
        # Comprobar el tiempo transcurrido
```

```

tiempo_actual = datetime.now()
tiempo_transcurrido = tiempo_actual - tiempo_inicio
if tiempo_transcurrido.total_seconds() >=
    duracion_segundos:
    print("Tiempo de captura alcanzado. Saliendo ...")
    )
break

# Lee el frame
ret, frame = cap.read()
if not ret:
    print("No se puede recibir frame. Saliendo ...")
    break

# Definir la region de interes (ROI) para incluir
# solo hasta la columna 550 y eliminar falsas
# detecciones
roi = frame[:, :550]

# Convertir la ROI a escala de grises
gray_roi = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)

# Comprueba si algun pixel supera el umbral de 50 en
# la ROI
if cv2.countNonZero(cv2.threshold(gray_roi, 50, 255,
cv2.THRESH_BINARY)[1]) > 0:
    # Genera un nombre de archivo basado en la fecha
    timestamp = tiempo_actual.strftime("%Y%m%d_%H%M%
S_%f")
    file_name = f"{carpeta_detecciones}/detection_{
timestamp}.jpg"

    contador_imagenes += 1
    if contador_imagenes >= limite_imagenes:
        print("Limite de imagenes alcanzado. Saliendo
        ...")
        break

    # Guarda la imagen
    cv2.imwrite(file_name, frame)
    print(f"Imagen con deteccion guardada: {file_name
}")

    # Reliza la inferencia
    results = model([file_name])  # devuelve una
        lista
    results[0].save(filename=f"{carpeta_inferencia}/
detection_{timestamp}.jpg")

```

```
# Muestra el frame
cv2.imshow("Frame", frame)

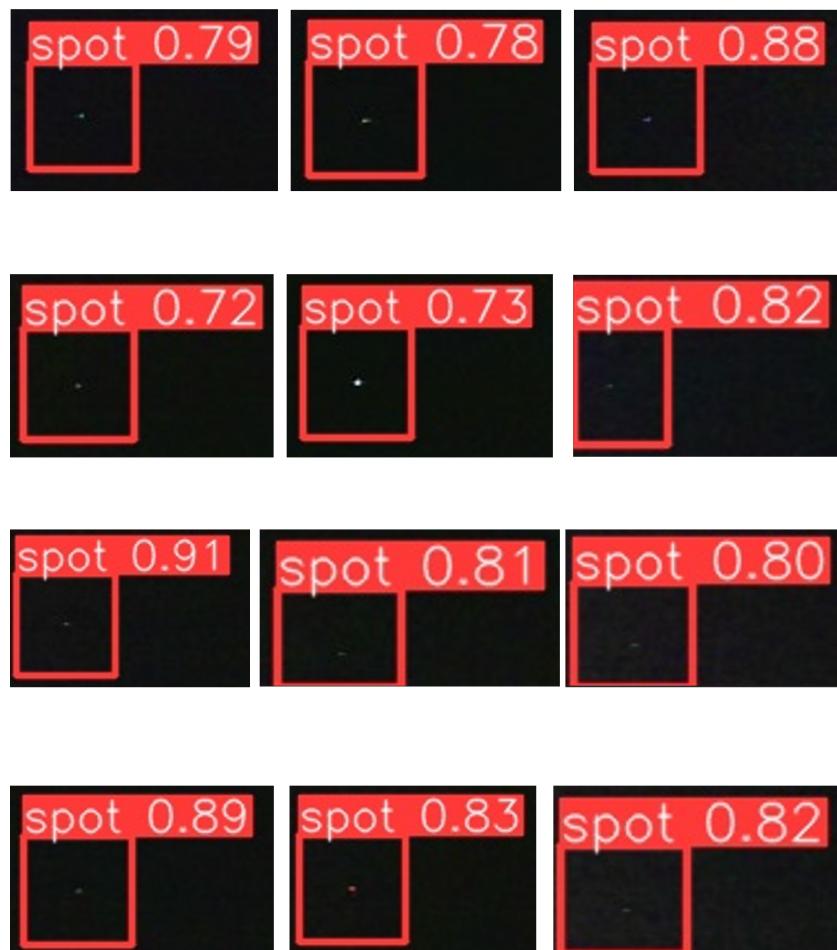
# Sale del bucle si se presiona la tecla 'q'
if cv2.waitKey(1) & 0xFF == ord('q'):
    print("Tecla 'q' presionada. Saliendo . . .")
    break

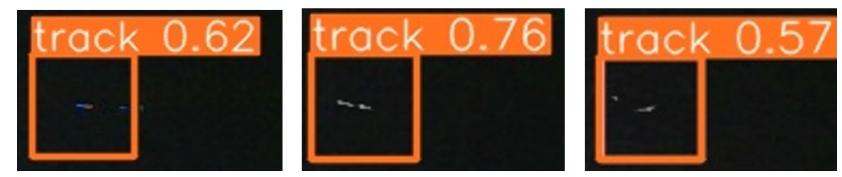
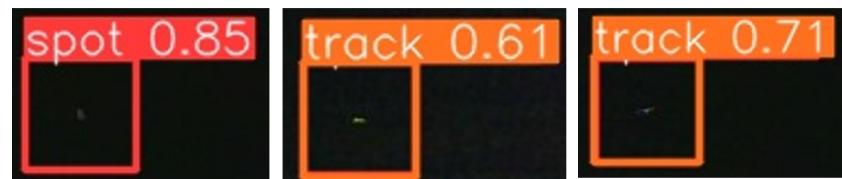
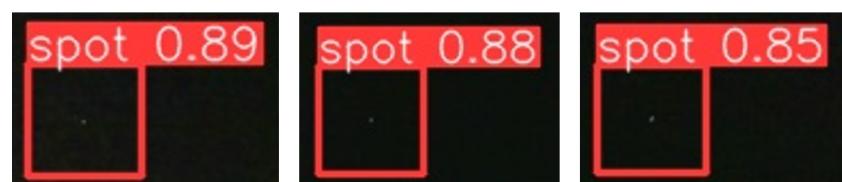
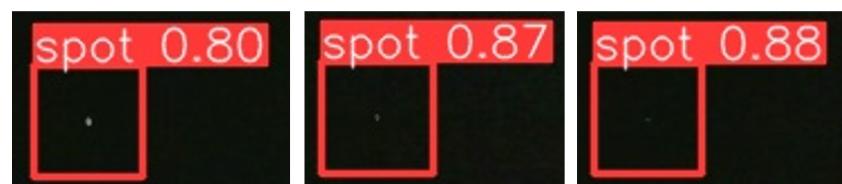
finally:
    # Libera la camara y cierra todas las ventanas
    cap.release()
    cv2.destroyAllWindows()
```

Apéndice B

Detecciones

En esta sección se presentan todas las detecciones obtenidas durante la experimentación.





Bibliografía

- [1] CERN. Victor Hess discovers cosmic rays. <https://timeline.web.cern.ch/victor-hess-discovers-cosmic-rays-0>. [Online; Consultado 02/2024].
- [2] Zbigniew Szadkowski and Zbigniew Piotr Szadkowski. *Cosmic Rays*. IntechOpen, 2018.
- [3] James W. Cronin. Cosmic rays: the most energetic particles in the universe. 71:S165–S172, 1999. <https://link.aps.org/doi/10.1103/RevModPhys.71.S165>.
- [4] Frank G. Schröder. *Instruments and Methods for the Radio Detection of High Energy Cosmic Rays*. Springer Theses, Berlin, Heidelberg, 1st edition, 2012.
- [5] F. Combes. *Active galactic nuclei*. John Wiley and Sons Inc, Hoboken, New Jersey, 2022.
- [6] Arman Tursunov et al. Supermassive black holes as possible sources of ultrahigh-energy cosmic rays. 895(1):14, 2020. <https://dx.doi.org/10.3847/1538-4357/ab8ae9>.
- [7] Naresh Dadhich. Magnetic penrose process and blanford-zanejk mechanism: A clarification. 2012. <https://doi.org/10.48550/arXiv.1210.1041>.
- [8] Arman Tursunov and Naresh Dadhich. Fifty years of energy extraction from rotating black hole: Revisiting magnetic penrose process. 2019. <http://dx.doi.org/10.3390/universe5050125>.
- [9] J. Matthews. A heitler model of extensive air showers. *Astroparticle Physics*, 22(5):387–397, 2005. <https://www.sciencedirect.com/science/article/pii/S0927650504001598>.

- [10] Lulu Liu. The speed and lifetime of cosmic ray muons. <https://web.mit.edu/lululiu/Public/pixx/not-pixx/muons.pdf>, 2007. [Online; Consultado 03/2024].
- [11] Piotr Homola et al. Cosmic-ray extremely distributed observatory. *Symmetry*, 12(11), 2020. <http://dx.doi.org/10.3390/sym12111835>.
- [12] Justin Vandenbroucke et al. Detecting particles with cell phones: the Distributed Electronic Cosmic-ray Observatory . *PoS*, ICRC2015:691, 2016. <https://doi.org/10.22323/1.236.0691>.
- [13] Hamamatsu. Si detectors for high energy particles. https://www.hamamatsu.com/content/dam/hamamatsu-photonics/sites/documents/99_SALES_LIBRARY/ssd/high_energy_kspd9002e.pdf. [Online; Consultado 03/2024].
- [14] Steve B. Howell. *Handbook of CCD astronomy*. Cambridge observing handbooks for research astronomers; 5. Cambridge University Press, Cambridge, second edition, 2006.
- [15] Patrick Martinez and Alain Klotz. *A practical guide to CCD astronomy*. Practical Astronomy handbooks; 8. Cambridge University Press, Cambridge, 1998.
- [16] Matthew Meehan et al. The particle detector in your pocket: The distributed electronic cosmic-ray observatory. 2017. <https://doi.org/10.48550/arXiv.1708.01281>.
- [17] Justin Vandenbroucke et al. Measurement of cosmic-ray muons with the distributed electronic cosmic-ray observatory, a network of smartphones. *Journal of Instrumentation*, 11, 2016. <http://dx.doi.org/10.1088/1748-0221/11/04/P04019>.
- [18] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width, 2017. <https://doi.org/10.48550/arXiv.1709.02540>.
- [19] Universal approximation theorem. https://en.wikipedia.org/wiki/Universal_approximation_theorem. [Online; Consultado 06/2024].
- [20] Irhum Shafkat. Intuitively Understanding Convolutions for Deep Learning. <https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1>. [Online; Consultado 05/2024].

- [21] Ani Aggarwal. YOLO Explained. <https://medium.com/analytics-vidhya/yolo-explained-5b6f4564f31>. [Online; Consultado 05/2024].
- [22] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. <https://doi.org/10.48550/arXiv.1506.02640>, 2016.