



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO**  
**POSGRADO EN CIENCIAS FÍSICAS**

**DESARROLLO DE ALGORITMOS DE IDENTIFICACIÓN DE TRAYECTORIAS  
DE MUONES EN EL OBSERVATORIO HAWC CON REDES NEURONALES.**

**TESIS**  
**QUE PARA OPTAR POR EL GRADO DE:**  
**MAESTRÍA EN CIENCIAS FÍSICAS**

**PRESENTA:**  
**JOSÉ ROBERTO ANGELES CAMACHO**

**DR. HERMES LEÓN VARGAS**  
**INSTITUTO DE FÍSICA, UNAM**

**DR. ANDRÉS SANDOVAL ESPINOSA**  
**INSTITUTO DE FÍSICA, UNAM**

**DRA. BEATRIZ ELIZABETH FUENTES MADARIAGA**  
**FACULTAD DE CIENCIAS, UNAM**

**Ciudad Universitaria, CD. MX, Marzo 2021**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

# Agradecimientos

Me gustaría agradecer a toda mi familia. En especial a mi madre Candelaria Camacho López que siempre me ha apoyado, a mi padre Roberto Angeles Hernández y a mi hermano Jesús Emmanuel Angeles Camacho. También me gustaría mencionar el apoyo de mis tíos Carmen Camacho López y Alejandro Nieto que siempre han estado conmigo.

A todos mis amigos, en especial a Miguel Ángel Esteban Reyes quien me ayudo en una parte de la identificación manual de las imágenes de trazas implementadas en este proyecto.

Además quisiera agradecer a mi asesor Hermes León Vargas por su paciencia y por la oportunidad de trabajar con él desde el servicio social. A la Dra. Beatriz Elizabeth Fuentes Madariaga y al Dr. Andrés Sandoval Espinosa que aceptaron ser miembros de mi comité tutor y supervisaron mi avance durante este posgrado. También agradezco a todos los sinodales por leer este trabajo y hacer las correcciones pertinentes.

Este trabajo fue realizado gracias al apoyo de los proyectos CONACyT y Beca PRIDIF-2021.

# Resumen

El observatorio HAWC (*High Altitud Water Cherenkov*) es un instrumento de investigación conformado por un arreglo de 300 detectores ubicado en una de las laderas del volcán Sierra Negra en el estado de Puebla. Este arreglo detecta cascadas atmosféricas producidas por rayos cósmicos y rayos gamma. Actualmente hay una línea de investigación alternativa que plantea utilizar a este observatorio como un detector indirecto de neutrinos. Esta línea de investigación busca señales de partículas secundarias producidas por la interacción entre neutrinos ultra energéticos y los núcleos que conforman al Pico de Orizaba. Cabe señalar que los muones generados en cascadas atmosféricas en la dirección de esta montaña representan la principal fuente de ruido para esta tarea. Una de las características tanto de estas señales de ruido como de las generadas por un neutrino, es que deben activar una línea de detectores del observatorio HAWC; además su trayectoria debe ser cuasi horizontal debido a que pasaron por el Pico de Orizaba. Por tal motivo nos vamos referir a este tipo de señales como trazas horizontales. En este trabajo se presenta una nueva técnica para discriminar entre eventos de trazas horizontales y cascadas atmosféricas, dicha técnica se basa en el reconocimiento de imágenes utilizando una red neuronal convolucional artificial. El estudio fue realizado con dos tipos de redes neuronales, una enfocada en la clasificación de imágenes y otra en detección de objetos. Después de analizar un total de 118476 eventos, los resultados que se obtuvieron indican que con ambos modelos se logra aumentar por un orden de magnitud el número de trazas que se identifican al compararlo con el algoritmo que actualmente se utiliza para esta tarea. Cabe mencionar que la red enfocada en la clasificación de imágenes obtuvo una tasa de falsos positivos de 2.8% del total de trazas que identifico. Por otro lado la red enfocada en la detección de objeto tuvo un porcentaje nulo de falsos positivos. La eficiencia de la clasificación (porcentaje de aciertos respecto al total de eventos probados) fue del 85.2% para la red enfocada a clasificación de imágenes y del 85.0% para la red enfocada a detección de objetos.

# Introducción

El observatorio HAWC es un instrumento de investigación que se dedica a estudiar al Universo en altas energías (entre 100 GeV y 100 TeV) [1]. Este observatorio se ubica en las laderas del volcán Sierra Negra en el estado de Puebla y está conformado por tanques llenos de agua de aproximadamente 5 m de altura y 7.3 m de diámetro. Cuando una partícula (ya sea un hadrón o un rayo gamma) llega del espacio exterior hacia la Tierra, esta interactúa con las partículas de la atmósfera y forma una cascada de partículas secundarias. Estas partículas llegan a los tanques del observatorio y por medio del efecto Cherenkov se genera luz que es captada por alguno de los 4 fotomultiplicadores que ahí se encuentran. Por medio del tiempo de detección y la distribución de carga en estos dispositivos, es posible determinar la dirección de la partícula inicial, y con ello generar mapas del cielo que nos dan información sobre las fuentes de rayos gamma en el Universo.

Al igual que los rayos gamma, los neutrinos de alta energía también apuntan hacia las fuentes más grandes de aceleración de partículas que hay en el Universo, y para altas energías estas partículas son una mejor fuente de estudio que los rayos gamma, ya que los fotones con energías superiores a los 100 TeV sufren una absorción por producción de pares con fotones de la radiación cósmica de fondo y con ello se reduce su camino libre medio [2]. La desventaja que tienen los neutrinos es que son difíciles de detectar y su tasa es prácticamente despreciable para energías del orden de los TeV. Sin embargo, hay varios métodos que se usan para la detección de estas partículas. Uno de ellos es la técnica *Earth-skimming* en donde se estudia la interacción de un neutrino y un nucleón en la superficie terrestre al intercambiar un bosón  $W^\pm$  dando como resultado la producción de un leptón cargado del mismo sabor del neutrino procedente. Debido a que la sección eficaz del neutrino y el nucleón es muy pequeña es necesario utilizar un objetivo de gran masa, como el suelo de la superficie terrestre o una montaña. Actualmente se está investigando la detección indirecta de neutrinos usando al observatorio HAWC por medio de la técnica *Earth-skimming*, en donde se usa al volcán Pico de Orizaba (que rodea a este observatorio) como objetivo de gran masa para producir la reacción neutrino-nucleón. El leptón con carga eléctrica que se produce viajará en la misma dirección del neutrino que lo originó, y para interacciones con la montaña la dirección será cuasi-horizontal. También es importante mencionar que los muones generados en cascadas atmosféricas en la dirección del Pico de Orizaba representan la principal fuente de ruido para detectar estas señales.

Hoy en día ya está implementado un algoritmo que discrimina entre eventos producidos por cascadas atmosféricas y por trazas horizontales. En la figura 1 se muestra una comparación visual entre estos dos tipos de eventos. Los círculos pequeños representan a los fotomultiplicadores que hay en cada tanque. Cada círculo de color representa un PMT que fue activado y su tamaño es proporcional a la carga que registró. Por otro lado la escala de color hace referencia al tiempo de activación. Como se puede observar en la imagen, cada uno de estos eventos tiene una forma característica que visualmente hace sencilla la distinción entre estos eventos. Actualmente el

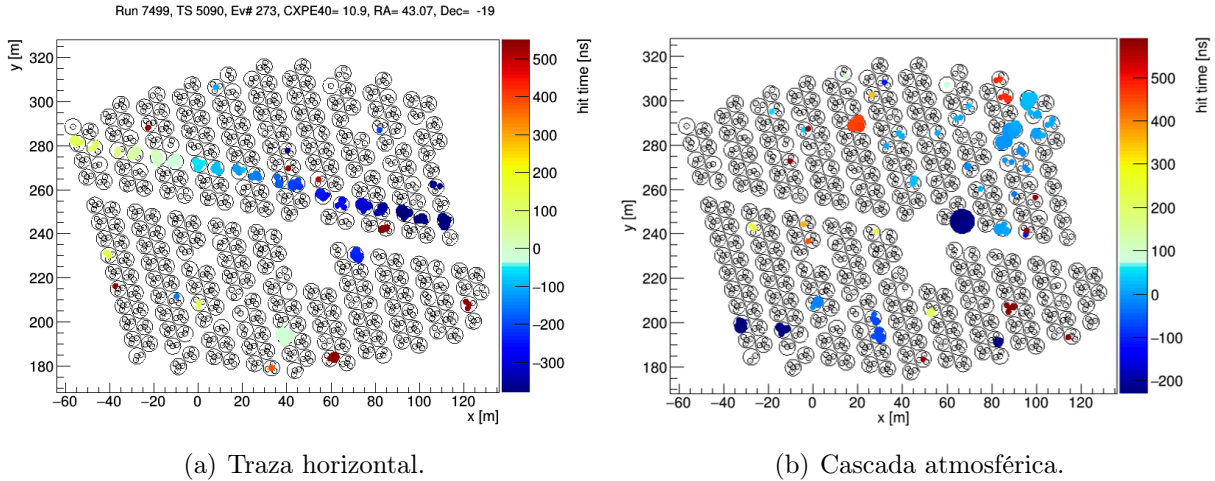


Figura 1: Comparación entre eventos de cascadas y trazas horizontales.

uso de redes neuronales artificiales para la clasificación de imágenes esta ampliamente optimizada, por tal motivo en este trabajo se estudio la posibilidad de aplicar estas herramientas para la discriminación de trazas horizontales. Cabe señalar que de una muestra de aproximadamente 2 millones de trazas de muones simulados, el algoritmo que actualmente se utiliza para discriminar este tipo de señales solamente logra identificar el 42 % de esta muestra [3]. Por tal motivo, se examinó la posibilidad de mejorar este porcentaje de acierto utilizando un red neuronal artificial.

La presente tesis comienza con una breve introducción para algunos aspectos de astropartículas, en donde se mencionan los fenómenos que provocan las señales de trazas y de cascadas que estamos estudiando. En el segundo capítulo se dan todas las características del funcionamiento del observatorio HAWC, para después dar de forma detallada la propuesta para la detección indirecta de neutrinos utilizando a este observatorio. En el siguiente capítulo se muestra la teoría detrás del funcionamiento de una red neuronal artificial enfocada en los modelos que se implementaron en este trabajo. Finalmente se describe el estudio utilizando estas herramientas en la discriminación de trazas horizontales y se compara con el algoritmo actual.

# Índice general

Agradecimientos . . . . .	II
Resumen . . . . .	III
Introducción . . . . .	IV
Lista de figuras . . . . .	VIII
Lista de tablas . . . . .	X
<b>1. Aspectos de astropartículas</b>	<b>1</b>
1.1. Preliminares . . . . .	1
1.1.1. Universo no térmico . . . . .	1
1.1.2. Preguntas abiertas . . . . .	2
1.2. Astronomía de rayos gamma . . . . .	4
1.2.1. Fuentes de rayos gamma . . . . .	4
1.2.2. Cascadas electromagnéticas . . . . .	6
1.3. Rayos Cósmicos . . . . .	7
1.3.1. Mecanismos de aceleración . . . . .	7
1.3.2. Fuentes de rayos cósmicos . . . . .	10
1.3.3. Espectro de energía . . . . .	11
1.3.4. Cascadas hadrónicas . . . . .	13
1.4. Astronomía de neutrinos . . . . .	14
1.4.1. Fuentes de neutrinos . . . . .	15
1.4.2. Espectro de energía para los neutrinos . . . . .	16
1.4.3. Detección de neutrinos . . . . .	17
<b>2. Observatorio HAWC</b>	<b>23</b>
2.1. Detectores utilizados en HAWC . . . . .	23
2.1.1. Detectores Cherenkov de agua (WCD) . . . . .	24
2.1.2. <i>Outrigger</i> . . . . .	25
2.2. Sistema de adquisición de datos . . . . .	26
2.2.1. Almacenamiento . . . . .	29
2.3. Reconstrucción de eventos . . . . .	29
2.3.1. Núcleo de la cascada . . . . .	30
2.3.2. Dirección . . . . .	31
2.3.3. Energía . . . . .	32
2.4. Software utilizado en HAWC . . . . .	33
<b>3. Planteamiento para la detección indirecta de neutrinos con HAWC</b>	<b>34</b>
3.1. Técnica <i>Earth skimming</i> . . . . .	34
3.2. Propuesta para la detección . . . . .	34
3.3. Viabilidad de la detección . . . . .	36

3.3.1.	<i>Trigger</i> de HAWC . . . . .	39
3.3.2.	Señales de Ruido . . . . .	39
3.4.	Reconstrucción de la dirección . . . . .	40
3.4.1.	Ángulo azimutal ( $\phi$ ) . . . . .	40
3.4.2.	Ángulo de elevación( $\theta$ ) . . . . .	40
3.5.	Algoritmo para detección de trazas . . . . .	42
3.5.1.	<i>Trigger</i> para trazas horizontales . . . . .	42
3.5.2.	Algoritmo de selección de trazas . . . . .	43
3.5.3.	Filtro para los candidatos a trazas . . . . .	43
<b>4.</b>	<b>Red Neuronal Artificial</b> . . . . .	<b>45</b>
4.1.	Red perceptrón multicapa . . . . .	45
4.1.1.	Estructura . . . . .	46
4.1.2.	Funcionamiento . . . . .	47
4.2.	Red neuronal convolucional . . . . .	51
4.2.1.	Convolución . . . . .	51
4.2.2.	<i>Pooling</i> o Agrupación . . . . .	53
4.2.3.	Funcionamiento . . . . .	54
4.3.	Detección de objetos . . . . .	57
4.3.1.	R-CNN . . . . .	59
4.3.2.	Fast-R-CNN . . . . .	60
4.3.3.	Faster-R-CNN . . . . .	61
<b>5.</b>	<b>Identificación de trazas con redes neuronales artificiales</b> . . . . .	<b>64</b>
5.1.	Prueba 1: Red convolucional . . . . .	64
5.1.1.	Imágenes de entrada . . . . .	64
5.1.2.	Estructura de la red . . . . .	66
5.1.3.	Imágenes de entrenamiento y de prueba . . . . .	66
5.1.4.	Resultado . . . . .	69
5.2.	Prueba 2: Detección de objetos . . . . .	72
5.2.1.	Imágenes de entrada . . . . .	72
5.2.2.	Estructura de la red . . . . .	73
5.2.3.	Imágenes de entrenamiento y de prueba . . . . .	73
5.2.4.	Resultados . . . . .	74
5.3.	Comparación con el algoritmo actual . . . . .	75
	<b>Conclusiones</b> . . . . .	<b>76</b>
	<b>A) Mecanismos de producción y aceleración para rayos gamma</b> . . . . .	<b>77</b>
	<b>B) Funciones de AERIE y principales resultados obtenidos por HAWC</b> . . . . .	<b>80</b>
	<b>C) Códigos utilizados para las redes neuronales artificiales</b> . . . . .	<b>83</b>
	<b>D) Gráficas complementarias para los modelo A y B</b> . . . . .	<b>92</b>
	<b>E) Trazas identificadas por la red</b> . . . . .	<b>94</b>
	<b>Bibliografía</b> . . . . .	<b>107</b>



# Índice de figuras

1.	Comparación entre eventos de cascadas y trazas horizontales. . . . .	V
1.1.	Espectro electromagnético . . . . .	4
1.2.	Sistema binario . . . . .	5
1.3.	Representación gráfica de manchas solares. . . . .	8
1.4.	Representación gráfica para aceleración con frentes de choque. . . . .	10
1.5.	Energía ganada por una partícula debido a una nube magnética de gas . . . . .	10
1.6.	Gráfica de Hillas . . . . .	11
1.7.	Espectro de energía de rayos cósmicos . . . . .	12
1.8.	Cascada atmosférica . . . . .	13
1.9.	Espectro de energía de neutrinos . . . . .	17
1.10.	Diagramas de Feynman para interacciones de corriente cargada y neutra . . . . .	17
1.11.	Diagrama para la dispersion neutrino electrón . . . . .	18
1.12.	Diagrama de un PMT . . . . .	20
2.1.	Imagen del observatorio HAWC . . . . .	23
2.2.	Detectores de agua Cherenkov WCD . . . . .	24
2.3.	Emisión de la radiación Cherenkov . . . . .	25
2.4.	Arreglo para los <i>outriggerer</i> . . . . .	26
2.5.	Conexión entre PMTs y el DAQ principal . . . . .	27
2.6.	Diagrama simplificado para un FEB analógico . . . . .	27
2.7.	Ejemplo para asociar un ToT a la señal . . . . .	28
2.8.	Gráfica del logaritmo del número de fotoelectrones en función del ToT . . . . .	28
2.9.	Sistema de adquisición de datos en HAWC . . . . .	29
2.10.	Esquema que muestra al núcleo de la cascada . . . . .	31
2.11.	Correlación entre carga y energía para un ángulo cenital ( $14.77^\circ < \theta < 15.42^\circ$ ) . . . . .	32
3.1.	Representación del uso de la técnica <i>Earth-skimming</i> en HAWC . . . . .	35
3.2.	Espectros de $\varphi_i$ para los modelos A,B y C . . . . .	37
3.3.	Representación para el ángulo azimutal . . . . .	41
3.4.	Representación para el ángulo de elevación . . . . .	41
3.5.	Representación de tanques vecinos. . . . .	42
3.6.	Ejemplo identificado como traza . . . . .	44
4.1.	Estructura de una red perceptrón multicapa . . . . .	46
4.2.	Red con una sola neurona en cada capa . . . . .	49
4.3.	Representación gráfica para encontrar un mínimo local de la función de costo . . . . .	51
4.4.	Entrada de una red convolucional . . . . .	51
4.5.	Representación del funcionamiento de la convolución . . . . .	52
4.6.	Representación del paso usado por un filtro para hacer la convolución . . . . .	52

4.7.	Cambio de la imagen de una letra al aplicar un filtro y una función de activación . .	53
4.8.	Cambio de la imagen de un número 7 al aplicar dos diferentes filtros . . . . .	53
4.9.	Ejemplo de <i>max pooling</i> y <i>average pooling</i> . . . . .	54
4.10.	Ejemplo de la primear capa de una red neuronal convolucional . . . . .	54
4.11.	Ejemplo de la segunda capa de una red neuronal convolucional . . . . .	55
4.12.	Aplanamiento y conexión con una red perceptrón multicapa . . . . .	55
4.13.	Representación general de la estructura de una red convolucional . . . . .	56
4.14.	Representación gráfica del producto interno entre una imagen a color y dos filtros .	57
4.15.	Representación de cuatro iteraciones para la búsqueda de un objeto . . . . .	58
4.16.	Representación del funcionamiento del algoritmo <i>selective search</i> . . . . .	59
4.17.	Esquema de funcionamiento del detector <i>R-CNN</i> . . . . .	60
4.18.	Esquema de funcionamiento del detector <i>Fast-RCNN</i> . . . . .	60
4.19.	Ejemplo de <i>RoI pooling</i> para un tamaño de $q = 4$ . . . . .	61
4.20.	Esquema de funcionamiento del detector <i>Faster-RCNN</i> . . . . .	61
4.21.	Ejemplo de propuesta de regiones en una imagen mediante anclajes . . . . .	62
4.22.	Representación de los áreas de intersección e unión en el par detección <i>ground-truth</i> .	63
5.1.	Simulación de una traza horizontal . . . . .	65
5.2.	Imágenes de trazas horizontales de eventos reales. . . . .	65
5.3.	Imagen de cascada atmosférica de datos reales. . . . .	66
5.4.	Estructura de la red VGG16 . . . . .	67
5.5.	Distribución de la variable $\beta$ para los candidatos del primer <i>subrun</i> del <i>Run 7499</i> . .	68
5.6.	Tipo de GPU accesible en <i>Google Colab</i> . . . . .	69
5.7.	Gráfica de costo y la tasa de aciertos para el modelo C. . . . .	69
5.8.	Gráfica de PR para el modelo C. Cada número que señala a cada punto representa un valor umbral. . . . .	71
5.9.	Costo y precisión para el modelo D. . . . .	72
5.10.	Gráfica PR para el modelo D. Cada número que señala a cada punto representa un valor umbral. . . . .	72
5.11.	Selección para una traza y una cascada usando <i>labelimg</i> . . . . .	73
5.12.	Imagen de salida para la red enfocada en detección de objetos. . . . .	74
5.13.	Costo para la red enfocada en detección de objetos. . . . .	74
5.14.	Bremsstrahlung y Compton inverso . . . . .	78
5.15.	Representación gráfica de manchas solares. . . . .	79
5.16.	Reconstrucción de Cascadas con AERIE . . . . .	80
5.17.	Mapa del cielo en rayos gamma . . . . .	81
5.18.	Comparación de funciones de distribución lateral . . . . .	82
5.19.	Estructura de la carpeta <i>deteccion-objetos</i> . . . . .	85
5.20.	Gráfica para el costo y la tasa de aciertos para el modelo A. . . . .	92
5.21.	Gráfica para el costo y la tasa de aciertos para el modelo B. . . . .	93
5.22.	Curva <i>precisión-recall</i> . . . . .	93

# Índice de tablas

1.1. Dominios de la astronomía de rayos gamma en función de los rangos de energía . . .	4
1.2. Flujo de neutrinos solares . . . . .	15
3.1. Valores de la constante de normalización $\phi_i$ y la energía $E_i$ . . . . .	36
3.2. Cálculo del flujo de leptones cargados tauónicos y muónicos . . . . .	37
3.3. Cálculo del flujo de leptones cargados tauónicos y muónicos . . . . .	38
3.4. Número de leptones cargados $\tau$ que se pueden detectar en HAWC por año . . . . .	38
3.5. Número de señales de ruido que se pueden detectar en HAWC por año . . . . .	39
3.6. Porcentaje de trazas simuladas que pasaron los 4 cortes mencionados . . . . .	44
4.1. Modelos preentrenados incluidos en API . . . . .	58
5.1. Evaluación de modelos para la red convolucional. . . . .	71
5.2. Trazas horizontales identificadas por los tres modelos de un total de 118476 eventos analizados. . . . .	75

# Capítulo 1

## Aspectos de astropartículas

La física de astropartículas es un campo interdisciplinario entre la astronomía y la física de partículas, que trata de revelar la naturaleza y estructura de la materia en el Universo [4]. Esta área de la física es relativamente nueva e históricamente tiene sus bases en la astronomía óptica. En la parte observacional en física de astropartículas se hace uso de todo tipo de técnicas para conocer las fuentes de radiación cósmica, la naturaleza de las partículas emitidas, así como las propiedades del medio interestelar. El conocimiento de la física de astropartículas se entiende como un medio vital para conocer la naturaleza de las estructuras más pequeñas y más grandes del Universo al mismo tiempo, ya que es necesario un conocimiento preciso de la física de partículas para comprender muchos contextos astrofísicos [5]. El observatorio HAWC es uno de estos experimentos que estudia al Universo por medio de la radiación gamma, y en esta tesis se trabajó en la mejora de los algoritmos de detección de señales horizontales que se registran en este observatorio. Por tal motivo y como punto de partida, me gustaría hablar primero sobre algunos aspectos de astropartículas. Primero hablaré sobre lo que entendemos como un Universo no térmico, para después pasar a temas relacionados con los rayos cósmicos y rayos gamma. Finalmente daré una breve descripción de la astronomía de neutrinos, que es la motivación que se encuentra detrás de este proyecto.

### 1.1. Preliminares

En esta sección primero se explicará el concepto de Universo no térmico, y como este se relaciona en el estudio de rayos cósmicos, rayos gammas y neutrinos. Después se mencionan algunas de las preguntas abiertas que actualmente se siguen estudiando en el área de astropartículas.

#### 1.1.1. Universo no térmico

Gran parte de la radiación que se propaga en el Cosmos y llega a la Tierra es radiación térmica generada en objetos calientes como estrellas. Sin embargo, se conoce que diversas partículas no resultan de estos procesos sino que se generan por otros mecanismos [6]. Cabe mencionar, que estas radiaciones también contribuyen de igual forma a la energía total del Cosmos, por ello es importante su estudio para entender la evolución del Universo. El ejemplo más conocido de partículas generadas por procesos no térmicos son los rayos cósmicos, que son partículas con carga eléctrica que llegan del espacio exterior hacia la Tierra. Debido a la presencia de campos magnéticos orientados aleatoriamente en el medio galáctico e intergaláctico, los rayos cósmicos son desviados, por tal motivo no se puede usar su dirección de llegada para conocer la fuente de donde provienen. Así que para estudiar los sitios de aceleración de estos rayos cósmicos, se usan partículas neutras que

no son desviadas por campos magnéticos [7]. Por tal motivo en secciones posteriores hablaremos de rayos gamma y neutrinos que son partículas que se producen como subproductos de los rayos cósmicos.

### 1.1.2. Preguntas abiertas

Ahora mostraremos algunas preguntas en el área de astropartículas que todavía no tienen respuesta y que actualmente se siguen estudiando.

- **Origen y aceleración de rayos cósmicos:** Una de las grandes preguntas en la física de astropartículas, es el origen de los rayos cósmicos de ultra alta energía. Aunque los rayos cósmicos se hayan descubierto hace más de 100 años, todavía sigue siendo un misterio como la naturaleza acelera a estas partículas hasta órdenes tan altos de energías (superiores a  $10^{20}$  eV). Estudios sugieren que rayos cósmicos con energías menores a  $10^{17}$  eV son acelerados por Supernovas que se encuentran en nuestra galaxia. Sin embargo, solo hay hipótesis de las fuentes que originan rayos cósmicos de energías superiores y que posiblemente se encuentren en otras galaxias [8].
- **Límite de Greinsen-Zatsepin-Kuzmin (GZK):** En 1991, el observatorio de rayos cósmicos *Fly's Eye* (que se encuentra en Utah Estados Unidos), registró un rayo cósmico que impactó a la atmósfera de la Tierra con una energía de  $3.2 \pm 0.9 \times 10^{20}$  eV [9]. Nunca antes se había registrado un evento de tan alta energía, y aunque los responsables del observatorio publicaron el descubrimiento refiriéndose a su detección sencillamente como «el evento»; en algún momento alguien decidió que esta extraordinaria partícula merecía un título más llamativo y terminó conociéndose como partícula *Oh-My-God*. Eventos con energías similares a esta partícula fueron registrados por otros experimentos que confirmaron el fenómeno.

Todavía no se sabe bien qué mecanismo utiliza la naturaleza para alcanzar dichas energías, además estos eventos tan energéticos deberían interactuar con la radiación de fondo de microondas y perder energía al viajar distancias extragalácticas (aproximadamente  $100 \times 10^6$  años luz). Dicha interacción nos da un límite para la energía de los rayos cósmicos que provienen de fuentes distantes, esta energía es de  $5 \times 10^{19}$  eV y a dicho efecto se le conoce como límite de Greinsen-Zatsepin-Kuzmin (GZK) [10]. Como se han registrado eventos superiores a este límite, significa que la fuente de emisión de estos eventos se encuentran relativamente cerca (en términos cosmológicos) o que nuestro entendimiento del límite GZK no es del todo correcto. Actualmente este límite sigue siendo un tema de investigación abierto [11].

- **Anisotropía de rayos cósmicos:** Como mencionamos anteriormente, los rayos cósmicos son partículas cargadas que llegan del espacio exterior hacia la Tierra, y estas provienen de todas las direcciones de forma aleatoria. Sin embargo, de cada 1,000 rayos cósmicos hay como máximo un solo rayo con una dirección no aleatoria. Este efecto del 0.1 % se denomina anisotropía de rayos cósmicos. Se cree que la anisotropía probablemente surge a raíz de la estructura del campo magnético de la Tierra y la ubicación de los aceleradores de rayos cósmicos más cercanos a la Tierra [12]. Sin embargo aún se publican estudios sobre este fenómeno para el caso de rayos cósmicos ultra energéticos. Por ejemplo, en el año 2019 la colaboración Pierre Auger presentó un estudio sobre esta anisotropía para rayos cósmicos de alta energía utilizando 15 años de datos [13].
- **Materia Oscura:** A partir de la observación de las velocidades orbitales de las estrellas en la Vía Láctea y las velocidades de las galaxias en los cúmulos galácticos, está claro que

la densidad de energía de la materia visible es insuficiente para describir correctamente su dinámica. Así que nuestra teoría que describe la gravedad está mal o hay materia que no estamos viendo. Esta última opción es la más aceptada, sin embargo, no hay un consenso claro sobre qué puede ser esta *materia oscura* y que hoy en día sigue siendo un tema de investigación altamente estudiado [5].

- **Aceleración del Universo:** Basándose en las ideas del Big Bang clásico, se podría suponer que el empuje inicial de la explosión sería ralentizado por la gravedad. Sin embargo, las observaciones sobre explosiones de Supernovas distantes (1998) parecen indicar que en las primeras épocas cosmológicas, la tasa de expansión era menor que la actual. El hallazgo de un Universo en aceleración, que ahora es generalmente aceptado, tiene importantes implicaciones para la cosmología. Una interpretación es que la mayor parte de la *materia oscura* que falta se almacena como *energía oscura* en un vacío dinámico, sin embargo todavía no hay respuesta clara que pueda explicar este fenómeno [5].
- **Neutrinos:** Neutrinos astrofísicos de alta energía (del orden de TeV y PeV) recientemente descubiertos, han abierto un nuevo régimen para probar nueva física. Ya que estas energías son las más altas que se han detectado y con ello se podrían observar diversos efectos a nuevas escalas. Además, dichos neutrinos viajan grandes distancias debido a su alta energía; por lo que durante su trayectoria podría haber pequeños efectos de nueva física que podrían detectarse cuando llegan a la Tierra [14].

Cabe mencionar que ya se han obtenido avances en el estudio de neutrinos de alta energía, por ejemplo, Ice Cube midió la sección transversal neutrino-nucleón en el rango de TeV-PeV [15]-[16]. Dichas mediciones son consistentes con predicciones del modelo estándar basadas en datos de colisionadores. Sin embargo, los neutrinos astrofísicos de energía superiores (del orden de EeV) podrían utilizarse para medir la misma sección transversal y con ello poder estudiar su dinámica de manera más profunda que usando colisionadores en la Tierra [14].

Numerosos modelos de nueva física tienen efectos que son proporcionales a alguna potencia de la energía del neutrino  $E$  y a la distancia propagada  $L$  que crecen de la forma  $\kappa_n E^n L$ . En donde el valor de las constantes  $\kappa, n$  dependen del modelo:

- $n = -1$  para el decaimiento de neutrinos.
- $n = 0$  para violación de Lorentz impar CPT o acoplamiento a un campo de torsión.
- $n = 1$  para violación de Lorentz par CPT o violación del principio de equivalencia.

Un experimento que observe neutrinos de alta energía provenientes de fuentes ubicadas a una distancia  $L$ , en principio, será capaz de probar los modelos antes mencionados con sensibilidades de  $\kappa_n \sim 4 \cdot 10^{-50} (E/EeV)^{-n} (L/Gpc)^{-1} EeV^{1-n}$  [14].

Actualmente, para observar algún fenómeno nuevo en la física de neutrinos astrofísicos, se hacen estudios sobre su espectro de energía, la distribución de la dirección de llegada y su composición, es decir, la proporción de cada sabor de neutrino sobre el flujo total que entra en la Tierra [14].

## 1.2. Astronomía de rayos gamma

La observación de estrellas en el rango espectral óptico pertenece al campo de la astronomía clásica [5]. Sin embargo, dicho rango solo cubre una pequeña parte del espectro electromagnético (ver figura 1.1). Cabe destacar, que todos los rangos en este espectro se usan para observaciones astronómicas. Pero la astronomía de rayos gamma abarca el rango de energía que va desde 1 MeV (aproximadamente) hasta las energías más altas de este espectro [7].

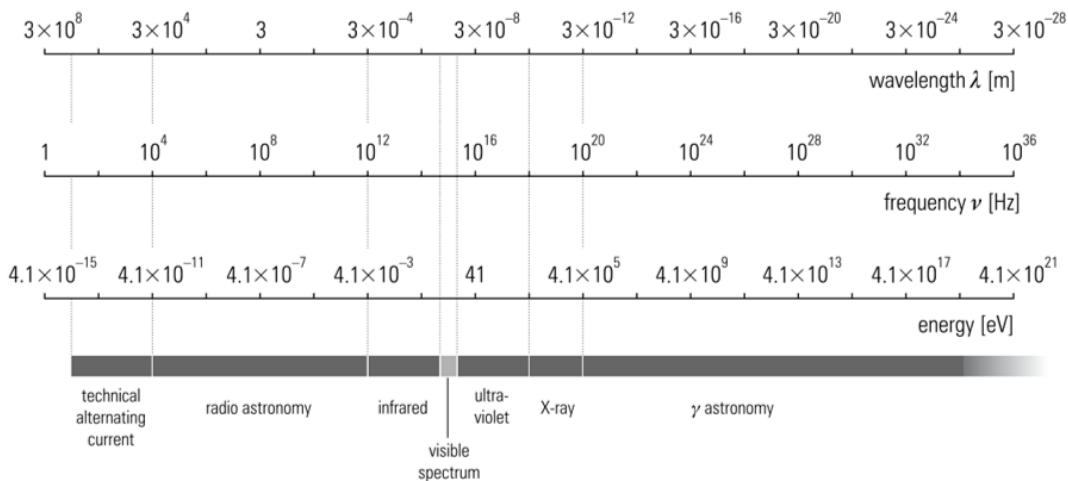


Figura 1.1: Espectro electromagnético. Imagen tomada de [5].

En esta área, los rayos gamma también se pueden clasificar acorde a la energía que tienen y que se muestran en la tabla 1.1. Cabe mencionar que los rayos gamma son partículas neutras, así que no son afectados por los campos electromagnéticos del espacio exterior. Esta es una ventaja sobre los rayos cósmicos, ya que la información de la dirección de la fuente en donde fueron emitidos no se pierde.

Dominio de energía	Rango de energía
Baja	1 MeV a 30 MeV
Alta	30 MeV a 50 GeV
Muy alta	50 GeV a 100 TeV
Ultra alta	100 TeV a 100 PeV
Extremadamente alta	Mayor a 100 PeV

Tabla 1.1: Dominios de la astronomía de rayos gamma en función de los rangos de energía. Tabla modificada de [7].

### 1.2.1. Fuentes de rayos gamma

Diferentes tipos de fuentes que emiten rayos gamma de muy alta energía se han observado. A continuación se mencionan las principales fuentes que emiten estas partículas:

- **Púlsares y púlsares de viento nebular (PWNe<sup>1</sup>):** En algunos casos en la explosión de una estrella, el remanente de Supernova se puede convertir en un pulsar o un PWNe.

<sup>1</sup>Siglas en inglés *Pulsar Wind Nebula*

Un púlsar es una estrella de neutrones que se encuentra en rotación y que posee un campo magnético muy intenso [7]. Por otro lado, un PWNe es un tipo de nebulosa que se encuentra dentro de las capas de un remanente de Supernova [7]. Este remanente se alimenta por los vientos generados por el púlsar central. Partículas cargadas se pueden acelerar en algunas zonas de la magnetósfera de la estrella de neutrones y por los procesos que se mencionan en el apéndice A, emitir rayos gamma de muy alta energía. La energía de rotación del púlsar, se pierde principalmente por la emisión de  $e^\pm$ . Estos electrones y positrones interactúan con el medio interestelar y generan escenarios de frentes de ondas de choque (aceleración de Fermi<sup>2</sup>). En dichos escenarios, rayos gamma también son producidos. Este tipo de fuente emite rayos gamma de muy alta energía (ver tabla 1.1) que pueden alcanzar energías de TeV [7].

- **Remanentes de Supernova:** Cuando una estrella masiva se queda sin combustible, se enfría. Esto hace que la presión caiga y la gravedad gane, por ello la estrella de repente se colapsa. Este colapso ocurre tan rápido que crea enormes ondas de choque que hacen que la parte exterior de la estrella explote. A esta explosión se le conoce como Supernova [5], y el remanente es material arrojado por dicha explosión que se expande por un cierto periodo de tiempo. Este material interactúa con el medio interestelar generando ondas de choque (mecanismo de Fermi) que aceleran rayos cósmicos. Como resultado de la aceleración de rayos cósmicos también se generan rayos gamma de muy alta energía (ver tabla 1.1) y se cree que son de origen hadrónico [7].
- **Sistemas Binarios** Estos sistemas principalmente son conformados por objetos pesados y compactos como por ejemplo una estrella de neutrones, agujeros negros o estrellas muy pesadas [5]. Una explicación para la producción de rayos gamma en estos sistemas, se puede mostrar mediante el siguiente ejemplo: consideremos un sistema formado por una estrella normal y un púlsar, en este caso la materia se arrastra permanentemente de la estrella normal y se convierte en un disco de acreción alrededor del púlsar (ver figura 1.2). Debido a este gran movimiento de plasma se obtienen fuertes campos electromagnéticos en la vecindad de la estrella de neutrones que pueden acelerar partículas cargadas y con ello generar rayos gamma. La energía de estos rayos emitidos por sistemas binarios puede variar, por ejemplo, se conocen mas de una docena de sistemas binarios que emiten gammas con energías que van de los 0.1 a los 100 GeV; sin embargo también se han encontrado alguno de estos sistemas que emiten gammas en un rango mayor a los 100 GeV [17].

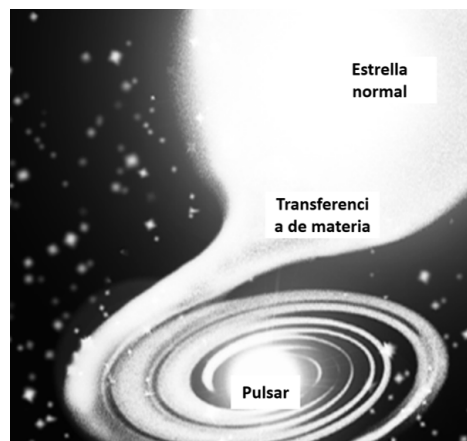


Figura 1.2: Formación de un disco de aceleración en un sistema binario. Imagen modificada de [5].

<sup>2</sup>La aceleración de Fermi se explica en la sección 1.3.1



- **Núcleos activos de Galaxia (AGN<sup>3</sup>):** Es un agujero negro super masivo que se encuentra en el centro de una galaxia con la característica de que es el objeto mas brillante de la galaxia [7]. Se ha observado que estos AGN emiten jets colimados de partículas con velocidades relativistas. Estas partículas cargadas de alta energía producen emisiones no térmicas desde radio hasta rayos gamma. La energía de los fotones emitidas por este tipo de fuentes va desde los MeV hasta los TeV, en donde se conocen mas de 1500 fuentes que emiten gammas con energías de GeV y mas de 60 fuentes que emiten gammas con energías de TeV (los cuales reciben el nombre de Blázares) [18]. Dependiendo del ángulo de visión, se dividen en diferentes categorías como Blázares, Radiogalaxias, Cuásares y las galaxias de Seyfert [7].
- **Galaxias Starburst:** Son galaxias con una tasa alta en la formación de estrellas [7]. Como consecuencias también se tiene una tasa alta en la generación de Supernovas que son fuentes de aceleración de rayos cósmicos y rayos gamma de muy alta energía (ver tabla 1.1).
- **Destello de rayos gamma (GRB<sup>4</sup>):** Son explosiones de rayos gamma extremadamente energéticos de origen extragaláctico. Estos se pueden dividir en 2 categorías [7], de corta duración (menores a 2 s) y de larga duración (mayores a 2 s). Se cree que el origen de estas dos categorías es la fusión de dos objetos compactos y la explosión de una Supernova de una estrella muy masiva, respectivamente. Estas fuentes además de emitir rayos gamma, también producen fotones en muchas otras longitudes de onda. Para este tipo de fuentes se han observado gammas en el rango de MeV hasta cientos de GeV, sin embargo no se ha detectado fotones con energías de TeV [19].

### 1.2.2. Cascadas electromagnéticas

La detección en la astronomía de rayos gamma se puede dividir en 2 categorías: experimentos espaciales y experimentos en la superficie terrestre. Debido a la atmósfera de la Tierra, la única forma de detectar rayos gamma de forma directa es mediante un instrumento que se encuentre en el espacio exterior. Sin embargo, el tamaño de estos detectores es pequeño debido a que se montan en satélites. Como consecuencia el área efectiva de estos instrumentos es relativamente pequeña (aproximadamente de  $1 m^2$ ), por lo tanto son adecuados para estudiar rayos gamma en el rango MeV-GeV. Sin embargo, para energías mayores a unos pocos cientos de GeV el flujo de rayos gamma se vuelve tan bajo que se requieren áreas y tiempos de exposición mucho más grandes para realizar mediciones significativas. Por ello para este caso se utilizan experimentos que se puedan construir en la superficie de la Tierra (con mayor área efectiva), como es el caso del experimento HAWC. Cuando un rayo gamma llega a la Tierra, este choca con las partículas que conforman la atmósfera, esto genera una cascada de partículas secundarias que se pueden observar mediante detectores en la superficie terrestre. Tanto la dirección como la energía del fotón que inició la cascada se pueden reconstruir utilizando distintos métodos.

En general se puede entender esta cascada mediante los siguientes pasos:

1. Un rayo gamma primario entra a la atmósfera y a través del mecanismo de producción de pares formará un electrón y un positrón [20]. Cabe mencionar que un rayo gamma con suficiente energía puede interactuar con los átomos del aire por medio de una reacción fotonuclear. Con dicha reacción muones son generados por medio de la formación de mesones y posteriormente su decaimiento. También se producen muones mediante el proceso de creación de pares ( $\mu^- \mu^+$ ),

---

<sup>3</sup>*Active Galactic Nuclei*

<sup>4</sup>En sus siglas en inglés *Gamma-Ray Bursts*

así que hay una componente hadrónica en este tipo de cascada [21]. Aunque el contenido de muones en una cascada electromagnética es muy pequeño, solamente representaría el 1 % de la cantidad de muones que se puede encontrar en una cascada iniciada por un hadrón [22].

2. Con la formación del electrón y el positrón en el paso anterior se formarán fotones mediante *Bremsstrahlung* o por aniquilación de pares. Cabe mencionar que también gran parte de los electrones y positrones perderán energía a través de colisiones con partículas del aire [23].
3. Los rayos gammas formados en el proceso anterior formaran más electrones y positrones por producción de pares y se irá repitiendo el proceso descrito en el paso dos. Cabe destacar que estos fotones también perderán energía por dispersión de Compton [23].
4. La energía de las partículas involucradas en estas interacciones irá disminuyendo hasta que la dispersión de Compton sea el proceso que domine en los fotones (superando a la producción de pares) y la pérdida de energía a través de colisiones sea mayor que a la producción de gammas por *Bremsstrahlung* y por aniquilación, para los electrones y positrones [23]. Así que la pérdida de energía por ionización empieza a dominar la cascada que se irá disipando en la atmósfera.

### 1.3. Rayos Cósmicos

Como se ha comentado anteriormente, los rayos cósmicos son partículas cargadas que llegan del espacio exterior hacia a la Tierra. En esta sección revisaremos algunos aspectos relevantes correspondientes a este tema. Primero se mencionan las fuentes y los mecanismos de aceleración que se conocen para estas partículas. También se muestra el espectro de energía que se mide en la Tierra, y abordaremos una forma de detectar a estos rayos cósmicos a partir de partículas secundarias que se generan cuando ingresan al planeta y que puede ser detectada en la superficie de la Tierra.

#### 1.3.1. Mecanismos de aceleración

A continuación se enlistan los principales mecanismos de aceleración para rayos cósmicos:

- **Manchas solares:** Las manchas solares son sitios en el Sol que tienen un campo magnético que depende del tiempo y cuya temperatura es un poco más baja en comparación con las zonas del Sol que la rodean [5]. Estas manchas aparecen más oscuras debido a que parte de la energía térmica se ha transformado en energía del campo magnético. En el inciso (a) de la figura 1.3, se muestra el esquema de una mancha solar con área igual a ( $A = \pi R^2$ ) y con un campo magnético variable  $B$ . Para este caso, el cambio del flujo magnético va a generar un potencial  $U$ :

$$-\frac{d\phi}{dt} = \oint E \cdot ds = U \quad (1.1)$$

Si se asume que el campo  $B$  es perpendicular al área, el flujo magnético está dado por:

$$\phi = \int B \cdot dA = B\pi R^2 \quad (1.2)$$

Un giro de una partícula cargada alrededor de la mancha solar obtendrá una energía de:

$$E = eU = e\pi R^2 \frac{dB}{dt} \quad (1.3)$$

Una mancha solar tiene una extensión de  $R = 10^9$  cm, con campos magnéticos de  $B = 2000$  Gauss que duran un día ( $dB/dt = 2000$  Gauss/día) [5], entonces:

$$E = 1.6 \cdot 10^{-19} \text{As} \pi 10^{14} \text{m}^2 \frac{0.2V_s}{86400 \text{s} \text{m}^2} = 0.73 \text{ GeV}$$

Se han observado partículas del Sol con energías de hasta 100 GeV. Sin embargo, esto puede representar el límite para el poder de aceleración de estrellas basado en el mecanismo del ciclotrón. Por lo tanto, este modelo puede explicar las energías observadas de forma correcta [5].

- Pares de manchas solares:** Éstas manchas comúnmente se forman en pares y con momento dipolar opuesto, además suelen acercarse y unirse para formar una sola mancha [5]. Para explicar lo que sucede en este proceso, supongamos que tenemos dos manchas solares con momento dipolar opuesto (tal y como se muestra en el inciso (b) de la figura 1.3). En dicha imagen, la primera mancha solar está en reposo y la segunda mancha se encuentra en movimiento. El dipolo magnético en movimiento genera un campo eléctrico perpendicular a la dirección del dipolo y perpendicular a la dirección del movimiento ( $\vec{E} = \vec{v} \times \vec{B}$ ) dicho campo eléctrico sería el responsable de acelerar partículas con carga. Bajo suposiciones realistas, las distancias entre estas manchas solares es de  $10^7$  m con campos magnéticos de 2000 Gauss y una velocidad relativa de  $10^7$  m/día. Por lo tanto, la energía de las partículas aceleradas por medio de este proceso está en el orden de Gigaelectronvoltios [5].

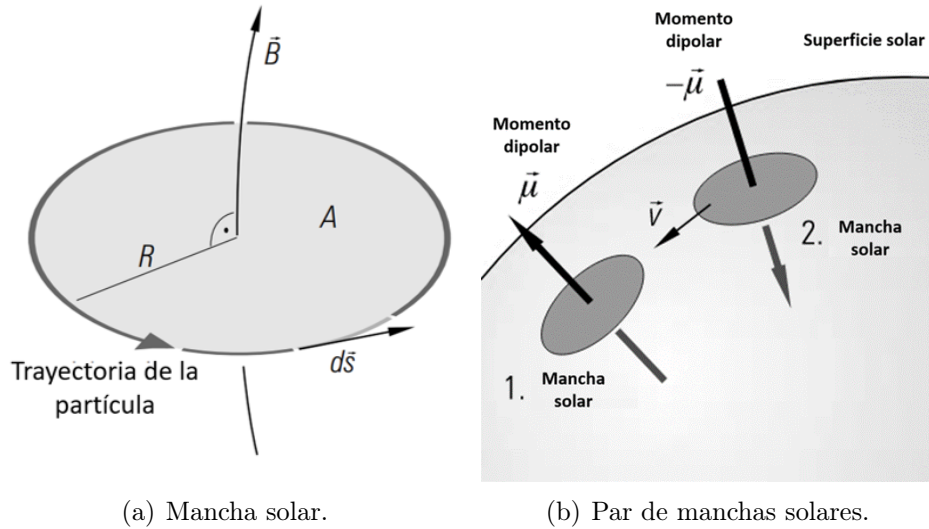


Figura 1.3: Representación gráfica de manchas solares. Imágenes modificadas de [5].

- Aceleración de Fermi de primer orden:** En este mecanismo los rayos cósmicos se encuentran atrapados en medio de frentes de ondas de choque [5]. Dichas ondas son generadas por procesos astrofísicos violentos, como es el caso de una Supernova. Para explicar el mecanismo de aceleración primero supongamos que la onda de choque se mueve a una velocidad  $u_1$ . Detrás de la onda de choque el gas retrocede con una velocidad  $u_2$ , así que el gas lleva una velocidad de  $u_1 - u_2$  en el sistema de referencia del laboratorio (ver inciso (a) de la figura 1.4). Una partícula con velocidad  $v$  que choca con el frente de onda y es reflejada, gana una energía de:

$$\Delta E = \frac{1}{2}m(v + (u_1 - u_2))^2 - \frac{1}{2}mv^2 = \frac{1}{2}m(2v(u_1 - u_2) + (u_1 - u_2)^2) \quad (1.4)$$

Tomando en cuenta que los términos lineales dominan la expresión anterior ( $v \gg u_1, u_2$  y que  $u_1 > u_2$ ), este simple modelo nos da una ganancia relativa de la energía por parte de la partícula de:

$$\frac{\Delta E}{E} \approx \frac{2(u_1 - u_2)}{v} \quad (1.5)$$

Considerando efectos relativistas y que la velocidad de la partícula se acerca a la velocidad de la luz, la expresión para la ganancia de la energía será:

$$\frac{\Delta E}{E} = \frac{4}{3} \frac{(u_1 - u_2)}{c} \quad (1.6)$$

Se obtienen resultados similares, si se supone que las partículas quedan atrapadas entre dos frentes de choque y estas se reflejan hacia adelante y hacia atrás rebotando en cada frente de onda (ver inciso (b) de la figura 1.4). Para este caso, la velocidad del frente de choque interior ( $v_2$ ) es mayor que la velocidad del frente de choque exterior ( $v_1$ ) debido a que esta última es desacelerada por la interacción con la materia interestelar. Una partícula con velocidad  $v$  que es reflejada en el frente de choque interno, gana una energía de:

$$\Delta E_1 = \frac{1}{2}m(v + v_2)^2 - \frac{1}{2}mv^2 = \frac{1}{2}m(v_2^2 + 2vv_2) \quad (1.7)$$

La reflexión en el frente de choque exterior conduce a una pérdida de energía de:

$$\Delta E_2 = \frac{1}{2}m(v - v_1)^2 - \frac{1}{2}mv^2 = \frac{1}{2}m(v_1^2 - 2vv_1) \quad (1.8)$$

Así que en promedio la partícula gana una energía de:

$$\Delta E = \frac{1}{2}m(v_1^2 + v_2^2 + 2v(v_2 - v_1)) = \frac{1}{2}m(v_1^2 + v_2^2 + 2v\Delta v) \quad (1.9)$$

Debido a que podemos despreciar los términos cuadráticos (ya que  $v_2 > v_1$ ), se tendrá que:

$$\frac{\Delta E}{E} \approx 2 \frac{\Delta v}{v} \quad (1.10)$$

A este tipo de aceleración se le conoce como mecanismo de Fermi de primer orden, en donde al usar las correcciones relativistas se pueden acelerar partículas con energías máximas de 100 TeV [5].

- **Aceleración de Fermi de segundo orden:** Este mecanismo describe la interacción de rayos cósmicos con nubes magnéticas. Para describir este mecanismo, supongamos que una partícula (con velocidad  $v$ ) es reflejada por una nube de gas que se mueve a velocidad  $u$ . En el caso en que  $v$  y  $u$  sean antiparalelos (ver inciso (a) de la figura 1.5), la partícula gana una energía de:

$$\Delta E_1 = \frac{1}{2}m(v + u)^2 - \frac{1}{2}mv^2 = \frac{1}{2}m(2uv + u^2) \quad (1.11)$$

En el caso en donde  $u$  y  $v$  sean paralelos (ver inciso (b) de la figura 1.5), la partícula pierde una energía de:

$$\Delta E_2 = \frac{1}{2}m(v - u)^2 - \frac{1}{2}mv^2 = \frac{1}{2}m(-2uv + u^2) \quad (1.12)$$

En promedio, la energía neta ganada por la partícula será:

$$\Delta E = \Delta E_1 + \Delta E_2 = mu^2 \quad (1.13)$$

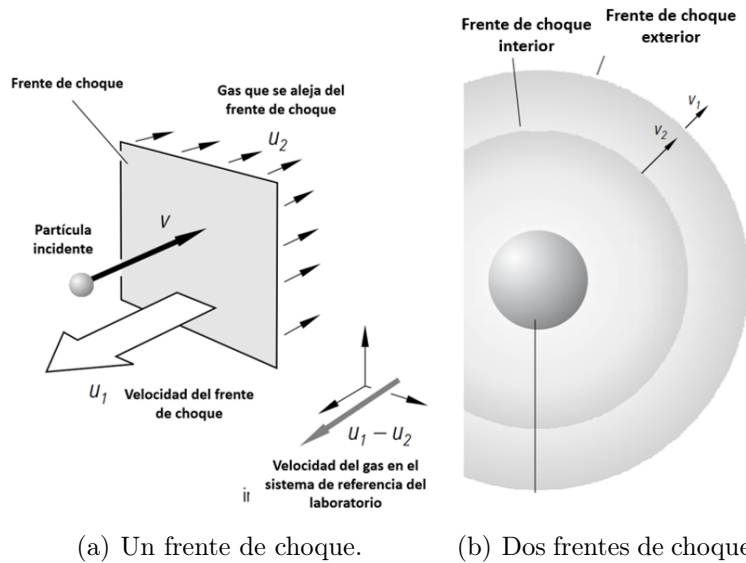


Figura 1.4: Representación gráfica para la aceleración de partículas por uno y dos frentes de choque. Imagen modificada de [5].

Así, que tendremos una ganancia de energía relativa de:

$$\frac{\Delta E}{E} = 2 \frac{u^2}{v^2} \quad (1.14)$$

Debido a que tenemos un término cuadrático en la velocidad de la nube, a este modelo se suele llamar mecanismo de Fermi de segundo orden. Además la ecuación también es válida para el caso relativista [5].

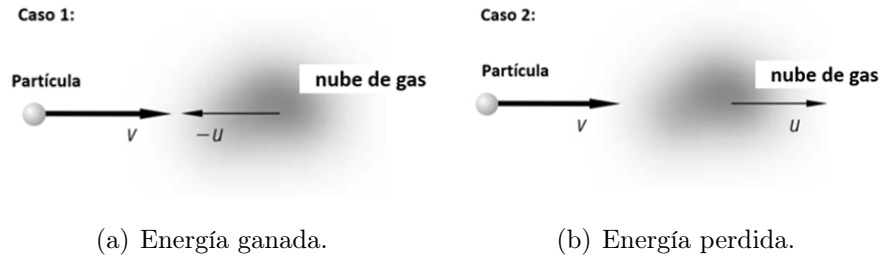


Figura 1.5: Representación de la energía ganada por una partícula debido a una nube magnética de gas. Imagen modificada de [5].

### 1.3.2. Fuentes de rayos cósmicos

Para mencionar los candidatos de fuentes de rayos cósmicos y ver como caracterizarlos, se hará uso de la gráfica de "Hillas" (ver figura 1.6). En dicha imagen, se muestran los distintos sitios de aceleración y se gráfica sus campos magnéticos ( $B$ ) en función de su tamaño físico ( $L$ ). También se observan 3 líneas rectas, estas representan los valores necesarios de  $B$  y  $L$  para acelerar diferentes partículas a 3 distintas energías.

- *Línea roja:* Representa los valores de  $B$  y  $L$  para acelerar protones a energías de 1 ZeV.

- *Línea roja punteada:* Representa los valores de  $B$  y  $L$  para acelerar protones a energías de 100 EeV.
- *Línea verde:* Representa los valores de  $B$  y  $L$  para acelerar núcleos de hierro a energías de 100 EeV.

En el gráfico de "Hillas" las fuentes que se encuentran por encima de unas de estas líneas, pueden acelerar rayos cósmicos a esa energía [7]. Estas fuentes se pueden clasificar en dos categorías:

- *Galácticas:* Son fuentes que se encuentran fuera del sistema solar, pero dentro de la galaxia. Las principales fuentes que se encuentran en esta categoría son: remanentes de Supernovas, nebulosas del viento pulsar, sistemas binarios, púlsares y estrellas jóvenes. La energía máxima que consiguen las partículas aceleradas en estas fuentes todavía es un tema de discusión, sin embargo se cree que solamente se podrían alcanzar energías en el orden de PeV [7].
- *Extragalácticas:* Son fuentes que se encuentran fuera de la galaxia, y se cree que son las responsables del rango de energía más alto de rayos cósmico que se observan en la Tierra. Las fuentes propuestas para integrar esta categoría son: núcleos activos de galaxias, galaxias estelares, cúmulos de galaxias y estallidos de rayos gamma [7].

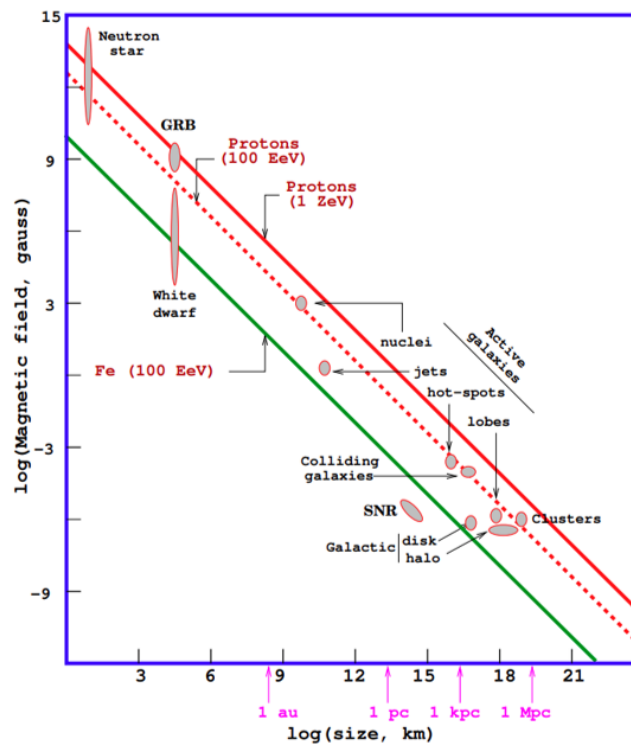


Figura 1.6: Gráfica de Hillas, muestra los candidatos de fuentes de aceleración para rayos cósmicos de 100 EeV a 100 ZeV. Imagen tomada de [7].

### 1.3.3. Espectro de energía

En la imagen 1.7 se muestra una gráfica del espectro de energía de rayos cósmicos que se observa en la Tierra. Dicha gráfica nos indica como se distribuye el flujo de partículas que se detecta con respecto a la energía. Como se puede ver, la cantidad de rayos cósmicos de baja energía es muy abundante y a medida que la energía va aumentando el flujo disminuye.

El comportamiento de este flujo, es posible describirlo con una ley de potencias y diferentes valores del índice espectral  $\alpha$  [24].

$$\frac{dN}{dE} = AE^{-\alpha} \quad (1.15)$$

El espectro abarca energías que van desde los  $10^8$  eV hasta los  $10^{21}$  eV (trece órdenes de magnitud). A continuación se hará una breve descripción de las diferentes zonas que se observan en este espectro:

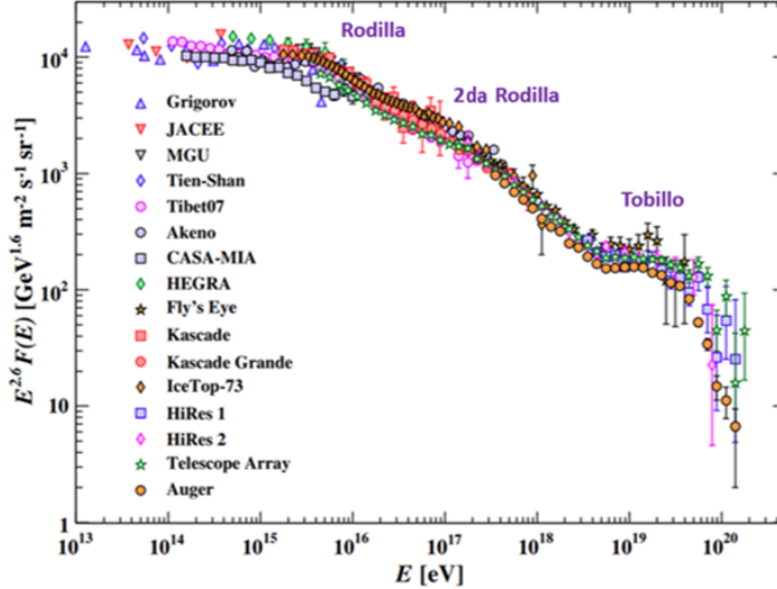


Figura 1.7: Espectro de energía de rayos cósmicos medido con diferentes experimentos. Imagen modificada de [7].

- Para la región con las energías más bajas (menores 1 GeV), se cree que son generados por erupciones solares, ya que el campo magnético solar bloquea la mayoría de las partículas (de este rango de energía) que vienen del exterior del sistema solar [7].
- La región que va de los 100 GeV a los 5 PeV se puede describir por una ley de potencia con un índice espectral de  $\alpha \sim 2.7$ , cabe destacar que la zona con un rango de energía de PeV se le conoce como rodilla. El origen de los rayos cósmicos de esta región (hasta la zona de la rodilla) se cree de origen galáctico y con Supernovas como principales fuentes de emisión de estas partículas [7].
- La región con energías de 5 PeV (aproximadamente) a 0.1 EeV sigue una ley de potencia con un índice espectral de  $\alpha \sim 3$ . Esta región se cree la zona de transición del origen galáctico al extragaláctico [7].
- Para la zona que va de los 0.1 EeV a los 5 EeV, el valor del índice espectral sube  $\alpha \sim 3.3$ . A esta región se le conoce como segunda rodilla, y su origen se asocia a transiciones a rayos cósmicos más pesados [7].
- El índice espectral baja a  $\alpha \sim 2.6$  en la zona de energía de 5 EeV a 40 EeV. A esta transición alrededor de los 5 EeV también se le conoce como tobillo. El origen para estos rayos cósmicos de alta energía se cree de origen extragaláctico [7].





En la figura 1.8 se puede observar un esquema que representa el proceso de formación de una cascada atmosférica generada por un hadrón, en dicha imagen también se muestra la componente electromagnética.

## 1.4. Astronomía de neutrinos

La astronomía basada en la observación de radiación electromagnética tiene el problema de que ese tipo de radiación es rápidamente absorbida por la materia. Además, de que los fotones de alta energía son atenuados por la radiación de fondo de microondas mediante el siguiente proceso:

$$\gamma + \gamma \rightarrow e^+ + e^- \quad (1.19)$$

Para el caso de partículas con carga eléctrica se había comentado que la información de su dirección se pierde, y que solo se mantiene para el caso de protones con alta energía ( $>10^{19}$  eV). Pero en ese rango de energía también hay una atenuación debido al límite GZK [13].

Las ventajas en el estudio de objetos astrofísicos basada en los neutrinos que emiten son las siguientes:

1. Al no tener carga no se desvían por campos electromagnéticos del espacio exterior, así que la información de su dirección se mantiene [5].
2. Los neutrinos son partículas estables que no decaen y pueden llegar a la Tierra desde sus fuentes [5].
3. Los neutrinos y los antineutrinos tienen interacciones distintas con la materia. Esto permitiría (en principio) averiguar si las partículas se originan a partir de una fuente de materia o antimateria [5].
4. Son altamente penetrantes, lo que permite estudiar el centro de las fuentes astrofísica y no solamente su superficie [5].
5. No se atenúan o absorben debido al polvo interestelar o por fotones infrarrojos o de cuerpo negro [5].

La gran desventaja que tiene esta astronomía, está en que los neutrinos tienen una sección transversal muy pequeña así que es difícil poder detectarlos. Para neutrinos solares en el rango de varios cientos de keV su sección transversal para la interacción nucleón-neutrino es de:

$$\sigma(\nu_e, N) \approx 10^{-45} \text{ cm}^2/\text{nucleón}$$

La probabilidad de interacción de estos neutrinos con nuestro planeta en incidencia central es

$$\phi = \sigma N_A d \rho \approx 4 \times 10^{-12}$$

En donde  $N_A$  es el número de Avogadro,  $d$  es el diámetro y  $\rho$  la densidad de la Tierra respectivamente. Por lo tanto, los experimentos para detectar neutrinos consideran objetos de grandes masas (por ejemplo, montañas) y tiempos de exposición prolongados. Sin embargo se debe mencionar que para altas energías, la sección transversal de interacción aumenta. A continuación, se mencionaran las diferentes fuentes de neutrinos que podemos medir en la Tierra.

### 1.4.1. Fuentes de neutrinos

La presencia de neutrinos que podemos detectar en la Tierra proviene de diferentes fuentes y que a continuación se mencionaran las principales:

- **Neutrinos atmosféricos:** Son generados en una cascada atmosférica a través de las líneas de decaimiento mostradas en las ecuaciones 1.16 y 1.17. Cabe mencionar, que los muones generados en estas cascadas también decaen y generan neutrinos:

$$\mu^+ \rightarrow e^+ + \nu_e + \bar{\nu}_\mu \quad (1.20)$$

$$\mu^- \rightarrow e^- + \bar{\nu}_e + \nu_\mu \quad (1.21)$$

Si contamos los neutrinos que se obtienen en estas 4 líneas de decaimiento, se espera tener la siguiente proporción:

$$\frac{N(\nu_\mu \bar{\nu}_\mu)}{N(\nu_e \bar{\nu}_e)} \approx 2$$

Sin embargo, el experimento Super-Kamiokande encontró un déficit en el número de neutrinos del muon. La diferencia entre los datos obtenidos y simulaciones Monte Carlo (MC) se reflejan en el siguiente ratio de proporción:

$$R = \frac{(N_\mu/N_e)_{data}}{(N_\mu/N_e)_{MC}} = 0.69 \pm 0.06$$

El déficit de neutrinos muónicos se puede explicar asumiendo que algunos de los neutrinos muónicos se transforman (oscilaciones de neutrinos) durante la propagación desde el punto de producción hasta el detector en un sabor de neutrino diferente, por ejemplo, en tau neutrinos [5].

- **Neutrinos solares:** El Sol es un reactor de fusión nuclear, en donde el hidrógeno se transforma en helio. La principal cadena de reacciones inicia mediante el siguiente proceso:



El 90% de los neutrinos solares son generados mediante la reacción anterior [27]. El 10% restantes se genera mediante las reacciones mostradas en la tabla 1.2, ahí también se muestra su contribución al flujo total de neutrinos que emite el Sol.

Fuente	Flujo ( $[cm^{-2}s^{-1}]$ )
pp	$5.95 \times 10^{10} (1 \pm 0.01)$
pep	$1.4 \times 10^8 (1 \pm 0.015)$
hep	$9.3 \times 10^3$
${}^7Be$	$4.77 \times 10^9 (1 \pm 0.10)$
${}^8B$	$5.05 \times 10^6 \left(1_{-0.16}^{+0.20}\right)$
${}^{13}N$	$5.48 \times 10^8 \left(1_{-0.17}^{+0.21}\right)$
${}^{15}O$	$4.80 \times 10^8 \left(1_{-0.19}^{+0.25}\right)$
${}^{17}F$	$5.63 \times 10^6 (1 \pm 0.25)$
Total	$6.54 \times 10^{10}$

Tabla 1.2: Flujo de neutrinos solares por reacción. Tabla tomada de [27].

Cabe mencionar que el Sol no produce antineutrinos ni los otros sabores de neutrinos [5].

- **Neutrinos geoterrestres:** Son neutrinos emitidos por la Tierra, estos se generan en las cadenas de desintegración de  $^{238}U$  y  $^{232}Th$ . Los geoneutrinos son antineutrinos de sabor electrónico, y su estudio es un área relativamente nueva con la que se puede obtener información acerca de las profundidades de la Tierra [28].
- **Neutrinos de Supernova:** Cuando una estrella colapsa debido a su propia gravedad, los electrones de la estrella se transforman en neutrones (ver ecuación 1.23) que produce una estrella de neutrones.



Durante el colapso hay una fase en la cual se llega a una temperatura de aproximadamente  $10^{11}K$ , en esta etapa protones térmicos producen pares de electrón-positrón que son absorbidos inmediatamente debido a la alta densidad de la materia circundante. Solamente el proceso de interacción débil con un  $Z$  virtual (ver ecuación 1.24) permite que la energía escape del núcleo en forma de neutrino.



En dicha reacción, los tres sabores de neutrinos se pueden formar.

- **Neutrinos galácticos y extragalácticos:** Anteriormente se ha comentado, que los rayos cósmicos se pueden generar en Supernovas, núcleos activos de galaxias o GRBs [28]. Dado que estas fuentes aceleran protones a muy alta energía pueden colisionar con otras partículas o con fotones dentro de la fuente, debido a estas interacciones se pueden generar piones que a su vez pueden producir neutrinos. Por lo tanto, las fuentes galácticas y extragalácticas que aceleran protones y que se mencionaron anteriormente, también son fuentes de neutrinos. Cabe destacar que la detección de este tipo de neutrinos es muy difícil, debido a que el número de estas partículas que llegan a la Tierra es muy pequeño y habría que distinguirlos entre los neutrinos solares y atmosféricas.

### 1.4.2. Espectro de energía para los neutrinos

En la figura 1.9 se muestra el espectro de energía para los neutrinos. Se puede observar en las energías menores a MeV neutrinos que fueron generados durante el Big Bang. Alrededor de un segundo después de esta gran explosión, debido a diversas interacciones las siguientes reacciones fueron generada:



La temperatura de estos neutrinos primordiales deberá ser de 1.9 K en el presente [5]. Cabe mencionar que es muy difícil que estos neutrinos de baja energía se puedan medir. Por otro lado las observaciones de neutrinos solares se encuentra en el rango de MeV y los neutrinos de Supernova se encuentra en el rango de 10 MeV aproximadamente [5]. También se muestra las energías para el caso de neutrinos atmosféricos. Por últimos, la forma del espectro e intensidad correspondiente a neutrinos por fuentes extragalácticos (AGN) solamente representa una estimación.

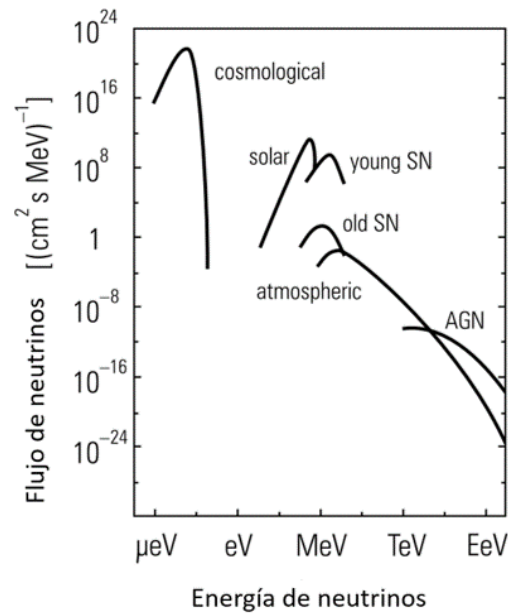


Figura 1.9: Espectro de energía de neutrinos. Imagen tomada de [5].

### 1.4.3. Detección de neutrinos

La forma en la que se puede detectar neutrinos de forma experimental es por medio de sus interacciones. Estas partículas participan en interacciones mediante el intercambio de bosones débiles. De acuerdo a la carga eléctrica del mediador estas interacciones son llamadas de corriente cargada y de corriente neutra con un correspondiente bosón cargado o neutro [29].

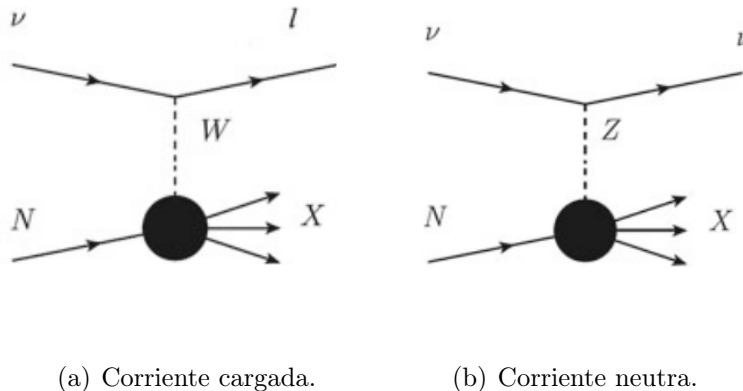


Figura 1.10: Diagrama de Feynman para interacciones de corriente cargada y neutra. Imagen tomada de [30].

- **Interacción de corriente cargada (CC):** En una interacción de corriente cargada, el neutrino se transforma en su compañero leptón (electrón, muón o tau). Sin embargo, si el neutrino no tiene suficiente energía para crear la masa de su compañero más pesado, la interacción de la corriente cargada no podrá realizarse. Estas interacciones están mediadas por bosones W con carga positiva o negativa [29].

$$\nu_l(\bar{\nu}_l) + N \rightarrow_{W_{\pm}} l^-(l^+) + X \quad (1.27)$$

En este caso  $N$  representa el núcleo objetivo y  $X$  los productos restantes. En el inciso a) de la figura 1.10 se muestra el diagrama de Feynman para esta interacción y un ejemplo es la reacción  $\bar{\nu}_e + p \rightarrow n + e^+$  mediada por el intercambio de un bosón  $W_-$  [31].

- **Interacción de corriente neutra (CN):** Esta interacción está mediada por el intercambio de una partícula sin carga eléctrica, llamada bosón  $Z_0$ . En este proceso, el neutrino colisiona transfiriendo parte de su energía y momento a la partícula objetivo. Los tres sabores de neutrinos pueden participar independientemente de la energía que posean sin perder su identidad [29].

$$\nu_l(\bar{\nu}_l) + N \rightarrow_{Z_0} \nu_l(\bar{\nu}_l) + X \quad (1.28)$$

En el inciso b) de la figura 1.10 se muestra el diagrama de Feynman para esta interacción y un ejemplo es la reacción  ${}^2H + \nu \rightarrow p + n + \nu$  mediada por el intercambio de un bosón  $Z_0$  [31].

Cabe mencionar que los neutrinos también interactúan con la materia a través de dispersiones. Así que estas partículas pueden dispersar un núcleo entero liberando un nucleón o múltiples nucleones de un objetivo. Al proceso de la dispersión de neutrinos a través de corriente cargada se conoce como "dispersión cuasielástica", mientras que para la dispersión a través de corriente neutra se le conoce tradicionalmente como "dispersión elástica"[30]. Un ejemplo es la dispersión entre un neutrino y un electrón, este proceso es puramente leptónico en el que un neutrino dispersa un electrón mediante el intercambio de un bosón virtual [32]. En la figura 1.11 se muestra el diagrama de Feynman para este proceso a través de una corriente cargada y una corriente neutra.

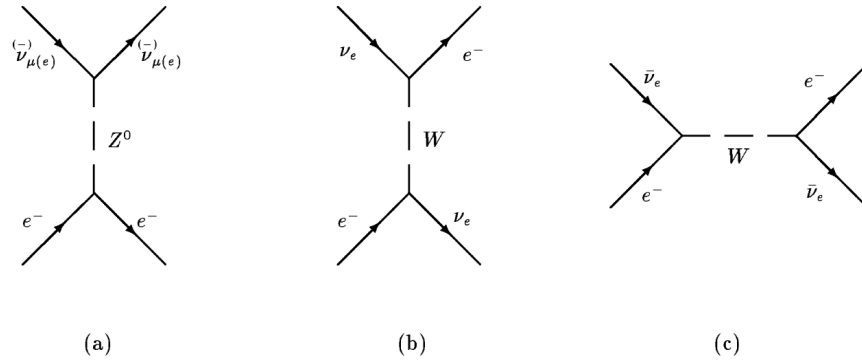


Figura 1.11: Diagrama de Feynman para los procesos de corriente neutra (CN) dispersión  $\nu - e$  (a), y corriente cargada (CC) dispersión  $\nu - e$  por un bosón  $W$  (b,c) . Imagen tomada de [32].

La cinemática para dispersión elástica entre un neutrino y un electrón está completamente descrita por una sola variable; por ejemplo con  $\theta_e$ , el ángulo del electrón saliente con respecto al haz de neutrinos. Sean  $E_\nu$  y  $E_e$  las energías del neutrino inicial y del electrón de salida respectivamente,  $m_e$  la masa del electrón y  $r = E_e/E_\nu$  la fracción de pérdida de energía del neutrino en el sistema del laboratorio [32]. Con la suposición de  $m_e \ll E_e$  y la aproximación para ángulos pequeños de  $\cos \theta_e$  se puede encontrar que:

$$E_e \theta_e^2 = 2m_e(1 - r)$$

y, como  $0 \leq r \leq 1$  la restricción experimentalmente importante será  $E_e \theta_e^2 < 2m_e$ . Esto significa que el electrón saliente se dispersa en una dirección extremadamente hacia adelante, que se usa experimentalmente para extraer sobre una base estadística, eventos de fondo, que tienen una distribución mucho más amplia en  $E_e \theta_e^2$ . Por otra parte, esta distinción hace que sea imposible medir

directamente la distribución  $r$  de las reacciones  $\nu - e$ , ya que las resoluciones alcanzables son justo en el mismo orden de magnitud que el límite cinemático. Cabe mencionar que experimentalmente es complicado observar los efectos de una dispersión neutrino electrón, ya que la sección transversal para esta interacción es muy pequeña ( $\approx 10^{-42} \text{cm}^{-2}$ ). Sin embargo desde los años 1980's sofisticados experimentos fueron desarrollados para esta tarea, teniendo suficiente estadística para medir la distribución cinemática y las secciones transversales diferenciales. Por ejemplo el experimento CHARM II logro medir parámetros electrodébiles con suficiente precisión para probar el modelo estándar en ordenes superiores [32]. Hoy en día la dispersión  $\nu - e$  se usa como una herramienta experimental para detectar neutrinos solares.

A continuación, se comentará de forma breve las principales técnicas mediante las cuales es posible detectar neutrinos.

**Centelladores:** Un centelleador es un material absorbente que tiene la propiedad adicional de convertir en luz una fracción de la energía depositada por la radiación ionizante [33]. Las partículas cargadas interactúan continuamente con los electrones del medio de centelleo a través de interacciones de Coulomb, lo que produce excitación o ionización atómicas. Las partículas neutras primero tendrán que experimentar una interacción directa con algún núcleo produciendo protones de retroceso o por fragmentos, que luego transferirán su energía al medio de la misma manera que las partículas cargadas primarias [33].

Los centelladores son sensibles a los antineutrinos electrónicos que inician la desintegración beta inversa de un protón ( $\bar{\nu}_e + p \rightarrow e^+ + n$ ) al ser compuestos orgánicos (cabe mencionar que también hay centelladores inorgánicos), estos materiales son ricos en núcleos de hidrógeno que actúan como objetivos para esta reacción. El positrón se aniquila rápidamente y produce dos rayos gamma; el neutrón se captura en un núcleo después de un corto tiempo (unos pocos microsegundos a unos pocos cientos de microsegundos), produciendo otra señal de rayos gamma. Esta coincidencia de una señal rápida (cuya energía proporciona la energía del antineutrino) y una señal retardada (cuya energía es característica del núcleo que captura el neutrón) permite que el experimento rechace el fondo de forma eficaz [31].

Los detectores de centelleo líquido tienen una buena resolución de tiempo y energía, pero no conservan la información direccional. En cierto modo, un centelleador puede considerarse como un cambiador de longitud de onda. Convierte la energía (o longitud de onda) de una partícula incidente o fotón energético (UV, rayos X o gamma-ray) en varios fotones de energía mucho menor (o longitud de onda más larga) en el rango visible o casi visible, que puede ser detectado por fotomultiplicadores, SiPM's ( Silicon Photomultiplier), fotodiodos o fotodiodos de avalancha [33].

Fotomultiplicadores: Estos aparatos se han utilizado para detectar fotones de baja energía en el rango de UV a visible, fotones de alta energía (rayos X y rayos gamma) y partículas ionizantes utilizando un centelleador [34]. En la figura 1.12 se muestra un diagrama de este instrumento y su funcionamiento se puede explicar mediante los siguientes pasos:

1. Los fotones incidentes excitan a los electrones del fotocátodo y arrancan fotoelectrones debido al efecto fotoeléctrico. La eficiencia cuántica describe la fotoemisión por parte del cátodo. Esta se define como la razón entre el número de fotoelectrones emitidos por el cátodo al número de fotones incidentes en la ventana, y generalmente se expresa como un porcentaje [34]. Esta fotoemisión tiene lugar de acuerdo con un cierto proceso de probabilidad. Los

fotones en longitudes de onda más cortas transportan mayor energía en comparación con aquellos en longitudes de onda más largas y contribuyen a un aumento en la probabilidad de fotoemisión. Como resultado, la máxima eficiencia cuantitativa se produce a una longitud de onda ligeramente más corta que la longitud de onda de la sensibilidad radiante máxima. La eficiencia cuántica depende de la longitud de onda de los fotones y generalmente es inferior al 35 % [34].

2. Los fotoelectrones son acelerados y focalizados hacia el primer dinodo por el electrodo de focalización.
3. Los fotoelectrones inciden sobre el dinodo y este emite electrones secundarios.
4. El proceso se repite en los sucesivos dinodos.
5. Los electrones secundarios salen del último dinodo y llegan al ánodo, obteniéndose así la corriente de salida.

La señal de salida está formada por pulsos de corriente, esta es procesada y enviada al sistema de adquisición.

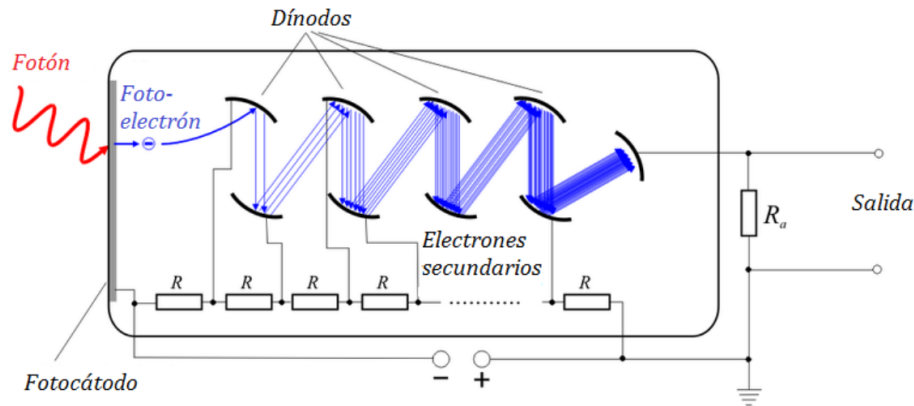


Figura 1.12: Diagrama de un tubo fotomultiplicador . Imagen tomada de [34].

**Experimentos:** Este tipo de detectores se han usado para estudiar la desaparición de  $\nu_e$  de reactores nucleares y para estudiar las oscilaciones de neutrinos solares (BOREXINO), además de estudiar la conversión de  $\nu_\mu$  en  $\nu_e$  en experimentos de larga distancia como NoVA y MINOS. También está los experimento KamLAND, MiniBooNE LENA y CLEAN (estos últimos 2 aun son propuestas) [31]. Los ejemplos antes mencionados solo son algunos de los experimentos que usan este tipo de materiales para la detección de neutrinos aunque la lista es aun mas grande.

**Métodos radioquímicos:** En este método un neutrino es capturado por un átomo de un isótopo radiactivo, que por medio de una desintegración beta dicho isótopo se convierte en otro elemento. Por ejemplo, en la siguiente reacción:  ${}^{71}\text{Ga} + \nu \rightarrow {}^{71}\text{Ge} + e^-$ . En este caso, el isótopo producido es inestable y volverá al elemento original. Los neutrinos se cuentan extrayendo el producto y observando estas desintegraciones [31].

- **Desventajas:** No se puede obtener información acerca de la dirección y la energía queda restringida a la del neutrino detectado, además de que se tiene una resolución del tiempo muy pobre que se encuentra en el orden de semanas.

- Experimentos: Homestake, SAGE y GALLEX/GNO [31].

**Detectores Cherenkov:** Este tipo de detectores usa la interacción de un neutrino con un medio (que puede ser agua, por ejemplo). Debido a dicha interacción se generan partículas cargadas que viajan más rápido que la luz en el medio utilizado, esto genera una onda de choque electromagnética. En el siguiente capítulo se hablará un poco más acerca de este efecto. Entre los experimentos que utilizan esta técnica son: Super-Kamiokande, IceCube, ANTARES y Baikal [31].

**Detectores de rastreo:** Con estos detectores se reconstruye el camino o traza de los leptones cargados que son producidos por interacciones de corrientes cargadas. En estos dispositivos también se emplean campos magnéticos que hacen que la trayectoria de la partícula se curve, permitiendo que el impulso y su signo de carga se puedan reconstruir [31].

- Ventajas: Este tipo de detectores se adaptan mejor a los neutrinos de mayor energía y son buenos para distinguir diferentes topologías de eventos en donde se involucran múltiples partículas.
- Experimentos: Actualmente en algunos experimentos se están utilizando los detectores de cámara de proyección de tiempo de argón líquido (LAr TPC por sus siglas en inglés *liquid Argon Time Projection Chamber*). La propuesta de estos detectores fue realizada por C. Rubbia en 1977 como una herramienta para obtener imágenes uniformes y de alta precisión de volúmenes de detectores masivos. El principio de funcionamiento del LAr TPC se basa en el hecho de que las trazas de ionización altamente purificadas se pueden transportar sin distorsiones mediante un campo eléctrico uniforme a distancias del orden de metros. El argón líquido es un medio ideal ya que proporciona alta densidad, altos rendimientos de ionización y centelleo, y es intrínsecamente seguro y económico. La lectura no destructiva de electrones de ionización por inducción de carga permite detectar la señal de los electrones que cruzan planos de cables subsecuentes con diferente orientación. Esto proporciona varias vistas proyectivas del mismo evento, lo que permite la reconstrucción de puntos espaciales (reconstrucción en 3D) y una medición calorimétrica precisa. Algunos experimentos que utilizan estos detectores son ICARUS, MicroBooNE, y SBND (en construcción). Cabe mencionar que estos dos últimos en realidad usan técnicas combinadas, MiniBooNE es principalmente un detector Cherenkov de aceite mineral, pero también observa el centelleo producido en el medio en menor cantidad. Por otro lado SBND usará una TPC de argón líquido (LAr) pero también observará el centelleo del LAr con un arreglo de fotomultiplicadores.

**Detectores de emulsión fotográfica:** Estos detectores están generalmente en forma de una placa de vidrio finamente recubierta con un medio transparente que contiene un compuesto de haluro de plata [35]. El paso de partículas subatómicas cargadas se registra en la emulsión de la misma manera que una película fotográfica en blanco y negro ordinaria registra una imagen. Después del revelado fotográfico queda un registro permanente de las trayectorias de las partículas cargadas y se puede observar a través de un microscopio [35].

Experimentos: El experimento OPERA (*Oscillation Project with Emulsion tRacking Apparatus*) es un ejemplo que utiliza esta técnica para la detección de neutrinos; en particular, usa una cámara de nube de emulsión (ECC por sus siglas en inglés *Emulsion Cloud Chamber*). El desafío del experimento es medir  $\nu_\tau$  que provienen de la oscilación de  $\nu_\mu$ . Para ello se requiere detectar un leptón  $\tau$  producido en la interacción de corriente cargada de un  $\nu_\tau$  [35]. Este experimento tiene aproximadamente 9 millones de películas de emulsión nuclear, tienen una área de  $10.2 \times 12.5 \text{ cm}^2$  y



está compuesta por una base plástica de  $205 \mu m$  con una capa sensible de  $44 \mu m$  en ambas caras de la base; además tiene un contenido de  $AgBr$  superior a las emulsiones fotográficas comerciales. La técnica ECC permite construir un detector modular que tiene alta masa (plomo) y alta resolución espacial y angular (emulsión) [36].

Por último mencionaremos que en el capítulo 3 se explica de forma un poco más detallada la forma en la que se plantea detectar neutrinos ultra energéticos por medio del observatorio HAWC.

# Capítulo 2

## Observatorio HAWC

El observatorio HAWC (*High Altitud Water Cherenkov*) es un instrumento de investigación conformado por un arreglo de 300 detectores de Cherenkov. Este arreglo está en operaciones desde el 20 de marzo del 2015 [1] y con él se pueden detectar cascadas atmosféricas producidas por rayos gamma y rayos cósmicos en un rango de cientos de GeV a cientos de TeV (aproximadamente) [7]. Se ubica en México dentro del Parque Nacional Pico de Orizaba en el estado de Puebla y está a una altura de 4100 m sobre el nivel del mar. Es importante mencionar que HAWC tiene un campo instantáneo de visión de 2 sr del cielo entre las declinaciones  $\delta \in [-26,+64]$ , cubriendo dos terceras partes del cielo cada 24 hrs y tiene un ciclo útil mayor al 95 por ciento [7].

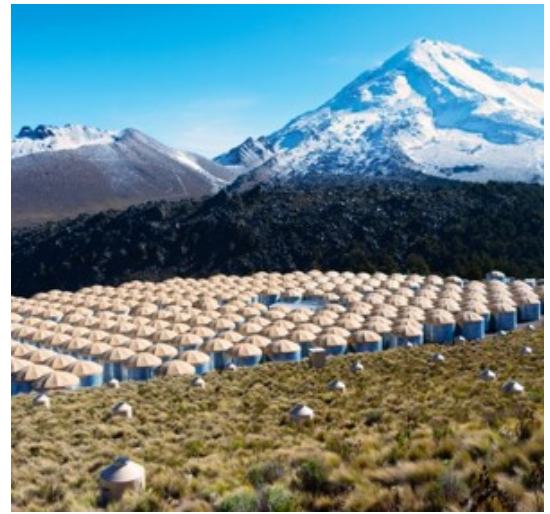


Figura 2.1: Imagen del observatorio HAWC. Tomada de [37].

A finales del 2018, HAWC completó una importante actualización que consistió en la adición de un arreglo de 345 pequeños tanques Cherenkov (los cuales denominaremos como *outriggerer*) que rodean a los 300 detectores originales [37]. Con esta actualización se mejora la reconstrucción del núcleo de la cascada y se incrementa el área efectiva en un rango de unos pocos TeV hasta más allá de 100 TeV [38].

En la imagen se muestra una foto del observatorio HAWC, en donde se puede observar que esta rodeado por dos montañas (el volcán Sierra Negra y el Pico de Orizaba) además de mostrar los detectores Cherenkov principales y los *outriggers*.

### 2.1. Detectores utilizados en HAWC

Actualmente el observatorio HAWC cuenta con dos tipos de detectores. Los *detectores de agua Cherenkov* (WCD<sup>1</sup>) y los detectores exteriores *outrigger*. A continuación se mencionara una descripción para cada uno de ellos.

---

<sup>1</sup>Por sus siglas en inglés *Water Cherenkov Detector*

### 2.1.1. Detectores Cherenkov de agua (WCD)

Este tipo de detector está formado por un cilindro hecho de acero corrugado que tiene una altura de 5 m y un diámetro de 7.3 m. En la parte superior se encuentra una cúpula de lona protectora cuya función es evitar el acumulamiento de agua y nieve. En el interior del cilindro se encuentra una bolsa que contiene aproximadamente 200 mil litros de agua ultra purificada que llega a una altura de 4.5 m del tanque [7]. En la parte inferior y dentro de esta bolsa se encuentra 4 fotomultiplicadores (PMTs) distribuidos tal y como se muestra en el inciso a) de la figura 2.2, dichos instrumentos son sensibles a las longitudes de onda en el rango visible y ultravioleta [1]. El PMT central es el modelo R7081 de la marca *Hamamatsu*, este dispositivo posee una alta eficiencia cuántica (el pico mencionado por el fabricante es del 40 % aproximadamente) y un diámetro de 10 pulgadas. Los fotomultiplicadores periféricos son del modelo R5912 de la misma marca y tienen un diámetro de 8 pulgadas con una eficiencia cuántica del 25 % aproximadamente [7].

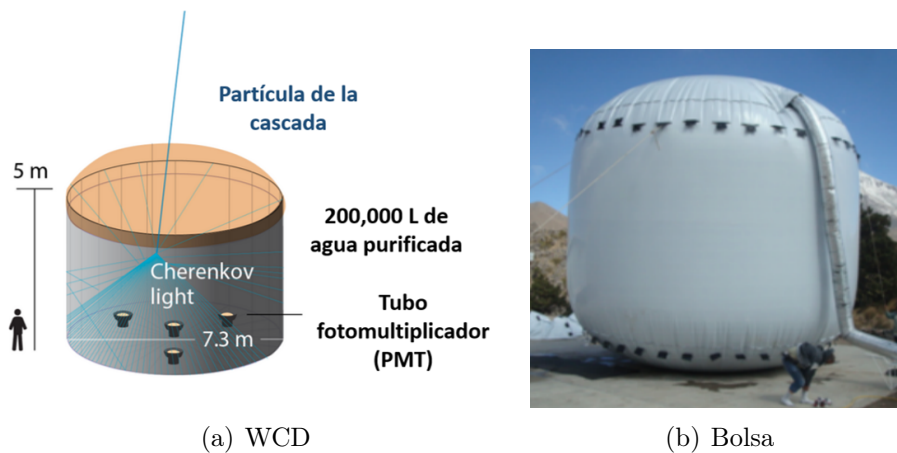


Figura 2.2: a) representación de un WCD, en donde se muestra una partícula que genera un cono de luz debido al efecto Cherenkov que es captada por los PMTs que se encuentran en el tanque. b) Foto de la bolsa utilizada para almacenar el agua. Imagen modificada de [7].

Cuando una partícula cargada incide en estos detectores, genera un cono de luz que podrá ser detectada por alguno de los cuatro fotomultiplicadores que se encuentran en el tanque. Dicho cono de luz se genera debido al el efecto Cherenkov. Este efecto ocurre cuando una partícula cargada se propaga en un medio dieléctrico más rápido que la velocidad de la luz en el mismo medio, dando como resultado la emisión del cono luz. La partícula incidente que se encuentra en estas condiciones forma una onda de choque, es decir, que las ondas que se van emitiendo por todo el camino de la partícula estarán en fase, y se tendrá una interferencia constructiva, dando como resultado un campo distinto de cero. Esto se puede entender mediante el principio de Huygens, en donde la radiación emitida solo se ve en un ángulo en particular, tal y como se ve en el inciso a de la figura 2.3, en donde las ondas generadas en los puntos  $P_1, P_2$  y  $P_3$  son coherentes y se combinan en un frente de onda plano en el segmento BC. Esta coherencia ocurre cuando la partícula recorre el segmento AB en el mismo tiempo en que la luz recorre el segmento AC [39]. Teniendo en cuenta lo anterior podemos expresar la longitud de dichos segmentos de la siguiente forma:

$$\begin{aligned}
 AB &= (\text{Velocidad de la Partícula})(\text{Tiempo}) = (\beta c)(\Delta t) \\
 AC &= (\text{Velocidad de la luz})(\text{Tiempo}) = (c/n)(\Delta t)
 \end{aligned}$$

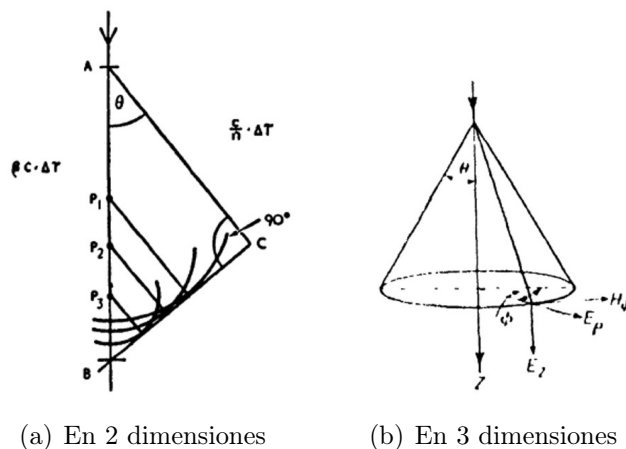


Figura 2.3: Emisión de la radiación Cherenkov. Imagen tomada de [39].

Por lo tanto se tendría que:

$$\cos \theta = \frac{AC}{AB} = \frac{1}{\beta n}$$

Así que el efecto Cherenkov tendrá las siguientes características:

- Para un medio con índice de refracción  $n$  habrá una velocidad umbral para que se pueda emitir radiación  $\beta_{min} = \frac{1}{n}$
- Para una partícula relativista con  $\beta = 1$  habrá un ángulo máximo de emisión  $\theta_{max} = \cos^{-1}(1/n)$
- La emisión de esta radiación formará un cono tal y como se observa en el inciso b de la figura 2.3.

Para el caso del agua con  $n = 1.33$  se puede ver que es necesario una  $\beta > 0.75$  para generar luz, además se obtiene un ángulo máximo de emisión de  $41.2^\circ$ . Los tanques de HAWC tienen un nivel de agua aproximadamente de  $z = 4 \text{ m}$ , tomando en cuenta que en este líquido se tiene una longitud de radiación de  $X_0 = 36.08 \text{ cm}$  [40] se puede estimar la energía de la componente electromagnética al llegar al fondo del tanque. Para ello utilizamos el modelo de Heitler en donde la energía de esta componente a una profundidad  $z$  será  $E = E_0/2^{z/X_0}$ , en donde  $E_0$  es la energía inicial de la partícula. Así que al fondo del tanque solo se tendrá  $1/2174.5$  de  $E_0$ . Para un electrón con una energía de  $2 \text{ keV}$  se tendrá que las partículas generadas en la cascada al fondo del tanque habrán reducido su energía a  $1 \text{ eV}$ . Por lo tanto esta profundidad en los tanques de HAWC asegura que la componente electromagnética de la cascada perderá toda su energía antes de llegar a la parte inferior del tanque y con ello se tendrá luz uniforme en los PMTs [7].

### 2.1.2. Outrigger

Son 345 tanques en forma de cilindro con un diámetro de  $1.55 \text{ m}$  y una altura de  $1.65 \text{ m}$ . Cada uno de estos tanques tiene un fotomultiplicador de un diámetro de  $8$  pulgadas y los tanques están separados entre sí por una distancia de  $12$  a  $18$  metros. Estos detectores se ubican alrededor del arreglo principal. Para fines del *trigger* y la adquisición de datos, estos detectores se dividen en  $5$  secciones con  $69$  *outriggers* en cada sección (ver figura 2.4). Cada detector se conecta con un nodo

que es el responsable del sistema de activación (*trigger*) y proporcionar energía eléctrica, además estos aparatos también se encargan de la calibración en cada una de las 5 secciones [7].

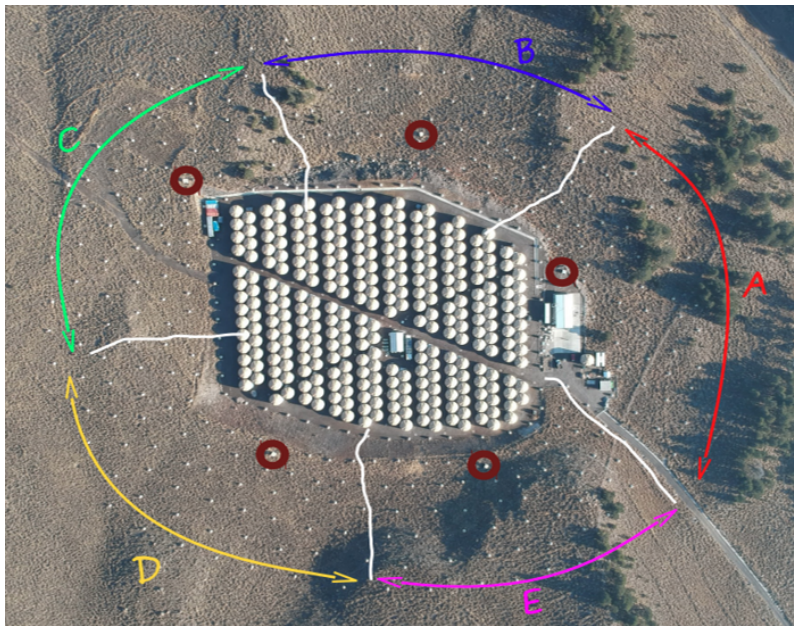


Figura 2.4: Las líneas blancas muestran las diferentes secciones (A, B, C, D y E) para el arreglo de los *outriggers*. Los círculos de color rojo oscuro muestran las ubicaciones de los nodos. Imagen tomada de [7].

## 2.2. Sistema de adquisición de datos

Se describirá de forma breve como es el sistema de adquisición de datos (DAQ<sup>2</sup>) en el observatorio HAWC. Todo lo correspondiente a la electrónica para este sistema se encuentra en una caseta en el centro del arreglo a la cual se le conoce como *counting house* y que se puede ver en la imagen 2.4. La forma en que funciona esta adquisición de datos se puede describir mediante los siguientes pasos:

1. La señal obtenida por los fotomultiplicadores se envía por medio de un primer cable coaxial RG-59 de una longitud de 65 ft, pasa por un conector SHV (*safe high voltage*) y continua por un segundo cable coaxial que está bajo tierra y tienen una longitud de 490 ft. Este llega a un módulo de protección para después ser conectado a un panel electrónico (*front-end-boards FEB*) por un tercer cable de 50 ft [41]. Esta conexión entre cables se puede ver representada en la imagen 2.5
2. La señal del PMT pasa por una caja desacopladora y con protección de descargas, después llega a un conjunto de placas electrónicas analógicas frontales (FEBs). En estos dispositivos primero se separa la señal del PMT de la línea de alto voltaje DC usando un capacitor (ver diagrama 2.6). Después hay una resistencia que ayuda a terminar la transmisión. Una vez que la señal pasó por dicha resistencia, va hacia a dos amplificadores para después pasar por un chip que compara la señal con los valores de un umbral alto (50 mV) y un umbral bajo (20 mV) [41]. Al final, las señales que pasan por estos umbrales son enviadas a placas digitales FEBs.

<sup>2</sup>Por sus siglas en inglés *Data Acquisition System*

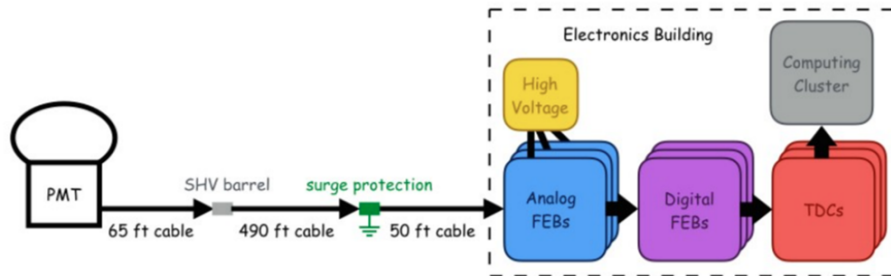


Figura 2.5: Esquema de la conexión entre PMTs y el DAQ principal. Imagen tomada de [41].

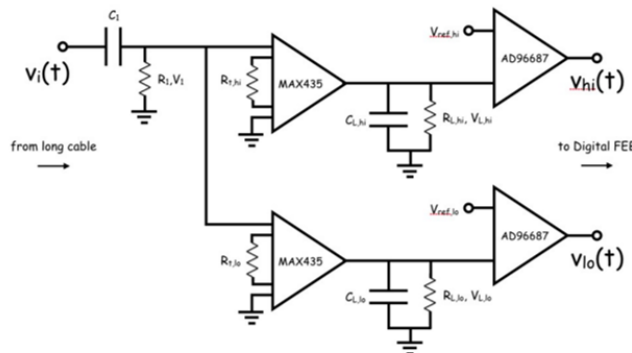


Figura 2.6: Diagrama simplificado para un FEB analógico. Imagen tomada de [41].

3. Las placas digitales FEBs, aceptan las salidas que pasaron por los umbrales mencionados en el paso anterior. A esta señal se le aplica un circuito lógico que combina el umbral bajo y alto en una sola forma de onda (*waveform*). Esta señal se procesa con un solo canal de convertidor de tiempo a digital (TDC) [41].
4. Los tiempos en los que las señales pulsadas de los PMT cruzan cada umbral se registran utilizando módulos de convertidor de tiempo a digital (TDC) CAEN VX1190A que tienen una precisión de 100 ps [7]. Esto se logra debido a que estos aparatos están conectados a un sistema GTC (*GPS Timing Control*) [7]. Dicho sistema ajusta un tiempo a los eventos registrados. Dependiendo del umbral cruzado por la señal observada (20 mV o 50 mV) [43], se caracteriza por un tiempo sobre el umbral (*Time over Threshold ToT*) alto o bajo. En la figura 2.7 se muestra un ejemplo en donde la señal paso por el umbral alto, por lo tanto se le asocia un ToT alto.
5. Mediante la técnica ToT se obtiene la carga depositada en cada PMT por evento, ya que la duración de la señal sobre el umbral (ToT) se puede poner en función del logaritmo de la carga generada en el PMT (Esta relación se puede observar en la imagen 2.8). En HAWC se consideran dos umbrales, un *high ToT* y un *low ToT*, para obtener la carga depositada en los PMTs.

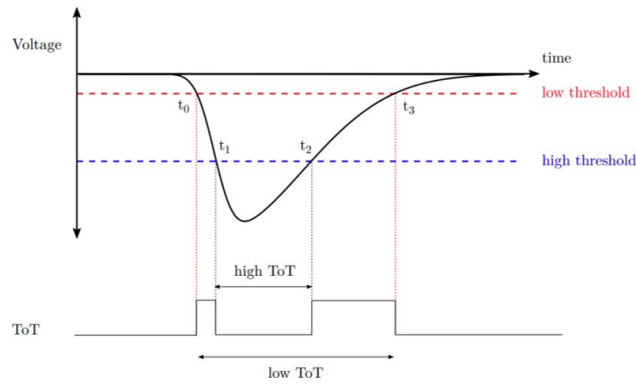


Figura 2.7: Gráfica que muestra la salida de la señal dada por el FEB, en donde se ve que pasa el umbral alto, así que se le asocia un ToT alto. Imagen tomada de [42].

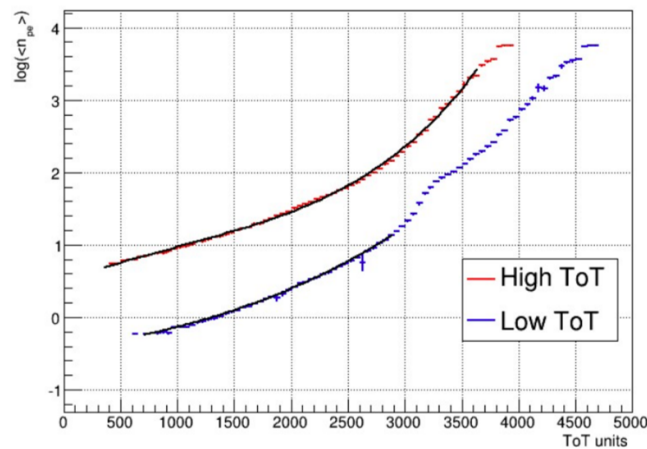


Figura 2.8: Gráfica en donde el logaritmo del número de fotoelectrones está en función del ToT. Imagen tomada de [43].

6. Por último la información de carga y tiempo obtenida por el TDC de cada PMT se guarda en un disco duro en formato XCDF (*The eXplicitly Compacted Data Format*), siempre y cuando esta cumpla con la condición de *trigger*.

*Trigger*: La condición de disparo consiste en que se activen al menos 28 PMT's en una ventana de tiempo de 150 ns [44].

Cabe destacar que después de aplicar el *trigger* la tasa de recolección de datos es de 15 – 20 MB/s. Esto resulta en una tasa de eventos de 24kHz y 2 TB de datos por día [44].

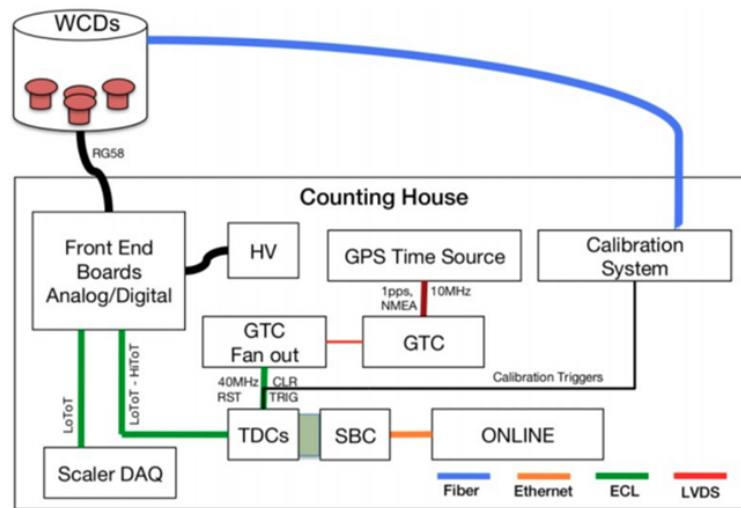


Figura 2.9: Esquema que muestra los dos sistemas de adquisición de datos en HAWC. Imagen tomada de [44].

Cabe destacar, que los pasos anteriores describen el funcionamiento del sistema de adquisición de datos principal del observatorio HAWC. Sin embargo, en este observatorio se encuentra otro sistema de adquisición de datos al que se le conoce como “*Scalar system*”. Este sistema cuenta el número de señales en cada PMT dentro de una ventana de 10 ms, con estos datos se hace una reconstrucción rápida utilizando el equipo de cómputo que hay en el *counting house* [7]. Este sistema se utiliza con el propósito de monitorear eventos de rayos gamma transitorios, AGN, GRBs y fluctuaciones en la tasa de rayos cósmicos de la actividad solar. Por lo tanto, si hay algún evento de interés se puede notificar a otros laboratorios [7]. En la imagen 2.9 se muestra un diagrama en general de los dos sistemas de adquisición de datos que se encuentran en HAWC.

### 2.2.1. Almacenamiento

Después de que se ha hecho la reconstrucción de las cascadas atmosféricas (o eventos), toda la información recabada es almacenada en archivos de tipo XCDF para su posterior análisis. Actualmente, de cada evento se guardan alrededor de 64 columnas correspondientes a diferentes variables [45]. Los datos son recabados por día en directorios llamados *runs*. Cada *run* se divide en *subruns* que son archivos XCDF que guardan todos los eventos dentro de una ventana de 2 minutos (recordemos que la tasa de eventos es de 24 KHz). Entonces cada *run* contiene un máximo de 692 *subruns* [45].

## 2.3. Reconstrucción de eventos

Como se mencionó anteriormente, una partícula del espacio exterior causa una cascada de partículas secundarias al chocar con las partículas que se encuentran en la atmósfera. En esta sección se explica la metodología que se usa en el observatorio HAWC para reconstrucción la dirección y la energía de la partícula que inicio la cascada. Cuando un evento es registrado por este observatorio, primero se hace una reconstrucción *online*. Esta reconstrucción se realiza en las instalaciones del observatorio y tiene como objetivo obtener resultados preliminares para identificar eventos transitorios [7]. Después se realiza una reconstrucción *offline* que es mas sofisticada y que



arroja una mejor reconstrucción. A continuación, se mencionará a grandes rasgos en que consiste esta reconstrucción *offline*, en donde se reconstruye el núcleo de la cascada, su dirección y su energía.

### 2.3.1. Núcleo de la cascada

El eje de la cascada es la trayectoria original de la partícula antes de interactuar con las partículas de la atmósfera. Así que el núcleo de la cascada es la intersección entre este eje y la sección del suelo en donde se deposita la máxima cantidad de energía (ver figura 2.10). Para hacer la reconstrucción del núcleo se utiliza una función que describe la distribución de carga lateral a lo largo del eje de la cascada. En HAWC se utiliza la función SFCF (Super Fast Core Fit) la cual se muestra en la siguiente expresión:

$$Q_i = A \left( \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|x-x_i|^2}{2\sigma^2}\right) + \frac{N}{\left(0.5 + \frac{|x-x_i|}{R_{MOL}}\right)^3} \right) \quad (2.1)$$

En donde:

- $x$  representa la posición del núcleo,  $x_i$  es la posición del  $i$ -ésimo PMT y el valor de  $A$  representa a la amplitud general.
- $\sigma$  es el ancho de la distribución Gaussiana, para las características de este observatorio se toma un valor de 10 m.
- $N$  es la normalización de la cola de la distribución NKG<sup>3</sup> (en el apéndice B se muestra más información sobre esta función). Esta tiene un valor de  $5 \times 10^{-5}$ .
- $R_{Mol}$  es el radio de Molière. Esta cantidad se define como el radio de un cilindro que contiene en promedio el 90 % de la energía depositada por la cascada [46]. Para HAWC esta cantidad es de aproximadamente 124 m.

Para hacer la reconstrucción del núcleo, se utilizan métodos numéricos para encontrar los valores de la posición del núcleo que minimicen la función chi cuadrada.

$$\chi^2 = \sum_{i=1}^{N_{PMT}} \frac{(q_i - Q_i)^2}{Q_i + \delta q_i^2} \quad (2.2)$$

El valor de  $q_i$  representa la carga medida por el  $i$ -ésimo PMT y su correspondiente incertidumbre está representada por el valor de  $\delta q_i$ . Cabe mencionar, que las mediciones de la carga obtenidas por los fotomultiplicadores están gobernadas por una estadística Poissoniana. Así que la incertidumbre de la carga esperada se podrá calcular como  $\sqrt{Q_i}$ . Por lo tanto, la incertidumbre total (valor del denominador de la ecuación 2.2) se calcula como la suma en cuadratura de la incertidumbre de la medición y de la carga esperada.

---

<sup>3</sup>Por sus siglas Nishimura-Kamata-Greisen

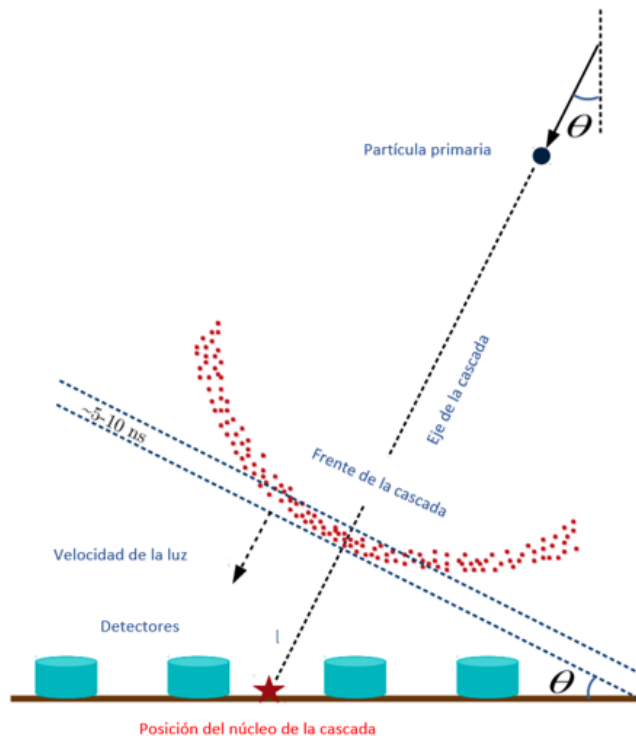


Figura 2.10: Esquema que muestra al núcleo de la cascada. Imagen modificada de [7].

### 2.3.2. Dirección

Como primera aproximación, se considera que el frente de la cascada es plano y viaja a una velocidad igual a la de la luz [46]. Con dicha consideración se puede reconstruir el eje de la cascada usando los tiempos de los PMTs que fueron activados. Sin embargo, se tienen que hacer dos importantes correcciones:

- La primera tiene que ver con la curvatura de la cascada. Las partículas más alejadas del núcleo viajan distancias más largas en comparación con las que se encuentran cerca de este. Sumado a su baja energía, sufrirán como resultados múltiples dispersiones que provocaran su atenuación [46]. Esto da como resultado un frente de cascada que se curva en los bordes. Por lo tanto, hay un retraso en los tiempos de los PMTs activados por las partículas de los bordes, ya que se consideró que el frente de onda fue plano. Así que se tiene que hacer una corrección para estos tiempos.
- Debido a la baja densidad de partículas en los bordes del frente de la cascada, se debe hacer otra corrección en los tiempos de muestreo [46].

Las dos correcciones anteriores ascienden a unos pocos ns para una distancia de 100 m desde el núcleo y se han optimizado con simulaciones y observaciones de la nebulosa del Cangrejo. Después de corregir estos efectos de bordes en los tiempos de los PMTs, el frente de la cascada se ajusta a un plano de llegada para poder estimar el ángulo cenital de llegada local ( $\theta$ ) y el ángulo azimutal ( $\phi$ ) [46].

### 2.3.3. Energía

Para la estimación de la energía de la partícula que generó la cascada, en HAWC se utilizan dos métodos que se explicaran a continuación:

- **Ground parameter (GP):** Para estimar la energía usando este método primero se calcula el radio óptimo ( $r_0$ ). Este radio se define como la distancia que hay entre el eje de la cascada y el lugar en donde se minimiza la incertidumbre de la densidad de energía [46]. Para HAWC se tiene un valor de  $r_0 = 40$  m y no dependen de la energía inicial de la partícula. El siguiente paso es evaluar al logaritmo de la función NKG ( $s_r$ ):

$$\begin{aligned} \text{Log}_{10}(s_r) = & \text{Log}(A) + s[\text{Log}_{10}(r/124.21) + \text{Log}_{10}(1 + r/124.21)] - 3\text{Log}_{10}(r/124.21) \\ & - 4.5\text{Log}_{10}(1 + r/124.21) \end{aligned} \quad (2.3)$$

para nuestro valor de  $r_0$  se tendrá como resultado la siguiente expresión:

$$\text{Log}_{10}(s_{r_0}) = \text{Log}_{10}(A) - 0.371024s + 0.930696 \quad (2.4)$$

En donde A es la amplitud del ajuste y  $s$  es un parámetro relacionado con la edad de la cascada. Después se hace el mejor ajuste de la ecuación anterior a los datos obtenidos en un evento [47]. El siguiente paso es asociar la energía con la señal a  $r_{40}$ . La técnica básica es usar simulaciones para crear un histograma 2D en donde se tenga en un eje el valor de  $\text{Log}_{10}(s_{r_0})$  y en el otro el valor del logaritmo de la energía real  $\text{Log}_{10}(E_{real})$ . La idea consiste en encontrar el mejor ajuste entre estas relaciones, así que buscamos una función  $f$  que dependa del logaritmo de  $s_{r_0}$  y que sea igual al logaritmo de la energía.

$$\text{Log}_{10}(E_{real}) = f(\text{Log}_{10}(s_{r_0})) \quad (2.5)$$

Sin embargo, la relación entre carga y energía depende del ángulo cenital del rayo gamma primario, debido a las cascadas atmosféricas con diferentes inclinaciones que deben atravesar antes de llegar al detector. Por lo tanto, la función de energía se convierte en una función tanto de la carga como del ángulo cenital:

$$\text{Log}_{10}(E_{real}) = f(\text{Log}_{10}(s_{r_0}), \theta) \quad (2.6)$$

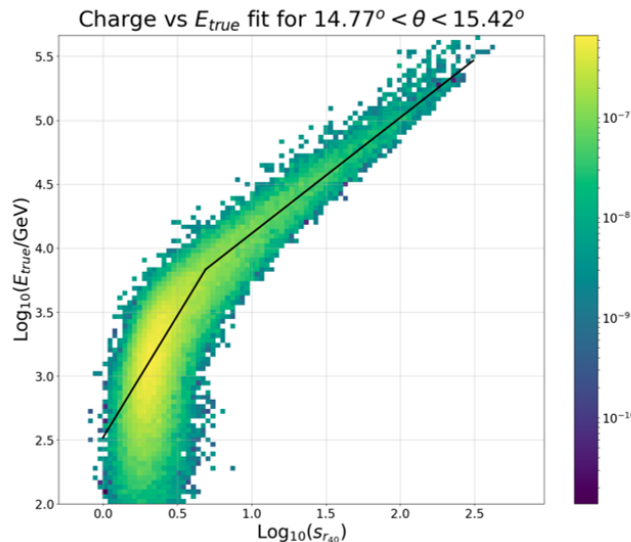


Figura 2.11: Correlación entre carga y energía para un ángulo cenital ( $14.77^\circ < \theta < 15.42^\circ$ ) [47].

En la imagen 2.11 se colocó un ejemplo de este histograma que muestra la correlación entre la carga efectiva y la energía  $E_{true}$  para un pequeño ángulo cenital ( $14.77^\circ < \theta < 15.42^\circ$ ). La línea de color negro representa al mejor ajuste. Los colores en el histograma denotan las probabilidades relativas asumiendo una fuente isotrópica en una declinación de  $20^\circ$ . Este ajuste se repite para cada intervalo cenital de modo que los mejores parámetros de ajuste se puedan parametrizar por el ángulo cenital [47].

- **Neural Network (NN):** Con esta técnica se implementa una red neuronal artificial para estimar la energía de eventos iniciado por rayos gamma. El modelo de red utilizado es un perceptrón multicapa que tiene dos capas ocultas y una función de activación de tipo logística. En la primera y segunda capa oculta se tienen 15 y 14 neuronas respectivamente [48]. La función de costo a minimizar para esta red es la siguiente:

$$D(w) = \frac{1}{2} \sum_{i=1}^n u_i \left[ \text{Log}_{10} \hat{E}(X_I; W) - \text{Log}_{10} E_i \right]^2 \quad (2.7)$$

En donde  $w$  representa a los pesos,  $n$  es el número de eventos,  $\hat{E}$  es la función para estimar la energía y que arroja la red,  $E_i$  es la energía simulada del evento  $i$ -ésimo y  $u_i$  es la importancia relativa del evento  $i$ -ésimo. Este valor de  $u_i$  se escoge para parecerse a un espectro de energía que siga una ley de potencia  $E^{-2}$ , ya que se observó que una red neuronal entrenada en tal espectro genera un RMS relativamente constante entre 1 y 100 TeV [48]. Las variables de entrada de esta red se eligieron de forma que describan tres características de la cascada: la cantidad de energía depositada en el detector, la medida en que la huella de la cascada en el suelo está contenida dentro del detector y el grado de atenuación de la cascada por la atmósfera [48]. En el capítulo 4 se comenta con un poco más de detalle los elementos de una red perceptrón multicapa y su funcionamiento.

## 2.4. Software utilizado en HAWC

*Analysis and Event Reconstruction Integrated Environment* (AERIE) es el *software* utilizado en HAWC para realizar el análisis de datos obtenidos por el experimento. Para poder utilizar todas las funciones de este programa es necesario instalar una serie de paquetes adicionales (también llamados *externals*). Así que el proceso de instalación del *software* usado en HAWC se puede dividir en dos partes: la instalación de los paquetes *externals* y la compilación de AERIE. Estos paquetes *externals* contienen programas y librerías necesarias para compilar el código base en HAWC. Actualmente boost, fftw, gsl, root, xerces, cmake, cfitsio, cealpix-cxx y photospline son los programas que conforman a los *externals* [49]. En este trabajo se realizó la instalación de AERIE en mi cuenta asociada al cluster de HAWC en el ICN-UNAM. Cabe señalar que todas las imágenes para el análisis hecho en este trabajo, fueron generadas por el *display* del observatorio implementado en AERIE. Para conocer otras funciones que tiene esta herramienta así como algunos de los principales resultados obtenidos en HAWC se puede consultar el apéndice B.

# Capítulo 3

## Planteamiento para la detección indirecta de neutrinos con HAWC

En este capítulo se describe el método propuesto para la detección indirecta de neutrinos ultra energéticos con el observatorio HAWC. Se comenta el tipo de señales que se esperan detectar, así como las señales de ruido con las que se tienen que lidiar. También se aborda el tema del algoritmo de selección que actualmente se utiliza para discriminar este tipo de señales.

### 3.1. Técnica *Earth skimming*

Esta propuesta, que plantea utilizar al observatorio HAWC como un detector indirecto de neutrinos, utiliza como base la técnica de *Earth-skimming*. Esta técnica plantea utilizar un volumen con una gran cantidad de masa, ya que en estos cuerpos es más factible que ocurra una interacción entre un neutrino y un nucleón a través de una corriente cargada. En dicha interacción se genera un leptón del mismo sabor del neutrino incidente, debido a que el nucleón que participa en esa interacción cambia de carga eléctrica, la cual es portada por un bosón  $W^\pm$  que decae en un leptón cargado [50]. En las reacciones (3.1) - (3.3) se muestran este tipo de interacciones.

$$\nu_e(\bar{\nu}_e) + n(p) \rightarrow e^-(e^+) + p(n) \quad (3.1)$$

$$\nu_\mu(\bar{\nu}_\mu) + n(p) \rightarrow \mu^-(\mu^+) + p(n) \quad (3.2)$$

$$\nu_\tau(\bar{\nu}_\tau) + n(p) \rightarrow \tau^-(\tau^+) + p(n) \quad (3.3)$$

Cabe mencionar, que ya se ha utilizado en algunos experimentos la de técnica de *Earth-skimming* para buscar la detección de neutrinos ultra energéticos. Como es el caso del experimento Pierre Auger, en donde sus detectores buscan señales de cascadas electromagnéticas iniciadas por el decaimiento de un  $\tau$  con energías en el orden de EeV [2]. También se encuentra la colaboración Ashra-1, que busca neutrinos emitidos por un GRB en el orden de PeV a EeV. Este experimento este situado en frente del volcán Mauna Kea en la isla Hawaii, en donde espera detectar leptones  $\tau$  generados por neutrinos ultra energéticos del mismo sabor al interactuar con los átomos de dicho volcán [2].

### 3.2. Propuesta para la detección

La propuesta que se esta llevando a cabo en el instituto de Física de la UNAM, consiste en utilizar al Pico de Orizaba como objetivo de gran masa. Así, cuando un neutrino pase por esta

montaña e interactúe con los átomos que se encuentren ahí; se puede producir un leptón cargado, y este o el producto de su descomposición pueda ser detectado por medio de los tanques del observatorio HAWC (en la figura 3.1 se muestra un diagrama de esos proceso). En esta propuesta se plantea hacer la reconstrucción directa de la trayectoria mientras estas partículas atraviesan los detectores de este observatorio. A continuación, se menciona la viabilidad en la detección de los tres tipos de leptones generados con esta técnica:

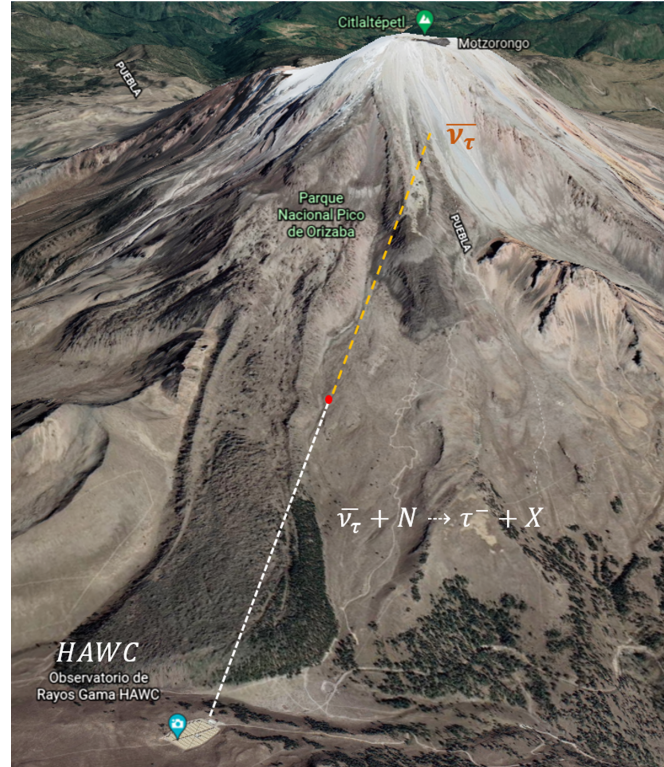


Figura 3.1: Representación del uso de la técnica *Earth-skimming* en HAWC. Imagen modificada de [51]

- **Neutrino del electrón:** Para la técnica *Earth-skimming* detectar estas partículas es desfavorable, debido a que en la interacción nucleón neutrino se producen electrones de alta energía. Estas partículas a su vez generan cascadas electromagnéticas que son absorbidas fácilmente por la montaña [2].
- **Neutrino del muón:** Para los  $\nu_\mu$  son aún muy complicadas de detectar, debido a que tienen una vida media relativamente grande que al combinarse con su alta energía solo se producirá una señal detectable para un muón ultra energético [2].
- **Neutrino del tau:** Para esta propuesta, los  $\nu_\tau$  son las más interesantes y más factibles de detectar. Porque los leptones  $\tau$  tienen una vida media relativamente corta (290.17 fs [52]), por ello aun si son muy energéticos, estos leptones cargados decaerán en partículas secundarias que podrán hacer la detección de la señal un poco más sencilla. Sin embargo, es importante mencionar que es muy complicado detectar neutrinos  $\tau$ , ya que hasta el año 2017 se tenían menos de 15 detecciones de este tipo de partículas [2].

Esta idea plantea medir neutrinos ultra energéticos, así que las señales que se esperan observar tendrán las siguientes características:

Modelo	$E_i$ [GeV]	$\phi_i$ [ $GeV^{-1} Km^{-2} sr^{-1} yr^{-1}$ ]
A	$10^7$	$6.2545 \times 10^{-5}$
B	$10^8$	$1.5780 \times 10^{-6}$
C	$10^9$	$1.5798 \times 10^{-8}$

Tabla 3.1: Valores de la constante de normalización  $\phi_i$  y la energía  $E_i$ . Tabla tomada de [2].

- El leptón que se produce viajará esencialmente en la misma dirección que el neutrino que lo generó. Por lo tanto, para que se registre una interacción entre este leptón y los detectores de HAWC, la trayectoria del neutrino que atravesó la montaña debe ser cuasi horizontal. Con esto se puede usar a los tanques de este observatorio como un detector de rastreo para este tipo de señales, a las cuales nos vamos a referir como trazas horizontales (que también incluyen a las señales de ruido).
- Debido a la gran energía de estas señales, se espera que generen una gran cantidad de luz Cherenkov. Esta cantidad supera el promedio de luz generada por muones atmosféricos (30 PEs) y de las señales de ruido (que están en un rango de 10-200 PEs) [2].
- La luz detectada debido a estas señales se espera que éste en el rango de miles a decenas de miles de PEs y dado el rango dinámico de los componentes electrónicos de HAWC, se podría anticipar que la luz recolectada podría usarse para al menos establecer un límite inferior en la energía del leptón cargado entrante.

### 3.3. Viabilidad de la detección

En la referencia [2] se realizó un estudio para ver las posibilidades del observatorio HAWC para poder medir este tipo de señales. Para ello se calculó el número de leptones cargados ( $N_L$ ) que se podían detectar en este observatorio mediante la siguiente expresión:

$$N_L = \Phi_L(A\Omega)_{eff}TD \quad (3.4)$$

En donde  $\Phi_L$  es el flujo de leptones cargados debido a neutrinos astrofísicos,  $(A\Omega)_{eff}$  es el área efectiva del observatorio,  $T$  es el tiempo de operación del experimento y  $D$  es el ciclo de trabajo para las observaciones. Este valor hace referencia a la fracción del tiempo  $T$  del experimento en el que se esta tomando datos. Para modelar el flujo de neutrinos diferencial en función de la energía se utiliza una modificación a la parametrización del flujo de neutrinos astrofísicos realizada por IceCube y que se muestra en la siguiente ecuación:

$$\varphi_i(E_\nu) = \frac{d\Phi_\nu(E_\nu)}{dE_\nu} = \phi_i \times \left[ \left( \frac{E_\nu}{E_i} \right)^{\alpha\eta} + \left( \frac{E_\nu}{E_i} \right)^{\beta\eta} \right]^{\frac{1}{\eta}} \quad (3.5)$$

Esta expresión toma en cuenta la suma de todos los sabores de neutrinos asumiendo que  $d\Phi_{\nu_e} = d\Phi_{\nu_\mu} = d\Phi_{\nu_\tau}$  y  $d\Phi_\nu = d\Phi_{\bar{\nu}}$ . Los valores de los parámetros son  $\alpha = -1$ ,  $\beta = -3$  y  $\eta = -1$ .  $E_i$  toma tres diferentes valores acorde a tres diferentes modelos (A,B y C) y que se pueden observar en la tabla 3.1. En la figura 3.2 se muestra el espectro de  $\varphi_i$  para cada modelo utilizando la misma normalización para el pico, aunque cada uno puede ser reescalado o sumado para descripciones dependientes del modelo. Los modelos A y B corresponden al espectro producido por BL Lac

AGNs, y combinaciones del modelo A y C seguirían la forma esperada de los neutrinos GZK producidos a partir de interacciones EBL y CMB.

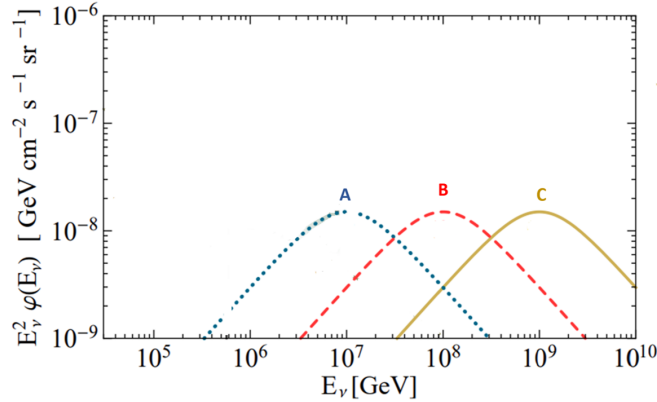


Figura 3.2: Espectros de  $\varphi_i$  para los modelos A,B y C . Imagen modificada de [53]

Por otro lado el flujo diferencial de leptones cargados que se esperan obtener mediante la técnica de *Earth-skimming* será:

$$\frac{d\Phi_L}{dE_L} = \frac{P_3(L)}{\beta_L \rho E_L} \int dE_\nu \frac{d\Phi_\nu(E_\nu)}{dE_\nu} \frac{1}{L_{CC}(E_\nu)} \exp\left[-\frac{A}{L_{CC}(E_\nu)}\right] \quad (3.6)$$

En donde  $P_3(L)$  es la probabilidad de que el leptón cargado pueda escapar del volcán,  $E_L$  es la energía del leptón,  $\rho$  es la densidad del material del volcán,  $\beta_L$  describe las pérdidas de energía radiativa de los leptones cargados a medida que viajan a través de la roca,  $L_{CC}$  es la longitud de interacción de corriente cargada y  $A$  es el ancho promedio del volcán. Así que para obtener este flujo de leptones cargados que logran atravesar la montaña, se integra de forma numérica la ecuación anterior considerando el flujo de neutrinos expuesto en 3.5 y se utilizan los valores descritos en [2] para los otros parámetros involucrados en la ecuación 3.6. Los resultados se presentan en la tabla 3.2.

Tau				Muón			
Energía neutrino ( $E_\nu$ )	$\Phi_{\tau+\bar{\tau}} [Km^{-2}sr^{-1}yr^{-1}]$			Energía neutrino ( $E_\nu$ )	$\Phi_{\mu+\bar{\mu}} [Km^{-2}sr^{-1}yr^{-1}]$		
Modelo	A	B	C	Modelo	A	B	C
10 PeV-100 PeV	0.4732	0.8269	0.1008	10 PeV-100 PeV	0.0476	0.0832	0.0101
100 PeV-10 EeV	0.0233	0.4251	0.3690	100 PeV-1 EeV	0.0015	0.0267	0.0185

Tabla 3.2: Cálculo del flujo de leptones cargados tauónicos y muónicos. Tabla tomada de [2].

Como se puede observar, el flujo de leptones cargados  $\tau$  es más grande que para el caso de los leptones  $\mu$  casi por un orden de magnitud, por tal motivo la detección de estos últimos es menos factible para este método. La razón de esta diferencia entre flujos se puede ver en la ecuación 3.6, donde el flujo de leptones cargados depende inversamente sobre las pérdidas de energía radiativa de estos leptones ( $\beta_L$ ); que son más de un orden de magnitud mayor para las  $\mu$ 's con respecto a las  $\tau$ 's [2]. Por otro lado, el área efectiva del observatorio HAWC se calculó por medio de simulaciones Monte Carlo. Para ello se tomó en cuenta la topografía que lo rodea (centrando la atención en el volcán Pico de Orizaba), por medio de un modelo geométrico sencillo. Los resultados se muestran



en la tabla 3.3 y se calcularon para diferentes condiciones de *trigger*, el cual corresponde a la longitud de la trayectoria que recorre un leptón cargado o el producto de su desintegración a lo largo del observatorio HAWC modelado como un prisma rectangular.

Trigger [m]	$A\Omega$ [ $Km^2$ sr]
$\geq 14$	0.013867
$\geq 21$	0.012544
$\geq 28$	0.010578
$\geq 35$	0.008418
$\geq 42$	0.006782

Tabla 3.3: Área efectiva para diferentes condiciones de *trigger*. Tabla tomada de [2].

Finalmente, en la tabla 3.3 para obtener el mayor flujo, dentro del rango permitido dado por sus incertidumbres estadísticas se muestra el número de leptones cargados  $\tau$  que se esperan detectar por año en el observatorio HAWC para dos bins de energía. Esos valores se calcularon a partir de las distintas áreas efectivas mostradas en la tabla 3.3 y tomando una suma de flujos entre el modelo A, C y el flujo de neutrinos astrofísicos modelado por IceCube (IC), el cual se define mediante la ecuación 3.7. En esta expresión se toman los parámetros de  $\phi = 0.3343 \text{ GeV}^{-1} Km^{-2} sr^{-1} yr^{-1}$  y  $\gamma = 1.71$ , esto para obtener el mayor flujo dentro del rango permitido dado por sus incertidumbres estadísticas [2].

$$\frac{d\Phi_\nu}{dE_\nu} = \phi \times \left( \frac{E_\nu}{100 \text{ TeV}} \right)^{-\gamma} \quad (3.7)$$

La última columna la tabla 3.3 muestra el tiempo necesario (en años) para la detección de un leptón cargado  $\tau$ .

Neutrino Energía ( $E_\nu$ )	Trigger [m]	N: $IC + \Phi_A + \Phi_B$	$\Delta T$ (yr)
10 PeV - 100 PeV	$\geq 14$	0.1492	6.7
	$\geq 21$	0.1349	7.4
	$\geq 28$	0.1138	8.8
	$\geq 35$	0.0906	11.0
	$\geq 42$	0.0730	13.7
100 PeV - 100 EeV	$\geq 14$	0.1505	6.6
	$\geq 28$	0.1321	7.6
	$\geq 35$	0.1148	8.7
	$\geq 42$	0.0914	10.9
	$\geq 21$	0.0736	13.6

Tabla 3.4: Número de leptones cargados  $\tau$  que se pueden detectar en HAWC por año. Tabla tomada de [2].

Como se puede observar en la tabla anterior, tomara alrededor de 8 años para encontrar una señal que atraviesan tres detectores de HAWC. Cabe mencionar que el estudio mostrado en [2] fue publicado en 2017 y a partir de esa fecha el funcionamiento del observatorio estaba planeando para funcionar por al menos 5 años. Sin embargo, con la implantación de nuevos detectores en 2018 (*outriggers*) y considerando su importancia como *trigger* para futuros instrumentos como el

CTA es probable que HAWC mantenga operaciones por al menos 10 años más. así que es viable medir alguna de estas señales de neutrinos ultra energéticos en el tiempo de vida del observatorio HAWC.

### 3.3.1. *Trigger* de HAWC

Como se comentó en el capítulo anterior, el *trigger* de HAWC consiste en que al menos 15 PMTs registren una señal por arriba de 2 umbrales (0.25 y 4 PEs) en una ventana de tiempo de 150 ns. Teniendo en cuenta que la suma de señales 112 PMTs en 60 segundos es de  $2.285 \times 10^8$ , se puede estimar que el número de PMTs activados en una ventana de 150 ns son aproximadamente 6. Por lo que solo haría falta que se activen 9 PMTs para cumplir la condición del *trigger* de HAWC. Entonces, si un leptón cargado o el producto de su descomposición pasa por al menos 3 WDC, activaría 12 fotomultiplicadores en una ventana de tiempo de aproximadamente 70 ns (tiempo en que pasaría por los 3 tanques). Así que se cumpliría con los 9 PMTs faltantes para cumplir con la condición de *trigger*. Por lo tanto, el *trigger* del observatorio HAWC se puede usar para la detección indirecta de neutrinos que se comentó anteriormente [2].

### 3.3.2. Señales de Ruido

Para esta propuesta se considera que la principal fuente de señales de ruido proviene de muones generados en cascadas atmosféricas en la dirección del Pico de Orizaba. En [2] se estudió éstas señales usando la caracterización del flujo de muones atmosféricos con energías superiores a los 15 TeV medido por IceCube:

$$\frac{d\Phi_\mu}{dE_\mu} = \phi_\mu \times \left( \frac{E_\mu}{10\text{TeV}} \right)^{-\delta} \quad (3.8)$$

Para este caso se toman los valores de los parámetros que maximizan el ruido, por ello se toma un flujo de muones de  $\phi_\mu = 4.6673 \times 10^7 \text{ TeV}^{-1} \text{ Km}^{-2} \text{ sr}^{-1} \text{ yr}^{-1}$  y un índice espectral ( $\delta$ ) de 3.73. Integrando la ecuación 3.8 y colocando este resultado para el flujo de leptones cargados en la fórmula 3.4, se puede hacer una estimación de estas señales de ruido. Los resultados se muestran en la tabla 3.5.

Neutrino Energía ( $E_\nu$ )	Trigger [m]	Ruido
10 PeV - 100 PeV	$\geq 14$	0.0145
	$\geq 21$	0.0131
	$\geq 28$	0.0111
	$\geq 35$	0.0088
	$\geq 42$	0.0071
100 PeV - 100 EeV	$\geq 14$	$< 10^{-5}$
	$\geq 28$	$< 10^{-5}$
	$\geq 35$	$< 10^{-5}$
	$\geq 42$	$< 10^{-5}$
	$\geq 21$	$< 10^{-5}$

Tabla 3.5: Número de señales de ruido que se pueden detectar en HAWC por año. Tabla tomada de [2].

Para separar estas señales de ruido se tomaron en cuenta las siguientes características de los detectores Cherenkov de HAWC.

- La cantidad de luz Cherenkov emitida por unidad de longitud en los tanques de HAWC es modelada por la ecuación de Frank-Tamm. Así que la cantidad de luz total será proporcional a la suma de las longitudes del camino de las partículas cargadas primarias y secundarias producidas (por radiación de frenado y producción de pares) que viaja dentro de un WCD.
- La suma de las longitudes de trayectoria, aumenta linealmente con la energía de la partícula primaria.

Así que la forma de discriminar estas señales de ruido sería observar la cantidad de luz total detectado en los WCD que pertenecen a una traza. Considerando que el momento de trazas cuasi-horizontales ( $\theta = 75$ ) es aproximadamente  $200 \text{ GeV}/c$ . Y si se esperan detectar leptones con energías del orden de PeV, entonces la cantidad de luz depositada en los detectores de HAWC será más grande (por un factor de  $10^3$ ) que la depositada por las señales de ruido.

### 3.4. Reconstrucción de la dirección

El sistema de referencia en HAWC ubica la posición de cada uno de sus detectores, cuando una traza horizontal activen una serie de tanques vecinos será posible reconstruir su dirección. La dirección de esta señal, se ubicará en el sistema de referencia del observatorio por medio de las coordenadas  $(\phi, \theta)$ . Por lo tanto, esta reconstrucción se basa en encontrar a estas dos coordenadas [54].

#### 3.4.1. Ángulo azimutal ( $\phi$ )

En la imagen 3.3 se puede observar una representación de dos tanques del observatorio en el plano  $(x, y)$ . En dicha figura se dibujó una recta que pase por el centro de estos tanques, después se trazan las líneas perpendicular a la recta anterior y que pasa por el centro de estos. Como resultado se van a generar 4 puntos ( $A, B, C$  y  $D$ ), que son la intersección entre estas líneas perpendiculares y la circunferencia de cada tanque. La línea roja se trazó al unir el punto  $A$  con el punto  $D$ , y la línea azul se trazó al unir el punto  $B$  con el punto  $C$ . Para una partícula con esa trayectoria, estas líneas representan dos direcciones en donde apenas se alcanza a dejar una señal. En otras palabras, la dirección de la traza que activó los tanques, estará delimitadas por estas dos direcciones. Los ejes del sistema de referencia en HAWC se han trasladado (sistema  $(x', y')$ , el sistema original se puede ver en la imagen 1), ahora el origen se encuentra en la intersección de las líneas de color azul y rojo. Los ejes de color verde son la posición de los puntos cardinales desde el punto de vista en HAWC, en donde hay un ángulo de inclinación entre el eje  $y'$  y la orientación del norte geográfico de  $\gamma = 15^\circ 33'$ . Tomando en cuenta lo anterior, se definen dos ángulos para la reconstrucción azimutal  $\phi_{min}$  y  $\phi_{max}$  [54], que se ven representados en la imagen 3.3.

#### 3.4.2. Ángulo de elevación( $\theta$ )

En la imagen 3.4 se puede observar una representación de cuatro tanques de HAWC pero ahora en el plano  $(x, z)$ . Para encontrar el ángulo de elevación de una partícula que genera una señal en estos tanques, nos fijamos en el centro del tanque inicial y final. Tomando en cuenta sus dimensiones ( $4.5 \text{ m}$  de altura y  $7.3 \text{ m}$  de diámetro) se pueden encontrar las coordenadas de los puntos A,B,C y D, estos puntos se encuentran en la esquina superior e inferior de cada tanque. Al igual que en el caso anterior, se formaron dos líneas, una al unir el punto A con el B y otra al unir el punto C con el D. Para el ángulo de elevación, estas serán las direcciones que delimitan el

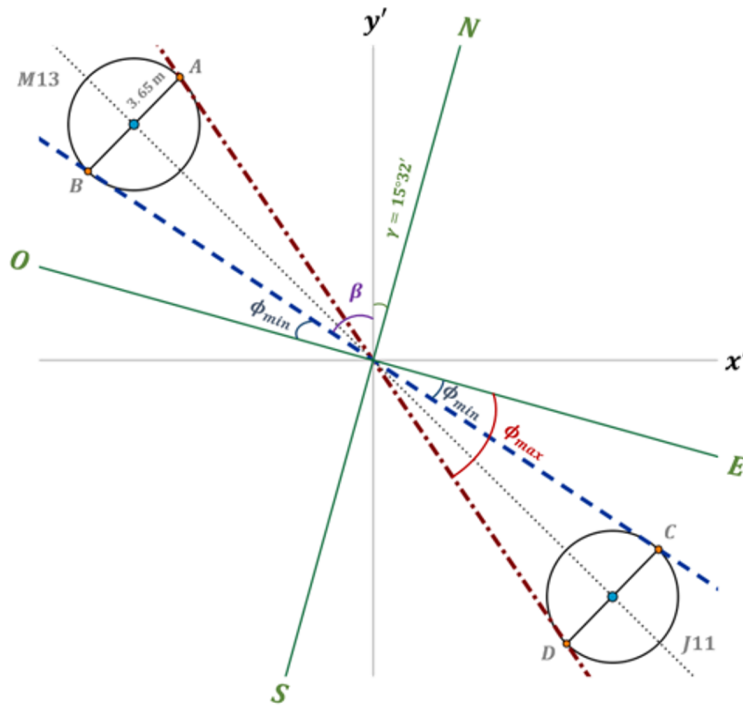


Figura 3.3: Representación para el ángulo azimutal, en donde se muestra a partir de donde se mide a los ángulos  $\phi_{min}$  y  $\phi_{max}$ .

paso de la partícula para que alcance a dejar una señal en los tanques. Para este caso también se definen dos ángulos para la reconstrucción  $\theta_{min}$  y  $\theta_{max}$  [54], que se ven representados en el inciso a) y en el inciso b) de la imagen 3.4 respectivamente.

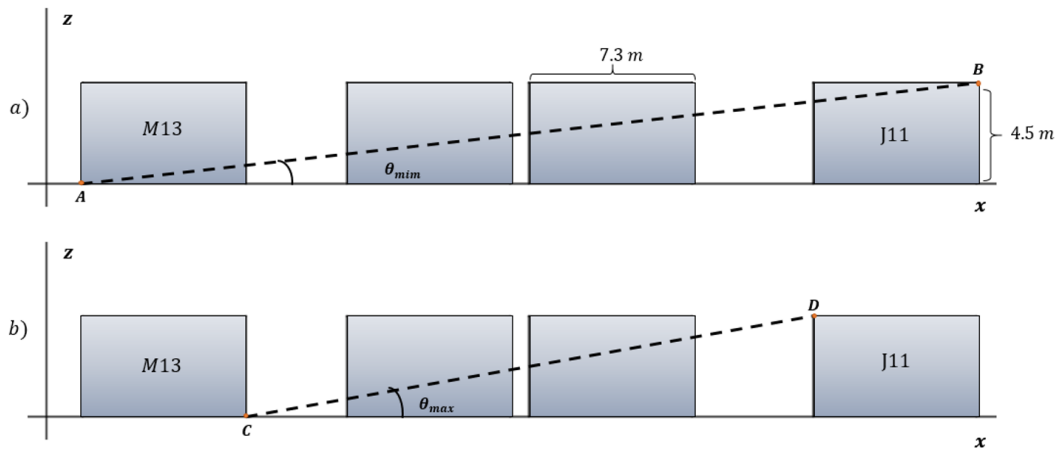


Figura 3.4: Representación para el ángulo de elevación, en donde se muestra a partir de donde se mide a los ángulos  $\theta_{min}$  y  $\theta_{max}$ .

Con esta reconstrucción se obtienen dos intervalos, uno para el ángulo azimutal y otro para el ángulo de elevación.

$$\begin{aligned} \phi &\in (\phi_{min}, \phi_{max}) \\ \theta &\in (\theta_{min}, \theta_{max}) \end{aligned}$$

En estos intervalos se encuentra la dirección de la partícula que generó la señal, y con ello se puede aproximar su posición.

### 3.5. Algoritmo para detección de trazas

El algoritmo que se utiliza para identificar trazas horizontales en los datos de HAWC se puede dividir en tres secciones: primero se establece un *trigger* o condición de disparo para trazas horizontales, después está el algoritmo en donde se establecen los criterios de selección de candidatos a trazas y por último se aplica una serie de filtros a estos candidatos. A continuación se hace una descripción de cada una de estas secciones basándonos en la referencia [55].

#### 3.5.1. *Trigger* para trazas horizontales

La manera en la que se procesa los datos obtenidos por HAWC mediante este *trigger* es la siguiente:

1. Se hace la reconstrucción estándar de los datos obtenidos por HAWC, en donde se reconstruye el plano de la cascada y otras variables de interés. Pero se descartan algunas variables (por ejemplo la energía de la cascada) esto para que la reconstrucción sea más rápida. En este paso los PMTs asociados al plano de la cascada se descartan.
2. Los PMTs que se activaron durante un evento se ordenan de acuerdo al tiempo que registraron para después ser procesados por este *trigger*. En este paso se observa el primer PMT que registró una señal. Después se busca otro PMT en un tanque vecino que se haya activado y que esté dentro de la ventana de tiempo establecida. Esta ventana será el tiempo en que tarda la luz en recorrer la distancia máxima de separación de dos PMTs vecinos que es (44.8 m).

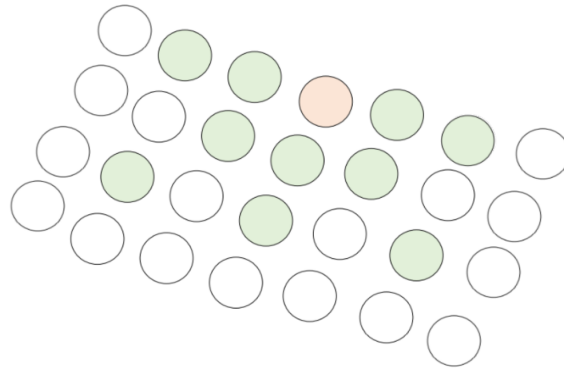


Figura 3.5: Representación de tanques vecinos para el primer tanque activado.

En la imagen 3.5 se muestra una representación de los tanques que se consideran vecinos (que están de color verde) del primer tanque que fue activado (que esta en color rojo). Cabe señalar que solo se guarda la información para tanques que registraron una señal de carga mayor a 4 PEs.

3. En este paso tenemos 2 condiciones en donde no se guarda la información. La primera es que la señal no cumple con la ventana de tiempo antes mencionada (150 ns). Y la segunda es que

la señal está separada por 2 tanques (esto se aplica también para propagación en diagonal). Cabe mencionar, que se permite más de un evento activado por traza en el mismo evento de cascada.

4. En este paso se establece que al menos se activen 9 PMTs para que se guarde el evento. La información que se guarda es la cantidad de carga total, el tiempo promedio y el número de PMTs que registraron señal en los WCDs que fueron activados. También solamente se guarda la información de tiempo y carga registrada por el PMT central.

Con este *trigger* se obtiene una gran cantidad de candidatos, en donde la mayoría se deben principalmente a pequeñas cascadas inclinadas que no pudieron ser identificadas por los algoritmos de reconstrucción. Los candidatos encontrados por esta condición de disparo (con aproximadamente una tasa de 16,7 kHz) se pasan a un segundo código que realiza la reconstrucción de la traza, utilizando como entrada los WCD seleccionados en esta parte.

### 3.5.2. Algoritmo de selección de trazas

Este algoritmo busca señales que se propaguen en línea recta a través de los detectores de HAWC. La velocidad con la que se activan dos tanques vecinos tiene que ser consistente con la velocidad de la luz, en donde el tiempo de activación entre estos tanques debe de estar dentro de una ventana de tiempo de 40 ns. Dicha ventana temporal fue optimizada utilizando simulaciones Monte Carlo de muones enviados en los tanques de este observatorio en distintas direcciones. También se agrega la condición de que al menos se activen dos fotomultiplicadores en cada tanque y se descartan los eventos en donde se activen más de 100 PMTs. En esta parte se hace la reconstrucción de los ángulos ( $\phi, \theta$ ) de la dirección de llegada a de la señal. La tasa de candidatos de trazas en los datos es de aproximadamente 420 Hz.

### 3.5.3. Filtro para los candidatos a trazas

La información obtenida en el algoritmo de selección anterior se hace pasar por una serie de filtros para que se eliminen las señales provocadas por cascadas muy inclinadas. Los filtros fueron seleccionaron en función del estudio de una gran cantidad de candidatos de trazas observados en un *display* (como el que se muestra en 3.6), a continuación se enlistan estos filtros:

- **Low charge:** Se define como la razón entre el número de tanques activados dividida por la longitud de la traza, y se establece que esta razón sea menor a 6.
- **High charge:** Se define como el número de píxeles<sup>1</sup> en el evento dividida por la longitud de la traza y se toma que esta cantidad sea menor a 1.5.
- **Track quality:** Se refiere a que al menos el 75 % de los píxeles que conforman la traza se pueda ajustar a una función lineal.
- **Minimum length:** Establece que solo se toman eventos con una longitud de traza mayor a 4 píxeles activados.

En la figura 3.6 se muestra un traza obtenida en los datos del observatorio HAWC que fue identificada con este algoritmo.

---

<sup>1</sup>Cada WCD activado se considera como un píxel.

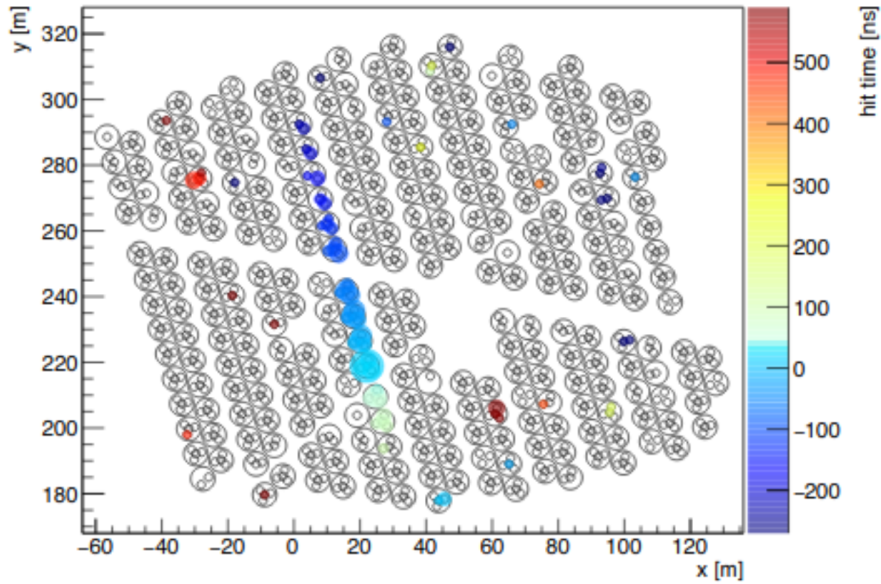


Figura 3.6: Ejemplo de evento identificado como traza con el algoritmo oficial de HAWC. Imagen tomada de [55].

<b>Tipo de corte</b>	<b>Muones</b>	<b>Porcentaje</b>
Sin cortes	2464812	100 %
<i>Low cahрге</i>	2449745	99.4 %
<i>High cahрге</i>	1457870	59.1 %
<i>Track quality</i>	2335724	94.8 %
<i>Minimum length</i>	1547687	62.8 %
<b>Todos los cortes</b>	1036137	42.0 %

Tabla 3.6: Porcentaje de trazas simuladas que pasaron los 4 cortes mencionados. Tabla tomada de [3].

Por último en la tabla 3.6 se muestra el efecto de aplicar los 4 cortes antes mencionados a una muestra de eventos simulados de un poco más de dos millones de muones horizontales que pasaron por los tanques de HAWC. En este caso solamente se obtiene un 42 por ciento de la muestra original. Por tal motivo, en este trabajo se hace un estudio para ver si el uso de una red neuronal artificial puede optimizar el número de trazas que se identifican.

# Capítulo 4

## Red Neuronal Artificial

Una red neuronal artificial puede verse como una función de  $R^n$  a  $R^m$ , en donde  $R^n$  son los datos de entradas y  $R^m$  son los datos de salida [56]. El usuario proporciona una lista de datos de entradas y salidas que tiene como objetivo entrenar a la red neuronal, como resultado se podrá dar cualquier dato de entrada y la red calculará el dato de salida en base al entrenamiento previo. Comúnmente los datos de entrada y salida son números reales acomodados en vectores. Por ejemplo, para una red que identifica imágenes de perros y gatos en blanco y negro de 20 por 20 píxeles, el dato de entrada sería un vector con 400 valores en donde cada valor indica la intensidad en escala de grises de cada píxel (un valor real entre 0 y 1). Por otro lado el dato de salida sería solo un número real, 0 para el caso de una identificación de un gato y 1 para el caso de un perro. Así que una red neuronal artificial es capaz de determinar relaciones no lineales entre un conjunto de datos, asociando patrones de entrada o salidas correspondientes. Por esto se han convertido en una herramienta muy popular y útil para resolver muchos problemas como clasificación, agrupamiento, regresión, reconocimiento de patrones, reducción de dimensión, predicción estructurada, traducción automática, detección de anomalías, toma de decisiones, visualización, visión por computadora entre otras [57]. En este trabajo, se implementó una red neuronal artificial para discriminar trazas horizontales en los datos de HAWC. Por tal motivo, en este capítulo se da la teoría correspondiente a este tema. Primero se explica el modelo de una red neuronal relativamente sencilla, el perceptrón multicapa. Después se describe el funcionamiento de una red convolucional, la cual es una red más compleja y que se especializa en clasificación de imágenes. Por último y tomando como base los dos modelos antes mencionados, se explica el enfoque de una red convolucional para detección de objetos. Cabe mencionar que en este escrito se trabajó con una red convolucional y una red enfocada en detección de objetos para la identificación de trazas.

### 4.1. Red perceptrón multicapa

El perceptrón multicapa es una red que generaliza al modelo de perceptrón simple. [58]. En el primer modelo, los datos de entrada son números y se representan por medio de letras  $(x_1, x_2, \dots, x_n)$ . La forma en la que se genera una señal de salida es por medio de pesos  $(w_1, w_2, \dots, w_n)$  que son números reales que indican la importancia de cada entrada para generar el dato de salida. Esta salida solamente podrá elegir entre dos valores, el 0 y el 1. Así que se calculará la suma del producto de los datos de entrada con sus pesos  $(\sum_{i=1}^n x_i w_i)$ , y el dato de salida dependerá de que el valor de esta suma sea mayor o menor a cierto valor umbral [58]. El perceptrón multicapa desarrollado en 1969 nace de las limitaciones de su predecesor al no poder simular todas las compuertas lógicas. La forma en la que el perceptrón multicapa resuelve este problema queda fuera de los temas de este



trabajo, pero puede consultarse en las referencias [59] y [60]. Por ahora solamente nos enfocaremos en describir la estructura de los componentes que conforman la red y se dará a grandes rasgos el algoritmo que emplea para su funcionamiento.

### 4.1.1. Estructura

La estructura de esta red se muestra en la figura 4.1 y los componentes que describen a esta red son los siguientes:

- Neurona:** La neurona artificial es la unidad elemental de la red, y en esencia es una función que recibe un dato de entrada y proporciona un dato de salida entre 0 y 1 ( $f : R \rightarrow [0, 1]$ ). Normalmente a este dato de salida se le conoce como valor de activación, por ejemplo, si el valor de una neurona es cercano a 1, se dice que la neurona es muy activa. En la imagen, a cada neurona se le representa por medio de un círculo, y la activación de cada neurona es representada mediante el símbolo  $a_k^l$ , en donde el índice  $l$  indica la capa en donde se encuentra ese valor de activación, y el símbolo  $k$  hace referencia a la neurona de la capa  $l$  de la cual se esta hablando. El valor de  $l$  estará entre 0 y  $L$ , en donde  $L$  indica el número de capas. Por otro lado, el valor de  $k$  estará entre 0 y  $N_l$ , en donde  $N_l$  indica el número de neuronas en la capa  $l$ .

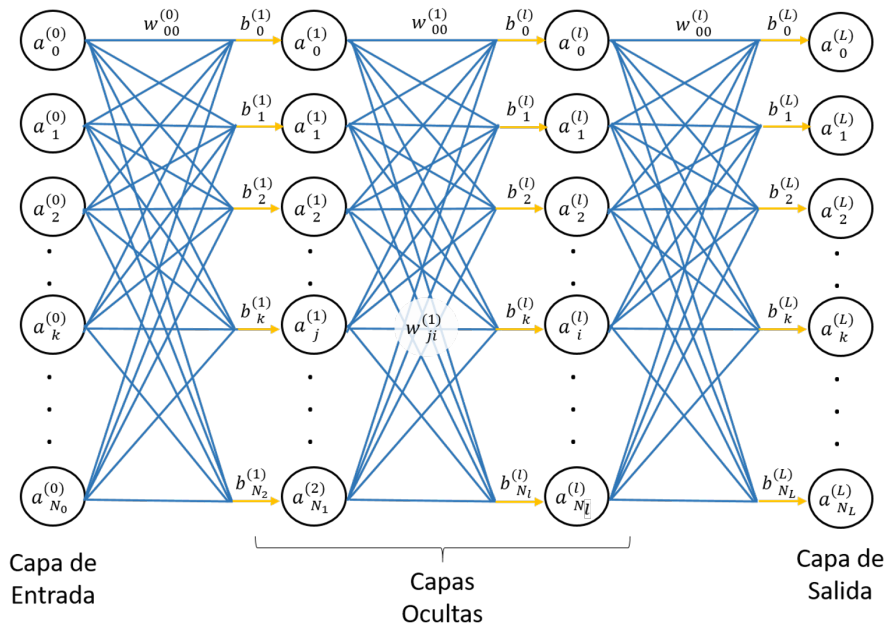


Figura 4.1: Estructura de una red perceptrón multicapa.

- Pesos:** Los pesos son representados en la imagen por las líneas de color azul. Estos valores se van modificando a medida que el usuario presente datos de entrenamientos, y son responsables de que la red funcione de manera correcta. Podemos visualizar a estos valores como cantidades que nos indican que tanta influencia tienen los datos de entrada de una capa previa en el dato de salida de una neurona en la siguiente capa. A estos pesos los denotaremos mediante el símbolo  $w_{ji}^l$ , el índice  $i$  estará entre 0 y  $N_l$ , y el valor de  $j$  estará entre 0 y  $N_{l+1}$ . Mas adelante se calculará de forma explícita la activación de una neurona artificial, en donde quedará un poco más claro el uso de estos índices.

- **Bias:** Los *bias* son representados en la imagen por medio de flechas amarillas y se pueden visualizar como valores que determinan que tan activa o inactiva es una neurona. Estas se representa mediante el símbolo  $b_k^l$ .
- **Capa de Entrada:** Es el conjunto de neuronas que contiene a los datos de entrada que el usuario proporcione, ya sea para entrenar o para probar la red neuronal. En la imagen las letras  $a_0^0, a_1^0, a_2^0, \dots, a_{N_0}^0$  representan a estos datos.
- **Capas Ocultas:** Recordemos que el punto clave para que una red funcione de manera correcta está en modificar los valores de pesos y *bias* a través del entrenamiento de la red. Pero estas modificaciones se verán reflejadas en el valor de activación de cada neurona. El perceptrón multicapa considera capas de neuronas entre los datos de entrada y los datos de salida, a las que usualmente se le llaman capas ocultas. Estas capas, son implementadas en este modelo para resolver problemas más sofisticados.

Tanto el número de capas ocultas como el número de neuronas que se utiliza en cada capa, dependerá del problema que el usuario este intentando resolver, realmente no hay una regla que nos indique el número de capas ocultas que debemos usar, comúnmente el usuario decide el número de capas a utilizar en base a su experiencia y a los resultados que va obteniendo [61].

- **Capa de Salida:** Es el conjunto de neuronas que contiene los datos de salida de la red. En la imagen las letras  $a_0^L, a_1^L, a_2^L, \dots, a_{N_L}^L$  representan a estos datos.

### 4.1.2. Funcionamiento

La explicación del funcionamiento de este tipo de red se basa en las referencias [62]-[64] y se puede explicar mediante los siguientes pasos:

- **Paso 1:** Se ponen los datos de entrada a la red, estos corresponden a los valores de activación de la primera capa. Después se asignan valores de números reales aleatorios entre -1 y 1 a los datos de pesos y *bias*. Este rango se elige debido a que los datos de entrada están entre cero y uno, así que se requiere valores positivos y negativos entre este rango.
- **Paso 2:** Se calcula el valor de activación de las neuronas de la primera capa oculta, para ello se utiliza la siguiente ecuación que sigue la notación mostrada en la sección anterior.

$$a_k^{(1)} = w_{k,j}^{(0)} \cdot a_j^{(0)} + b_j^{(1)} \quad (4.1)$$

Con  $1 \leq k \leq N_1$  y  $1 \leq j \leq N_0$ , que en su forma matricial es:

$$\begin{bmatrix} a_0^{(1)} \\ a_1^{(1)} \\ \vdots \\ a_{N_1}^{(1)} \end{bmatrix} = \sigma \left( \begin{bmatrix} w_{0,0}^{(0)} & w_{0,1}^{(0)} & \dots & w_{0,N_0}^{(0)} \\ w_{1,0}^{(0)} & w_{1,1}^{(0)} & \dots & w_{1,N_0}^{(0)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N_1,0}^{(0)} & w_{N_1,1}^{(0)} & \dots & w_{N_1,N_0}^{(0)} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_{N_0}^{(0)} \end{bmatrix} + \begin{bmatrix} b_0^{(1)} \\ b_1^{(1)} \\ \vdots \\ b_{N_1}^{(1)} \end{bmatrix} \right) \quad (4.2)$$

En donde:

$$\sigma \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \sigma(x) \\ \sigma(y) \\ \sigma(z) \end{pmatrix}$$

$\sigma$  representa a la función de activación, esta función sirve para que las cantidades en cada neurona tengan valores entre 0 y 1 ( $f : R \rightarrow [0, 1]$ ). Actualmente se utiliza una gran variedad de funciones para desempeñar esta tarea, entre las más utilizadas se encuentra la función sigmoide, la función escalón y la función ReLu. Las principales características que debe de tener esta función son las siguientes:

- Las redes neuronales se entrenan mediante el proceso de descenso de gradientes (el cual se explica en el paso 4 y 5), por lo que la función de activación debe ser diferenciable o al menos diferenciable en partes.
- La salida de la función de activación debe ser simétrica en cero para que los gradientes no se desplacen hacia una dirección particular.
- Las funciones de activación se aplican después de cada capa y deben calcularse millones de veces en redes profundas. Por lo tanto, deberían ser computacionalmente económicos de calcular.

Cabe mencionar que si representamos a la matriz que contiene a todo los pesos para calcular la activación de la primera capa con la letra  $M^1$ , y a los dos vectores columna como  $a^0$  y  $b^0$  para los valores de activación y para las bias respectivamente. La ecuación para obtener el valor de activación de la primera capa se puede escribir como:

$$a^1 = \sigma (M^1 a^0 + b^1)$$

El valor de activación para las otras capas se puede representar mediante las siguiente dos ecuaciones:

$$\begin{aligned} a^2 &= \sigma (M^2 a^1 + b^2) \\ a^3 &= \sigma (M^3 a^2 + b^3) \end{aligned}$$

Finalmente se puede seguir usando la ecuación anterior para ir calculando los valores de activación de las siguientes capas hasta llegar a la última.

- **Paso 3:** Una vez que tenemos los valores de activación de la última capa, definimos la función de costo para un solo evento mediante la siguiente expresión:

$$C_0 = \sum_{k=0}^{N_L} (a_k^{(L)}(w_1, w_2, \dots, w_F, b_1, b_2, \dots, b_G) - y_k)^2 \quad (4.3)$$

En este caso  $F$  y  $G$  representan el número total de pesos y bias correspondientemente. Por otro lado  $y_i$  son las etiquetas de nuestros eventos, es decir, los valores de salida que se esperan obtener y que el usuario proporciona a la red. El valor de la función de costo para todos los eventos ( $n$ ) será un promedio:

$$C = \frac{1}{n} \sum_{j=0}^{n-1} C_j \quad (4.4)$$

En donde  $C_i$  representa el costo calculado a partir del dato de entrada y de salida  $i$ -ésimo. Aquí podemos notar que si el valor de la función de costo se acerca a cero, significa que la red esta haciendo la clasificación de forma correcta. Así que el proceso de entrenamiento consiste en minimizar la función de costo. Y esta función depende de los valores de pesos y *bias*.

- Paso 4:** Para una red sencilla con 6 datos de entrada, 2 capas ocultas con 8 neuronas cada una y un solo valor de salida se tendrá un total de 120 valores de pesos ( $6 \times 8 + 8 \times 8 + 8 \times 1$ ) y 17 valores de *bias* ( $8 + 8 + 1$ ) dando un total de 137 variables. Así que para este ejemplo, la función de costo dependerá de este número de variables. Por lo tanto, el primer paso para calcular el mínimo de la función de costo consiste en encontrar el valor de su gradiente. Para ello se usa el método de propagación hacia atrás. Para explicar este modelo usaremos como ejemplo la red neuronal más sencilla de dos capas ocultas y que se puede observar en la imagen 4.2.

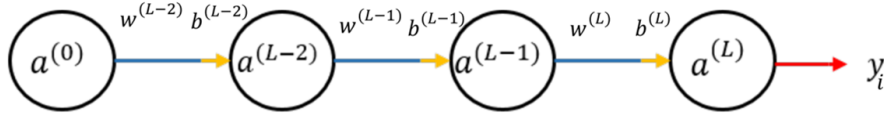


Figura 4.2: Ejemplo de una red neuronal que solamente tiene una neurona en cada capa.

Entonces nos fijamos en el caso en donde entra a la red un evento y queremos calcular cuanto varia la función de costo con respecto al último peso y que representa un elemento del vector gradiente. Si utilizamos la regla de la cadena podemos expresar esta variación como:

$$\frac{\partial C_0}{\partial w^{(L)}} = \frac{\partial C_0}{\partial a^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial z^{(L)}}{\partial w^{(L)}} \quad (4.5)$$

donde la variable  $z^{(L)}$  se introduce para simplificar la notación y se define como:

$$z^{(L)} = a^{L-1}w^L + b^L \quad (4.6)$$

Derivando las ecuaciones 4.2, 4.3 y 4.6 para este caso tendremos que:

$$\frac{\partial C_0}{\partial w^{(L)}} = a^{(L-1)}\sigma'(z^{(L)})2(a^{(L)} - y_0) \quad (4.7)$$

De forma similar podemos calcular que tan sensible es la función de costo para un pequeño cambio en el valor de  $b^{(L)}$ :

$$\frac{\partial C_0}{\partial b^{(L)}} = \frac{\partial C_0}{\partial a^{(L)}} \frac{\partial a^{(L)}}{\partial z^{(L)}} \frac{\partial z^{(L)}}{\partial b^{(L)}} = \sigma'(z^{(L)})2(a^{(L)} - y_0) \quad (4.8)$$

Con esto obtenemos la variación de la función de costo con respecto al peso  $w^{(L)}$  y al *bias*  $b^{(L)}$  usando el primer evento del entrenamiento. Para los otros evento, la ecuaciones anteriores se reescriben como:

$$\frac{\partial C_i}{\partial w^{(L)}} = a^{(L-1)}\sigma'(z^{(L)})2(a^{(L)} - y_i) \quad (4.9)$$

$$\frac{\partial C_i}{\partial b^{(L)}} = \sigma'(z^{(L)})2(a^{(L)} - y_i) \quad (4.10)$$

Así que para calcular la derivada parcial con respecto al peso y al *bias*  $L$  de la función de costo, habrá que considerar todos los eventos y hacer un promedio.

$$\frac{\partial C}{\partial w^{(L)}} = \frac{1}{n} \sum_{i=0}^{n-1} \frac{\partial C_i}{\partial w^{(L)}} \quad (4.11)$$

$$\frac{\partial C}{\partial b^{(L)}} = \frac{1}{n} \sum_{i=0}^{n-1} \frac{\partial C_i}{\partial b^{(L)}} \quad (4.12)$$

Con esto obtenemos dos coordenada del gradiente que buscamos:

$$\nabla C = \begin{pmatrix} \frac{\partial C}{\partial w^{(0)}} \\ \vdots \\ \frac{\partial C}{\partial w^{(L)}} \\ \frac{\partial C}{\partial b^{(0)}} \\ \vdots \\ \frac{\partial C}{\partial b^{(L)}} \end{pmatrix} \quad (4.13)$$

Se pueden usar los valores calculados en 4.11 y 4.12 junto con la misma idea de usar la regla de la cadena para ir calculando de forma recursiva los otros valores  $(\frac{\partial C}{\partial w^{(L-1)}}, \dots, \frac{\partial C}{\partial w^{(0)}}, \dots, \frac{\partial C}{\partial b^{(0)}})$  del gradiente de la función de costo. Cabe mencionar que para el caso de una red con una estructura más complicada, las ecuaciones 4.11 y 4.12 no se complican demasiado simplemente habrá que agregar otro índice.

$$\frac{\partial C_i}{\partial w_{jk}^{(L)}} = \frac{\partial C_i}{\partial a_j^{(L)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial z_j^{(L)}}{\partial w_{jk}^{(L)}} \quad (4.14)$$

$$\frac{\partial C_i}{\partial b_k^{(L)}} = \frac{\partial C_i}{\partial a_j^{(L)}} \frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \frac{\partial z_j^{(L)}}{\partial b_k^{(L)}} \quad (4.15)$$

Así que en este paso se calculó el gradiente de la función de costo en función de los pesos y *bias*.

- **Paso 5:** Ahora se procede a minimizar la función de costo, para ello se suma el vector  $-\nabla C$  a los valores de pesos y *bias* que colocaremos en un mismo vector  $v$ , para que la función se vaya acercando a cero. Para ilustrar esto usaremos la red de 137 variables (120 pesos y 17 *bias*) mencionada en el paso 4.

$$\tilde{v} = v - \nabla C = \begin{pmatrix} v_0 \\ \vdots \\ v_{136} \end{pmatrix} - \begin{pmatrix} \frac{\partial C}{\partial v_0} \\ \vdots \\ \frac{\partial C}{\partial v_{136}} \end{pmatrix} = \begin{pmatrix} w_0 \\ \vdots \\ w_{119} \\ b_0 \\ \vdots \\ b_{16} \end{pmatrix} - \begin{pmatrix} \frac{\partial C}{\partial w_{00}^{(0)}} \\ \vdots \\ \frac{\partial C}{\partial w_{70}^{(L)}} \\ \frac{\partial C}{\partial b_0^{(0)}} \\ \vdots \\ \frac{\partial C}{\partial b_0^{(L)}} \end{pmatrix}. \quad (4.16)$$

Se repetirá el proceso anterior hasta llegar a un mínimo local de la función de costo. La figura 4.3 muestra una representación de este proceso, en donde se gráfica una función de costo de 2 dimensiones y cada que se repite el proceso descrito en la ecuación 4.16 se ilustra con un paso (también se le suele llamar época) que se va acercando a un mínimo local.

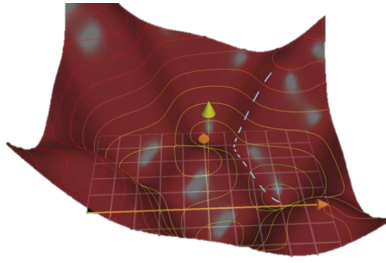


Figura 4.3: Representación gráfica para encontrar un mínimo local de la función de costo. Imagen tomada de [63].

Así que con este método se está minimizando la función de costo y con cada época su valor se acerca a cero. Esto significa que cada vez la salida que calcula la red neuronal es más parecida al resultado que deseamos obtener.

Con estos pasos se explica el funcionamiento de una red perceptrón multicapa, ahora se procede a explicar una red un poco más compleja y que será el tema de la siguiente sección.

## 4.2. Red neuronal convolucional

Como se comentó anteriormente, este tipo de red se especializa en clasificación de imágenes. Por ejemplo, se puede usar para distinguir entre imágenes de perros y gatos; incluso son muy efectivas en la parte de detección de objetos (identificar algún elemento dentro de una imagen con ciertas características). Como una red convolucional se especializa en el análisis de imágenes, la entrada de este tipo de red será una imagen. En la figura 4.4 se muestra una imagen de 6 por 6 píxeles, como está en escala de grises cada píxel tiene el valor entre 0 y 1 de la intensidad del color negro. Estos valores de escala de grises se colocan en una matriz de 6 por 6. Una vez que especificamos los datos de entrada en la red convolucional, intervienen 2 procesos que son el pilar fundamental para el aprendizaje de la red. Estos procesos son la convolución y el *pooling* o agrupación, que son los temas que se verán a continuación.

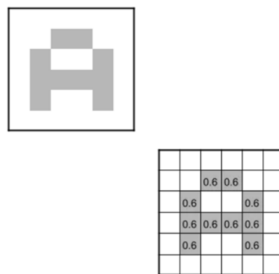


Figura 4.4: Entrada de una red convolucional. Imagen tomada de [65].

### 4.2.1. Convolución

Como primer punto, se debe comentar que el elemento fundamental para llevar a cabo este proceso es por medio de un filtro. Y un filtro solamente es una matriz cuadrada, cabe mencionar que en algunas referencias a este filtro también se le conoce como *kernel*. Así que la convolución

es el proceso en donde el filtro va recorriendo de izquierda-derecha y de arriba-abajo todos los píxeles de la imagen de entrada, aplica un producto escalar y genera una nueva matriz de salida [65]. Por ejemplo, en la imagen 4.5 se tiene un filtro de 3 por 3 (que es el cuadro de color azul) que está analizando una zona de la imagen inicial (que es la que se encuentra de color rojo) aplica un producto escalar, que es multiplicar el valor de cada entrada y después sumarlo para generar un píxel de la imagen de salida (que es el que se señala de color verde). Conforme se va moviendo el filtro va generando otros píxeles de la imagen de salida.

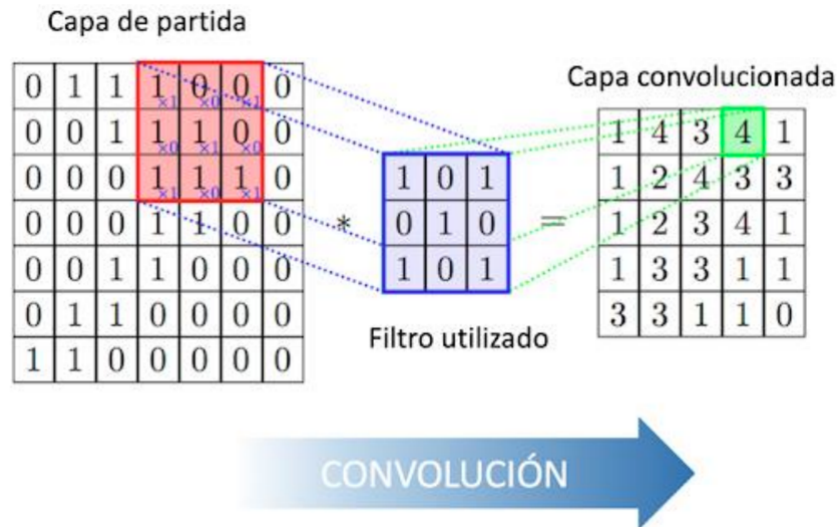


Figura 4.5: Representación del funcionamiento de la convolución. Imagen tomada de [66].

Otra variable involucrada es a lo que se le conoce como *stride* o paso, que indica el número de píxeles de separación que usará el filtro para realizar la convolución [67]. En el cuadro de la izquierda de la figura 4.6 se usa un paso igual a 1, en donde el filtro se va recorriendo un píxel; y en el cuadro de la derecha se usa un paso igual a 2. Para este caso el filtro va dejando 2 píxeles de separación en cada convolución.

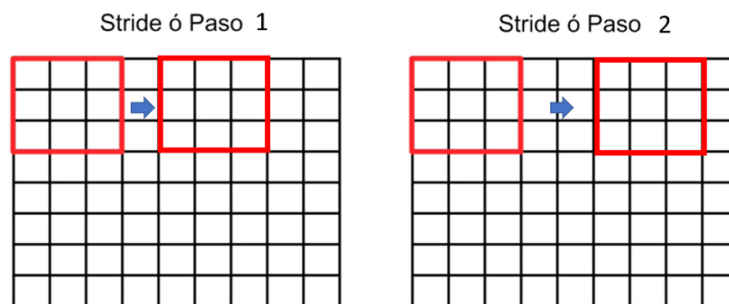


Figura 4.6: Representación del paso usado por un filtro para hacer la convolución. Imagen modificada de [67].

Después de pasar el filtro por la imagen, se aplica una función de activación para que los valores de los píxeles de la imagen estén entre 0 y 1. En este caso se aplicó la función RELU (*Rectified Lineal Unit*) la cual se define en la ecuación 4.17 y el procedimiento completo se representa en la

imagen 4.7.

$$f(x) = \max(0, x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \quad (4.17)$$

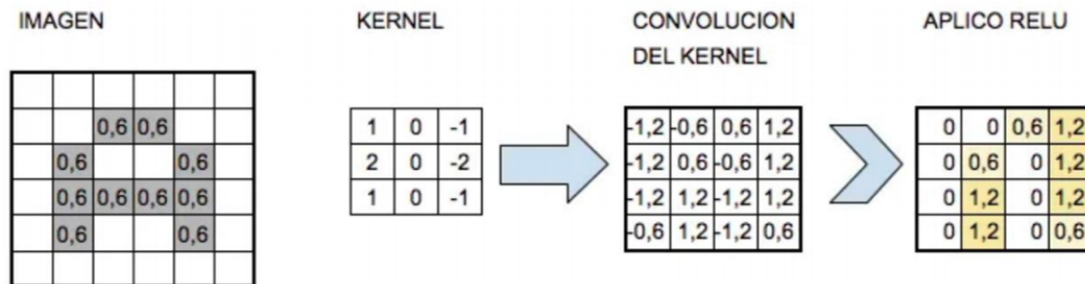


Figura 4.7: Cambio de la imagen de una letra al aplicar un filtro y una función de activación (RELU). Imagen tomada de [65].

Es importante mencionar, que este tipo de red se especializa en el análisis de imágenes debido al uso de filtros. En la figura 4.8 se muestra el efecto de utilizar dos diferentes filtros sobre la imagen de entrada, que corresponde al número 7. Cada filtro corresponde a una matriz de 3 por 3 con distintos valores, y a cada valor se le asocia un color diferente; por ejemplo al -1 se le da el color negro, al 1 el blanco y al cero el gris. Después pasamos cada filtro por la imagen original y aplicamos el proceso de convolución. Como resultado, para el primer filtro obtenemos una imagen que resalta los bordes horizontales y para el segundo filtro obtenemos una imagen que resalta los bordes verticales.



Figura 4.8: Cambio de la imagen de un número 7 al aplicar dos diferentes filtros. Tomada de [69].

### 4.2.2. Pooling o Agrupación

Este procedimiento consiste en reducir el tamaño de la imagen obtenida después de aplicar el filtro [65]. Los dos tipos de *pooling* más conocidos son: el *max pooling* y el *average pooling*. El



primero consiste en elegir un cuadrante de la imagen y tomar el valor más alto. En la imagen de la izquierda de la figura 4.9 se muestra un ejemplo, en donde se analiza una zona de la imagen conformada por 4 píxeles y se toma el valor más alto para generar una imagen más pequeña. El *average pooling* consiste en elegir un cuadrante de la imagen y realizar un promedio entre esos valores. En la imagen de la derecha de la figura 4.9, se analiza una zona de la imagen conformada por 4 píxeles y se toma el promedio de esos valores para generar una imagen más pequeña.

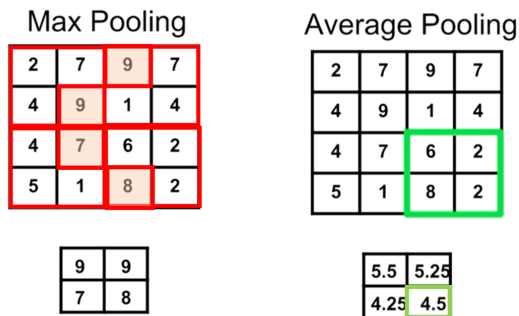


Figura 4.9: Ejemplo de *max pooling* y *average pooling*. Imagen modificada de [67].

### 4.2.3. Funcionamiento

El funcionamiento de este tipo de red se puede explicar mediante los siguientes pasos:

- Paso 1:** En la figura 4.10 se muestra un ejemplo de la primera capa de una red convolucional. En dicha figura se muestra una imagen de entrada de 28 por 28 píxeles, a esta imagen se le aplican 32 filtros de 3 por 3. Después de la convolución se habrán generado 32 imágenes de salida de 28 por 28 píxeles Y a cada imagen de salida se le aplica una función para tener valores entre cero y uno. Por último, se aplica una capa de *max pooling* a cada imagen para obtener 32 imágenes pero ahora de 14 por 14 píxeles.

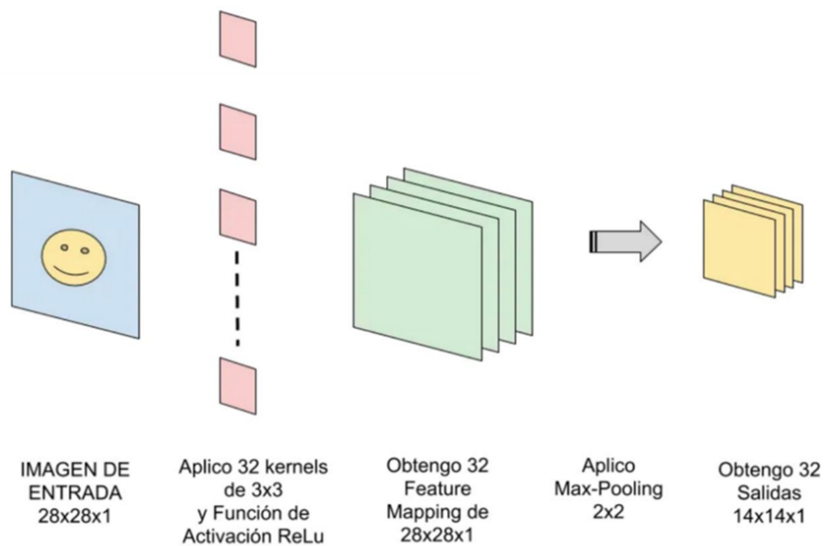


Figura 4.10: Ejemplo de la primear capa de una red neuronal convolucional. Tomada de [69].

- Paso 2:** En la figura 4.11 se muestra un ejemplo de la segunda capa de una red convolucional. La entrada en este caso es el resultado obtenido en la primera capa, que eran 32 imágenes de 14 por 14 píxeles. Después a estas 32 imágenes se le aplican 64 filtros. Pero para poder realizar el producto escalar ahora nuestros filtros serán tridimensionales, es decir, tendrán un ancho y una altura de 3 píxeles, pero tendrán una profundidad de 32; para que el producto interno este bien definido con las imágenes anteriores. Después de la convolución tendremos 64 imágenes de 14 por 14 píxeles. Igualmente, se le aplica una función para tener valores entre 0 y 1 y después se le aplica un *max pooling* de 2 por 2; como resultado obtenemos 64 imágenes de 7 por 7 píxeles. Las siguientes capas convolucionales serán muy parecidas y seguirán estructuras similares a este caso.

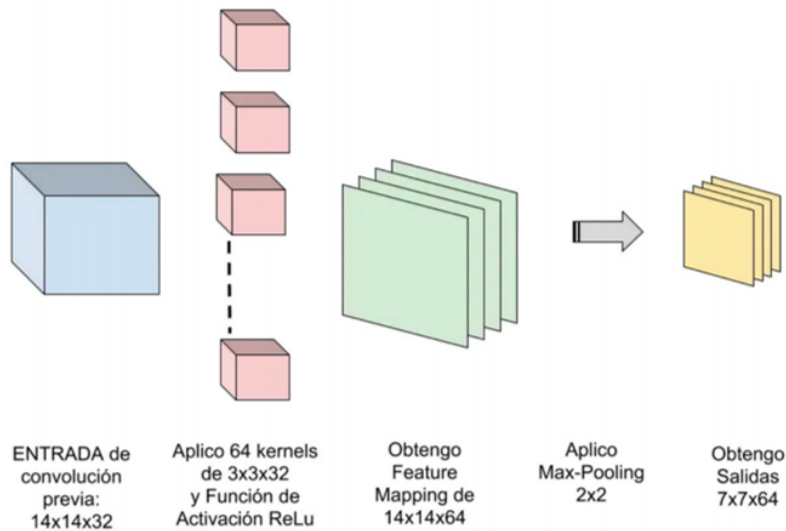


Figura 4.11: Ejemplo de la segunda capa de una red neuronal convolucional. Tomada de [69].

- Paso 3:** Finalmente, en la última etapa de la red convolucional; el valor numérico del píxel de cada imagen o del grupo de imágenes se mapea a un vector. En la figura 4.12 se observa un ejemplo, en donde una imagen de 3 por 3 se mapea a un vector de 9 componente. A este proceso se le conoce como *flatten* o aplanamiento (esta es una capa *fully connected*). El vector obtenido después del aplanamiento, es la capa de entrada de una red perceptrón multicapa que pasa por una serie de capas ocultas para arrojar finalmente un resultado.

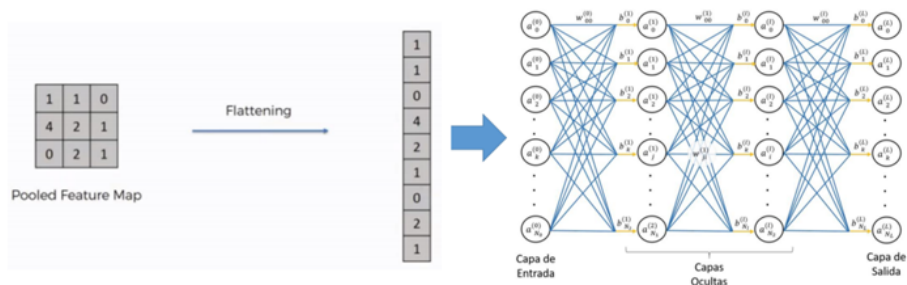


Figura 4.12: Aplanamiento y conexión con una red perceptrón multicapa. Modificada de [65].

Así que en la figura 4.13 se muestra la estructura general de una red convolucional, en donde la imagen de entrada es la de un auto que pasa por una primera convolución. Después se le aplica una capa de *pooling*, luego pasa por una segunda convolución para después pasar por otra capa de *pooling*. Luego se le aplica el aplanamiento, para tener un vector que pasa por una red perceptrón multicapa; para finalmente arrojar el resultado que nos dice la categoría a la cual corresponde.

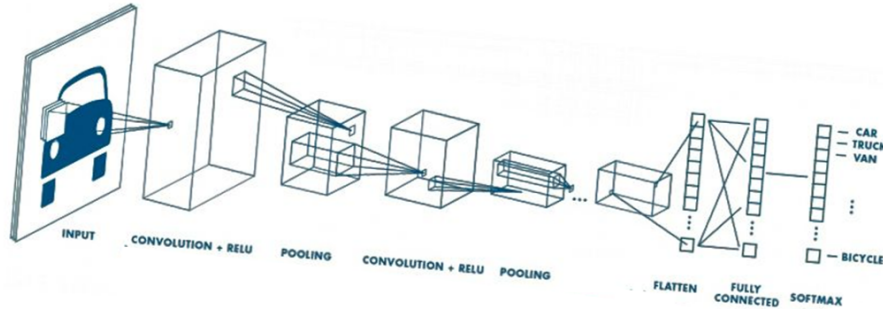


Figura 4.13: Representación general de la estructura de una red convolucional. Tomada de [70].

Cabe destacar que en la mayoría de los casos se utilizan imágenes a color en una red convolucional. Para este caso la entrada de la red serán 3 imágenes, una de color rojo, otra de color verde y una última de color azul. Esto debido a que las imágenes a color están en formato RGB (*red, green y blue*). En la figura 4.14 se ilustra el producto interno para este caso, entre una imagen de color y dos filtros. Aquí cada imagen de entrada se denotan como  $(x(1, 1, i))$  con  $0 \leq i \leq 2$ , y como tenemos 2 filtros se van a generar dos imágenes de salida (denotadas como  $(o(1, 1, i))$  con  $1 \leq i \leq 2$ ). En esta situación cada filtro estará formado por 3 matrices de 3 por 3, denotados como  $(w_j(1, 1, i))$  con  $1 \leq i \leq 2$  y  $0 \leq j \leq 1$ . también se incluye un valor de *bias* para cada filtro  $(b_j(1, 1, 0))$  con  $0 \leq j \leq 1$ . Al denotar la sección de la imagen de entrada resaltada en la figura 4.14 como  $(x(r, r, i))$  el valor de salida para la imagen  $o(1, 1, 1)$  resaltado en color verde en la misma figura se obtiene mediante la siguiente expresión:

$$o(r, r, 1) = b_2(1, 1, 0) + \sum_{i=0}^2 x(r, r, i) \cdot w_2(1, 1, i) \quad (4.18)$$

Los otros píxeles para las dos imágenes de salida se obtienen al ir recorriendo los filtros por toda la imagen (proceso de convolución).

Es importante mencionar que en una red perceptrón multicapa, como se vio anteriormente con cada imagen o evento de entrenamiento se van ajustando los valores de pesos y *bias* para ir obteniendo un mejor resultado. Para el caso de una red convolucional, con cada imagen de entrenamiento se ajusta los valores que hay en cada filtro (también conocidos como pesos de los *kernels*) y de las *bias*. Esto es una gran ventaja al momento del proceso de aprendizaje. Por ejemplo, en la primera convolución de un filtro de 3 por 3, eso son sólo 9 parámetros que debemos ajustar; para el caso de 32 filtros serian un total de 288 parámetros. En comparación con los pesos entre dos capas de neuronas del tipo perceptrón multicapa: una de 748 y otra de 6272 en donde están todas interconectarlas con todas y eso equivaldría a tener que entrenar y ajustar más de 4,5 millones de pesos (sólo para 1 capa) [65].

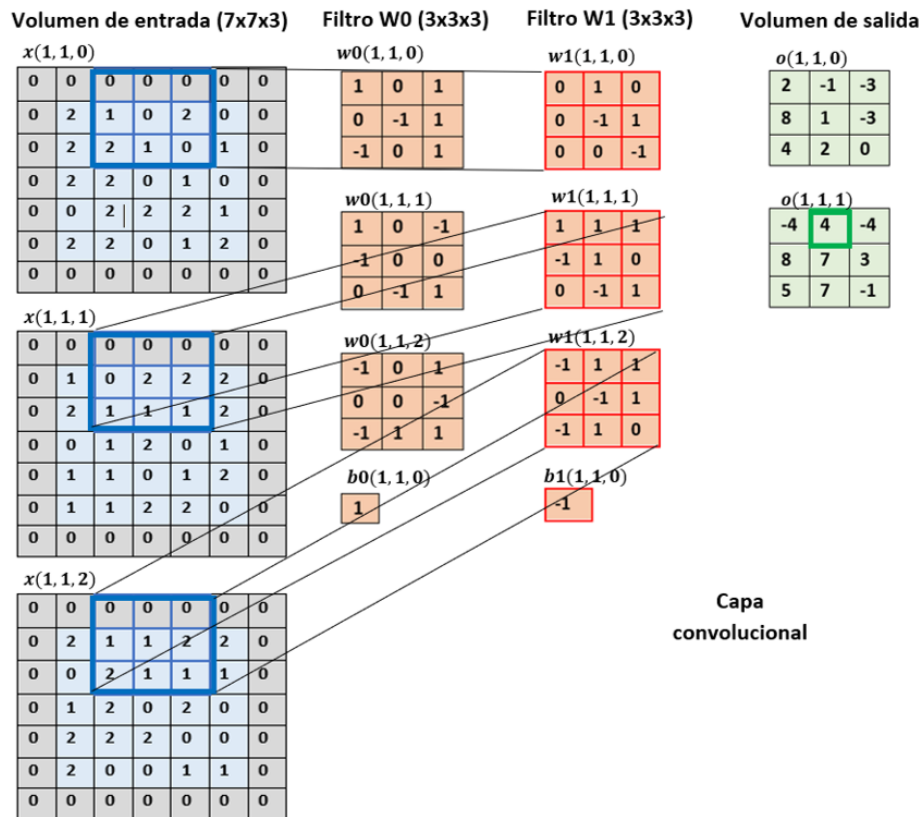


Figura 4.14: Representación gráfica del producto interno entre una imagen a color y dos filtros. Modificada de [71].

### 4.3. Detección de objetos

La detección de objetos utilizando un algoritmo de *machine learning*, implica usar una red neuronal convolucional que detecte una cantidad específica de objetos sobre una imagen. Este tipo de red será capaz de identificar múltiples objetos (si así se desea), dar la posición  $(x, y)$  del objeto en la figura (o su centro) y dibujar un rectángulo a su alrededor [72].

Para comentar de forma intuitiva como funcionan este tipo de algoritmos, supongamos que tenemos una red convolucional que distingue entre imágenes individuales de perros y gatos. Si le pasamos a esta red una imagen con dos perros, es probable que no pueda hacer la clasificación de forma correcta; ya que puede que no detecte ni siquiera a uno. Lo mismo ocurre si le pasamos una imagen con un perro y un gato. Así que una solución para corregir este problema, es escoger un área más pequeña dentro de la imagen e ir moviendo la región observada de izquierda a derecha y de arriba a bajo hasta que red convolucional detecte una imagen. Este procedimiento se ilustra en la figura 4.15, en donde la región observada se muestra con rectángulo de color verde. Con esta idea se soluciona el problema anterior, sin embargo, también tiene los siguientes inconvenientes:

- Si el movimiento de la región observada es de muy pocos píxeles, se pueden estar detectando al mismo perro múltiples veces.
- Surge una problemática de poder distinguir entre animales si estos se encuentran muy cercanos.

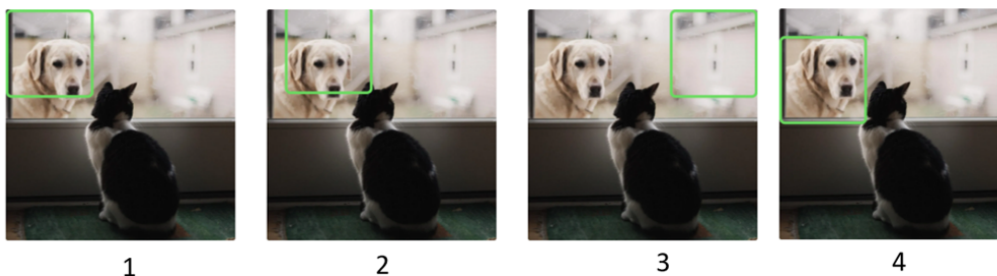


Figura 4.15: Representación de cuatro iteraciones para la búsqueda de un objeto. Tomada de [75].

Para lidiar con los problemas mencionados anteriormente, se utiliza una métrica específica para la detección de imágenes con la que se puede evaluar al mismo tiempo si la clase de objeto es correcta y si la posición del rectángulo es buena. Esta métrica se conoce como *mAP* y se define mediante la expresión 4.19 [72]. En dicha ecuación *TP* hace referencia a los verdaderos positivos y *FP* a los falsos positivos. Cabe mencionar que un mayor valor para esta métrica, indica una mayor precisión en el modelo.

$$mAP = \frac{1}{|clases|} \sum_{c \in clases} \frac{TP(c)}{TP(c) + FP(c)} \quad (4.19)$$

La API (Interfaz de programación de aplicaciones) de detección de objetos, es un marco de trabajo de código abierto creado sobre TensorFlow<sup>1</sup> que facilita la construcción, el entrenamiento y la implementación de modelos de detección de objetos [74]. Ya existen modelos previamente entrenados en dicha interfaz, a los que se refieren como *Model Zoo*. Esto incluye una colección de modelos previamente entrenados. Estos modelos se pueden usar para inferencias, si estamos interesados en categorías solo en este conjunto de datos. También son útiles para inicializar modelos al entrenar en un nuevo conjunto de datos. Las diversas arquitecturas utilizadas en el modelo preentrenado se describen en la siguiente tabla:

Modelo	Velocidad	mAP
ssd_mobilenet_v1_coco	Fast	21
ssd_inception_v2_coco	Fast	24
rfcn_resnet101_coco	Medium	30
faster_rcnn_resnet101_coco	Medium	32
faster_rcnn_inception_resnet_v2_atrous_coco	Slow	37

Tabla 4.1: Modelos preentrenados incluidos en API. Tabla tomada de [74].

En este trabajo se usó el modelo *faster\_rcnn\_resnet101\_coco*, por tal motivo se mostrará de forma breve el funcionamiento de las redes: *R-CNN*, *Fast-R-CNN* y *Faster-R-CNN*, que son la base del modelo que se utilizó.

<sup>1</sup>Es una biblioteca de código abierto para aprendizaje automático desarrollado por Google, para más inf. consultar [73].

### 4.3.1. R-CNN

El modelo R-CNN (Region-based Convolutional Network) fue desarrollado en el año 2014 [75] y fue pionero en la tarea de detección de objetos. La explicación del funcionamiento de este modelo se puede dar mediante los siguientes pasos, que también son representado de forma gráfica en la figura 4.17.

1. Se coloca una imagen como entrada de la red.
2. De la imagen de entrada se seleccionan regiones de interés, que son zonas en la imagen en donde podría estar el objeto a detectar. Para este algoritmo se extraen un poco más de 2000 regiones de forma aleatoria [76]. Para la elección de estas zonas de interés, se emplea el método *selective search*. Dicho algoritmo selecciona muchas regiones diminutas para posteriormente ir incrementando el tamaño de las mismas [77]. La manera en la que se hace esta tarea, es fusionando regiones similares. La similitud entre la región  $a$  y  $b$  se define como:

$$S(a, b) = S_{texture}(a, b) + S_{size}(a, b) \quad (4.20)$$

Donde  $S_{texture}(a, b)$  mide la similitud visual y se define como la intersección de los valores del histograma obtenidos a partir de las mediciones de textura proporcionadas por el algoritmo *SHIFT-like* [77]. Por otro lado,  $S_{size}(a, b)$  se utiliza para fusionar regiones más pequeñas. Esto con la finalidad de evitar que una sola región mayor abarque a todas las demás. En la figura 4.16 se muestra una representación gráfica de este proceso. En donde la imagen inicial tienen un gran número de regiones, después se van combinando estas zonas hasta llegar a las regiones de interés seleccionadas por el rectángulo verde



Figura 4.16: Representación del funcionamiento del algoritmo *selective search*. Tomada de [77].

3. En este paso las regiones de interés serán la entrada a una red convolucional. Cabe mencionar que el tamaño de estas zonas se ajusta a la entrada de la red.
4. Por último, la salida de la red convolucional nos dará la clasificación del objeto.

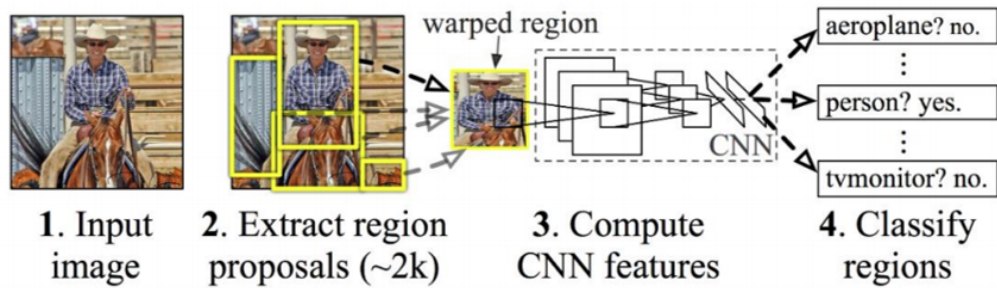


Figura 4.17: Esquema de funcionamiento del detector *R-CNN*. Tomada de [76].

### 4.3.2. Fast-R-CNN

Un inconveniente para la *R-CNN* es que una red debe analizar alrededor de 2000 regiones por cada imagen, eso hace que el tiempo de procesamiento sea relativamente largo. Para aumentar la velocidad con la que se analiza cada imagen, en el año 2015 se introdujo este modelo [75]. La explicación del funcionamiento para la red *fast-R-CNN* se puede dar de manera resumida mediante los siguientes pasos, que también son representados de forma gráfica en la figura 4.18.

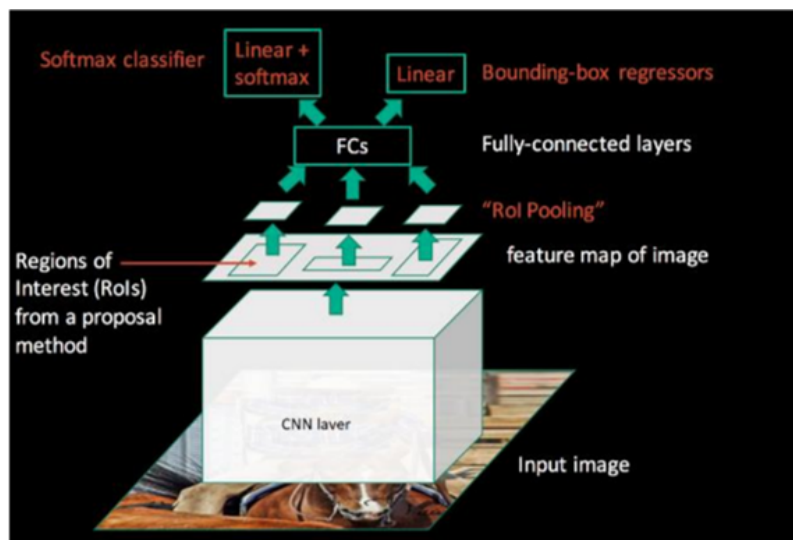


Figura 4.18: Esquema de funcionamiento del detector *Fast-RCNN*. Tomado de [77].

1. La imagen de entrada es procesada por una red convolucional la cual obtiene vectores de característica de la imagen.
2. Los mapas de vectores de características obtenidos en la capa de CNN son utilizados para obtener las regiones de interés.
3. Estas regiones de interés son pasadas por una capa de *region of interest (RoI) pooling*, que hace que tengan el mismo tamaño. El *RoI pooling* consiste en fijar un número concreto  $q$ , partir las regiones de interés en  $q$  partes iguales, o lo más parecidas posibles, y realizar un *max pooling* en esas  $q$  partes [77]. En la figura 4.19 se muestra un esquema de este proceso.

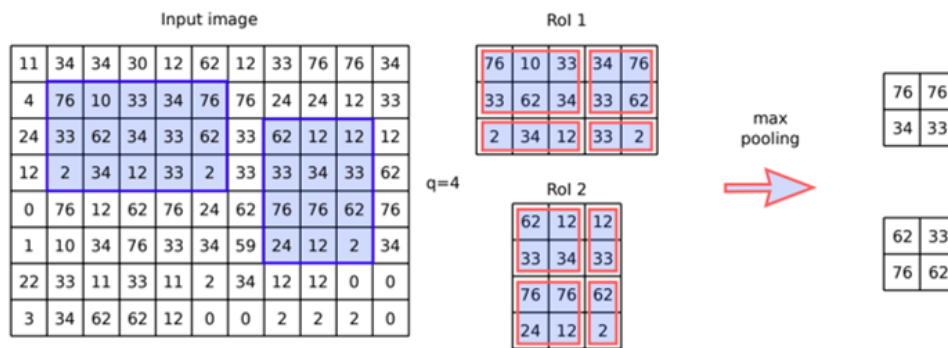


Figura 4.19: Ejemplo de *RoI pooling* para un tamaño de  $q = 4$ . Tomada de [77].

4. La salida obtenida en la capa de *RoI pooling*, se mapea a un solo vector (*fully connected*).
5. La capa *fully connected* obtenida en el paso anterior se divide en 2 capas iguales. Una de ellas produce estimaciones de probabilidades *softmax* sobre  $n$  clases de objetos más una clase de tipo *background*. La otra capa genera mediante regresión lineal, cuatro valores reales para cada una de las  $n$  clases de objetos. Estos valores son la posición de los vértices del rectángulo que delimita al objeto.

### 4.3.3. Faster-R-CNN

Esta estructura también fue elaborada en el año 2015 [75]. La explicación del funcionamiento de esta red se puede entender mediante los siguientes pasos, que también son representado de forma gráfica en la figura 4.20.

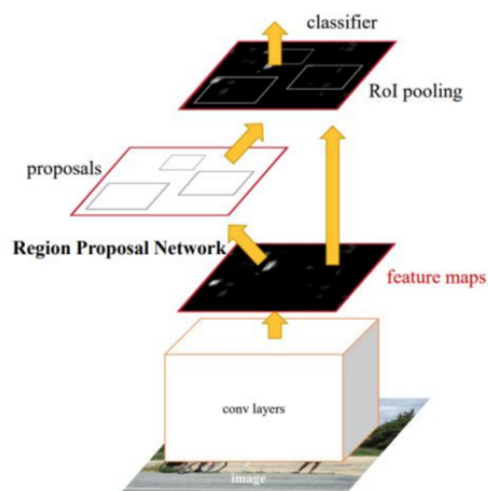


Figura 4.20: Esquema de funcionamiento del detector *Faster-RCNN*. Tomado de [77].

1. Se pasa la imagen de entrada por una red convolucional que extrae las principales características de la imagen.
2. El mapa de características obtenidos en el paso anterior se introduce en otra red convolucional que arroja las regiones de interés de la figura. A esta red especializada para esta tarea se



le conoce como red de propuesta de interés (RPN). Este tipo de red toma como entrada una ventana espacial de  $n \times n$  del mapa de características. Para localizar un objeto en este mapa, se realiza un barrido de diferentes cuadros delimitadores, los cuales se conocen como anclajes. Como primer paso, se colocan sobre la imagen puntos de anclaje distribuidos de forma simétrica. Sobre cada punto se coloca un número  $k$  de anclajes a diferentes escalas y diferentes relaciones de aspecto [77]. En la figura 4.21 se muestran puntos de anclajes uniformemente distribuidos y sobre cada punto se coloca un número  $k = 9$  de anclajes, 3 escalas y 3 relaciones de aspecto.

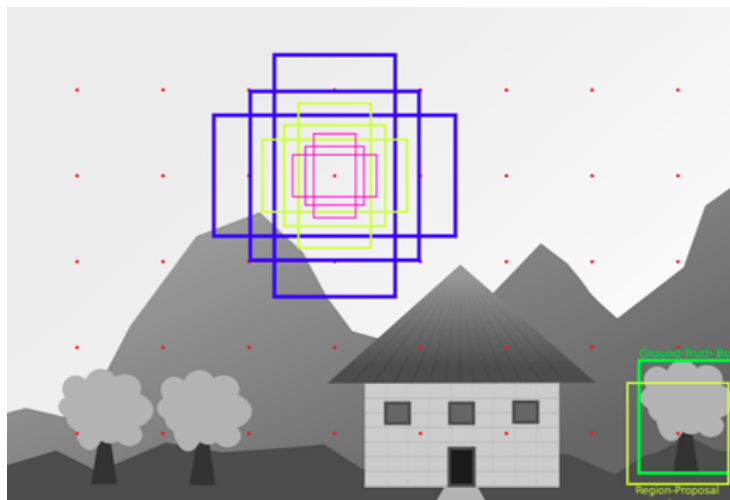


Figura 4.21: Ejemplo de propuesta de regiones en una imagen mediante anclajes. Tomada de [77].

Un *dataset* es todo el conjunto de datos que disponemos para nuestro experimento. Cuando el *dataset* es el resultado de una observación directa, también se le denomina *ground-truth*. Teniendo esto en cuenta a cada anclaje se le asigna una clase, el cual indica si lo que contiene es un objeto o es el fondo de la imagen. Para asignar cada clase se utiliza un umbral en el valor de la intersección entre la unión del anclaje (IoU) y el *ground-truth*. Cabe mencionar, que esta intersección sobre la unión es una métrica que indica cuanto se parecen dos conjuntos. En detección de objetos, cuanto se acerca el área de una detección, o el anclaje en este caso, a la etiqueta en los datos. Este concepto se representa en la figura 4.22 y se define en la ecuación 4.21 [77].

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (4.21)$$

A los anclajes con un  $IoU > 0.7$  se les asigna una etiqueta positiva, en el caso especial en el que ningún anclaje supere el valor de 0.7 (esto puede ocurrir por ejemplo porque las relaciones de aspecto utilizadas no son adecuadas) se etiquetaran como objetos los anclajes con mayor valor IoU. Los anclajes con un valor de  $IoU < 0.3$  se etiquetan como fondo y el resto, los anclajes que no son ni fondo ni objeto, no se utilizan para el entrenamiento. Finalmente, como se comentó anteriormente la salida que arroja esta red serán las regiones de interés de la imagen.

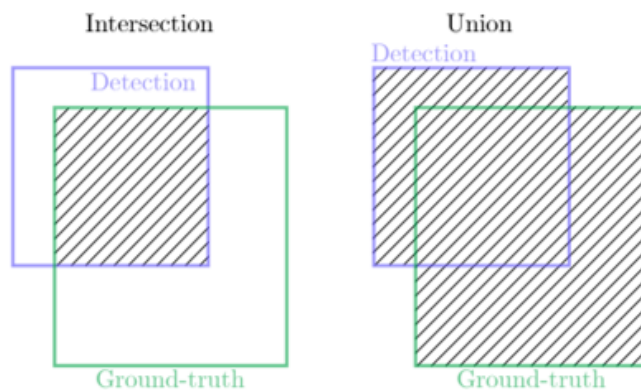


Figura 4.22: Representación de los áreas de intersección e unión en el par detección *ground-truth*.

3. Las regiones de interés obtenidas anteriormente se procesan de la misma forma que para una red *Fast-R-CNN*.

Con esta noción de redes neuronales se procede a explicar el análisis de este trabajo en el siguiente capítulo.

# Capítulo 5

## Identificación de trazas con redes neuronales artificiales

En este trabajo, se implementaron dos tipos de redes neuronales para discriminar entre eventos de trazas horizontales y cascadas atmosféricas. El primer tipo, fue una red convolucional enfocada en la clasificación de imágenes; el análisis y los resultados para este caso se muestran en la primera parte de este capítulo. Posteriormente, se muestra el estudio realizado con el otro caso; que fue una red neuronal orientada a la detección de objetos. Finalmente, se muestra una comparación entre las trazas detectadas con estos dos métodos y el algoritmo que actualmente se utiliza para identificar trazas horizontales y que se comentó en el capítulo 3.

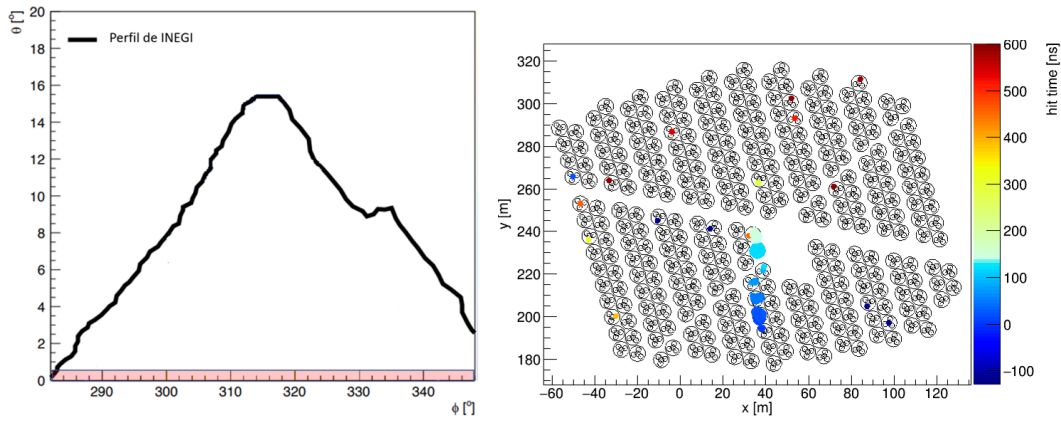
### 5.1. Prueba 1: Red convolucional

En esta primera prueba, se utilizó una red convolucional para clasificar trazas horizontales y cascadas atmosféricas. Debido a que solo teníamos 2 clases, se implementó un clasificador binario. A continuación se describen las imágenes de entrada para este tipo de red neuronal.

#### 5.1.1. Imágenes de entrada

Se procede a describir el tipo de imágenes utilizadas para la entrada de la red neuronal, además de señalar cómo se obtuvieron las imágenes para cada categoría. Cabe mencionar, que todas las imágenes se obtuvieron a partir del *display* oficial de HAWC que viene implementado en AERIE. Solamente se realizó una pequeña modificación al código para que al momento de que se desplegará la imagen se guardará de forma automática.

- **Trazas horizontales:** Para esta categoría se usaron tres diferentes formas para generar este tipo de imágenes:
  - **Simulaciones:** Se utilizaron simulaciones Monte Carlo de trazas de muones horizontales con una energía de 100 GeV. Estas partículas tenían un ángulo de elevación de  $0.28^\circ$  y se simularon un total de 36000 eventos. La comparación de este ángulo de elevación con el terreno se muestra en el inciso (a) de la imagen 5.1, la banda roja indica la elevación de los muones simulados y se compara con el perfil del Pico de Orizaba. Para el ángulo azimutal, se fueron tomando señales en intervalos de 10 grados hasta llegar a 360 grados. Con esto se cubrió las diferentes direcciones por las que se podían detectar a este tipo de señales. Un ejemplo de este tipo de imágenes se muestra en el inciso (b) de la figura 5.1.



(a) Ángulo de elevación (banda roja) de trazas simuladas. (b) Simulación de una traza horizontal.

Figura 5.1: Evento simulado de una traza horizontal ( $0.28^\circ$  de elevación) en los tanques de HAWC (imagen (a)). Comparación con el terreno de las trazas simuladas (imagen (b) modificada de [55]).

- Eventos reales clasificados a mano: Se identificaron de forma manual trazas de un conjunto de aproximadamente 15 mil imágenes. Estas imágenes se obtuvieron a partir de los candidatos proporcionados por el *trigger* para trazas horizontales del primer y tercer *subrun* del *run* 7499. En el inciso a) de la figura 5.2 se muestra un ejemplo para este caso.
- Eventos reales obtenidos con el algoritmo oficial: Estas imágenes se obtuvieron al analizar eventos a partir del *subrun* 3 del *run* 7499 utilizando el algoritmo actual para identificar trazas horizontales. Un ejemplo de una imagen obtenida con este método se muestra en el inciso b) de la figura 5.2.

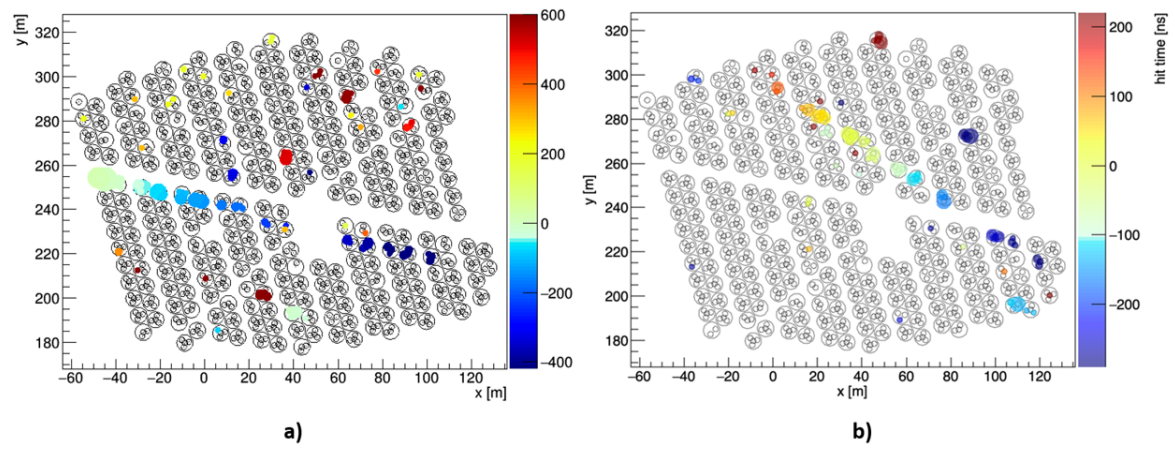


Figura 5.2: Imágenes de trazas horizontales de eventos reales.

Es importante mencionar, que hay algunas diferencias entre las imágenes obtenidas con estos tres métodos. Por ejemplo, en la imagen obtenida con simulaciones se toman en cuenta a los 300 detectores de HAWC. Sin embargo, para el caso de las otras dos figuras no se toman en cuenta a los detectores que no están en funcionamiento. Además, los colores obtenidos en el inciso b) de la figura 5.2 son más claros que los que se muestran en el inciso a) de esa misma

figura. Este hecho afecta los resultados en la red a la hora de hacer el entrenamiento y se discute en la sección 5.1.3.

- **Cascadas:** Estas imágenes se identificaron de forma manual a partir de los candidatos proporcionados por el *trigger* para trazas horizontales del primer *subrun* del *run* 7499. En el inciso a) de la figura 5.3 se muestra un ejemplo para este caso.

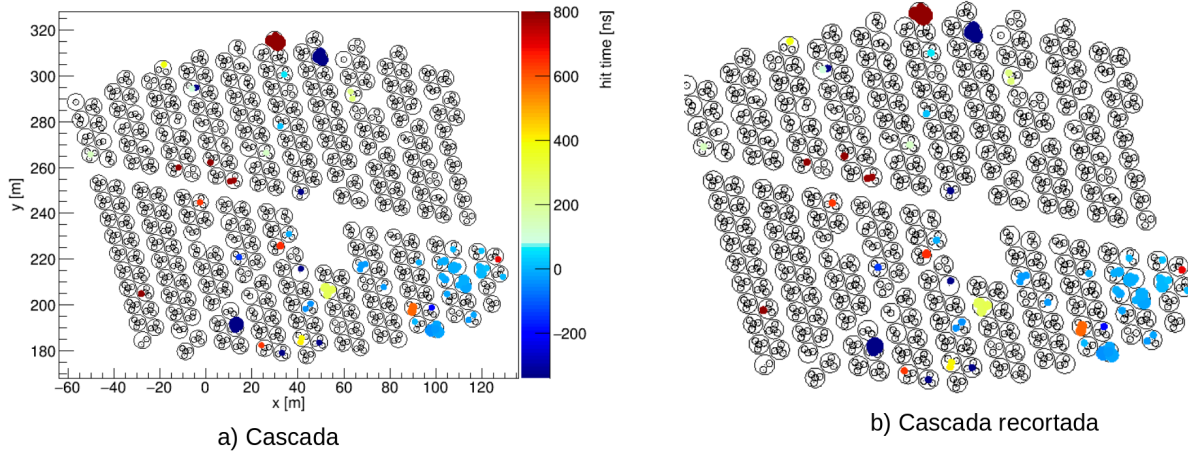


Figura 5.3: Imagen de cascada atmosférica de datos reales.

Es importante mencionar, que se recortaron tanto las imágenes de cascada como de trazas horizontales; de forma que no se vieran los ejes que indican la posición de los PMTs, así como la escala de color. Se tomó esta decisión, al observar una mejora en los resultados de la red para un caso en donde no sea hacia este recorte y uno en el que si se realizaba. Un ejemplo de una imagen recortada se muestra en el inciso b) de la figura 5.3.

### 5.1.2. Estructura de la red

La estructura de la red utilizada fue la VGG16. Esta red fue diseñada por Simonyan y Zisserman en el año 2014 [78]. Cabe mencionar que este modelo se trata de una red preentrenada, es decir, que los valores de los filtros ya han sido calculados en base a un conjunto de datos. Este modelo contiene 13 capas convolucionales y 3 capas *fully connected*, dando un total de 16 capas (por ello lleva ese nombre). La entrada para esta red será una imagen a color de 224 por 224 píxeles, que pasa por las 16 capas que se habían comentado anteriormente y que se muestran de forma gráfica en la figura 5.4. Finalmente, la salida de esta red pasa por la función sigmoide que nos da un resultado entre cero y uno. Es importante señalar que se eligió esta estructura de red después de probar con algunos modelos que no estaban preentrenados. En todos los casos la red VGG16 arrojaba los mejores resultados. En nuestro caso el entrenamiento se realizó para 50 pasos o épocas.

### 5.1.3. Imágenes de entrenamiento y de prueba

Para las imágenes de entrenamiento se trabajó con tres diferentes conjuntos, ya que diferentes pruebas indicaron que la proporción de las imágenes utilizadas generaban diferentes resultados. Los tres diferentes grupos de imágenes utilizados fueron los siguientes:

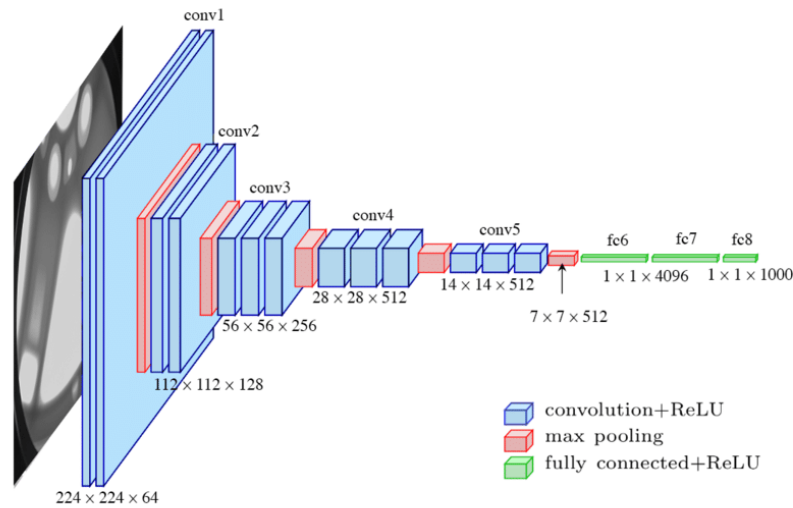


Figura 5.4: Estructura de la red VGG16. Imagen tomada de [78].

- **Conjunto 1:** Se utilizaron 234 imágenes de trazas y 3025 imágenes de cascadas. Las imágenes para las trazas corresponden a 78 imágenes de cada categoría (simulaciones, reales obtenidas con el algoritmo actual y reales que fueron clasificadas a mano). Como se puede observar se utilizó la misma proporción para cada categoría. Este conjunto se propuso con la idea de nivelar los pocos eventos que se tenían de trazas reales con eventos obtenidos por el algoritmo actual y simulaciones. A la red que uso estas imágenes para el entrenamiento la llamaremos modelo A.
- **Conjunto 2:** Se utilizaron 1973 imágenes de trazas y 3920 imágenes de cascadas. Las imágenes para las trazas corresponden a 73 imágenes de trazas reales clasificadas a mano y a 1900 simulaciones. La idea de este modelo surgió después de entrenar una red con solamente simulaciones para las trazas. Para ese caso la red no podía distinguir muchos eventos de trazas reales (debido a que muchas veces también contienen pequeñas cascadas). Sin embargo, se vio una mejora al incluir eventos de trazas reales. A la red que utilizó estas imágenes para el entrenamiento le llamaremos modelo B.
- **Conjunto 3:** Se utilizaron 1050 imágenes de trazas clasificadas a mano y 5627 imágenes de cascadas también elegidas a mano. En el primer *subrun* del *Run 7499*, el *trigger* para trazas horizontales seleccionó un total de 118562 candidatos a trazas. Como es muy tardado revisar a ojo esta cantidad de eventos, se puso una restricción a estos candidatos. Para ello se utilizó la longitud promedio y el tiempo de propagación de la señal registrada por el *trigger* de trazas horizontales, con estos datos se podía calcular el valor de la beta relativista ( $\beta = v/c$ ). Las señales de trazas deben tener el valor de esta variable alrededor de 1 con un cierto ancho. Por otro lado, las cascadas atmosféricas tendrán una distribución mucho más ancha. Se observó la distribución de esta variable para los eventos del primer *subrun* (mostrados en el histograma 5.5) y se decidió aplicar un corte para descartar todos los eventos en donde el valor de beta no estuviera en la siguiente región:  $0.5 < \beta < 1.5$ . Así que solamente se revisaron aproximadamente 15000 eventos que pasaron este filtro para obtener las trazas de eventos reales usados en este conjunto. A la red que utilizó estas imágenes para el entrenamiento le llamaremos modelo C.

Para los datos de prueba se utilizaron dos conjuntos:

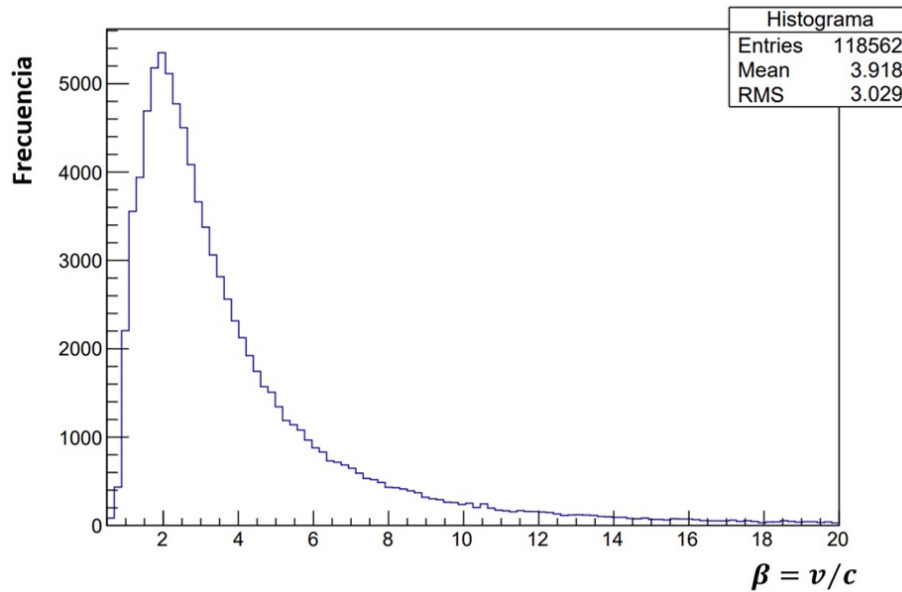


Figura 5.5: Distribución de la variable  $\beta$  para los candidatos del primer *subrun* del *Run* 7499.

- **Conjunto 1:** Está formado por 116 imágenes de trazas reales obtenidas a mano y 116 imágenes de cascadas. Este conjunto se utilizó para calcular la función de costo y la tasa de acierto en cada una de las épocas de entrenamiento. Cabe destacar que no se usaron más eventos debido a las pocas imágenes que se tenía de eventos reales obtenidas a mano, en donde la mayor cantidad se usó para el entrenamiento. Aquí se usó una misma proporción entre imágenes de cascadas y trazas.
- **Conjunto 2:** Está formado por 100 imágenes de eventos de trazas obtenidas a mano y 1000 imágenes de cascada que pasaron por el filtro de  $\beta$ , es decir, son eventos en los cuáles a pesar de hacer el corte en  $\beta$  resultaron cascadas. La proporción de este conjunto se hizo con la idea de simular una proporción similar entre el número de trazas y cascadas que se encuentran en los datos reales.

Se puede notar que no se utilizó ningún evento de trazas simuladas y de trazas obtenidas con el algoritmo actual para probar la red. Esto se debe a las características que tienen estas imágenes (mencionadas anteriormente) y que las distingue de los eventos obtenidos con el *display* de HAWC utilizado en este trabajo. Por tal motivo, los tres modelos clasificaron correctamente 1000 imágenes de cada una de estas categorías. El código empleado para la red se muestra en el apéndice C. Es importante mencionar que debido a la cantidad de imágenes y a la complejidad de la red utilizada, los tiempos de entrenamiento son relativamente largos. Por tal motivo, todos los entrenamientos se llevaron a cabo en la plataforma de *Google Colab* [82]. Esta es una plataforma en internet que permite ejecutar código en *python* y que además da acceso de forma gratuita a GPU por un tiempo limitado (12 horas). En la figura 5.6 se muestra el tipo de GPU al que se puede acceder en esta plataforma (para ese ejemplo fue una *NVIDIA Tesla T4*). Para nuestro caso, los entrenamientos para este tipo de red duraban entre 10 a 20 minutos aproximadamente, en donde la mayoría de las veces se usaba el mismo GPU mencionado anteriormente.

```

from tensorflow.python.client import device_lib
device_lib.list_local_devices()

[{"name": "/device:CPU:0",
  device_type: "CPU",
  memory_limit: 268435456,
  locality {
  }
  incarnation: 13312032622213023874, name: "/device:XLA_CPU:0",
  device_type: "XLA_CPU",
  memory_limit: 17179869184,
  locality {
  }
  incarnation: 17211719027912066112,
  physical_device_desc: "device: XLA_CPU device", name: "/device:XLA_GPU:0",
  device_type: "XLA_GPU",
  memory_limit: 17179869184,
  locality {
  }
  incarnation: 12998984426264930814,
  physical_device_desc: "device: XLA_GPU device", name: "/device:GPU:0",
  device_type: "GPU",
  memory_limit: 14755228544,
  locality {
    bus_id: 1
    links {
    }
  }
  incarnation: 10308131387956119549,
  physical_device_desc: "device: 0, name: Tesla T4, pci bus id: 0000:00:04.0, compute capability: 7.5"]}

```

Figura 5.6: Tipo de GPU accesible en *Google Colab*.

### 5.1.4. Resultado

Con el fin de hacer más clara la explicación de esta sección, solamente se mostrarán las gráficas representativas para el modelo C, que fue el que obtuvo mejores resultados. Sin embargo, la información completa de las gráficas para los otros dos modelos se muestran en el apéndice D.

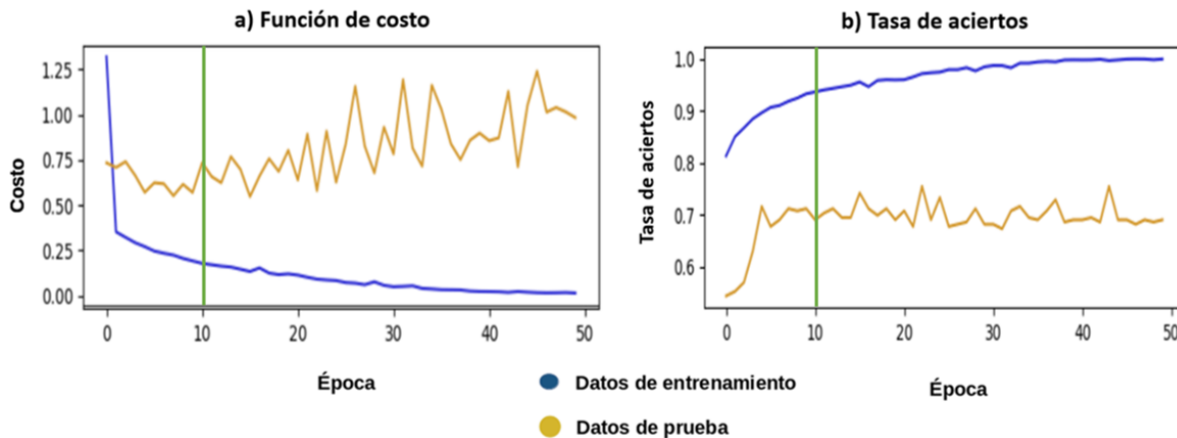


Figura 5.7: Gráfica de costo y la tasa de aciertos para el modelo C.

En primer instancia, en la figura 5.7 se muestra la gráfica para la función de costo y la tasa de aciertos del modelo C durante el entrenamiento. Se puede observar que el costo para los datos de entrenamiento va disminuyendo en cada época, sin embargo para los datos de prueba (conjunto 1) empiezan a aumentar a partir de la época 10. Esto es una clara muestra de que la red se está especializando en los datos de entrenamiento (sobreajuste). Para corregir esta situación, solamente se toma el modelo entrenado hasta la época 10 (indicada por la línea vertical verde en la gráfica anterior). En cuanto a la tasa de aciertos, se puede ver que más del 90% de las imágenes de entrenamiento se clasificaron de forma correcta. Por otro lado, para las imágenes de prueba solo se alcanza un 70% de aciertos. Esto se debe a que muchas veces una imagen de una traza también contiene una cascada, así que la red la clasifica como tal. Sin embargo, aunque esto parezca un valor bajo; en este trabajo nos interesa más el porcentaje de trazas que la red clasifica de forma correcta. Así podremos comparar esta cantidad con el número de trazas que actualmente se identifican y ver



si hay alguna mejora. Por tal motivo, vamos a pasar a la red nuestro segundo conjunto de prueba. Y para ver que tan bien funciona el modelo sobre este conjunto, primero definimos los 4 tipos de predicciones que puede arrojar la red:

- Verdadero positivo (TP): Ejemplo positivo, predicho positivo.
- Falso Positivo (FP): Ejemplo negativo, predicho positivo.
- Verdadero Negativo (TN): Ejemplo negativo, predicho negativo.
- Falso Negativo (FN): Ejemplo positivo, predicho negativo.

Para evaluar que tan buena es una clasificación binaria, se suelen emplear las siguientes variables:

- **Precisión (P)**: Esta variable nos indica que proporción de lo eventos clasificados como positivos realmente lo son.

$$P = \frac{TP}{TP + FP} \quad (5.1)$$

- **Recall (R)**: Nos indica que proporción de todos los positivos se clasifican como tal.

$$R = \frac{TP}{TP + FN} \quad (5.2)$$

- **Taza de falsos positivos (FPR)**: Refleja que proporción de todos los negativos se clasifican como positivos.

$$FPR = \frac{FP}{FP + TN} \quad (5.3)$$

Para este trabajo, se tomó como positivo a un evento clasificado como traza y como negativo a un evento clasificado como cascada. Anteriormente se comentó que la salida de nuestra red es un número entre 0 y 1. Si el evento es más cercano a cero significa que es más probable que sea una traza. Por otro lado, si es más cercano a 1 significa que el evento es una cascada. Para poder calcular las variables anteriormente mencionadas, se necesita definir un valor umbral de aceptación. Por ejemplo, si se tiene un umbral de 0.4 todos los eventos menores a este valor serán clasificados como positivos. Así que el valor umbral es un número real entre 0 y 1 a partir del cual se considera que un evento es positivo o negativo. Como se comentó anteriormente, en este trabajo se busca obtener un porcentaje alto de eventos de trazas que realmente sean trazas, es decir, que se busca un valor alto en la *precisión*. Por tal motivo se decidió hacer una gráfica de *recall* contra *precisión* (también conocida como curva *PR* [79]) para distintos valores umbrales y que se muestra en la figura 5.8. En este tipo de gráfica se busca el valor umbral que nos de un número alto tanto en el *recall* como en la *precisión*. Por tal motivo, en la tabla 5.1 se muestra los valores umbrales que arrojaron los mejores resultados para la *precisión* y el *recall* de los tres modelos analizados. Además se añade la tasa de falsos positivos. En dicha tabla podemos observar que el modelo A y B obtuvieron un valor más alto para la *precisión* si lo comparamos con el modelo C. Sin embargo, también obtuvieron un *recall* muy bajo, es decir, que el modelo A solamente identificó 7 imágenes y el modelo B solamente una imagen de un total de 100 imágenes de trazas que pasamos por la red. Por otro lado, el modelo C identificó de forma correcta 35 de 100 imágenes de trazas. Aunque este modelo haya identificado más trazas que el modelo A y B debe tomarse en cuenta que tuvo la mayor tasa de falsos positivos, que fue de 2%. Aunque parezca un valor muy bajo significa que de 1000 imágenes de cascadas veinte fueron clasificadas de forma incorrecta. Cabe mencionar que este

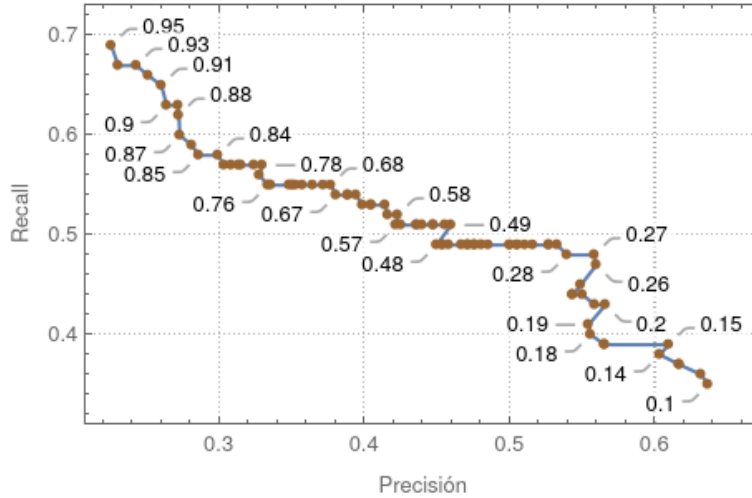


Figura 5.8: Gráfica de PR para el modelo C. Cada número que señala a cada punto representa un valor umbral.

Modelo	Umbral	Precisión %	Recall %	FPR %
A	0.30	87.5	7	0.1
B	0.19	100	1	0
C	0.10	63	35	2
D	0.10	100	11	0
Detección de objetos	-	100	11	0

Tabla 5.1: Evaluación de modelos para la red convolucional.

conjunto de prueba fue diseñado para que siguiera una proporción entre trazas y cascada similar a los que se encuentran en los datos reales. Es decir, que de 1100 eventos analizados 1000 serán de cascada y 100 de trazas. Así que para este ejemplo el modelo C habrá identificado 55 trazas, de la cuales 35 son correctas y 20 son incorrectas. Por lo tanto tenemos una proporción muy similar entre falsos positivos y verdaderos positivos. Esto no es muy eficiente a la hora de analizar miles o cientos de miles de eventos. Por tal motivo se propone un cuarto modelo:

- Modelo D:** Se pasan por la red del modelo C un total de 8426 imágenes de cascada que pasaron el filtro en  $\beta$ . Y se añade a los datos de entrenamiento de este modelo todos los falsos positivos que se encontraron. Con esto le enseñamos a la red que los datos que está clasificando como traza realmente son cascadas. Cabe mencionar que el conjunto 2 que se está utilizando para probar la red no se utilizó para este proceso.

El costo y la tasa de acierto para el modelo D se muestran en la figura 5.9. Esta es muy parecida a la del modelo C, solamente que se obtiene un sobreajuste a partir de la época 20. Por tal motivo solo se usa el modelo hasta esa época. La gráfica de *recall* contra *precisión* se muestra en la figura 5.10. Aquí ya se nota una diferencia si la comparamos con los resultados del modelo C, ya que en esta gráfica se obtiene un valor de *precisión* del 100% y un *recall* del 11% para el umbral de 0.1. Aunque el número de trazas identificadas es menor que el modelo anterior (11 de 100) mejora el valor de la *precisión*. Por ello, en esta ocasión no se obtiene ningún falso positivo. Así que este

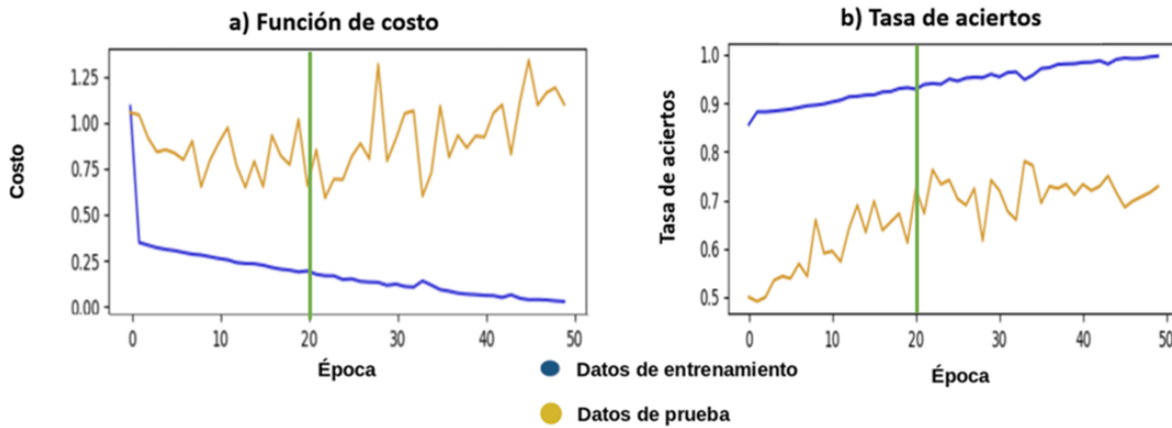


Figura 5.9: Costo y precisión para el modelo D.

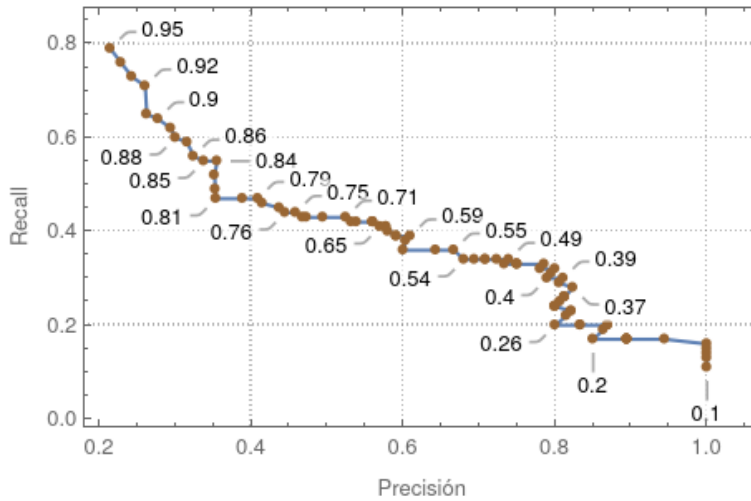


Figura 5.10: Gráfica PR para el modelo D. Cada número que señala a cada punto representa un valor umbral.

modelo será el que se use para compararlo con el algoritmo que actualmente se utiliza para la identificación de trazas horizontales.

## 5.2. Prueba 2: Detección de objetos

En esta segunda prueba, se utilizó una red convolucional enfocada en detección de objetos. También en este tipo de red se tendrán dos clases. Así que para una imagen de entrada, se buscará una figura que corresponda a una traza o a una cascada. A continuación se describen los datos de entrada que se utilizaron para entrenar este tipo de red.

### 5.2.1. Imágenes de entrada

Al igual que en la red convolucional anterior, se utilizaron dos tipos de imágenes de entrada. Las características de estas imágenes se describen a continuación:

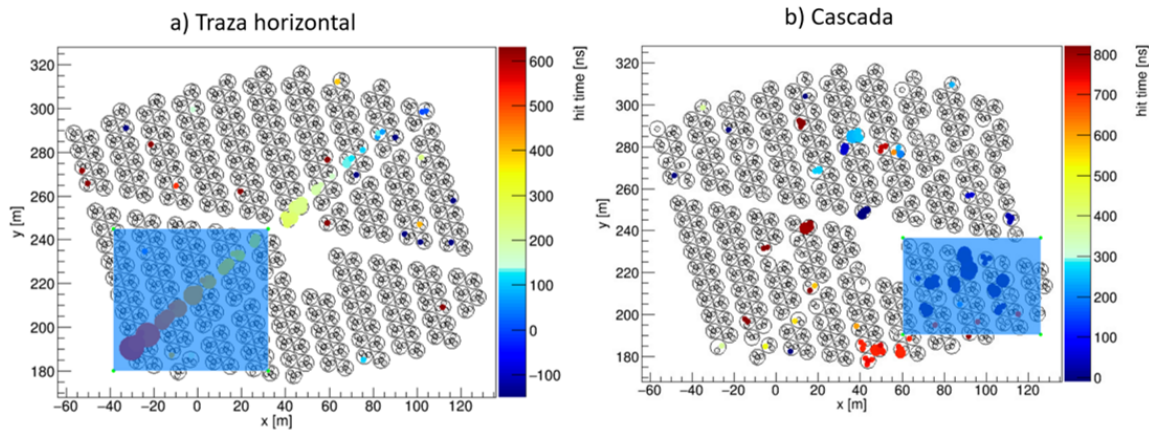


Figura 5.11: Selección para una traza y una cascada usando *labelimg*.

- **Traza horizontal:** Para este tipo de imágenes se utilizaron simulaciones de trazas horizontales con las mismas características que en la red convolucional anterior. Para este caso, en cada imagen se tenía que seleccionar la región que se intentaba detectar. Para esta tarea se empleó el programa *labelimg*, el cual permite dibujar un rectángulo sobre el objeto que se desea detectar. Un ejemplo de esto se muestra en el inciso a) de la figura 5.11. Después de hacer dicha selección con el programa antes mencionado, se podía generar un archivo en formato *xml*. Este archivo contenía la posición en píxeles sobre la imagen del rectángulo formado, la categoría a la cual corresponde (que en este caso sería traza) y el nombre de la imagen que se está analizando.
- **Cascada:** Para esta categoría se usaron imágenes de cascada identificadas de manera manual, al igual que las trazas se utilizó el programa *labelimg* para seleccionar la zona de la imagen que se busca detectar. Un ejemplo de esto se muestra en el inciso b) de la figura 5.11.

### 5.2.2. Estructura de la red

El modelo de red utilizado para la detección de objetos fue el *faster\_rcnn\_resnet101\_coco*. Como se comentó anteriormente, este es un modelo preentrenado y que se encuentra dentro de los modelos para la detección de objetos API. Se escogió este modelo debido a que tiene un valor de *mAP* alto (en comparación a los otros modelos API, ver tabla 4.1) y su velocidad de ejecución es relativamente corta. Además dicho modelo cuenta con una amplia documentación bibliográfica para su implementación. En este trabajo se utilizaron como base las referencias [80]-[83] para poder ejecutar este tipo de modelo. Al igual que la red anterior, el entrenamiento se realizó en la plataforma de *Google Colab*. Dicho entrenamiento constó de 70000 pasos o épocas y duró aproximadamente 10 horas.

### 5.2.3. Imágenes de entrenamiento y de prueba

Para este modelo solamente se utilizó un conjunto para los datos de entrenamiento. El cual consistió en 2866 imágenes de cascadas y 2860 imágenes de trazas. Por otro lado, para probar este modelo se utilizaron las mismas imágenes utilizadas en el conjunto 2 que se comentó en la sección anterior, el cual consistía en 100 imágenes de trazas y 1000 imágenes de cascadas.

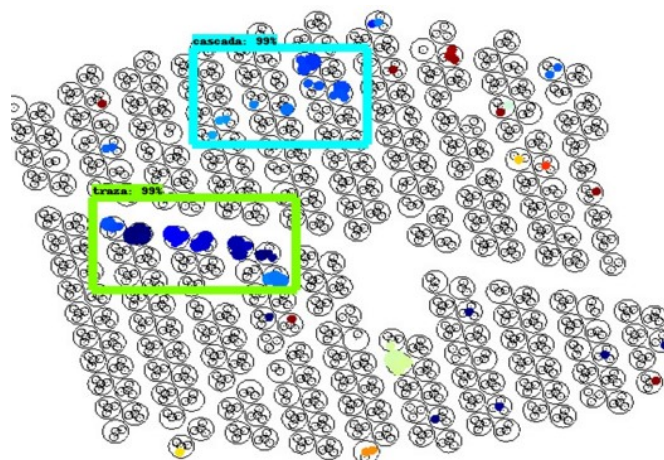


Figura 5.12: Imagen de salida para la red enfocada en detección de objetos.

### 5.2.4. Resultados

La salida para este tipo de red es una imagen con una región que contiene al objeto detectado. Un ejemplo de este tipo de imagen se muestra en la figura 5.12. Como se puede observar, el cuadro azul indica una detección de cascada y el cuadro verde indica una detección de traza. Además, cada uno de estos rectángulos contiene un porcentaje de confiabilidad que indica que tan segura esta la red de lo que identifica es realmente lo que estamos buscando. Para evitar una tasa alta de falsos positivos, se tomó que la red solo pueda presentar objetos con un porcentaje mayor o igual al 99%. Una de las ventajas de este modelo, es que proporciona la posición de la traza detectada. Además, en principio se podría detectar más de una sola traza en una imagen, aunque en los eventos analizados no se observó este caso. La función de costo para este modelo se muestra en la figura 5.13. La línea más clara es el valor del costo, después de aplicar una función de *smoothing* a estos valores (llamada *exponential moving average*) se obtiene la línea que se resalta en naranja. Esto nos da una mejor visualización que nos permite ver que se obtiene el comportamiento esperado para esta función, ya que su valor va disminuyendo conforme se aumenta los pasos de entrenamiento. Aunque es visible que tiene más subidas y bajadas que en el caso de la red anterior. Esto se debe a que esta red es un poco más compleja y por ello también necesita más pasos de entrenamiento, así que para este modelo se utilizó el modelo entrenado hasta el paso 70 mil.

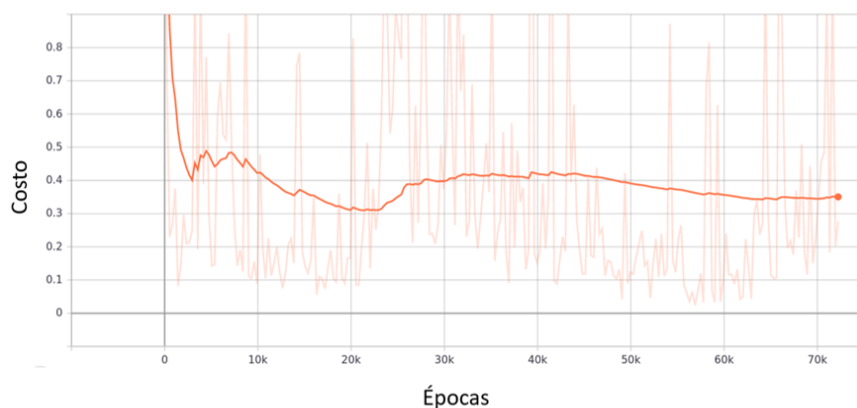


Figura 5.13: Costo para la red enfocada en detección de objetos.

Una vez teniendo el modelo entrenado, se pasó a esta red los datos de prueba. Debido a que esta vez solamente nos tenemos que fijar si la red hizo bien o mal la detección, no es necesario definir ningún umbral. Así que tenemos un clasificador binario al cual le calculamos las variables de *precisión*, *recall* y *FPR*. Estas también se muestra en la tabla 5.1 en donde se puede observar que se obtuvo los mismo valores que el modelo D, aunque cabe señalar que no todos los 11 eventos de trazas identificados por esta red fueron iguales a los del modelo D. Para observar si había una diferencia se aumento en 100 el número de eventos de trazas. Así que utilizando una muestra con 1000 eventos de cascadas y 200 eventos de trazas, el modelo D identificó de forma correcta todos los eventos de cascadas pero solamente 22 eventos de trazas; dando una eficiencias (porcentaje de aciertos respecto al total de eventos probados) del 85.2%. Por otro lado la red enfocada en detección de objetos identificó de forma correcta todos los eventos de cascadas pero solamente 20 eventos de trazas; dando una eficiencias del 85.0%. Con esto se observa que ambas red son buenas discriminado eventos de cascadas.

Finalmente procedemos a realizar la comparación entre el modelo D, la red de detección de objetos y el algoritmo actual sobre una misma muestra de datos. Este es el tema de la siguiente sección.

### 5.3. Comparación con el algoritmo actual

Para realizar la comparación entre los 3 algoritmos que se tienen, se eligió como datos de muestra todos los eventos proporcionado por el *trigger* de trazas para los eventos del segundo *subrun* del *run* 7499. Esto nos daba un total de 118476 candidatos a analizar. Posteriormente a todos estos candidatos se les aplicó el corte en  $\beta$ , dando como resultado un total de 10668 eventos que pasaron este filtro. Así que se pasó esta cantidad de imágenes por los dos modelos de red. En la tabla 5.2 se muestran todas las trazas horizontales identificada por los tres modelos. Como se puede observar, tanto el modelo D como el de detección de objetos identificaron una mayor cantidad de trazas horizontales en comparación con el algoritmo actual. Aunque cabe señalar que el modelo D obtuvo 3 falsos positivos, que representan un 2.8% del total de trazas que identifico. Además, la red para la detección de objetos identifico 11 trazas menos que el modelo D; siendo que en la prueba realizada sobre el conjunto 2 obtenía un mismo valor para el *recall*. Esto se debe a que se necesitan más 1100 imágenes en el conjunto de prueba para encontrar los porcentajes adecuados. Aunque aumentar el número de imágenes de este conjunto es algo tardado debido a que se requiere una clasificación manual. Sin embargo, no se obtiene porcentajes muy alejados. Para la detección de objeto se siguió manteniendo un porcentaje nulo en los falsos positivos y para el modelo D fue menor al 3%. Cabe mencionar que el apéndice E se maestra las imágenes de los eventos encontrados por el modelo D.

Modelo	Tazas identificadas	Falsos positivos
Algoritmo Actual	9	0
D	103	3
Detección de objetos	92	0

Tabla 5.2: Trazas horizontales identificadas por los tres modelos de un total de 118476 eventos analizados.

# Conclusiones

En este trabajo se implementaron dos modelos de redes neuronales, una enfocada en detección de objetos y otra en clasificación de imágenes. Ambos algoritmos se utilizaron para discriminar entre eventos de trazas horizontales y de cascadas atmosféricas. En la muestra de prueba, que consistió en todos los eventos del segundo *subrun* del *run* 7499 que pasaron el *trigger* para trazas horizontales y el corte en beta (10668 eventos). Se observó para ambos modelos, una mejora de un orden de magnitud en el número de trazas que actualmente se identifican. Aunque este es un resultado preliminar, luce bastante prometedor. Sin embargo, debido a la gran cantidad de eventos que se esperan analizar para un escenario real (para un *run* completo serían alrededor 7 millones de imágenes) todavía se tendrían que pasar por la red una muestra más grande de eventos para conocer con certeza el porcentaje de falsos positivos que se podrían encontrar; que en esta ocasión fue de un 2.8% del total de imágenes identificadas para la red enfocada a la clasificación de imágenes. En este sentido la red enfocada en detección de objetos luce más prometedora, ya que en las pruebas hechas en este trabajo el porcentaje de falso positivos fue nulo y logro identificar más trazas que el algoritmo actual. Por lo tanto, en planes a futuro se espera hacer el conteo de trazas identificadas para un *run* entero y ver que tanto cambian los porcentajes para los falsos positivos.

Con ambos modelos se obtuvo un aumento en el número de trazas que se identifican. Con este incremento se puede mejorar la estadística en trabajos como los presentados en [55], en donde se estudia la sombra en muones del Pico de Orizaba a través del análisis en trazas horizontales. Además se podría incrementar el número de candidatos para encontrar una señal generada por un neutrino ultra energético.

# A) Mecanismos de producción y aceleración para rayos gamma

A continuación se mencionan los procesos involucrados en la emisión de rayos gammas.

- **Bremsstrahlung:** Una partícula cargada es desacelerada y desviada al pasar en un campo eléctrico generado por otra partícula cargada (que podría ser un núcleo atómico o un electrón ver inciso a) de figura 5.14 ). La energía perdida durante la desaceleración produce un fotón, a este proceso se le conoce como *Bremsstrahlung* o radiación de frenado [7]. Para el caso de un electrón, la tasa de pérdida de energía cuando estos son desacelerados se puede describir mediante la siguiente expresión:

$$-\frac{dE}{dt} = \frac{cm_p n}{X_0} E \quad (5.4)$$

Donde  $E$  es la energía del electrón,  $m_p$  es la masa del protón,  $c$  es la velocidad de la luz y  $n$  es el número de densidad del gas que se encuentra en el ambiente.  $X_0$  es la longitud de radiación, que se define como la distancia media en la que un electrón pierde  $1/e$  de su energía. El espectro de energía del fotón emitido mediante este proceso es continuo, y disminuye como  $1/E_\gamma$  a altas energías [5].

- **Dispersión inversa de Compton:** En este mecanismo un electrón acelerado a altas energías y con velocidades relativistas, colisiona con un fotón que gana energía y es acelerado debido a este choque [7]. En el ambiente astrofísico, los fotones que pueden ser acelerados provienen de la radiación de fondo de microondas, radiación infrarroja producida por el polvo del medio interestelar y fotones emitidos por la luz de las estrellas. La sección transversal ( $\sigma_{IC}$ ) para este proceso, se puede colocar en función de la energía del electrón  $E_e$  y la energía inicial del fotón  $\omega_0 = h\nu$ . Para ello se define el término de Klein-Nishina (KN)  $\epsilon_0 = \omega_0/E_e$ , así que la sección transversal se puede expresar como:

$$\sigma_{IC} = \frac{3\sigma_T}{8\epsilon_0} \left[ \left(1 - \frac{2}{\epsilon_0} - \frac{2}{\epsilon_0^2}\right) \ln(1 + 2\epsilon_0) + \frac{1}{2} + \frac{4}{\epsilon_0} - \frac{1}{2(1 + 2\epsilon_0)^2} \right] \quad (5.5)$$

Donde  $\sigma_T$  es la sección transversal de dispersión de Thomson, que se define como  $\sigma_T = (8\pi/3)r_e^2$  con  $r_e$  indica el valor del radio clásico del electrón. Dependiendo de la energía del electrón y del fotón la dispersión inversa de Compton se puede dividir en 2 régimen [7]: El régimen de Thompson (donde  $\epsilon_0 \ll 1$ ) y el régimen KN (donde  $\epsilon_0 \gg 1$ ). La tasa de pérdida de energía de los electrones para este proceso en el régimen de Thompson, se puede describir mediante la siguiente expresión:

$$-\frac{dE_e}{dt} = \frac{4}{3}\sigma_T c U_{rad} \left(\frac{\nu\gamma}{c}\right)^2 \quad (5.6)$$

En este caso  $U_{rad}$  es la densidad de los campos de radiación ambiental y  $\gamma$  es el factor de Lorentz del electrón en movimiento con velocidad  $\nu$ . La ecuación 5.6 se puede resolver en



términos  $E_e$  y en el régimen de Thompson es proporcional a  $E_e^2$ . Una expresión similar se puede obtener en el régimen de KN donde la tasa de pérdida de energía de los electrones es independiente de su energía ( $E_e$ ) [7]. Este mecanismo se muestra en el inciso b) de la figura 5.14.

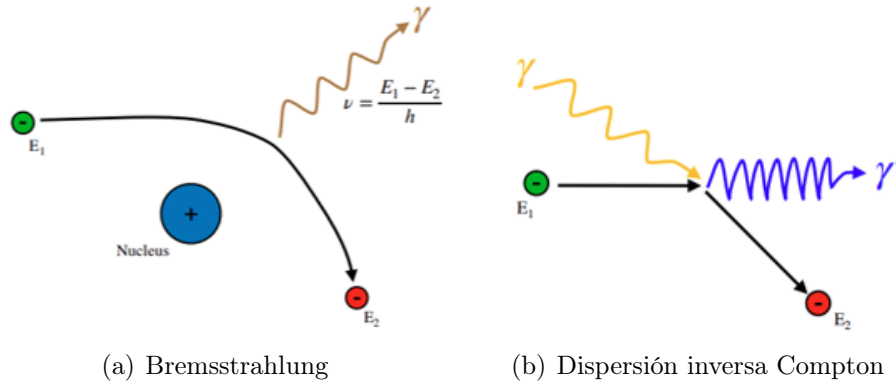


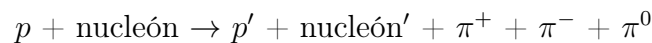
Figura 5.14: Representación gráfica para Bremsstrahlung y Compton inverso. Imagen tomada de [7].

- Radiación de sincrotrón:** Cuando un electrón se mueve en un campo magnético este se acelera. El electrón describe una trayectoria en forma de espiral alrededor de las líneas del campo magnético durante su movimiento (ver inciso a) de figura 5.15). Dicha trayectoria se obtiene porque la dirección de la velocidad del electrón es perpendicular al campo magnético. Debido a este movimiento circular, el electrón se acelera y produce radiación electromagnética, a esta radiación se le conoce como radiación de sincrotrón [5]. La tasa promedio de pérdida de energía del electron debido a este proceso, se puede describir por la siguiente ecuación:

$$-\frac{dE_e}{dt} = \frac{4}{3}\sigma_T c U_{mag} \left(\frac{\nu\gamma}{c}\right)^2 \quad (5.7)$$

En este caso  $U_{mag} = B^2/\mu_0$  es la densidad de energía del campo magnético ambiental con  $\mu_0$  representando el valor de la permeabilidad del campo magnético en el vacío. Por otro lado la potencia irradiada  $P$  por un electrón de energía  $E$  en un campo magnético  $B$  será proporcional a  $E^2 B^2$  [5].

- Decaimiento del pión:** Protones acelerados en fuentes astrofísicas, puede producir piones neutro y con carga. Esto mediante las interacciones protón-protón y protón-nucleón [5]. Un posible proceso sería:



Los piones cargados decaen en muones y neutrinos, mientras que los piones neutros decaen rápidamente ( $\tau = 8.4 \times 10^{-17}$  s) en dos fotones.

$$\pi^0 \rightarrow \gamma + \gamma$$

Si un pión neutro no está en movimiento, los fotones emitidos tienen cada uno la mitad de la energía del pión cuando estaba en reposo  $m_0^{\pi^0} \approx 135$  MeV. Para un pión neutro que se encuentra en movimiento los fotones emitidos obtienen diferentes energías dependiendo de

su dirección de emisión con respecto a la dirección de vuelo del pión neutro. En procesos astrofísicos, los piones formados obtienen velocidades relativistas. Por lo tanto, los rayos gamma generados en este proceso se encuentran en el rango de 50 GeV A 100 TeV [7]. Este mecanismo se muestra en el inciso b) de la figura 5.15.

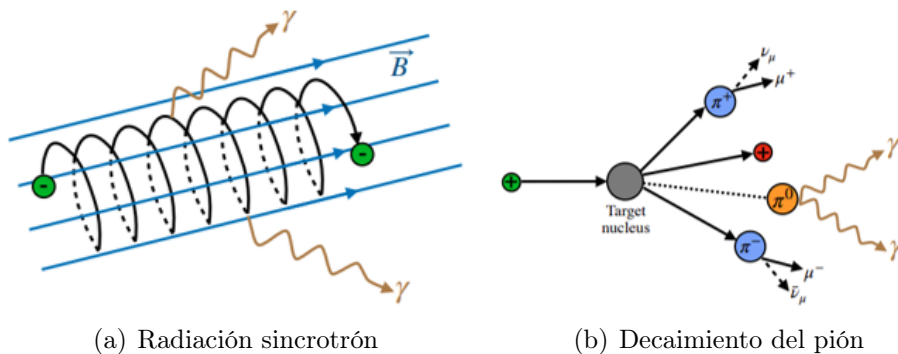


Figura 5.15: Representación gráfica para radiación sincrotrón y decaimiento del pión. Imagen tomada de [7].

## B) Funciones de AERIE y principales resultados obtenidos por HAWC

Algunas de las funciones que tiene AERIE para el análisis de datos:

- **Reconstrucción de Cascadas:** A partir de los datos obtenidos por el observatorio, y que han sido clasificado como posibles candidatos a cascadas atmosféricas, se puede utilizar este *software* para realizar una reconstrucción de esta información, en donde se obtendrá tanto el núcleo de la cascada como la dirección en la que llega. En la imagen 5.16 se muestra un ejemplo de esta reconstrucción, en donde la estrella que está en el centro del círculo (con un radio de 40 m) representa el núcleo de la cascada, la escala de colores indica el tiempo en que se detecta la señal y el tamaño de cada marcador es proporcional a los fotoelectrones detectados. Por medio de estos diagramas se pueden distinguir las cascadas generadas por un protón o por un fotón. Las cascadas generadas por un protón se caracterizan por tener señales grandes lejos del núcleo de la cascada, debido a que las partículas que aquí se generan son muy penetrantes (ver inciso a) de la figura 5.16), y las cascadas generadas por un fotón tiene la características de ser mas simétricas o uniformes y tener señales grandes cerca del núcleo de la cascada. Esto se debe a que los fotones y electrones creados aquí no tienen un gran momento transversal ( ver inciso b) de la figura 5.16).

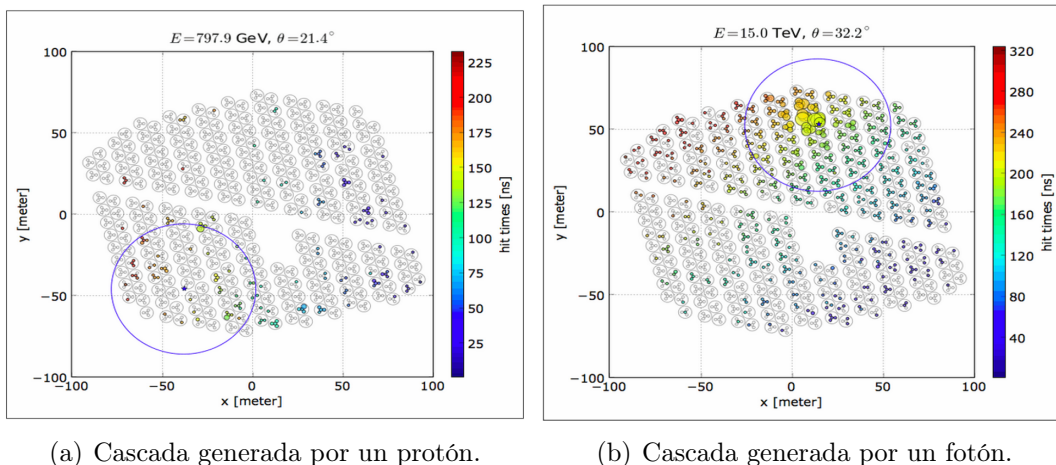


Figura 5.16: Reconstrucción de Cascadas con AERIE. Imagen tomada de [1].

- **Mapas del cielo:** Se pueden hacer mapas del cielo a partir de los datos de cascada que han sido reconstruidos, como el que se muestra en la imagen 5.17, que nos da información de las fuentes en el Universo que generan rayos gammas.

- **Simulaciones:** Se pueden simular cascadas atmosféricas que llegan al observatorio, o la respuesta de los detectores al paso de una partícula en específico.

Entra algunos de los resultados destacados publicados por parte de la colaboración se encuentran los siguientes:

- En el año 2013 se observó la sombra de la luna en Rayos Cósmicos, que fue un estudio en donde únicamente se utilizaron 30 tanques ya que la construcción del observatorio no había terminado , y que atrajo la atención de medios internacionales como la BBC [1].
- Se observó un fenómeno transitorio, en donde se encontró un aumento en la emisión de rayos gammas en la galaxia Makarian 501 el 6 de abril del 2016 [1].
- Presentación de un mapa del cielo de rayos gammas en altas energías (ver imagen 5.17), en donde se observan nuevas fuentes de emisión a lo largo de la Vía Láctea, así como objetos extragalácticos como los Makarian 421 y 501 [1].
- En el año 2017 HAWC realizó mapas detallados en rayos gamma de dos pulsares, dichos objetos fueron considerados como los mejores candidatos para producir el exceso de positrones que se mide en la Tierra. Estas mediciones muestran que estos pulsares no son las fuentes del flujo anómalo de positrones detectados en la Tierra. Con ello se eliminan a dos de los más probables candidatos que podrían producir el exceso de positrones [1].
- En 2018 los investigadores en HAWC realizaron la primera observación de chorros del microquásar SS 433. Este es un sistema binario ubicado dentro de la nebulosa W50 [1].
- En 2020 HAWC detectó la emisión de rayos gamma proveniente del remanente de supernova G106.3+2.7. Al combinar los resultados de HAWC con mediciones previas obtenidas por VERITAS, se encontró que la emisión de rayos gamma continúa hasta energías mayores a 100 TeV. Los rayos gamma detectados por HAWC podrían provenir ya sea de protones con energías de al menos 800 TeV o de electrones con energía de al menos 270 TeV. Estos resultados proveen evidencia de que G106.3+2.7 podría ser un Pevatrón [1].

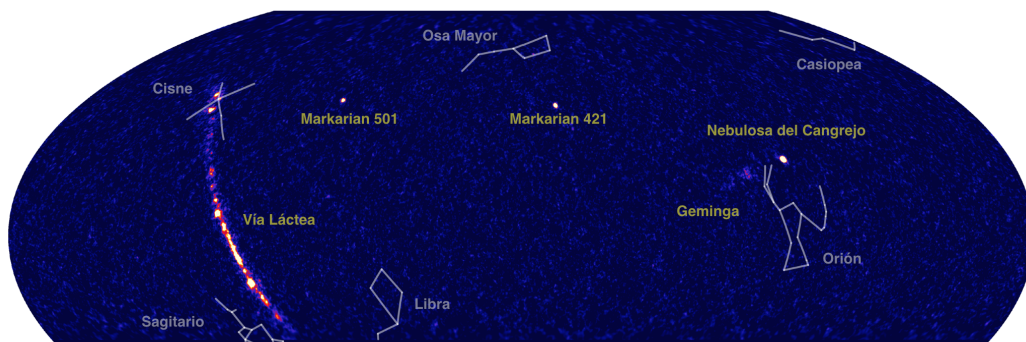


Figura 5.17: Mapa del cielo en rayos gamma. Imagen tomada de [9].

En este centro de investigación se encuentran estudiando muchos temas interesantes como la búsqueda de nuevas fuentes de emisión de rayos gammas, la búsqueda de señales de aniquilación de materia oscura y diferentes estudios sobre el Sol, por mencionar algunas líneas de investigación.

Además existe la propuesta de utilizar este observatorio como un detector de neutrinos, que es el tema en el que se va a centrar el siguiente capítulo.

### Función NKG

La teoría de la propagación lateral de las partículas electromagnéticas fue desarrollada por Greisen, Katamata y Nishimura [46]. En cada punto de interacción, los procesos de producción de pares y bremsstrahlung emiten partículas que no están alineadas con precisión a lo largo de la dirección primaria, y una mayor dispersión de Coulomb de los  $e^\pm$  da como resultado una mancha lateral del frente de partículas. La aproximación de la densidad de electrones a una distancia  $r$  desde el eje de la cascada viene dada por la función NKG citeveinticuatro.

$$\rho_e(r) = \frac{N_e}{2\pi R_m^2} C(s) \left(\frac{r}{R_m}\right)^{(s-2)} \left(1 + \frac{r}{R_m}\right)^{(s-9/2)} \quad (5.8)$$

En este caso  $N_e$  es el número total de electrones en la cascada y  $C(s) = 0.366s^2(2.07 - s)^{1.25}$  es una expresión de normalización tal que  $\int_0^\infty 2\pi r \rho(r) dr = N_e$ . El parámetro  $s$  es la edad de la cascada. Por definición la cascada inicia a  $s = 0$  y alcanza su maximo a  $s = 1$ . Por ultimo  $R_m$  es el radio de Molière. La forma radial de la función NKG significa que las cascadas electromagnéticas están densamente colimadas, con una densidad de partículas que disminuye rápidamente al aumentar la distancia lateral desde el eje de la ducha [46].

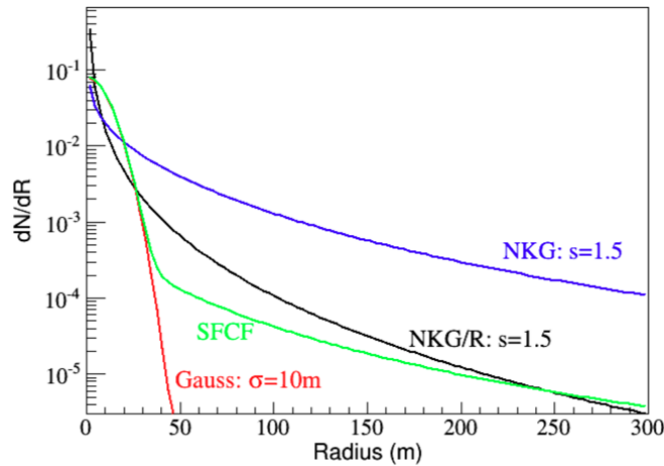


Figura 5.18: Comparación de funciones de distribución lateral [7].

En la figura 5.18 se muestran la función de distribución lateral para el numero promedio esperado de partículas (NKG) y la energía de la cascada electromagnética iniciada por un rayo gamma (NKG/R) como función del radio de la cascada medido en e plano de esta. Ambas utilizan un parámetro  $s = 1, 5$ . También se muestra la función de distribución lateral SFCF utilizada para ajustar las posiciones del núcleo de la cascada en HAWC. Esta se aproxima a la energía electromagnética esperada de la cascada de la distribución NKG/R en el límite de grandes distancias desde el eje de la cascada y coincide con un gaussiano bidimensional con un ancho de 10 m a pequeñas distancias desde el eje de la cascada [41].

## C) Códigos utilizados para las redes neuronales artificiales

En esta sección se colocan los códigos empleados para el entrenamiento de los dos tipos de redes neuronales estudiadas en este proyecto. Se empieza por el código para la red convolucional enfocada a la clasificación de imágenes. Este código fue desarrollado en *python* utilizando las librerías de *Tensorflow 2.0*. La implementación de este código se basó principalmente en la referencia [84]. El código utilizado para el entrenamiento fue el siguiente:

```
1 import sys
2 from matplotlib import pyplot
3 from keras.utils import to_categorical
4 from keras.applications.vgg16 import VGG16
5 from keras.models import Model
6 from keras.layers import Dense
7 from keras.layers import Flatten
8 from keras.optimizers import SGD
9 from keras.preprocessing.image import ImageDataGenerator
10
11 # definicion del modelo para la cnn
12 def define_model():
13     # cargamos el modelo
14     model = VGG16(include_top=False, input_shape=(224, 224, 3))
15     # se marcan las capas cargadas como no transferibles
16     for layer in model.layers:
17         layer.trainable = False
18     # se colocan nuevas capas para la clasificacion
19     flat1 = Flatten()(model.layers[-1].output)
20     class1 = Dense(128, activation='relu', kernel_initializer='he_uniform')(flat1)
21     output = Dense(1, activation='sigmoid')(class1)
22     # definimos el nuevo modelo
23     model = Model(inputs=model.inputs, outputs=output)
24     # compilamos el modelo
25     opt = SGD(lr=0.001, momentum=0.9)
26     model.compile(optimizer=opt, loss='binary_crossentropy', metrics=['accuracy'])
27     return model
28
29 # se define una funcion para graficar el costo y la tasa de aciertos
30 def summarize_diagnostics(history):
31     # grafica para el costo
32     pyplot.subplot(211)
33     pyplot.title('Cross Entropy Loss')
34     pyplot.plot(history.history['loss'], color='blue', label='train')
35     pyplot.plot(history.history['val_loss'], color='orange', label='test')
36     # grafica para la tasa de aciertos
37     pyplot.subplot(212)
38     pyplot.title('Classification Accuracy')
```

```

39  pyplot.plot(history.history['accuracy'], color='blue', label='train')
40  pyplot.plot(history.history['val_accuracy'], color='orange', label='test')
41  # comandos para guardar las graficas
42  filename = sys.argv[0].split('/')[0]
43  pyplot.savefig(filename + '_plot.png')
44  pyplot.close()
45
46  # Se ejecuta el test harness para evaluar el modelo
47  def run_test_harness():
48      # se define el modelo
49      model = define_model()
50      # se crea el generador de datos
51      datagen = ImageDataGenerator(featurewise_center=True)
52      # especificaciones para las imagenes
53      datagen.mean = [123.68, 116.779, 103.939]
54      # se especifican las carpetas que contienen las imagenes de prueba y
      # entrenamiento.
55      train_it = datagen.flow_from_directory('dataset_traza_vs_cascada/train/',
56      class_mode='binary', batch_size=64, target_size=(224, 224))
57      test_it = datagen.flow_from_directory('dataset_traza_vs_cascada/test/',
58      class_mode='binary', batch_size=64, target_size=(224, 224))
59      # modelo de ajuste
60      history = model.fit_generator(train_it, steps_per_epoch=len(train_it), epochs=20,
61      verbose=1)
62      # se guarda el modelo
63      model.save('VGG16_500_2000.h5')
64      # evaluacion del modelo
65      _, acc = model.evaluate_generator(test_it, steps=len(test_it), verbose=0)
66      print('> %.3f' % (acc * 100.0))
67      # resumen del entrenamiento
68      summarize_diagnostics(history)
69
70  # ejecucion del modelo
71  run_test_harness()

```

Ahora procedamos a colocar el código utilizado para hacer la predicción de un conjunto de imágenes.

```

1  # prediccion para una nueva imagen
2  from keras.preprocessing.image import load_img
3  from keras.preprocessing.image import img_to_array
4  from keras.models import load_model
5
6
7  # cargamos y preparamos la imagen de entrada
8  def load_image(filename):
9      # cargamos la imagen
10     img = load_img(filename, target_size=(224, 224))
11     # se convierte a una matriz
12     img = img_to_array(img)
13     # se ajusta la forma de imagen para que coincida con la entrada de la red
14     img = img.reshape(1, 224, 224, 3)
15     # se coloca la posicion del centro de la imagen
16     img = img.astype('float32')
17     img = img - [123.68, 116.779, 103.939]
18     return img
19
20 # cargamos el modelo entrenado y hacemos la prediccion, el ciclo for corre sobre el

```

```

    numero de imagenes a analizar
21
22 for i in range(1,102):
23     # cargamos la imagen
24     img = load_image(str(i)+".png")
25     # cargamos el modelo
26     model = load_model('gato_perro.h5')
27     # se hace la prediccion
28     result = model.predict(img)
29     print(result[0],i)
30     # se guarda en un archivo de texto
31     archivo = open("datos.txt","a")
32     numero = result[0]
33     archivo.write('\n'+ '%s'%numero)

```

Para el caso de la red neuronal enfocada en la detección de objetos, la estructura de la carpeta que tenía el código de la red se muestra en la siguiente figura:

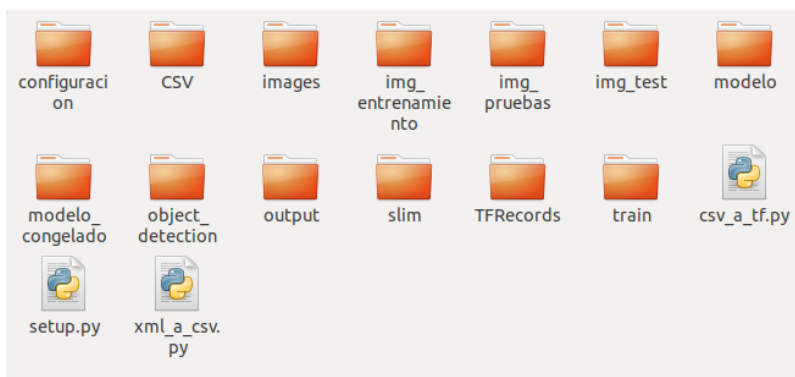


Figura 5.19: Estructura de la carpeta *deteccion-objetos*.

Ahora se procede a explicar el contenido de cada una estas carpetas:

- **configuración:** Esta carpeta contiene los archivos que indican cuantas categorías se van a usar. Para nuestro caso fueron 2, cascadas y trazas. El archivo de configuración para estas categorías tenía la siguiente forma:

```

1     item {
2     id: 1
3     name: 'traza'
4     }
5
6     item {
7     id: 2
8     name: 'cascada'
9     }
10

```

- **images:** Contiene todas las imágenes (tanto las de entrenamiento como las de prueba). También tiene los archivos en formato xml que generó el programa *labelimg*.
- **imag\_entrenamiento:** Contiene las imágenes de entrenamiento.
- **imag\_test:** Contiene las imágenes de prueba (conjunto 1).



- **imag\_prueba:** Contiene las imágenes de prueba (conjunto 2).
- **output:** En esta carpeta se encuentran las imágenes de prueba (del conjunto 2) después de pasar por la red.
- **CSV:** Utilizando el archivo *xml\_a\_csv.py*, se coloca la información de la posición del rectángulo que contienen los archivos *xml* en una hoja en formato *csv*. En esta carpeta se guarda ese archivo.
- **TFRecords:** Utilizando el archivo *csv\_a\_tf.py*, se convierte los archivos *csv* a formato *record*. Estos archivos se guardan en esta carpeta y son los que lee la red.
- **modelo:** En esta carpeta se encuentra el modelo de la red preentrenado *faster\_rcnn\_resnet101\_coco*, además de su archivo de configuración. En este archivo se especifica el número de clases a detectar, así como la ubicación de sus archivos de configuración. Además de indicar el tamaño de la imagen de entrada y el número de pasos de entrenamiento. También se indica la ubicación de los datos de entrenamiento.
- **object\_detection:** En esta carpeta se encuentran los programas para probar y entrenar de la red. El código para iniciar el entrenamiento es el siguiente:

```

1 import functools
2 import json
3 import os
4 import tensorflow as tf
5
6 from google.protobuf import text_format
7
8 from object_detection import trainer
9 from object_detection.builders import input_reader_builder
10 from object_detection.builders import model_builder
11 from object_detection.protos import input_reader_pb2
12 from object_detection.protos import model_pb2
13 from object_detection.protos import pipeline_pb2
14 from object_detection.protos import train_pb2
15
16 tf.logging.set_verbosity(tf.logging.INFO)
17
18 flags = tf.app.flags
19 flags.DEFINE_string('master', '', 'BNS name of the TensorFlow master to use.')
20 flags.DEFINE_integer('task', 0, 'task id')
21 flags.DEFINE_integer('num_clones', 1, 'Number of clones to deploy per worker.'
22 )
23 flags.DEFINE_boolean('clone_on_cpu', False,
24 'Force clones to be deployed on CPU. Note that even if
25 'set to False (allowing ops to run on gpu), some ops may
26 'still be run on the CPU if they have no GPU kernel.')
27 flags.DEFINE_integer('worker_replicas', 1, 'Number of worker+trainer
28 'replicas.')
29 flags.DEFINE_integer('ps_tasks', 0,
30 'Number of parameter server tasks. If None, does not use
31 'a parameter server.')
32 flags.DEFINE_string('train_dir', '',

```

```

32         'Directory to save the checkpoints and training summaries.
33     ')
34 flags.DEFINE_string('pipeline_config_path', '',
35                     'Path to a pipeline_pb2.TrainEvalPipelineConfig config
36                     'file. If provided, other configs are ignored')
37
38 flags.DEFINE_string('train_config_path', '',
39                     'Path to a train_pb2.TrainConfig config file.')
40 flags.DEFINE_string('input_config_path', '',
41                     'Path to an input_reader_pb2.InputReader config file.')
42 flags.DEFINE_string('model_config_path', '',
43                     'Path to a model_pb2.DetectionModel config file.')
44
45 FLAGS = flags.FLAGS
46
47
48 def get_configs_from_pipeline_file():
49     pipeline_config = pipeline_pb2.TrainEvalPipelineConfig()
50     with tf.gfile.GFile(FLAGS.pipeline_config_path, 'r') as f:
51         text_format.Merge(f.read(), pipeline_config)
52
53     model_config = pipeline_config.model
54     train_config = pipeline_config.train_config
55     input_config = pipeline_config.train_input_reader
56
57     return model_config, train_config, input_config
58
59
60 def get_configs_from_multiple_files():
61     train_config = train_pb2.TrainConfig()
62     with tf.gfile.GFile(FLAGS.train_config_path, 'r') as f:
63         text_format.Merge(f.read(), train_config)
64
65     model_config = model_pb2.DetectionModel()
66     with tf.gfile.GFile(FLAGS.model_config_path, 'r') as f:
67         text_format.Merge(f.read(), model_config)
68
69     input_config = input_reader_pb2.InputReader()
70     with tf.gfile.GFile(FLAGS.input_config_path, 'r') as f:
71         text_format.Merge(f.read(), input_config)
72
73     return model_config, train_config, input_config
74
75
76 def main(_):
77     assert FLAGS.train_dir, "'train_dir' is missing.'
78     if FLAGS.pipeline_config_path:
79         model_config, train_config, input_config = get_configs_from_pipeline_file
80         ()
81     else:
82         model_config, train_config, input_config = get_configs_from_multiple_files
83         ()
84
85     model_fn = functools.partial(
86         model_builder.build,
87         model_config=model_config,

```

```

86     is_training=True)
87
88     create_input_dict_fn = functools.partial(
89         input_reader_builder.build, input_config)
90
91     env = json.loads(os.environ.get('TF_CONFIG', '{}'))
92     cluster_data = env.get('cluster', None)
93     cluster = tf.train.ClusterSpec(cluster_data) if cluster_data else None
94     task_data = env.get('task', None) or {'type': 'master', 'index': 0}
95     task_info = type('TaskSpec', (object,)), task_data)
96
97     # Parameters for a single worker.
98     ps_tasks = 0
99     worker_replicas = 1
100    worker_job_name = 'lonely_worker'
101    task = 0
102    is_chief = True
103    master = ''
104
105    if cluster_data and 'worker' in cluster_data:
106        # Number of total worker replicas include "worker"s and the "master".
107        worker_replicas = len(cluster_data['worker']) + 1
108    if cluster_data and 'ps' in cluster_data:
109        ps_tasks = len(cluster_data['ps'])
110
111    if worker_replicas > 1 and ps_tasks < 1:
112        raise ValueError('At least 1 ps task is needed for distributed training.')
113
114    if worker_replicas >= 1 and ps_tasks > 0:
115        # Set up distributed training.
116        server = tf.train.Server(tf.train.ClusterSpec(cluster), protocol='grpc',
117                                job_name=task_info.type,
118                                task_index=task_info.index)
119        if task_info.type == 'ps':
120            server.join()
121            return
122
123        worker_job_name = '%/task:%d' % (task_info.type, task_info.index)
124        task = task_info.index
125        is_chief = (task_info.type == 'master')
126        master = server.target
127
128    trainer.train(create_input_dict_fn, model_fn, train_config, master, task,
129                 FLAGS.num_clones, worker_replicas, FLAGS.clone_on_cpu,
130                 ps_tasks,
131                 worker_job_name, is_chief, FLAGS.train_dir)
132
133    if __name__ == '__main__':
134        tf.app.run()
135

```

Por otro lado, el código para colocar las imágenes de prueba es el siguiente:

```

1 import numpy as np
2 import os
3 import six.moves.urllib as urllib
4 import sys

```

```

5 import tarfile
6 import tensorflow as tf
7 import zipfile
8 import json
9 import time
10 import glob
11 from io import StringIO
12 from PIL import Image
13 import matplotlib.pyplot as plt
14 from utils import visualization_utils as vis_util
15 from utils import label_map_util
16 from multiprocessing.dummy import Pool as ThreadPool
17 import argparse, sys
18
19 parser = argparse.ArgumentParser()
20
21 parser.add_argument('--labels', help='Directorio a label_map.pbtxt', default =
    'configuracion/label_map.pbtxt')
22 parser.add_argument('--images', help='Directorio de las imagenes a procesar',
    default = 'img_pruebas')
23 parser.add_argument('--model', help='Directorio al modelo congelado', default
    = 'modelo_congelado')
24 args = parser.parse_args()
25
26 MAX_NUMBER_OF_BOXES = 30
27 MINIMUM_CONFIDENCE = 0.99
28
29 PATH_TO_LABELS = args.labels
30 PATH_TO_TEST_IMAGES_DIR = args.images
31
32 label_map = label_map_util.load_labelmap(PATH_TO_LABELS)
33 categories = label_map_util.convert_label_map_to_categories(label_map,
    max_num_classes=sys.maxsize, use_display_name=True)
34 CATEGORY_INDEX = label_map_util.create_category_index(categories)
35
36 # Path to frozen detection graph. This is the actual model that is used for
    the object detection.
37 MODEL_NAME = args.model
38 PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
39
40 def load_image_into_numpy_array(image):
41     (im_width, im_height) = image.size
42     return np.array(image.getdata()).reshape(
43         (im_height, im_width, 3)).astype(np.uint8)
44
45 def detect_objects(image_path):
46     image = Image.open(image_path)
47     image_np = load_image_into_numpy_array(image)
48     image_np_expanded = np.expand_dims(image_np, axis=0)
49
50     (boxes, scores, classes, num) = sess.run([detection_boxes,
    detection_scores, detection_classes, num_detections], feed_dict={
    image_tensor: image_np_expanded})
51
52     vis_util.visualize_boxes_and_labels_on_image_array(
53         image_np,
54         np.squeeze(boxes),

```

```

55     np.squeeze(classes).astype(np.int32),
56     np.squeeze(scores),
57     CATEGORY_INDEX,
58     min_score_thresh=MINIMUM_CONFIDENCE,
59     use_normalized_coordinates=True,
60     line_thickness=8)
61 fig = plt.figure()
62 fig.set_size_inches(16, 9)
63 ax = plt.Axes(fig, [0., 0., 1., 1.])
64 ax.set_axis_off()
65 fig.add_axes(ax)
66
67 plt.imshow(image_np, aspect = 'auto')
68 plt.savefig('output/{}'.format(image_path), dpi = 62)
69 print(boxes)
70 plt.close(fig)
71
72 # TEST_IMAGE_PATHS = [ os.path.join(PATH_TO_TEST_IMAGES_DIR, 'image-{}.jpg'.
73     format(i)) for i in range(1, 4) ]
74 TEST_IMAGE_PATHS = glob.glob(os.path.join(PATH_TO_TEST_IMAGES_DIR, '*.jpeg'))
75 print(TEST_IMAGE_PATHS)
76
77 # Load model into memory
78 print('Loading model...')
79 detection_graph = tf.Graph()
80 with detection_graph.as_default():
81     od_graph_def = tf.GraphDef()
82     with tf.gfile.GFile(PATH_TO_CKPT, 'rb') as fid:
83         serialized_graph = fid.read()
84         od_graph_def.ParseFromString(serialized_graph)
85         tf.import_graph_def(od_graph_def, name='')
86
87 print('detecting...')
88 with detection_graph.as_default():
89     with tf.Session(graph=detection_graph) as sess:
90         image_tensor = detection_graph.get_tensor_by_name('image_tensor:0')
91         detection_boxes = detection_graph.get_tensor_by_name('detection_boxes
92 :0')
93         detection_scores = detection_graph.get_tensor_by_name('
94 detection_scores:0')
95         detection_classes = detection_graph.get_tensor_by_name('
96 detection_classes:0')
97         num_detections = detection_graph.get_tensor_by_name('num_detections:0'
98 )
99
100     for image_path in TEST_IMAGE_PATHS:
101         detect_objects(image_path)

```

- **train:** Aquí se guarda el modelo entrenado.
- **modelo \_ congelado:** Se cambia el formato del modelo generado en el entrenamiento para que se pueda utilizar para la predicción en las imágenes de prueba.
- **slim:** En esta carpeta se encuentran archivos que apuntan a variables de ambiente para que funcionen los programas del entrenamiento y de prueba.

Como se menciono anteriormente, todos los códigos se ejecutaron en la plataforma de *Google Colab*, así que dejamos los enlaces de acceso:

Red convolucional para la clasificación de imágenes:

<https://colab.research.google.com/drive/11SBHBiMdSyBf9FxRBU2TMBwZijK8uinv?usp=sharing>

Red convolucional para la detección de imágenes:

[https://colab.research.google.com/drive/1XG1EqD\\_Xdt1yqDdqy8fQc7upwLNmL21V?usp=sharing](https://colab.research.google.com/drive/1XG1EqD_Xdt1yqDdqy8fQc7upwLNmL21V?usp=sharing)

## D) Gráficas complementarias para los modelos A y B

En esta sección se muestran las gráficas de costo, la tasa de ciertos y la curva PR para los modelos A y B. En primer lugar, se muestra la gráfica del costo y la tasa de aciertos para el modelo A (ver figura 5.20). Como se puede observar en este modelo se obtiene un sobreajuste en la época 8, por tal motivo se emplea este modelo hasta dicha época. También es evidente que la tasa de acierto para los datos de prueba es muy baja, apenas supera el 55 % de imágenes clasificadas de forma correcta. Esto se debe a que este tipo de red no distingue bien las trazas de eventos reales, por eso solamente clasifica bien un número pequeño de esta categoría. Por otro lado, en la figura 5.21 se muestra las mismas gráficas pero para el modelo B. Aquí se obtiene un sobreajuste en la época 10 y al igual que el caso anterior se tiene una tasa de aciertos para los datos de prueba que apenas supera el 55 %. Este modelo también clasifica muy pocas imágenes de traza de forma correcta.

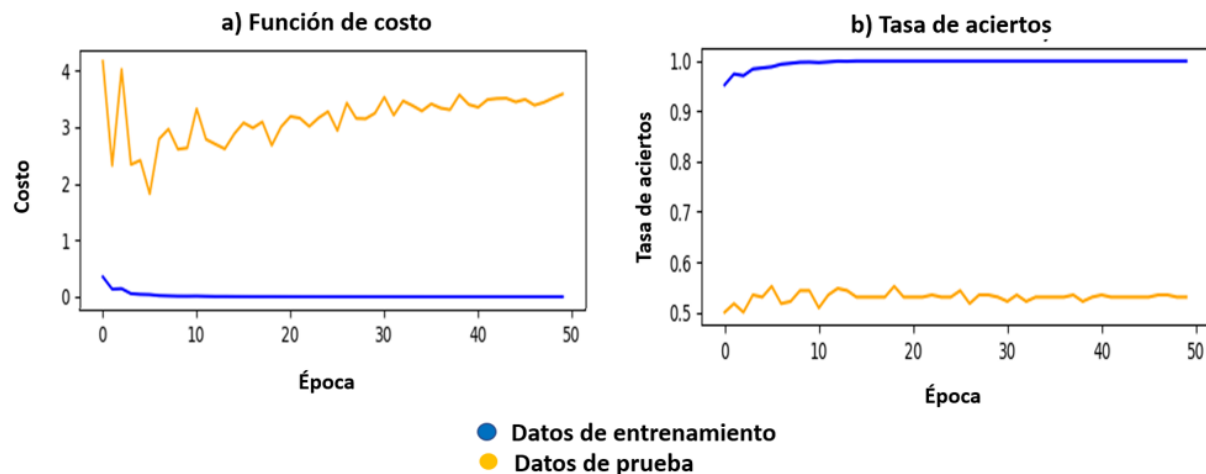


Figura 5.20: Gráfica para el costo y la tasa de aciertos para el modelo A.

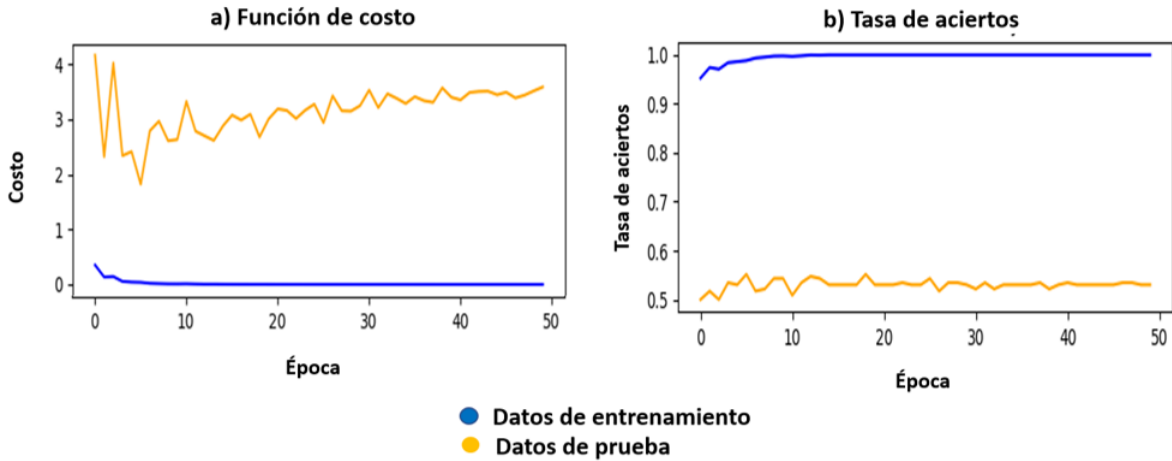


Figura 5.21: Gráfica para el costo y la tasa de aciertos para el modelo B.

Las gráficas de *precisión* contra *recall* para el modelo A y B se muestran en la figuras 5.22. En estas gráficas buscamos el valor umbral que nos de el valor mas alto para la *precisión*. Para el modelo A se obtiene la mayor *precisión* (del 87.5%) en el umbral de 0.3 y para el modelo B se obtiene en el umbral de 0.19 que tuvo una *precisión* del (100%). Estos fueron los valores reportados en la tabla 5.1. En la figura 5.22 para el modelo A se puede observar que hay un retroceso en el

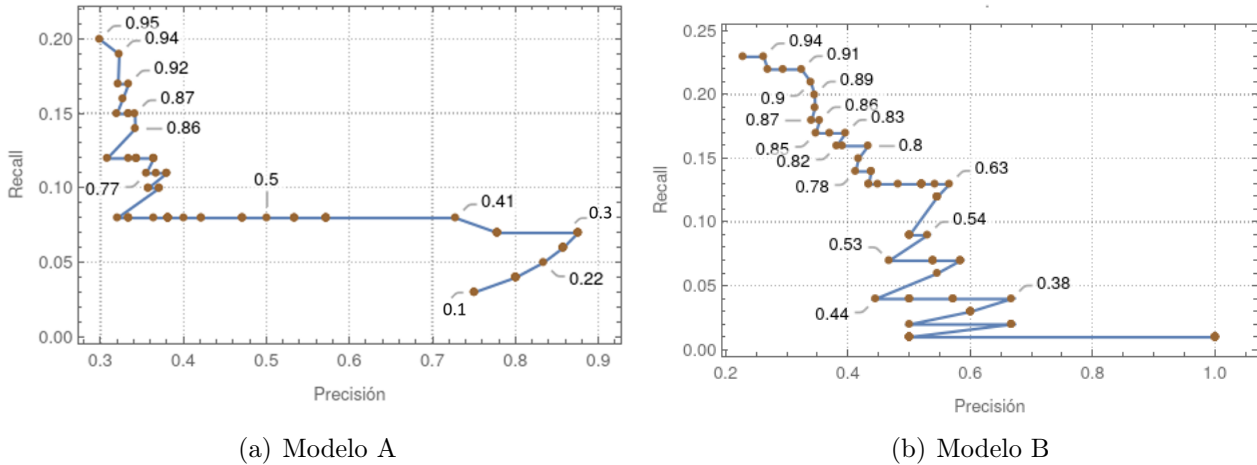


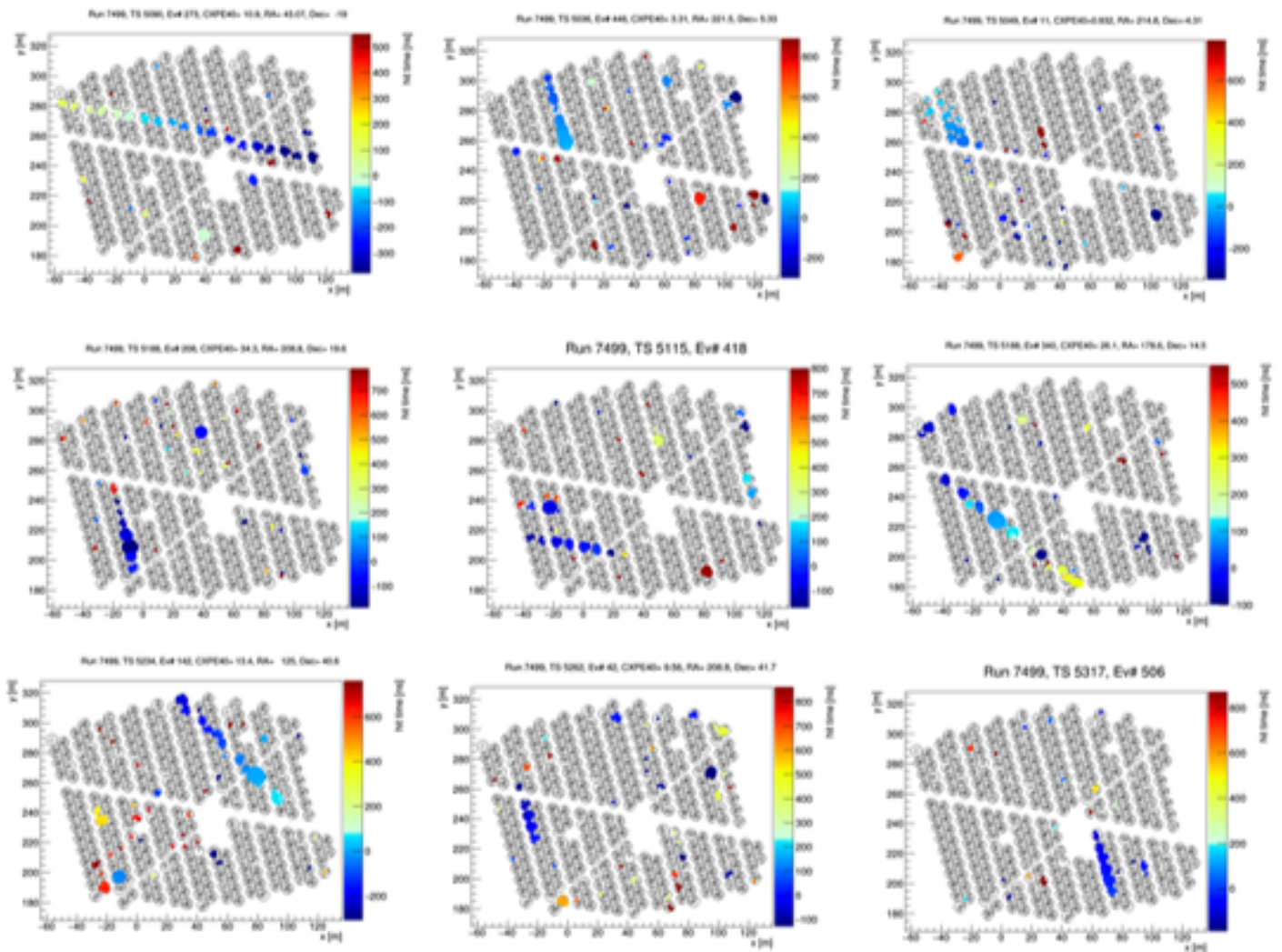
Figura 5.22: Curva *precisión-recall*

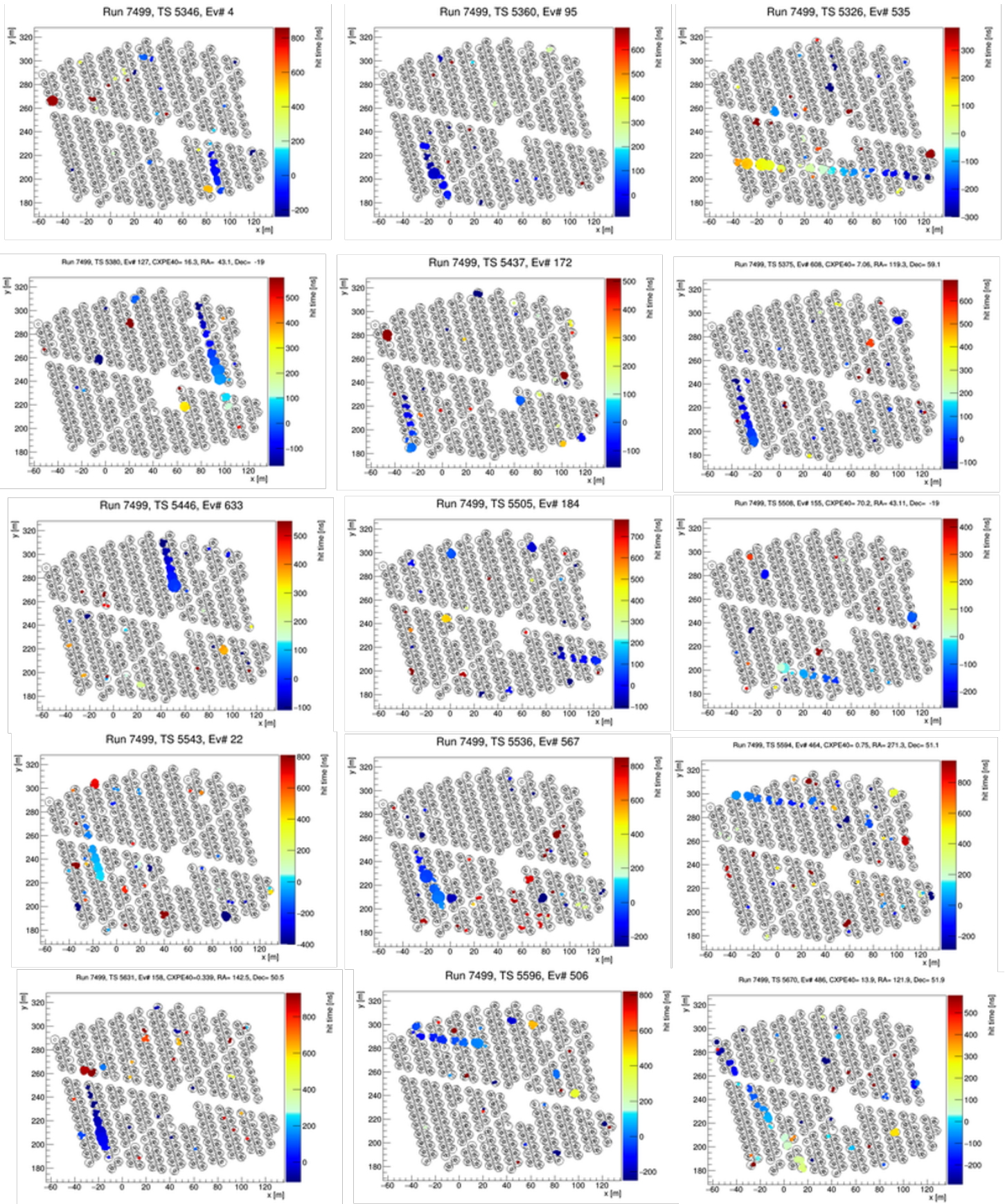
valor de la *precisión* para el caso en donde el valor umbral es menor a 0.3, en ese caso se obtuvieron 7 verdaderos positivos y un falso positivo. Lo que sucede en esta situación fue que al disminuir el valor umbral la cantidad de verdaderos positivos fue disminuyendo mientras que la cantidad de falsos positivos se mantenía igual hasta llegar al umbral más bajo (0.1), en este umbral se tenía 3 verdaderos positivos y un falso negativo dando una *precisión* de 0.75 menor a la que se obtuvo en el umbral 0.3.

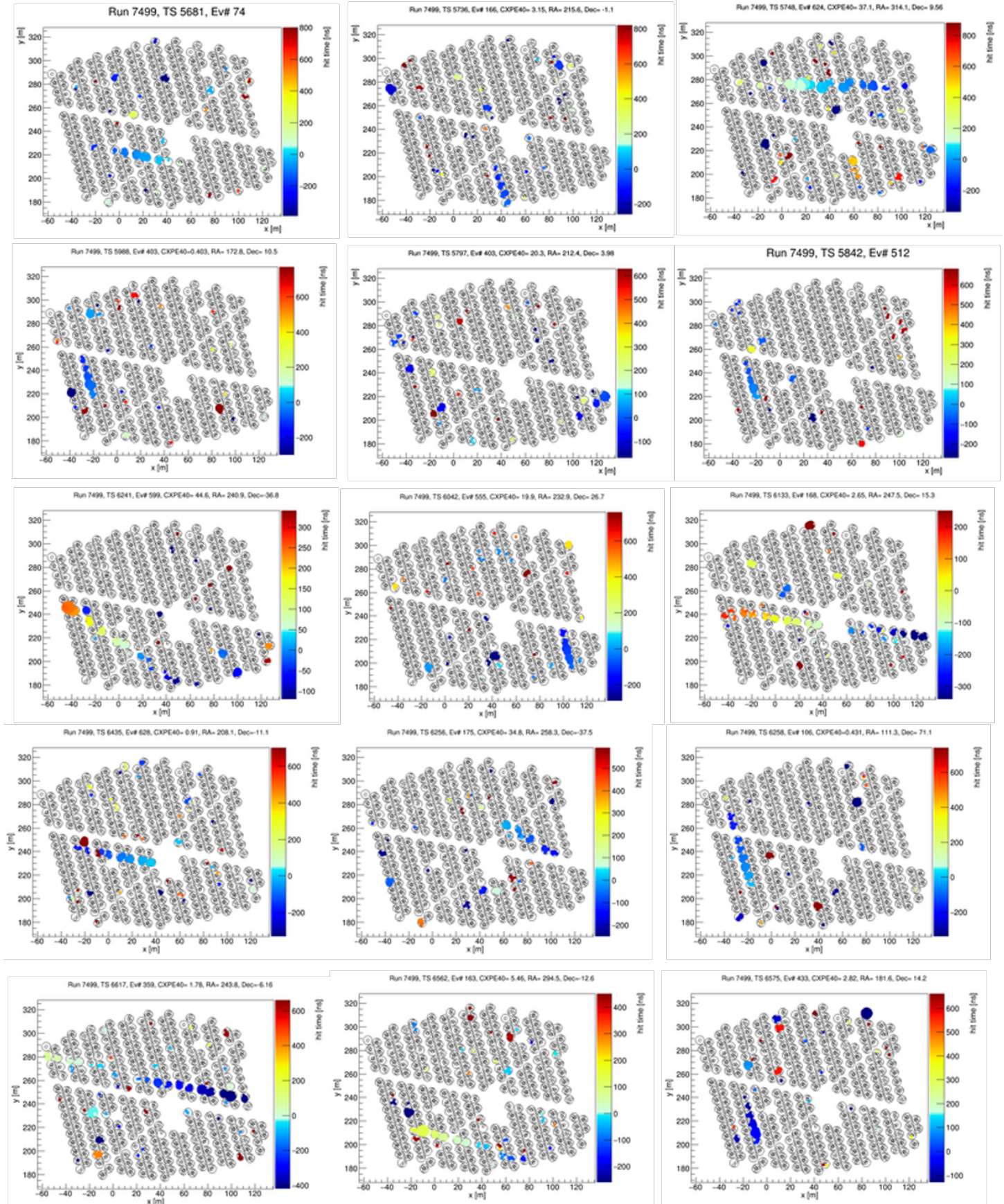


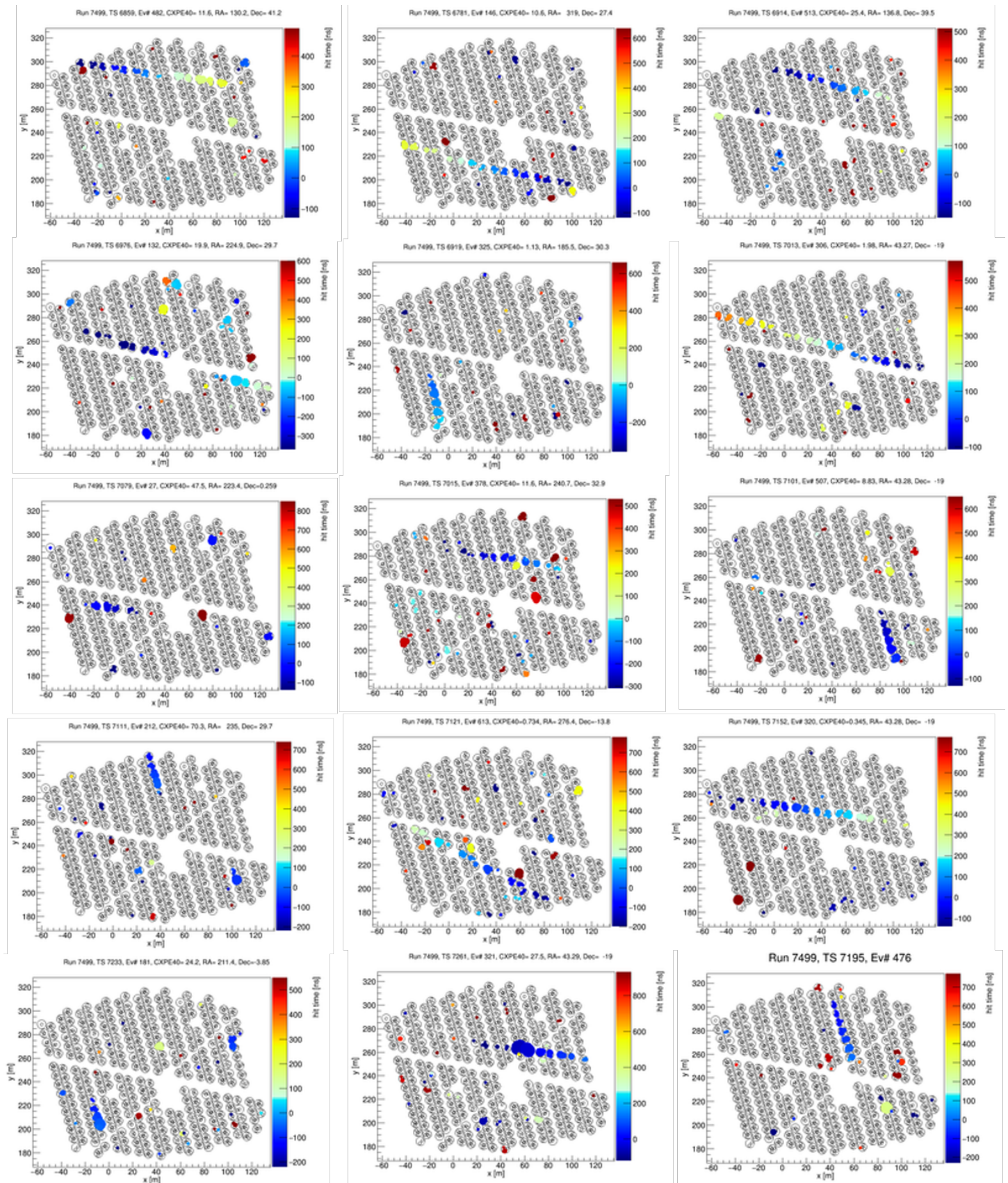
## E) Trazas identificadas por la red

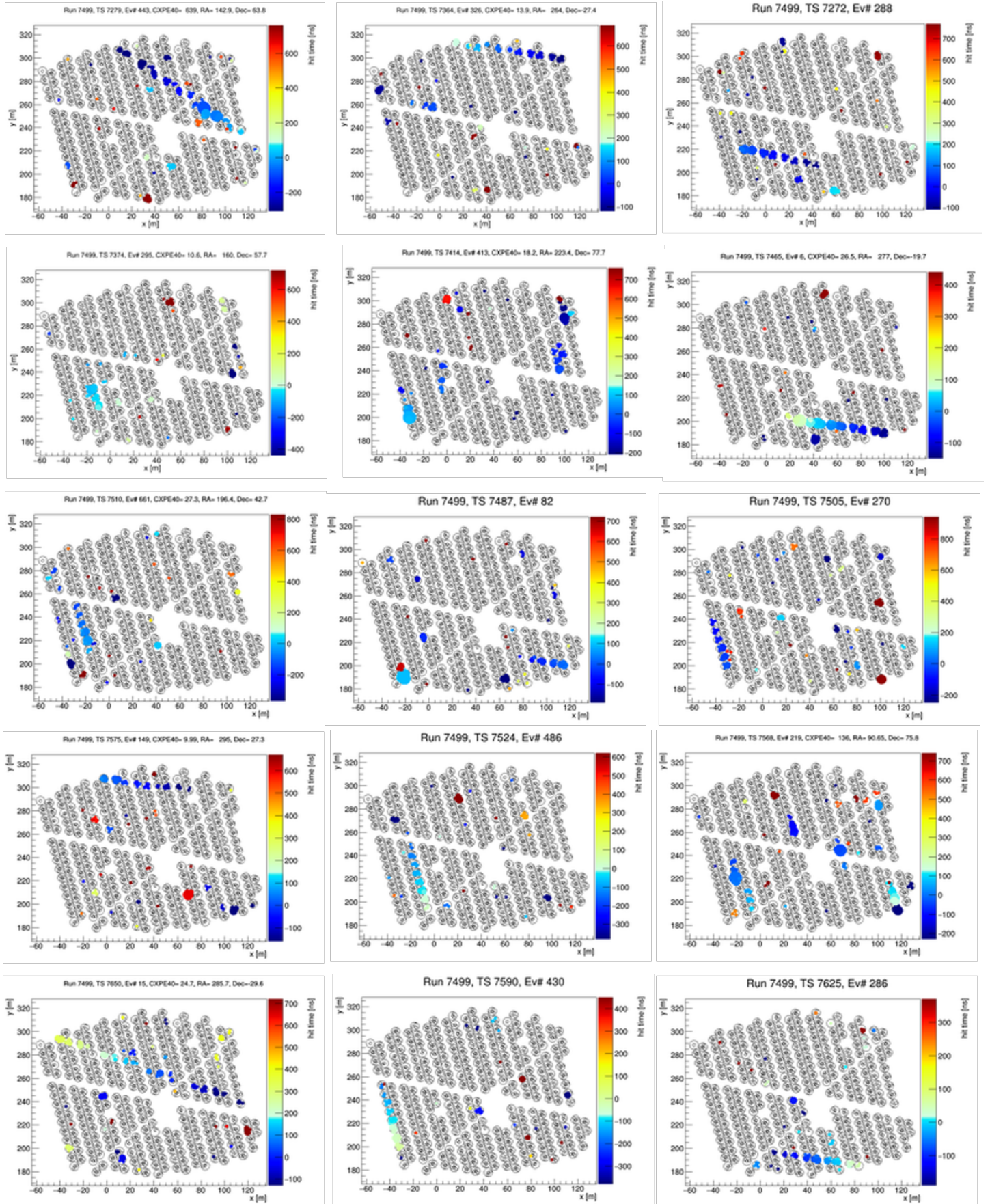
En esta sección se muestran todos los eventos que la red neuronal enfocada en la clasificación de imágenes identifico como una traza horizontal. Esto son los resultados después de analizar los eventos del segundo subrun del run 7499.

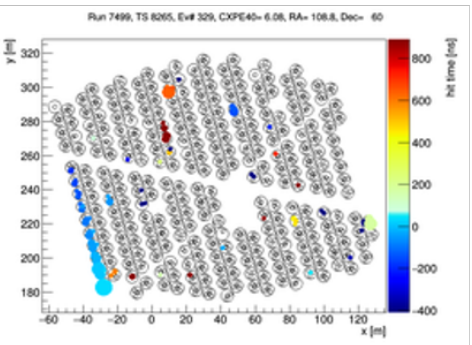
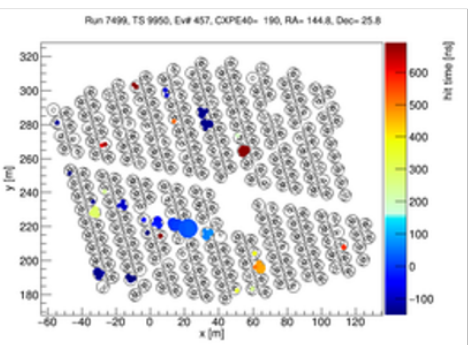
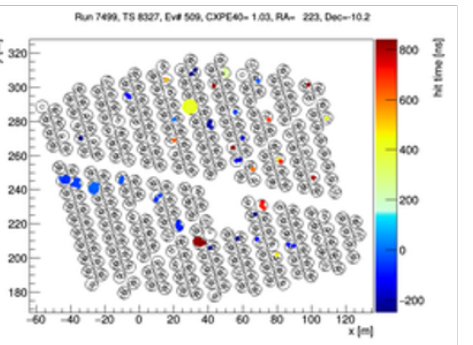
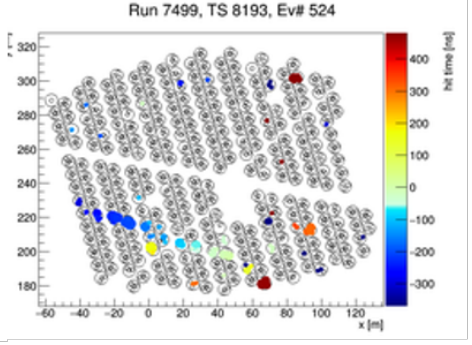
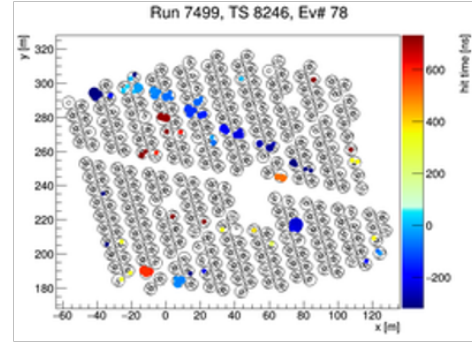
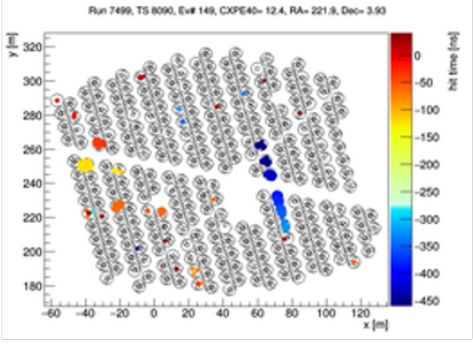
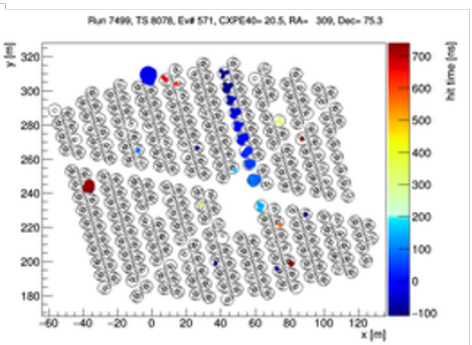
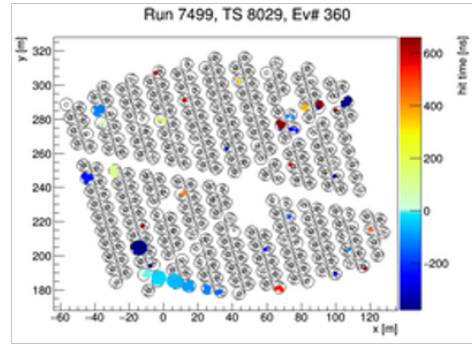
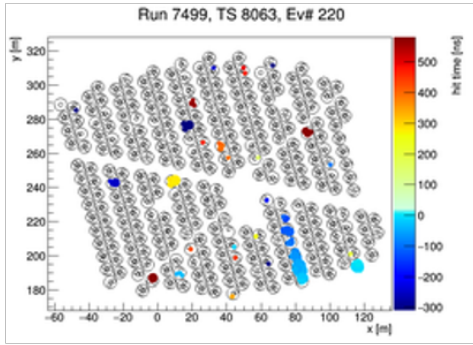
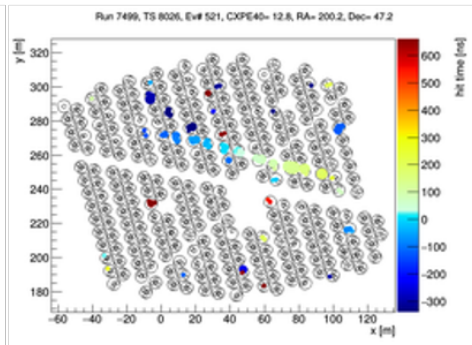
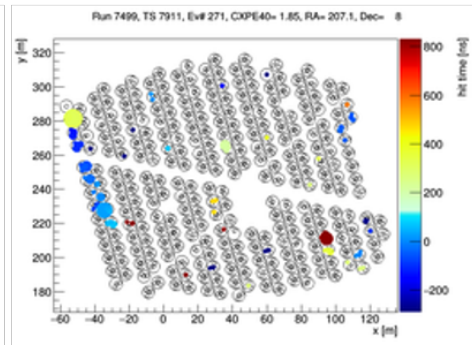
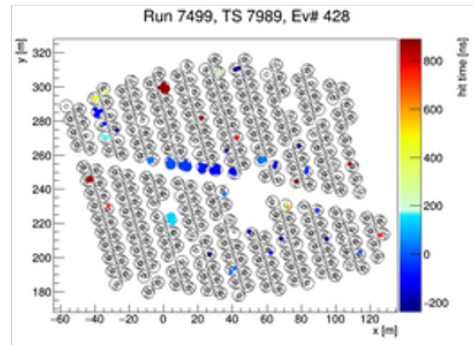
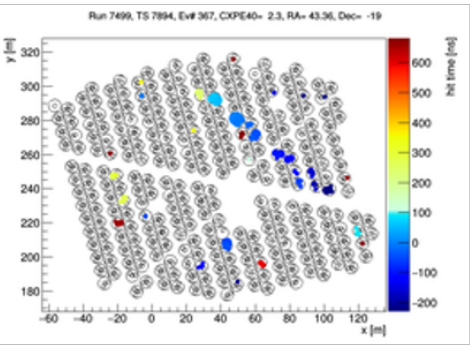
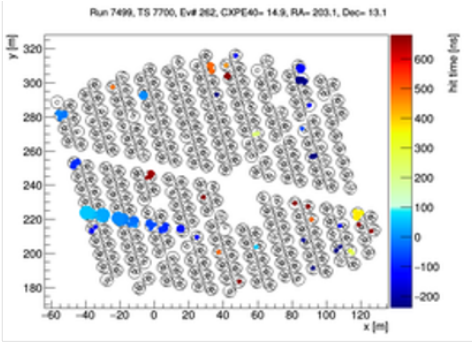
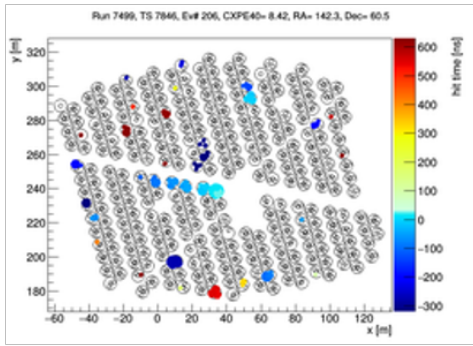


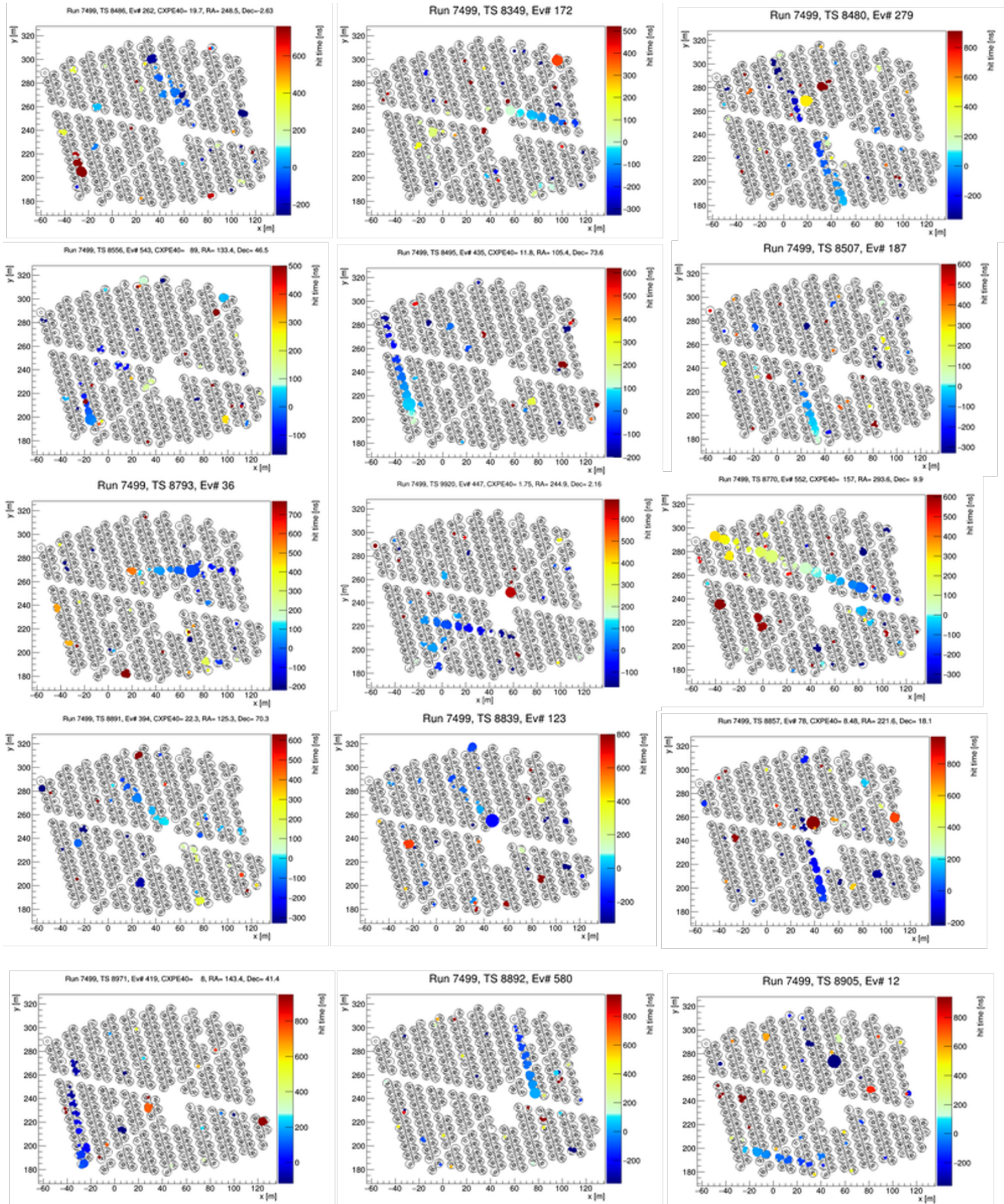


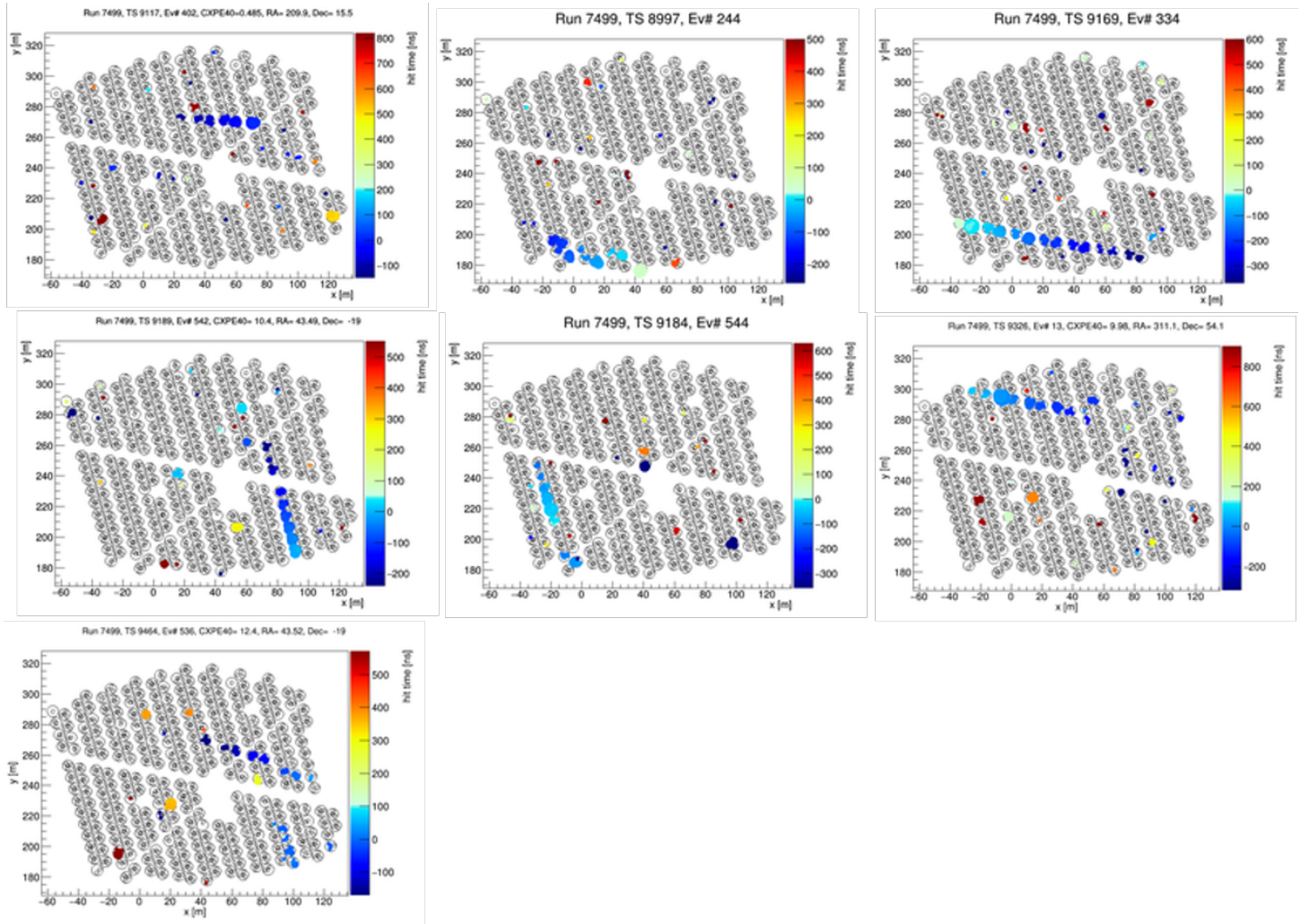














# Bibliografía

- [1] <https://www.hawc-observatory.org/>. Recuperado el 22/10/2020.
- [2] LEÓN V. HERMES ET AL. *Capability of the HAWC Gamma-Ray Observatory for the Indirect Detection of Ultrahigh-Energy Neutrinos*. En: Hindawi volumen 2017, pp.34-47.
- [3] LEÓN V. HERMES. *Observing Earth-skimming neutrinos at HAWC*. Documento interno de HAWC 2019.
- [4] CIRKEL-BARTELT V. *History of astroparticle physics and its components*. Living Rev. Relativ. 2008;11:2. doi:10.12942/lrr-2008-2.
- [5] C. GRUPEN. (2005). *Astroparticle Physics* Springer-Verlag Berlin Heidelberg.
- [6] W. HOFMANN(2012). *Astrophysics with H.E.S.S.* Recuperado el: 7-septiembre-2020, de H.E.S.S. sitio web: <https://www.mpi-hd.mpg.de/hfm/HESS/pages/about/physics/>
- [7] VIKAS JOSHI. (2019). *Reconstruction and Analysis of Highest Energy Gamma-Rays and its Application to Pulsar Wind Nebulae*.(Tesis de doctorado).Ruperto-Carola University of Heidelberg.pp.1-40.
- [8] FRANK G. (2012). *Instruments and Methods for the Radio Detection of High Energy Cosmic Rays* Springer.Karlsruhe, Alemania. pp.vii-10.
- [9] JOHN WALKER. (1994). THE OH-MY-GOD PARTICLE. Recuperado el: 7-septiembre-2020, de sitio web: <https://www.fourmilab.ch/documents/OhMyGodParticle/>
- [10] WATSON A. A.. (2013). *High-Energy Cosmic Rays and the Greisen-Zatsepin-Kuzmin Effect*. En *Reports on Progress in Physics*. Vol 77. <https://arxiv.org/abs/1310.0325>.
- [11] JOHN BAEZ. (2012). OPEN QUESTIONS IN PHYSICS. Recuperado el: 7-septiembre-2020, de sitio web: [https://math.ucr.edu/home/baez/physics/General/open\\_questions.html#cosmology](https://math.ucr.edu/home/baez/physics/General/open_questions.html#cosmology)
- [12] (2018). HAWC AND ICECUBE JOIN EFFORTS TO ANALYZE THE ANISOTROPY OF COSMIC RAYS. Recuperado el: 7-septiembre-2020, de HAWC sitio web: <https://www.hawc-observatory.org/news/>
- [13] CACCIANIGA L. ET AL.(2019) *Anisotropies of the highest energy cosmic-ray events recorded by the Pierre Auger Observatory in 15 years of operation*. En PoS, ICRC2019. <https://inspirehep.net/literature/1819408>
- [14] RAFAEL ALVES BATISTA.(2019) *Open Questions in Cosmic-Ray Research at Ultrahigh Energies*. En: astro-ph.HE arXiv:1903.06714v1, pp.1-20.

- [15] M. G. AARTSEN ET AL. IceCube Collab. Nature 551 (2017) 596. <https://www.nature.com/articles/nature24459>
- [16] M. BUSTAMANTE AND A. CONNOLLY Phys. Rev. Lett. 122 (2019) 041101. <https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.122.041101>
- [17] GUILLAUME DUBUS. (2015). Gamma-ray emission from binaries in context. Comptes Rendus Physique, Volume 16, Issues 6–7, Pages 661-673. [sciencedirect.com/science/article/pii/S1631070515001498](https://www.sciencedirect.com/science/article/pii/S1631070515001498)
- [18] CHARLES D. D. Y BERRIE G (2016). Active galactic nuclei at  $\gamma$  ray energies. arXiv:1602.06592 [astro-ph.HE]
- [19] MEHR UN NISA. (2019). Constraining TeV-scale Astrophysical Foregrounds for Dark Matter Searches with HAWC. (Tesis de doctorado). University of Rochester. pp.1-45.
- [20] HART, R. (2009). *Electromagnetic Cascade*. Recuperado el 24-10-2020, de Hardhack Sitio web: [http://hardhack.org.au/electromagnetic\\_cascades](http://hardhack.org.au/electromagnetic_cascades)
- [21] PETER K. F.. (2010). Muons in gamma showers. En NASA. Goddard Space Flight Center 19th Intern. Cosmic Ray Conf., Vol. 7 p 219-222 (SEE N85-35968 24-93).
- [22] STANEV, T. ET AL. (1986). Extensive Air Showers. Springer, Alemania. pp. 15-25
- [23] MINAYA, I. (2011). Composición de la radiación cósmica de ultra-alta energía a partir de los datos del Observatorio Pierre Auger. (Tesis de Maestría). Facultad de Ciencias Físicas de la Universidad Complutense de Madrid, España.
- [24] COLÍN F. PAULINA (2018). La sombra del sol observada por el detector de rayos gamma HAWC (Tesis de licenciatura). Universidad Nacional Autónoma de México .pp.1-40.
- [25] GRIEDER, P. (2010). Introduction, Facts and Phenomenology. En *Extensive Air Shower*. (pp.3-24). Berlín: Springer.
- [26] DAVIES, B. (2016). *Cosmic Rays*. Recuperado el 26-10-2020, de Antarctic Glaciers Sitio web: <http://www.antarcticglaciers.org/glacial-geology/dating-glacial-sediments-2/cosmic-rays>
- [27] GIUNTI C. Y CHUNG W. K. (2007). Fundamentals of Neutrino Physics and Astrophysics. Oxford Scholarship Online. pp. 361.
- [28] ÁLVARES H. PAOLA (2019). Estudio de los neutrinos de altas energías reportados por el observatorio IceCube en la dirección de las burbujas de Fermi (Tesis de Maestría). Universidad Nacional Autónoma de México .pp.1-35.
- [29] PAOLA Á. H. (2019). Estudio de los neutrinos de altas energías reportados por el observatorio ICECUBE en la dirección de las burbujas de Fermi. (Tesis de Maestría). México, UNAM. pp.1-35.
- [30] VLADIMIR K.. (2015). Neutrino Astrophysics with the ANTARES Telescope. (Tesis de doctorado). Italia, Università degli Studi di Genova. pp.1-40.
- [31] T. T. C. 2013, [t2k-experiment.org/neutrinos/neutrino-detection](http://t2k-experiment.org/neutrinos/neutrino-detection).

- [32] GABY R. Y ROLF B. (1993). Neutrino electron scattering. Publicado en Mod.Phys.Lett.A 8 (1993) 1067-1088, Mod. Phys. Lett. A8 (1993) 1067-1088 and CERN Geneva - CERN-PPE-93-065.
- [33] CHRISTIAN W. F. Y HERWIG SCHOPPER. (2011). Particle Physics Reference Volume 2: Detectors for Particles and Radiatio. Springer open. Geneva, Suiza .pp.45-55.
- [34] MÁXIMO C. Y MARIANO S. (2014). PMT y Centelladores. (presentación), Argentina. Recuperado el 19/04/2021 en: <http://materias.df.uba.ar/14aa2017c2/files/2014/04/PMT-y-centelladores.pdf>
- [35] BRITANNICA, T. EDITORS OF ENCYCLOPAEDIA. Nuclear photographic emulsion. Encyclopedi Britannica. Recuperado el 19/04/2021 en: <https://www.britannica.com/science/nuclear-photographic-emulsion>
- [36] DONATO D. F. (2008). Nuclear emulsions in the OPERA experiment. Invited talk at the 24th International Conference on Nuclear Tracks in Solids, Bologna. <https://arxiv.org/pdf/0812.0451.pdf>
- [37] VINCENT MARANDON. HAWC – A very high energy gamma-ray observatory. Pagina web: <https://www.mpi-hd.mpg.de/mpi/de/forschung/abteilungen-und-gruppen/nichtthermische-astrophysik/projekte/hawc> . Recuperado el 22/10/2020.
- [38] V.MARANDON ET AL. *Latest news from the HAWC outrigger array*. ICRC 2019 arXiv:1908.07634v1 [astro-ph.IM] 20 Aug 2019.
- [39] JELLEY, J. (1958). Introduction, En *Cerenkov Radiation and its applications*.(pp.3-6).Gran Bretaña: United Kingdom Atomic Energy Authority.
- [40] GROOM D. (2014). Atomic and nuclear properties of water (liquid). [https://pdg.lbl.gov/2015/AtomicNuclearProperties/HTML/water\\_liquid.html](https://pdg.lbl.gov/2015/AtomicNuclearProperties/HTML/water_liquid.html) Revisado el 20 de abril 2021.
- [41] JOSHUA WOOD. (2016). An All-Sky Search for Bursts of Very High Energy Gamma Rays with HAWC (Tesis de Doctorado). University of Maryland.pp.53-106.
- [42] STEPHENS MARINELL S. (2019). Constraints on Lorentz-invariance violation with the HAWC observatory (Tesis de Doctorado). Michigan State University. pp.22-31.
- [43] AVILA ROJAS D. O. (2016). Sensibilidad del observatorio HAWC a rayos gamma de alta energía. (Tesis de Licenciatura). Universidad Nacional Autónoma de México .pp.17-20.
- [44] ZHIXIANG REN (2018). Improve the Low Energy Sensitivity of the HAWC Observatory. (Tesis de Doctorado). University of New Mexico .pp.19-29.
- [45] ROSALES, G. (2017). Caracterización de la modulación azimutal del flujo de cascadas atmosféricas con el observatorio HAWC . (Tesis de licenciatura). UNAM. .pp.27-31.
- [46] ZIGFRIED H. A. (2017). Cosmic Ray Observations at the TeV Scale with the HAWC Observatory . (Tesis de Doctorado). Wisconsin Madison. .pp.27-31.
- [47] KELLY M. (2018). A survey of the highest-energy astrophysical sources with the HAWC observatory . (Tesis de Doctorado). The Eberly College of Science. pp.43-65.

- [48] THE HAWC COLABORATION (2019). The Crab Nebula spectrum extending up to 100 TeV as seen by the HAWC Gama-Ray Observatory. *The Astrophysical Journal*, Volume 881, Número 2.
- [49] [https://private.hawc-observatory.org/wiki/index.php/Install\\_HAWC\\_software](https://private.hawc-observatory.org/wiki/index.php/Install_HAWC_software) . Pagina interna de HAWC. Recuperado el 8/11/2020.
- [50] HERNÁNDEZ, M. (2017). Caracterización de la interacción de rayos cósmicos con volcanes por medio de una simulación Monte Carlo con GEANT4. (Tesis de licenciatura). UNAM, México. pp.26-28.
- [51] Google Earth (s.f.). [Observatorio de Rayos Gama HAWC, Puebla, México]. Revisado 25 de abril 2021. <https://www.google.com/intl/es-419/earth/>
- [52] EPIFANOV D. (2014). Tau lifetime and decays. arXiv:1407.7196 [hep-ex].
- [53] M. D. KISTLER Y R. LAHA (2016). Multi-PeV Signals from a New Astrophysical Neutrino Flux Beyond the Glashow Resonance. <https://arxiv.org/abs/1605.08781>
- [54] LEÓN V. HERMES. *Finding muon tracks in (10 seconds) Raw data*. Documento interno de HAWC 2016.
- [55] LEÓN V. HERMES ET AL. *Prospects of Earth-skimming neutrino detection with HAWC*. arXiv:1908.07622v1 [physics.ins-det] 20 Aug 2019.
- [56] SANDERSON, G. [3Blue1Brown]. (20 de diciembre 2020). But what \*is\* a Neural Network? | Chapter 1, deep learning. [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=aircAruvNkK&t=465s>
- [57] OLGA D. (2017). 10 Applications of Artificial Neural Networks in Natural Language Processing. Recuperado el 25/04/2021 de <https://medium.com/@datamonsters/artificial-neural-networks-in-natural-language-processing-bcf62aa9151a>: :text=Since %20artificial %20neural %20networks %20allow, %2C %20anomaly %
- [58] MICHAEL A. NIELSEN, *Neuronal Networks and Deep Learning*, Determination Press, 2015.
- [59] LARA, F., *Fundamento de redes neuronales artificiales*, Centro de instrumentos, UNAM.
- [60] PALMER P. ET AL. *¿Qué son las redes neuronales artificiales? Aplicaciones realizadas en el ámbito de las adicciones*. En: Adicciones 1999 Vol. 11 Num. 3, pp.243-255.
- [61] HEATON, J. *The Number of Hidden Layers*. Recuperado el 20-12-2020, de Heaton Research Sitio <http://www.heatonresearch.com/2017/06/01/hidden-layers.html>
- [62] SANDERSON, G. [3Blue1Brown]. (20 de diciembre 2020). Gradient descent, how neural networks learn | Chapter 2, deep learning. [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=IHZwWFHWa-w&t=262s>
- [63] SANDERSON, G. [3Blue1Brown]. (20 de diciembre 2020). What is backpropagation really doing? | Chapter 3, deep learning. [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=Ilg3gGewQ5U>

- [64] SANDERSON, G. [3Blue1Brown]. (20 de diciembre 2020). Backpropagation calculus | Appendix to deep learning chapter 3.[Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=tIeHLnjs5U8&t=511s>
- [65] <https://www.juanbarrios.com/redes-neurales-convolucionales/> consultado 20 de diciembre 2020.
- [66] CALVO, D.. (2017). Red neuronal convolucional. Recuperado de <https://www.diegocalvo.es/red-neuronal-convolucional/>
- [67] [AMP Tech]. (20 de diciembre 2020). Redes neuronales convolucionales CNN (Clasificación de imágenes).[Archivo de video]. Recuperado de [https://www.youtube.com/watch?v=ns2L2T6wvAYab\\_channel=AMPTech](https://www.youtube.com/watch?v=ns2L2T6wvAYab_channel=AMPTech)
- [68] [Deeplizard]. (20 de diciembre 2020). Convolutional Neural Networks (CNNs) explained.[Archivo de video]. Recuperado de [https://www.youtube.com/watch?v=YRhxdVk\\_sIsab\\_channel=deeplizard](https://www.youtube.com/watch?v=YRhxdVk_sIsab_channel=deeplizard)
- [69] [Deeplizard]. (20 de diciembre 2020). Convolutional Neural Networks (CNNs) explained.[Archivo de video]. Recuperado de [https://www.youtube.com/watch?v=YRhxdVk\\_sIsab\\_channel=deeplizard](https://www.youtube.com/watch?v=YRhxdVk_sIsab_channel=deeplizard)
- [70] SANTAMARIA L.. ET AL. *Reconocimiento de genes en secuencias de ADN por medio de imágenes*. XVIII Conferencia de la Asociación Española para la Inteligencia Artificial.
- [71] CONTRERAS, O. (2018). Desarrollo de una red neuronal convolucional para el procesamiento de imágenes placentarias. (Tesis de licenciatura). UNAM. .pp.17-30.
- [72] <https://www.aprendemachinelearning.com/modelos-de-deteccion-de-objetos/>. consultado 20 de diciembre 2020.
- [73] <https://www.tensorflow.org/> . Consultado 25 de abril 2021.
- [74] <https://cloud.google.com/solutions/creating-object-detection-application-tensorflow?hl=es-419> . Consultado 20 de diciembre 2020.
- [75] PARTHASARATHY D. *A Brief History of CNNs in Image Segmentation: From R-CNN to Mask R-CNN* .Recuperado el 20-12-2020, Sitio <https://blog.athelas.com/a-brief-history-of-cnns-in-image-segmentation-from-r-cnn-to-mask-r-cnn-34ea83205de4>
- [76] LÓPEZ, P. (2019). Detección de actividades anómalas en espacios públicos mediante redes neuronales profundas. (Tesis de Maestría). Universidad de Alcalá. .pp.21-48.
- [77] ARRIOLA, I. (2018).Detección de objetos basada en Deep Learning y aplicada a vehículos autónomos. (Tesis de Maestría). KISA/ICSI. .pp.25-29.
- [78] [http://personal.cimat.mx:8181/mrivera/cursos/aprendizaje\\_profundo/preentrenadas/preentrenadas.html](http://personal.cimat.mx:8181/mrivera/cursos/aprendizaje_profundo/preentrenadas/preentrenadas.html). Consultado 20 de diciembre 2020.
- [79] RUIZ, J.. Presentación en línea. <https://www.cs.us.es/cursos/rac-2018/temas/tema-07.pdf> . Consultado 21 de diciembre 2020.

- [80] [AMP Tech]. (20 de diciembre 2020). Detección de objetos con tensorflow (Parte 1).[Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=SJRP0IRfPj0t=1s>
- [81] [AMP Tech]. (20 de diciembre 2020). Detección de objetos con tensorflow (Parte 2).[Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=EKe05rMG-Wwt=4s>
- [82] Sitio oficial: <https://colab.research.google.com/notebooks/intro.ipynb#recent=true>
- [83] KAPKAR N. *TensorFlow 2 Object Detection API With Google Colab* .Recuperado el 21-12-2020, Sitio <https://medium.com/swlh/tensorflow-2-object-detection-api-with-google-colab-b2af171e81cc>
- [84] BROWNLEE J. *How to Classify Photos of Dogs and Cats (with 97% accuracy)*.Recuperado el 21-12-2020, Sitio <https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-photos-of-dogs-and-cats/>