

TESIS CARRERA DE MAESTRÍA EN FÍSICA

**IDENTIFICACIÓN DE INTERACCIONES DE
PARTÍCULAS EN SENSORES DE IMÁGENES DE
SILICIO**

D. F. Balmaceda
Maestría en Física

Dr. X. Bertou
Director

Dr. M. Sofo Haro
Co-director

Miembros del Jurado

Dr. M. Gómez Berisso (Instituto Balseiro, CNEA),
Dr. J. Lipovetzky (Instituto Balseiro, CNEA),
Ing. R. Cibils (INVAP)

Noviembre de 2019

Laboratorio Detección de Partículas y Radiación

Instituto Balseiro
Universidad Nacional de Cuyo
Comisión Nacional de Energía Atómica
Argentina

A mi familia.
A mi novia.
A mis amigos.
A todo el universo.

Índice de símbolos

ADC	Convertor analógico-digital
ANDES	Agua Negra Deep Experiment Site
APS	Sensor de pixeles activos
AP	Punto de acceso
BLC	Calibración del nivel de negro
CCD	Dispositivo de carga acoplada
CMOS	Semiconductor de metal-oxido complementario
CNN	Red neuronal convolucional
CONNIE	Coherent Neutrino Nucleus Interaction Experiment
CSI	Interfaz serie para cámaras
CSS	Hoja de estilo en cascada
DAMIC	Dark Matter in CCD
DHCP	Protocolo de configuración dinámica de host
DPC	Corrección de pixeles defectuosos
FPS	Cuadros por segundos
HDMI	Interfaz multimedia de alta definición
HTML	Lenguaje de marcas de hipertexto
I²C	Circuito inter-integrado
JS	Javascript
LBS	Bit menos significativo
LED	Diodo emisor de Luz
MCC	Metallurgical Corporation of China
MIPI	Interfaz de procesador de la industria móvil
ML	Machine learning

NAT	Traducción de direcciones de redes
OMS	Organización Mundial de la Salud
PWA	Aplicación web progresiva
QPU	Unidad de procesamiento cuádruple
Radmonpi	Monitor de radiación para Raspberry Pi
ReLU	Unidad lineal rectificadora
SBC	Computadora de placa simple
SD	Secure Digital
SSH	Secure Shell
USB	Bus universal en serie

Índice de contenidos

Índice de símbolos	v
Índice de contenidos	vii
Índice de figuras	xi
Índice de tablas	xiii
Resumen	xv
Abstract	xvii
1. Introducción	1
1.1. Sensores CMOS	2
1.1.1. Principio de funcionamiento	2
1.1.2. Construcción de la imagen	3
1.1.3. Filtro de Bayer	4
1.1.4. Píxeles dañados	4
1.2. Sensores CCD	5
2. Desarrollo de Radmonpi, monitor de radiación de partículas	7
2.1. Hardware utilizado	7
2.1.1. Sensor utilizado	7
2.1.2. Raspberry Pi, SBC utilizado	8
2.2. Librerías previas	9
2.2.1. Raspiraw	9
2.2.2. Primera versión de radmonpi[1]	9
2.2.3. Interfaz web (detecGUI)	10
2.3. Nuevo Radmonpi (Radmonpi++)	10
2.3.1. Configuración inicial	10
2.3.2. Utilización del programa con interfaz en consola (CLI)	11
2.3.3. Generación de una máscara de píxeles defectuosos	12

2.3.4.	Interfaz web en <i>Bokeh</i>	13
2.3.5.	Interfaz web en <i>React</i>	14
3.	Medición de Radón	15
3.1.	Introducción y motivación	15
3.2.	Configuración experimental	16
3.2.1.	Máscara de pixeles defectuosos utilizada	17
3.3.	Primeras mediciones	18
3.3.1.	Número de cuentas como función del tiempo	18
3.3.2.	Distribución espacial de los eventos	19
3.3.3.	Forma de los eventos	20
3.4.	Medición de medio mes	21
3.4.1.	Numéro de cuentas como función del tiempo	21
3.4.2.	Distribución espacial de los eventos	22
3.5.	Conclusiones	22
3.5.1.	Primera medición	22
3.5.2.	Mediciones de medio mes	23
4.	Medición de Pb con sensores CCD	25
4.1.	Introducción y Motivación	25
4.2.	Configuración Experimental	26
4.3.	Resultados	27
4.3.1.	Primera imagen del sensor	27
4.3.2.	Medición con fuente	28
4.3.3.	Posición de los eventos de Pb	29
4.3.4.	Medición sin fuente	29
4.4.	Mediciones con sensores CMOS	31
4.4.1.	Configuración Experimental	31
4.4.2.	Medición sin Pb	31
4.4.3.	Medición con Pb	32
4.5.	Conclusiones	32
5.	Reconocimiento de eventos con técnicas de Machine Learning	33
5.1.	Introducción y motivación	33
5.1.1.	Usos y motivación	33
5.1.2.	Principios de Inteligencia Artificial	34
5.1.3.	El perceptrón	35
5.1.4.	Modelo multicapa	36
5.1.5.	Redes convolucionales	37
5.1.6.	Algoritmo de <i>backpropagation</i> y métricas a definir	37

5.2. Simulación de los datos de entrada, eventos y ruidos	38
5.2.1. Simulación de los eventos	38
5.2.2. Simulación del ruido y función de validación	39
5.3. Descripción de la red neuronal empleada	40
5.4. Resultados	40
5.4.1. Distribución energética de la simulación para eventos de $27 e^-$	40
5.4.2. Entrenamiento de la red para eventos de $27 e^-$	41
5.4.3. Entrenamiento de la red para distintos valores de E	42
5.5. Conclusiones	44
6. Conclusiones, palabras finales y trabajos futuros	45
6.1. Software empleado	45
6.2. Medición de Rn	46
6.3. Medición de Pb en Agua	46
6.4. Técnicas de ML aplicadas al análisis de eventos	46
A. Utilización de las QPU	49
A.1. Descripción y motivación	49
A.2. Simulación	49
A.3. Resultados y Conclusión	50
B. Mediciones del flujo de muones en Sierra Grande	51
B.1. Introducción y motivación	51
B.2. Equipo de medición y algoritmo empleado	51
B.3. Interfaz web desarrollada	52
B.4. Medición de la actividad de Rn con <i>Wave</i>	52
Bibliografía	55
Agradecimientos	59

Índice de figuras

1.1. Un celda de tres transistores (3T) típica para un sensor CMOS-APS . . .	3
1.2. Fotografía del escenario de prueba para comprobar la correcta obtención de los datos <i>raw</i> mediante el algoritmo implementado.	4
1.3. Esquema de un filtro de Bayer típico y sus efectos en fotografía.	5
2.1. Raspberry 3B+ Utilizada	9
2.2. Interfaz web de Radmonpi desarrollada en <i>Bokeh</i>	13
3.1. Cadena de decaimiento del ^{222}Rn	15
3.2. Dispositivo experimental implementado para la medición de ^{222}Rn	17
3.3. Histograma del promedio de las imágenes para determinar el nivel de fondo y la máscara de pixeles defectuosos	18
3.4. Nivel de Rn y cantidad de partículas alfa detectadas como función del tiempo.	19
3.5. Distribución espacial de los eventos correspondientes a partículas alfas detectadas sobre el sensor.	19
3.6. Dos eventos de partículas alfas detectados con formas distintas.	20
3.7. Dependencia temporal de la cantidad de eventos de partículas alfa durante 15 días.	21
3.8. Distribución espacial de 799 eventos sobre el sensor.	22
4.1. Porcentaje de muertes atribuidas a la intoxicación por Pb por países. . .	25
4.2. Sensor CCD Utilizado y disposición de las láminas de Cu y Pb sobre el sensor.	26
4.3. Configuraciones experimentales utilizadas	27
4.4. Primera imagen del sensor CCD con fuente radioactiva.	27
4.5. Segmentaciones de las trazas dadas por diversas partículas en el sensor CCD.	28
4.6. Espectro leído el CCD, con fuente radiactiva.	28
4.7. Distribución espacial de eventos 12 keV y 13.5 keV sobre el CCD con fuente.	29

4.8. Espectro leído el CCD, sin fuente radiactiva	30
4.9. Espectro leído el CCD en la mitad derecha, sin fuente radiactiva	30
4.10. Configuración experimental para la medición de Pb con CMOS.	31
4.11. Eventos en vacío detectados por el sensor CMOS	32
4.12. Eventos detectados por el sensor CMOS con el ladrillo de Pb.	32
5.1. Representación esquemática de un perceptrón, con 4 neuronas de entrada y una función de activación $g(\cdot)$	35
5.2. Representación esquemática de una red neuronal multicapa densa, con n entradas, p capas y m salidas. Con azul se encuentra señalados los pesos de la primera neurona a las neuronas de la segunda capa.	36
5.3. Eventos y ruidos simulados para $E=27e$	39
5.4. Histograma de cargas de los conjuntos simulados para $E=27e$	41
5.5. Precisión para el conjunto de entrenamiento y validación como función de la época	41
5.6. 5 eventos (resp. ruidos) tomados al azar, clasificados como ruidos (resp. eventos) por la red neuronal.	42
5.7. Histograma de cargas de las entradas mal clasificados de los conjuntos simulados para $E=27e$	42
5.8. Histograma de cargas centrales de las entradas mal clasificados de los conjuntos simulados para $E=27e$	43
5.9. Precisión obtenida en la validación como función de la energía del conjunto de entrenamiento. Además se muestra la fracción de eventos y los ruidos de validación bien clasificados.	43
B.1. Capturas de pantalla de la interfaz web utilizada durante el experimento de Sierra Grande. Se muestra tanto su versión web (izquierda), como su versión <i>mobile</i> (derecha).	52
B.2. Actividad de radón medida por el <i>Wave</i> durante la semana del experimento de Sierra Grande.	53

Índice de tablas

1.1. Organización de la información típica de un grupo de 4 píxeles para una profundidad de 10 bits por píxel.	4
2.1. Características técnicas de los sensores disponibles en las distintas versiones de <i>raspicam</i>	7
2.2. Especificaciones técnicas de los modelos de Raspberry Pi utilizados . . .	8
5.1. Descripción de la red neuronal empleada	40

Resumen

Se desarrolló un código en C que permite obtener y analizar las imágenes *raw* obtenidas por sensores CMOS, para la detección de interacción de partículas con el sensor. Además de la interfaz por consola, se desarrollaron dos interfaces web que permiten visualizar la información procesada.

Gracias a este software, se fabricó un prototipo de un monitor del nivel de actividad de radón. Para ello, se construyó un concentrador de iones con un material conductor, un soporte impreso en 3D, y un sensor CMOS. Los datos obtenidos fueron procesados por una raspberry pi, modelo 3B+. Se demostró el correcto funcionamiento del código y del concentrador de iones para la detección de partículas alfa, provenientes de partículas inestables previamente depositadas sobre el sensor. Se analizó la forma de los eventos de las partículas alfas detectadas, y su distribución sobre el sensor. Las mediciones se contrastaron con un medidor de radón disponible comercialmente, observándose cierta inercia en la medición dada por dicho sensor.

Además, con el fin de detectar concentraciones de plomo en agua, se utilizaron sensores CCDs para la detección de los picos L_α y L_β del plomo. Para ello se colocó una lámina de plomo sobre una mitad del sensor y una lámina de cobre sobre la otra mitad. El cobre utilizado permitió una correcta calibración de la energía de los eventos detectados. Se calculó el flujo eventos que se espera para los picos L_α y L_β del plomo, con estos resultados se extrapoló al sensores CMOS utilizado anterior, y se adaptó un experimento su verificación.

Por otro lado, se utilizaron técnicas de Machine Learning para la clasificación de imágenes en eventos y ruidos. Estas imágenes fueron simuladas tomando un ruido gaussiano para cada pixel y, en el caso de las imágenes de eventos, distribuyendo n cargas unitarias sobre el centro del sensor, siguiendo distribución gaussiana. Todas las imágenes sufrieron un proceso de pre-selección. Se implementó una red neuronal convolucional con una única capa convolucional y una capa densa y se observó la dependencia de la precisión como función de la cantidad de cargas n distribuidas en el conjunto de datos de entrenamiento.

Palabras clave: SENSOR CMOS, CCD, MEDICIÓN DE RADÓN, DETECCIÓN DE PLOMO, REDES NEURONALES ARTIFICIALES, DETECCIÓN DE PARTÍCULAS

Abstract

A C program has been developed to acquire and analyze raw images from a CMOS sensor, to detect particles interactions with the sensor itself. In addition to the command-line interface, two webs interfaces were developed that allow viewing the processed information.

Thanks to this software, a prototype of a radon detector were manufactured. Using an ion concentrator made of conductive tape, a 3D printed stand, and a CMOS image sensor. The data obtained were processed by a raspberry pi 3B+.

Unstable particles were deposited on the sensor, thanks to the ion concentrator. Those particles decay, emitting an alpha which has been detected by the program developed.

The shape of the alpha particles' detected events and their distribution on the sensor were analyzed. The measurements were contrasted with a commercially available radon meter, observing some inertia in the measurement given by said sensor.

To build a lead in water meter, CCD sensors were used to measure L_α and L_β peaks of lead. To do this, a lead sheet was placed on one half of the sensor and a copper sheet on the other half. The copper was used to calibrate the energy of the detected events. The expected event flux by the L_α and L_β peaks of lead was calculated and then extrapolated to the CMOS sensor previously used. This experiment was adapted to the CMOS sensor.

Using Machine Learning techniques, simulated images from events and noises were classified. Those images were simulated by taking a Gaussian noise for each pixel and, in the case of events, by taking n unit charges over the centre of the sensor, following a Gaussian distribution. All those images were validated by a pre-selection process. An artificial neural network with a single convolutional layer and a dense layer was implemented, the dependence of the accuracy as a function of the n charges from the training set was observed.

Keywords: CMOS SENSOR, CCD, LEAD DETECTION, RADON MEASUREMENT, ARTIFICIAL NEURAL NETWORK, PARTICLE DETECTION

Capítulo 1

Introducción

“If I have seen further it is by standing on the shoulders of giants”
— Isaac Newton.

Existen diversos métodos para “ver” partículas, aunque todas ellas muestran los efectos que producen y no las partículas en sí. Por ejemplo, las cámaras de niebla (*cloud chamber*), utilizan vapor para dibujar trazas al pasar una partícula cargada. Un método usual para la detección de partículas consiste en medir la energía depositada por la partícula, al atravesar un medio. En el presente trabajo, nos basaremos en dicho método.

Las partículas cargadas que se mueven por un material interactúan electromagnéticamente con los electrones de los átomos del material, esta interacción excita o ioniza a los átomos, resultando en una pérdida de energía de la partícula. A partir de esto, se define el poder de frenado (*stopping power*) como la energía perdida por unidad de longitud desplazada sobre el material. La ecuación 1.1 es la fórmula de Bethe (en unidades gaussianas) que modela el comportamiento, a energías medias y altas, que presenta una partícula de velocidad $v = \beta c$ sobre un material con densidad electrónica n y un potencial de ionización medio I .

$$-\frac{dE}{dx} = \frac{4\pi e^4 n z^2}{m_e c^2 \beta^2} \left[\ln \left(\frac{2m_e c^2 \beta^2}{I \cdot (1 - \beta^2)} \right) - \beta^2 \right] \quad (1.1)$$

En esta fórmula se tienen en cuenta los efectos relativistas, pero es posible considerar el límite no relativista tomando $\beta \ll 1$. El resultado de esta simplificación se muestra en la ecuación 1.2.

$$-\frac{dE}{dx} = \frac{4\pi n z^2 e^4}{m_e v^2} \ln \frac{2m_e v^2}{I} \quad (1.2)$$

El rango de penetración de una partícula de energía inicial E estará dado por

$$R(E) = \int_0^E \frac{1}{-dE'/dx} dE'$$

Debido a esto, una manera de detectar partículas consiste en medir la energía que depositen, esta energía se podrá medir en forma de calor, vibraciones o ionizaciones. Se propone utilizar sensores de imágenes de silicio para la detección de partículas, los cuales utilizan la energía depositada por la partícula para generar pares electrón-hueco.

1.1. Sensores CMOS

Los sensores CMOS de pixeles activos (APS) disponibles comercialmente son utilizados en fotografía para capturar luz (generalmente del espectro visible). La tecnología CMOS que utilizan hace referencia a un conjunto de familias lógicas basadas en transistores complementarios Metal Óxido Semiconductor (CMOS), los cuales utilizan transistores de tipo pMOS y de tipo nMOS, de tal forma que el único consumo en reposo se debe a las corrientes parasitarias.

Estos sensores han adquirido gran relevancia en el sector tecnológico y se encuentran disponibles de forma masiva en *smartphones*, ordenadores portátiles y consolas de videojuegos, por nombrar algunos. En el ámbito científico, debido al bajo consumo energético y a su bajo costo, se comenzaron a utilizar como alternativa a los CCDs (sección 1.2) [2][3] utilizados en los experimentos de física como DAMIC[4] y CONNIE[5], donde la zona de vaciamiento es más profunda aumentando la sensibilidad, ya que las mediciones requieren alto grado de precisión y sensibilidad.

1.1.1. Principio de funcionamiento

Los sensores de imágenes CMOS consisten en un arreglo matricial de fotodetectores, que producen una corriente de electrones que varía en función de la intensidad de luz recibida, basándose en el efecto fotoeléctrico para la generación de pares electrón-hueco.

En general se utilizan celdas de tres transistores (3T) por pixel, la Fig. 1.1 representa el circuito típico de una celda 3T. Estas celdas contienen una compuerta *reset*, que conecta al fotodiodo a la tensión V_{rst} , limpiando toda la carga integrada. El transistor de lectura T_{sf} actúa como amplificador que permite la lectura sin remover la carga acumulada. El transistor de selección T_{sel} permite la lectura de una única fila.

Si bien son menos utilizadas, existen otros tipos de celdas nT, en particular las 4T implementan una compuerta de transferencia que permiten transmitir la carga acumulada.

A pesar de estar orientadas a la detección de luz (fotones) para su uso en fotografía, cualquier partícula que deposite una energía E en el sensor, es capaz de generar E/a

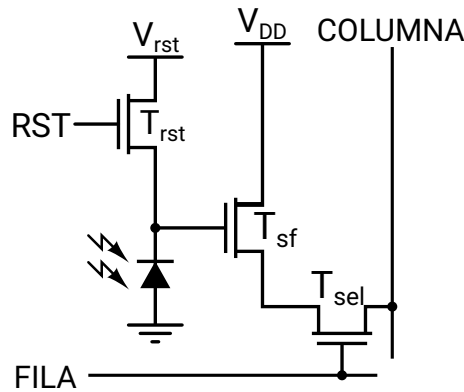


Figura 1.1: Un celda de tres transistores (3T) típica para un sensor CMOS-APS

pares de electrón-hueco siendo a la energía promedio para generar un par electrón-hueco en el material¹. El valor máximo de electrones hasta llegar a saturación es conocido como *full well capacity*.

La lectura de los pixeles se hace por filas y no de forma global. Una vez realizada la lectura, se activan las compuertas de reinicio de la fila y se repite para la siguiente fila. Esto significa que los intervalos de tiempo de integración no son los mismos para todas las filas del sensor, esto da lugar un efecto gelatina (*rolling shutter*) sobre la imagen final.

Uno de los estándares utilizados para la transmisión de la información contenida en los pixeles y el pulso del reloj es el CSI-2 (interfaz serial de cámaras)[7], definido por MIPI, la cual implementa hasta 4 líneas para la transmisión de datos y una para el control del reloj.

1.1.2. Construcción de la imagen

La codificación de la información de los pixeles para la transmisión desde el sensor depende de la profundidad de bits por pixel del mismo. El canal de datos típico para los sensores de 10 bits por pixel presenta la siguiente estructura: En primer lugar, se agrupan 4 pixeles seguidos que corresponden a una misma fila; estos 4 pixeles forman una palabra de 40 bits (5 bytes). Los 8 bits más significativos del i -ésimo pixel del grupo se encuentran en el i -ésimo byte de la palabra. El quinto byte contiene los dos bits menos significativos de cada pixel del grupo. La Tabla 1.1 denota la estructura de un grupo de 4 pixeles en una palabra de 40 bits.

Para disminuir la profundidad de bits a 8, simplemente se debe ignorar el quinto byte en la palabra de 40 bits. Para utilizar los 10 bits por pixel que provee el sensor, se implementó un algoritmo que corte el quinto byte y lo agregue a cada pixel correspondiente. Utilizando un escenario de prueba (Fig. 1.2a), se comprobó el resultado obtenido con dicho algoritmo (Fig. 1.2b).

¹3.7 eV en el caso del silicio [6].

		Bits							
		8	7	6	5	4	3	2	1
B y t e	1	10	9	8	7	6	5	4	3
	2	10	9	8	7	6	5	4	3
	3	10	9	8	7	6	5	4	3
	4	10	9	8	7	6	5	4	3
	5	2	1	2	1	2	1	2	1

Tabla 1.1: Organización de la información típica de 4 píxeles para una profundidad de 10 bits por píxel. El color hace referencia a un píxel del grupo y el número indica el orden del bit, siendo 1 el LSB.



(a) Imagen obtenida con *Rapistill*.
(procesada)

(b) Mediante algoritmo implementado.
(raw)

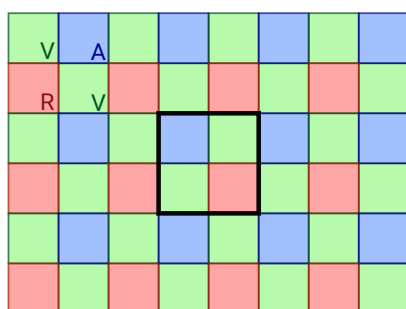
Figura 1.2: Fotografía del escenario de prueba para comprobar la correcta obtención de los datos *raw* mediante el algoritmo implementado.

1.1.3. Filtro de Bayer

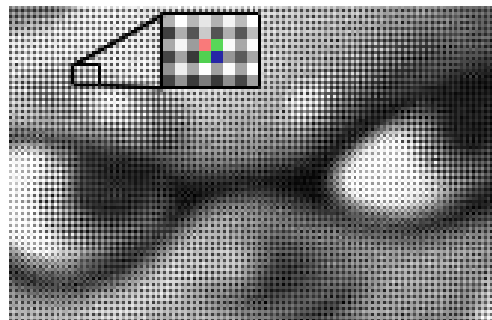
Los sensores CMOS destinados para fotografías a color poseen un filtro de Bayer, el mismo consiste en un arreglo de filtros rojos, verdes y azules, típicamente dispuestos como se muestra en la Fig. 1.3a. Gracias a este filtro, cada píxel posee la información de un único rango de longitudes de onda, permitiendo una composición de la imagen en 3 colores diferentes, de manera análoga al ojo humano. En la Fig. 1.3b se muestra una sección de la Fig. 1.2b, en la que se pueden apreciar los efectos del filtro de Bayer sobre la imagen final.

1.1.4. Píxeles dañados

Los sensores CMOS están compuestos por millones de píxeles y muchos de ellos pueden dañarse por distintos fallos en sus componentes electrónicos; de acuerdo al comportamiento ante la exposición de la luz, pueden clasificarse los píxeles dañados[8] como:



(a) Esquema de un filtro de Bayer.



(b) Ampliación de la Fig. 1.2b.

Figura 1.3: A la izquierda se observa un esquema de un filtro de Bayer típico de los sensores de imágenes CMOS. Cada 4 píxeles, hay 2 filtros verdes, 1 rojo y 1 azul. El color verde es utilizado dos veces debido a la sensibilidad al verde del ojo humano.

A la derecha se pueden apreciar los efectos producidos por el filtro de Bayer sobre la imagen del escenario de prueba. En este caso los píxeles oscuros son los que corresponden al color azul, ya el color predominante es el rojo, como se aprecia en la Fig. 1.2a.

Píxeles muertos

Son aquellos en los que la salida es siempre la mínima posible, sin importar la intensidad a la que estén sometidos. Estos píxeles reducen el área efectiva del sensor y son capaces de atenuar la señal de un evento que contenga dicho píxel.

Píxeles atascados

Al contrario de los píxeles muertos, estos presentan un valor cercano al valor de saturación. Son fácilmente detectables al capturar sucesivas fotos en oscuridad (o bien, en vacío, como nos referiremos en adelante), pero su identificación es importante ya que los mismos pueden confundirse con eventos puntuales (de 1 ó 2 píxeles).

Píxeles calientes

Estos píxeles pueden aparecer con un valor más alto que otros píxeles normales, dependiendo del tiempo de exposición, la temperatura del sensor y la intensidad de luz recibida; sin necesariamente llegar a saturación.

1.2. Sensores CCD

Tanto los CCD como los sensores CMOS están formados, en esencia, por semiconductores de metal-óxido; y también se basan en el efecto fotoeléctrico para generar pares electrón-hueco. La principal diferencia entre los CCD y los sensores CMOS es a la hora de leer dicha carga acumulada.

Para ello, estos sensores son capaces de transmitir la carga almacenada a celdas vecinas. Esta carga puede ser transportada vertical y horizontalmente para luego realizar

la medición de la carga almacenada por dicho pixel.

En general, los sensores CCD presentan un mayor rango dinámico y un menor ruido de lectura que los CMOS, a costa de un precio mayor. Además, presenta mayor uniformidad entre los pixeles, haciendo que sea menos susceptible a encontrarse pixeles defectuosos.

Capítulo 2

Desarrollo de Radmonpi, monitor de radiación de partículas

“Sometimes science is more art than science, Morty. A lot of people don’t get that.”

— Rick. Rick and Morty

2.1. Hardware utilizado

2.1.1. Sensor utilizado [9][10]

El principal sensor utilizado fue desarrollado por Omnivision, modelo OV5647. El mismo posee 5038848 pixeles activos dispuestos en 2592 filas y 1944 columnas. El tamaño de los pixeles es de $1.4\mu\text{m} \times 1.4\mu\text{m}$, lo que da una superficie total activa de $3.63\text{mm} \times 2.72\text{mm}$. Las señales de los pixeles puede leerse con ADCs de 10 bits integrados en el mismo chip, por lo que estos poseen un valor adimensional entre 0 y 1023. Todas estas características pueden verse en la tabla 2.1.

Además presenta una serie de pixeles en la periferia que son utilizados para la calibración del nivel de negro (BLC), estos pixeles presentan un filtro que bloquea la

Tabla 2.1: Características técnicas de los sensores disponibles en las distintas versiones de *raspicam*.

Características	OV5647	IMX219
Tamaño de pixel	$1.4\mu\text{m} \times 1.4\mu\text{m}$	$1.12\mu\text{m} \times 1.12\mu\text{m}$
Profundidad de bits	10 bits (0 - 1023)	10 bits (0 - 1023)
Resolución	2592 x 1944	3280 x 2464
Resolución total	5038848	8081920
Rango de temperaturas	0°C a 50°C	-20°C a 60°C
Frecuencia de reloj	6 MHz a 27 MHz	6 MHz a 27 MHz
Transferencia a resolución máxima	15 fps	30 fps

luz visible. En el presente trabajo se desactivó la calibración del nivel de negro debido a que dicho procesamiento modifica los valores *raw* dados por la cámara.

El sensor también presenta un corrector de píxeles defectuosos (DPC), que permite identificar píxeles atascados (definidos en la sección 1.1.4) y promediarlos con los píxeles vecinos. Este tipo de procesamiento sobre la imagen bloquea señales provenientes de un único píxel como pueden ser los eventos provenientes de fotones de rayos X.

El sensor OV5647 se encuentra en la versión 1.3 de la *Raspicam*, una cámara CMOS disponible comercialmente y adaptada para su uso en una Raspberry Pi (compatible con cualquier versión). El precio de la cámara ronda las £17. También se comprobó el funcionamiento del sensor IMX219, desarrollado por Sony, que se encuentra en la versión 2.1 de la *Raspicam*.

2.1.2. Raspberry Pi, SBC utilizado

Raspberry Pi es una computadora de placa reducida (SBC) de bajo costo producida en el Reino Unido, se encuentra disponible comercialmente desde febrero del 2012. Este tipo de computadoras tuvieron un gran auge en los últimos años, debido a la facilidad de uso, a la diversidad de aplicaciones, el bajo costo y su reducido tamaño y peso. Estas características las hacen ideales para su uso en educación y para el diseño de prototipos de dispositivos pequeños, que no requieren un poder de procesamiento elevado.

En el presente trabajo se utilizaron los modelos 1B y 3B+ de *Raspberry Pi*, sus especificaciones se encuentran en la Tabla 2.2. Si bien *Raspberry Pi* soporta varios sistemas operativos (para arquitecturas ARM), se utilizó *Raspbian* debido a la facilidad de instalación y al soporte que este presenta.

	Raspberry Pi 1B	Raspberry Pi 3B+
Fecha de lanzamiento	Abril, 2012	Marzo, 2018
SoC	Broadcom BCM2835	Broadcom BCM2837
CPU	700 MHz ARM11	1.4GHz quad-core ARMv8
Memoria RAM	512 MB	1 GiB
Conectividad	Ethernet	Ethernet, Wifi 802.11n/ac, Bluetooth 4.2
USB 2.0 disponibles	2	4
Almacenamiento	Tarjeta SD	Tarjeta Micro SD

Tabla 2.2: Especificaciones técnicas de los modelos de Raspberry Pi utilizados

En el caso de disponer conectividad wifi, se configuró un punto de acceso (AP) con servidor DHCP para simplificar la comunicación. Para crear el AP se utilizó el software *hostapd*, y para configurar el servidor DHCP se utilizó *dnsmasq*. Además se permitió el enrutamiento entre las interfaces cableadas e inalámbricas, a modo de NAT, para permitir conectividad a internet a los clientes conectados al AP.

Para facilitar la conexión de la cámara, se utilizó un adaptador con salida HDMI tal como puede verse en la Fig. 2.1



Figura 2.1: Raspberry 3B+ utilizada con su respectivo adaptador HDMI para la conexión con la cámara.

Debido al SoC que presentan la raspberry pi 3B+, se analizó la posibilidad de hacer uso de las QPU del fabricante Broadcom, tal como se muestra en el apéndice A

2.2. Librerías previas

2.2.1. Raspiraw

Este programa escrito en lenguaje C permite utilizar los sensores ADV7282M, IMX219 y OV5647 para obtener imágenes sin procesar (datos *raw*). Raspiraw permite establecer los registros de los sensores y configurar distintos parámetros como el tiempo de exposición, ganancia digital, cantidad de cuadros por segundo y el tiempo de ejecución del programa.

Los datos *raw* pueden ser guardados en su totalidad o saltando una cantidad de cuadros definida por el usuario. Esta característica es útil para guardar imágenes en una tarjeta micro SD, debido a su baja velocidad de escritura. En caso de guardar todos los cuadros, se recomienda guardarlos en memoria RAM, debido a su velocidad de acceso.

El código fuente es libre, bajo la licencia BSD de 3 cláusulas, y se encuentra disponible en GitHub. Fue desarrollado por un usuario de la plataforma bajo el pseudónimo “6by9”.

2.2.2. Primera versión de radmonpi[1]

Es una modificación de *raspiraw* desarrollada por Fabricio Alcalde y Clara Galimberti durante la beca de verano 2018 del Instituto Balseiro. La misma utiliza la librería OpenCV para detectar pixeles que superen un umbral, guardando una región contigua

a aquellos pixeles que superen dicho umbral (región de interés). También es capaz de calcular y mostrar la carga del evento, el área, la posición y el tiempo de los eventos.

Para discriminar eventos utiliza pixeles adyacentes que superen un cierto umbral definido por el usuario. Además es posible establecer un tamaño de pixeles mínimos para considerar un conjunto de pixeles consecutivos como un evento.

Debido a la dependencia con OpenCV utiliza una profundidad de bits de 8 bits, por lo que la señal dada por el sensor está limitada entre 0 y 255 unidades de ADC.

2.2.3. Interfaz web (detecGUI)

Radmonpi incluye un *relay* de datos que crea un puerto TCP al cuál se pueden conectar varios clientes para recibir los datos de los eventos detectados. Esto es especialmente útil para evitar la conexión SSH para la obtención de los datos, permitiendo que múltiples clientes puedan acceder a los mismos datos, protegiendo la seguridad del resto del sistema.

Además se incluyó una interfaz gráfica web hecha en python, con *Bokeh*. En esta interfaz web se muestran tres gráficos:

- Eventos por segundos. Obtenido como $\frac{100}{t_i - t_{(i-100)}}$, siendo t_i el tiempo del evento i .
- Histograma de area de los últimos 1000 eventos.
- Histograma de carga de los últimos 1000 eventos.

2.3. Nuevo Radmonpi (Radmonpi++)

La nueva versión se propone como una modificación de la versión desarrollada por Fabricio y Clara, en la que se elimina la dependencia de la librería OpenCV, permitiendo hacer uso de los 10 bits del sensor. Además se definen mecanismos para la detección de pixeles defectuosos para favorecer la detección de interacciones con menor area.

La documentación sobre el modo de utilización del programa y la correcta configuración puede encontrarse en el repositorio en GitHub <https://github.com/DBFritz/Radmonpi>

2.3.1. Configuración inicial

Una vez instalado el sistema operativo *Raspbian*, se habilitaron los puertos de comunicación I²C. Además, se configuró la Raspberry Pi como AP wifi, permitiendo una comunicación inalámbrica para el análisis de los datos.

2.3.2. Utilización del programa con interfaz en consola (CLI)

El programa por consola utiliza el protocolo de comunicación serial I²C para la comunicación con la cámara.

En el código fuente están definidos distintos modos de capturas, en los cuales se definen los registros a escribir. El modo por defecto presenta una larga exposición (4 s) y elimina los filtros internos del sensor de ajuste del nivel de negro y de compensación de pixeles defectuosos (sección 2.1.1). Estos registros son escritos al inicio del programa, y quedan fijos durante toda la medición.

Mediante el uso de múltiples hilos de ejecución (*multithreading*), el programa es capaz de exponer el sensor mientras se procesan las imágenes. El procesamiento consiste en la detección de pixeles que superen un valor umbral fijado previamente (llamado valor de *trigger*), cuando se encuentra dicho pixel se añaden los pixeles vecinos que superen otro valor umbral menor al anterior (llamado valor de *threshold*) y se repite este último paso para los pixeles añadidos, sin evaluar los pixeles ya añadidos previamente. Esta lista de pixeles constituyen un evento, y sobre esta se calculan los siguientes valores.

1. **Tiempo del evento:** Tiempo entre el inicio del programa y la detección del evento. La incerteza en el tiempo de la interacción es justamente el tiempo de exposición, ya que es imposible distinguir distintos tiempos en un mismo cuadro (imagen).
2. **Tamaño del evento:** Cantidad de pixeles (tomando los 8 pixeles adyacentes) que superan un cierto umbral fijado (*threshold*), estos pixeles son los que conforman el evento siempre y cuando no se superpongan los pixeles provenientes a distintas partículas.
3. **Cantidad de pixeles saturados:** Determinar la cantidad de pixeles saturados es importante ya que existe una zona de trabajo en la cual la energía depositada por una partícula es lineal con el valor dado por el pixel. No es posible garantizar esta linealidad cuando los pixeles están saturados.
4. **Carga del evento:** Definida como la suma de los valores de los pixeles que conforma el evento, que es proporcional a la energía depositada por la partícula en la zona activa del sensor, siempre que los pixeles no se encuentren saturados ni los eventos se solapen.
5. **Centro de masa del evento:** Definida como un promedio de las posiciones de los pixeles del evento, ponderado por los valores de los pixeles.

A su vez el programa puede recibir los siguientes argumentos (todos son opcionales):

- **Ganancia:** Fija el valor de ganancia en el registro del sensor, el puede variar entre 0 y $2^{n_g} - 1$, con n_g el número de bits en el registro de ganancia (10 para el sensor OV5647). Las unidades no se encuentran calibradas y la relación depende del sensor.
- **Tiempo de captura (*timeout*):** Tiempo de ejecución del programa, pasado el mismo se desactiva la captura del sensor y el programa finaliza.
- **Tiempo de exposición:** Es el tiempo de adquisición de un cuadro. Para poder procesar todos los cuadros es necesario que este sea mayor al tiempo de procesamiento.
- **Mostrar tiempos:** Muestra en pantalla la duración de cada tarea, lo cual resulta útil para comprobar que el tiempo de exposición fijado sea el correcto y para tener en cuenta la cota inferior en el tiempo de exposición para evitar desincronización.
- **Tamaño mínimo y máximos de eventos:** Al especificar el tamaño mínimo se permite el filtrado de eventos más puntuales como los provenientes de rayos X, además de reducir considerablemente el ruido. El tamaño máximo permite agregar estabilidad al detector ante cambios repentinos en la luminosidad expuesta al sensor.
- **Guardar eventos:** Las imagenes de los eventos pueden ser guardadas con en formato .pgm, Netpbm en escala de grises y en caracteres imprimibles (ASCII).
- **Valores umbrales para la detección:** También se puede ajustar el valor mínimo para la detección de un evento (valor de *trigger*). Una vez alcanzado este valor, se toman todos los pixeles adyacentes que superen otro valor fijado (definido como valor de *threshold*).
- **Cargar máscara de pixeles defectuosos:** Para la detección de eventos con un número bajo de tamaño de pixeles resulta útil cargar una máscara previa de pixeles defectuosos, estos archivos son generados con el generador de máscaras (ver sección 2.3.3).

2.3.3. Generación de una máscara de pixeles defectuosos

El algoritmo de determinación de pixeles defectuosos establece un único valor umbral (al contrario de la detección), y un número mínimo de fotogramas en los que un pixel particular debe superar ese valor. Este algoritmo consiste en armar un arreglo de contadores del tamaño de la imagen, cada contador identifica un único pixel y se incrementa cuando el valor del píxel es mayor al valor de umbral. Al finalizar la adquisición (mediante un tiempo ajustado previamente o mediante la señal de interrupción

del sistema), se imprimen en pantalla la lista de las coordenadas de los pixeles defectuosos, es decir, aquellos pixeles que superaron el valor umbral en más de la cantidad de imágenes mínimas seteadas.

Este programa también permite exportar el histograma de los valores de todos los pixeles para hacer un análisis más profundo sobre las imágenes adquiridas en este proceso.

Tanto la generación de la máscara de pixeles defectuosos, como el monitoreo en tiempo real de partículas pueden ser ejecutados e interpretados desde la interfaz web desarrollada en *Bokeh*.

2.3.4. Interfaz web en *Bokeh*

Para la visualización de los datos de forma remota se desarrolló una interfaz web al programa, que es capaz de ajustar ciertos parámetros y mostrar resultados en tiempo real. El código es soportado gracias a *Bokeh*, una librería de Python que permite visualizar datos de forma interactiva, usando estándares de los navegadores más modernos. En la Fig. 2.2 se muestra la interfaz ejecutándose en la versión 56 de Opera.

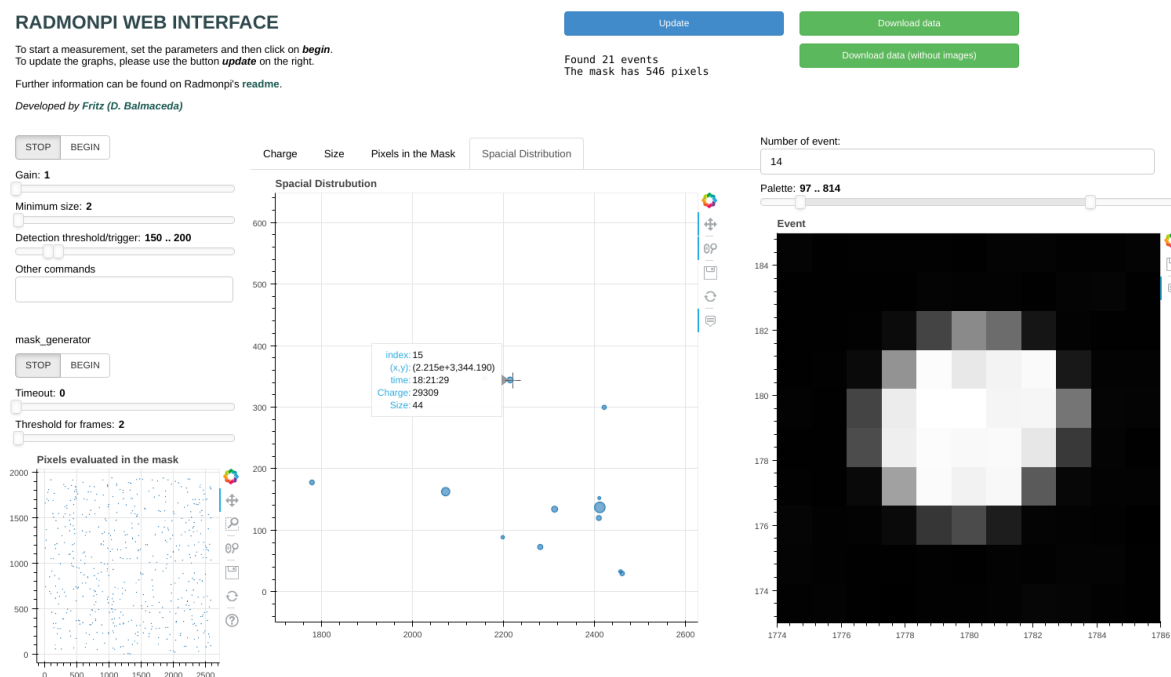


Figura 2.2: Interfaz web de Radmonpi desarrollada en *Bokeh*.

En esta interfaz pueden ajustarse los valores de ganancia, tamaño mínimo y los valores para la detección de eventos. Además permite la generación de máscaras, especificando previamente los parámetros necesarios; una vez finalizada la generación de la máscara, es posible ver los pixeles que contiene en el gráfico correspondiente.

Desde la interfaz web se puede observar la distribución espacial de los eventos sobre el sensor, y los histogramas de carga de los eventos, de tamaños de los eventos y de

valores de los pixeles al obtener la máscara. Este último permite determinar la cantidad de pixeles que se encuentran saturados, lo cual es importante ya que se espera que los mismos no se encuentren saturados para la detección de los eventos (el sensor debe estar a oscuras). A su vez, gracias a esto, es posible estimar el número de pixeles defectuosos que se esperan encontrar en la máscara al definir un cierto umbral.

La interfaz web también está equipada con un visualizador de eventos, en los que, al especificar un índice, se muestra, en escala de grises, los valores de los pixeles del evento y su entorno. Además de especificar la posición de cada pixel sobre el sensor, también se especifican el tiempo de detección, la carga y el tamaño del evento.

Al utilizar el servidor web que provee *Bokeh*, y al ejecutar código de procesamiento del lado del servidor, el rendimiento de dicha interfaz puede verse disminuido. En base a eso, se decidió desarrollar una interfaz web modular en la librería *React*.

2.3.5. Interfaz web en *React*

En base a los problemas que presenta *Bokeh*, se diseñó una interfaz web modular para visualizar los datos obtenidos de *Radmonpi*. Para ello, se utilizó el *framework React*, desarrollado por *Facebook*. El mismo permite desarrollar PWAs en *Javascript* (JS), incluyendo sintaxis de HTML como objetos (*JSX*). *React* necesita tener instalado *NodeJS* para su desarrollo; pero una vez compilado, se generan códigos estáticos en HTML, CSS y JS que se pueden montar en cualquier servidor (programación *serverless*).

Para el desarrollo de la interfaz se utilizó la librería de *Material UI*, desarrollada por *Google*. Esta define componentes que se pueden integrar en una aplicación web para la visualización, siendo amigables tanto para interfaces web como móviles. Para realizar los gráficos se utilizó la librería *recharts*.

Debido a su modularidad, es que esta interfaz se adaptó en el experimento de medición del flujo de muones en la mina de Sierra Grande, tal como se describe en el apéndice [B](#).

Capítulo 3

Medición de Radón

“If you tell the truth you don’t have to remember anything”
— Mark Twain (1835-1910), escritor.

3.1. Introducción y motivación

El radón (símbolo químico Rn) es el gas noble con número atómico 86, cuyo isótopo más abundante es el ^{222}Rn , que es un producto de la cadena de decaimiento del ^{238}U . A su vez, este isótopo también es inestable y su vida media es 3.82 días, la mayor de todos los isótopos del Rn. Los decaimientos sucesivos, con sus respectivas vidas medias y sus energías liberadas, se muestran en la Fig. 3.1.

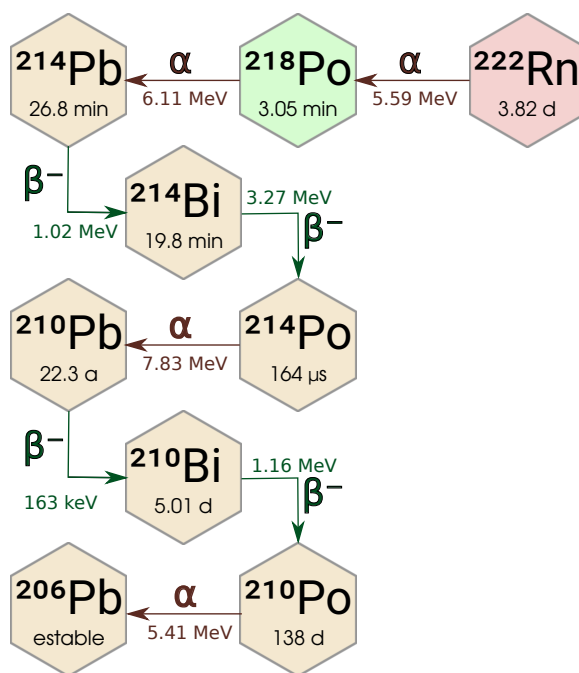


Figura 3.1: Cadena de decaimiento del ^{222}Rn . Debajo de cada elemento se especifica la vida media.

El radón es la segunda causa principal de cáncer de pulmón, la OMS establece que de un 3% a un 15% de los casos de cáncer de pulmón son causados por exposición al Rn. Se atribuye aproximadamente unas 21 mil muertes por año debido a exposición de Rn. No existe un nivel determinado a partir del cual la actividad de Rn puede considerarse como peligroso, pero distintas organizaciones utilizan valores umbrales que pueden variar entre los 100 Bq/m³ y 200 Bq/m³. En particular, la OMS recomienda no permanecer en ambientes con una actividad de Rn mayor a 200 Bq/m³.

Existen varios medidores de radón disponibles comercialmente, encontrándose una amplia gama de precios y características. En particular, para la medición del nivel de Rn del ambiente se utilizó el medidor *Wave* producido por *Airthings*, cuyo costo es de €199 y es capaz de monitorear el nivel de Rn, la temperatura y la humedad. Presenta tres LEDs de colores rojo, amarillo y verde que denotan niveles de Rn mayor a 150 Bq/m³, entre 100 Bq/m³ y 150 Bq/m³ y menor a 100 Bq/m³, respectivamente. A su vez, se sincroniza con dispositivos móviles con Android o iOS mediante la aplicación desarrollada a tal fin, permitiendo al usuario observar los datos medidos. Funciona mediante dos baterías AAA y la duración estimada de la batería es de 1.5 años.

Haciendo uso del software Radmonpi, descrito en la sección 2.3, es posible detectar partículas alfas provenientes de la cadena de decaimiento del ²²²Rn (Fig. 3.1), si previamente se depositan las partículas inestables sobre el sensor [11]. Se depositará ²¹⁸Po producido por ²²²Rn.

3.2. Configuración experimental

Para depositar elementos inestables que emitan partículas alfas se implementó un concentrador de iones, para esto es necesario que las partículas a depositar estén eléctricamente cargadas. El concentrador de iones consiste en placas metálicas con simetría de revolución dispuestas en forma de embudo, el potencial eléctrico de las placas decrece conforme decrece el radio. En particular, y a modo de prototipo, se utilizaron 3 potenciales distintos. El potencial más alto fue de 2000 V, la etapa intermedia fue de 1950 V y por último se ajustó a tierra la base del concentrador. Para asegurar el potencial de las placas se utilizó una fuente de alta tensión y un divisor resistivo de 10 MΩ.

El ²²²Rn decae emitiendo una partícula alfa, y el ²¹⁸Po producto de este decaimiento, queda cargado eléctricamente. Gracias al concentrador de iones, se deposita el ²¹⁸Po sobre el sensor, de tal forma que, cuando este decaiga, pueda observarse una partícula alfa sobre el sensor.

El sensor utilizado en este experimento fue el Omnivision OV5647, descrito en la sección 2.1.1, al que previamente se le removió la lente óptica para evitar que las partículas alfas se atenúen sin llegar a ser detectadas. Se colocó el sensor en un porta-

sensor impreso en 3D[12], el mismo también fijaba la posición del concentrador de iones. En la parte más baja del porta-sensor se colocó una lámina de cobre, la cual se fijó en 0 V. El dispositivo terminado se puede ver en la Fig. 3.2.



Figura 3.2: Dispositivo experimental implementado y utilizado para la medición de ^{222}Rn .

Todo el conjunto se colocó en un amplio contenedor opaco de madera, de forma tal que todo el sistema quede en oscuridad y no sea necesario una ventilación adicional sobre el sistema. Se colocó la *Raspberry Pi* fuera del contenedor opaco y se utilizaron las aberturas por la cual pasan los cables, para garantizar una ventilación.

3.2.1. Máscara de pixeles defectuosos utilizada

La máscara creada en vacío contenía 934 pixeles y se obtuvo con un valor umbral (*threshold*) de 356 unidades de ADC, fue necesario que un pixel superara dicho valor en más de 3 de los 50 fotogramas tomados, dichos valores garantizaron una correcta adquisición de las partículas alfas.

En la Fig. 3.3 se muestra el histograma del fondo, normalizado en las 50 imágenes adquiridas, de tal forma que el área total del histograma corresponde a la resolución del sensor. En dicha figura también se muestra la diferencia entre el total de los pixeles y la integral del histograma (denominada función de distribución desacumulativa), esto da una idea de la cantidad de pixeles que se espera encontrar en una máscara para distintos valores umbrales. Como se puede ver, para un umbral de 356 unidades de ADC, se espera del orden de 900 pixeles defectuosos.

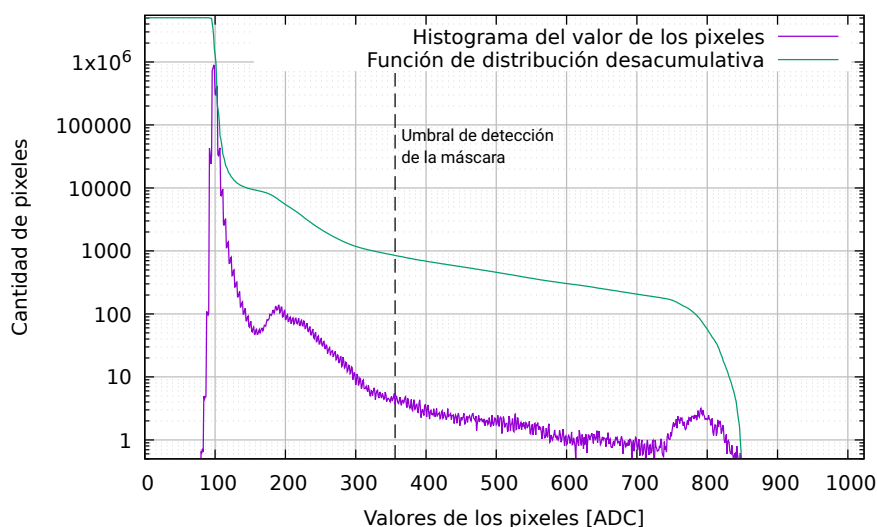


Figura 3.3: Histograma del promedio de las imágenes para determinar el nivel de fondo y la máscara de pixeles defectuosos. Además se muestra la función de distribución desacumulativa. El histograma se encuentra normalizado con el número de imágenes adquiridas.

Se observan tres grupos distintos de pixeles. En primer lugar, un grupo que conforma la mayoría de los pixeles, con una distribución centrada cerca de las 100 unidades de ADC. Un grupo de pixeles con una de distribución no simétrica que abarca desde las 170 hasta las 300 unidades de ADC, seguido de un decaimiento exponencial; este conjunto de pixeles corresponde al grupo de pixeles calientes. Por último, se observa un grupo reducido de pixeles cuyo valores ronda entre 740 y 840 unidades de ADC; este último grupo corresponde a los pixeles atascados y todos ellos fueron añadidos a la máscara.

3.3. Primeras mediciones

Se aplicó alta tensión durante las primeras 43 horas de exposición del sensor, y 17 horas sin alta tensión; durante este tiempo se detectaron 118 eventos provenientes de partículas alfa.

3.3.1. Número de cuentas como función del tiempo

La Fig. 3.4 muestra la cantidad de partículas alfa detectadas, considerando una ventana de tiempo de 4 horas; además se muestra el número acumulado total de cuentas.

Para comprobar el correcto funcionamiento del concentrador de iones, a las 43 horas de medición se apagó la fuente de alta tensión. Se observa que la cantidad de eventos detectados luego del apagado de la fuente resultan despreciables.

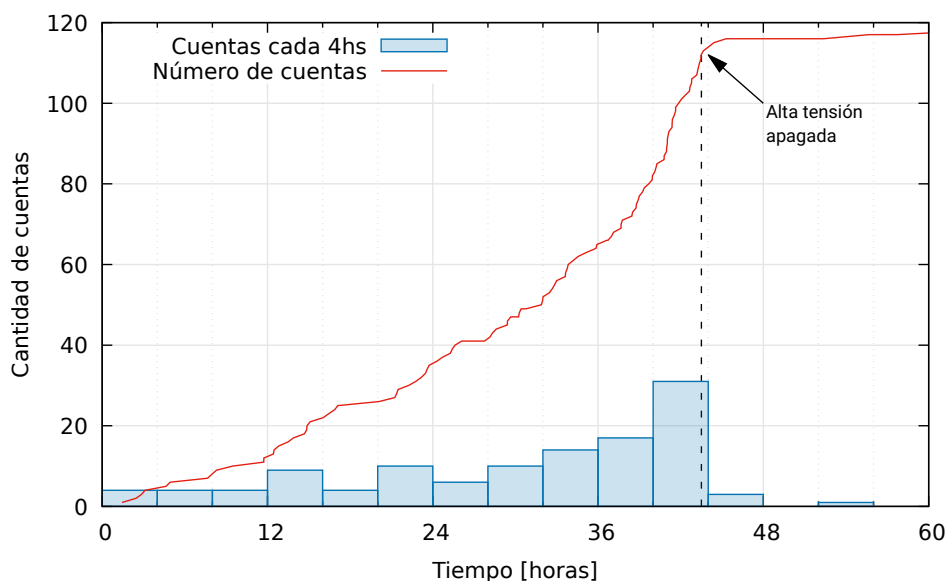


Figura 3.4: Cantidad de partículas alfa detectadas como función del tiempo. Se muestra el número de cuentas, tanto de forma acumulativa (con línea continua), como tomando una ventana de tiempo de 4 horas (en barras). A las 43 horas se observa un cambio rápido en el número de cuentas debido al apagado de la fuente de alta tensión.

3.3.2. Distribución espacial de los eventos

Para analizar la distribución espacial de los eventos detectados se graficó las posiciones de todos los eventos provenientes de partículas alfas detectadas, la Fig. 3.5 muestra la posición de los eventos sobre el sensor.

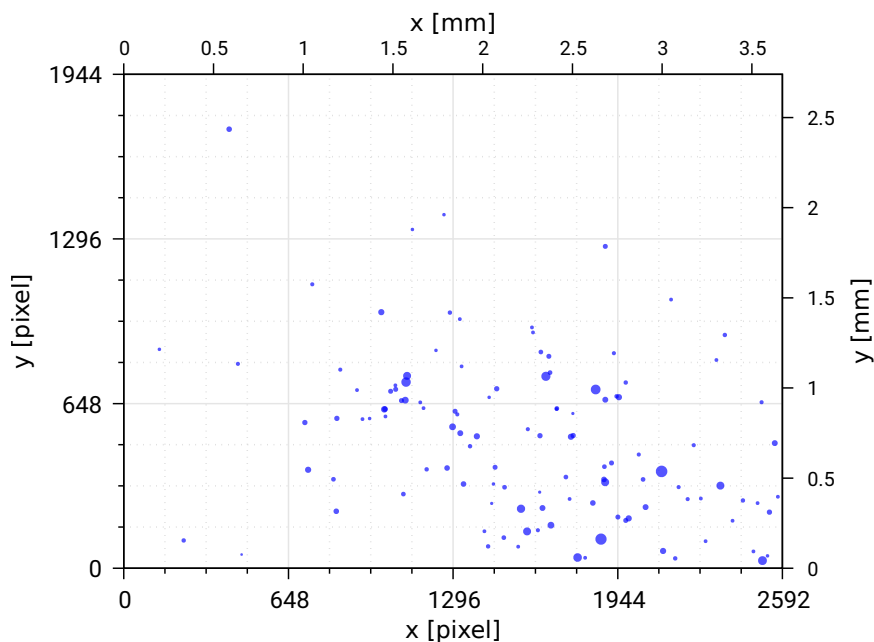


Figura 3.5: Distribución espacial de los eventos correspondientes a partículas alfas detectadas sobre el sensor, durante una exposición de 43 horas. El área de los círculos es proporcional al área del evento.

En base a esto, se observa una región de la cámara con mayor densidad de eventos, lo que indica un correcto funcionamiento del concentrador de iones. Por otro lado, se observa una incorrecta alineación entre el centro del sensor y la máxima densidad de eventos. Si bien esto significa una disminución en el número de cuentas que se podrían detectar, permite una mejor identificación sobre la distribución espacial dada por el concentrador, asumiendo que el mismo presenta simetría radial.

3.3.3. Forma de los eventos

El tamaño y la forma de los eventos permitió discriminar entre eventos provenientes de partículas alfas de los producidos por otras partículas. A su vez, las formas de los eventos de partículas alfas son diversas entre sí, aunque todas ellas corresponden a óvalos. En la Fig. 3.6 se muestran dos eventos de partículas alfas con trazas diferentes.

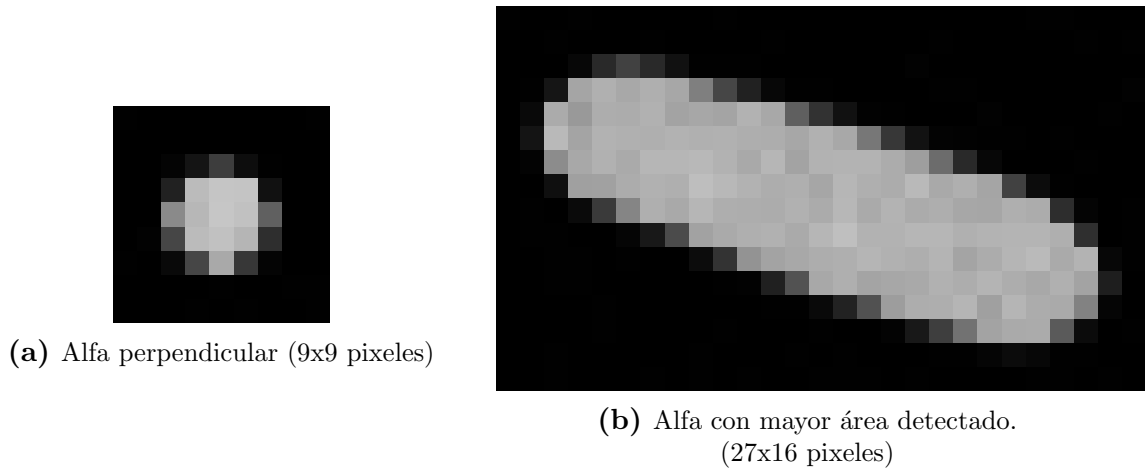


Figura 3.6: Dos eventos provenientes de partículas alfas, debajo de cada una se especifica el número de pixeles totales en la imagen. Los tamaños de los eventos fueron 18 y 160, respectivamente.

Las diferencias entre las trazas de estas partículas están dadas por el ángulo de incidencia sobre el detector. Los iones de ^{218}Po están depositados sobre el sensor, pero al decaer, las partículas alfas pueden ser emitidas en cualquier dirección. Las partículas que ingresan perpendicular al sensor dejan una traza más chica y completamente circular. Aquellas que ingresan al sensor con un ángulo de incidencia dejan una traza cuya longitud dependerá del ángulo con respecto a la normal.

La mayor longitud en la traza se obtendrá cuando la partícula ingrese horizontalmente; consideremos que la partícula con una mayor traza detectada (Fig. 3.6b), cumple esta condición. Esta partícula abarca (24 ± 2) pixeles en el eje x , y (14 ± 2) pixeles en el eje y , lo que nos da una longitud total de (28 ± 2) pixeles, o bien (39 ± 3) μm .

Teniendo en cuenta el poder de frenado (dado por la ecuación 1.2) de una partícula alfa en el silicio, a la energía de emisión del ^{222}Rn (5.6 MeV), es posible estimar la energía depositada. Estimando un poder de frenado constante de (130 ± 10) keV/ μm

[13], obtenemos una energía depositada de (5.1 ± 0.5) MeV, que es menor que la energía del alfa emitido. Esto es coherente con lo esperado, ya que la partícula puede no haber entrado perfectamente horizontal, el poder de frenado no es constante para todo el rango de energía y parte de ella debería quedar depositada sobre el Filtro de Bayer.

3.4. Medición de medio mes

En base a las primeras mediciones, se procedió a realizar una medición 358 horas (casi 15 días), tiempo durante el cual se encontraron 799 eventos de partículas alfa.

3.4.1. Número de cuentas como función del tiempo

En la Fig. 3.7 se observa el número de eventos de partículas alfa como función del tiempo, ya sea de forma acumulativa, como tomando ventanas de tiempo de 4 horas. En la misma Fig. también se gráfica el nivel de Rn obtenido con el medidor *Wave* de *Airthings*.

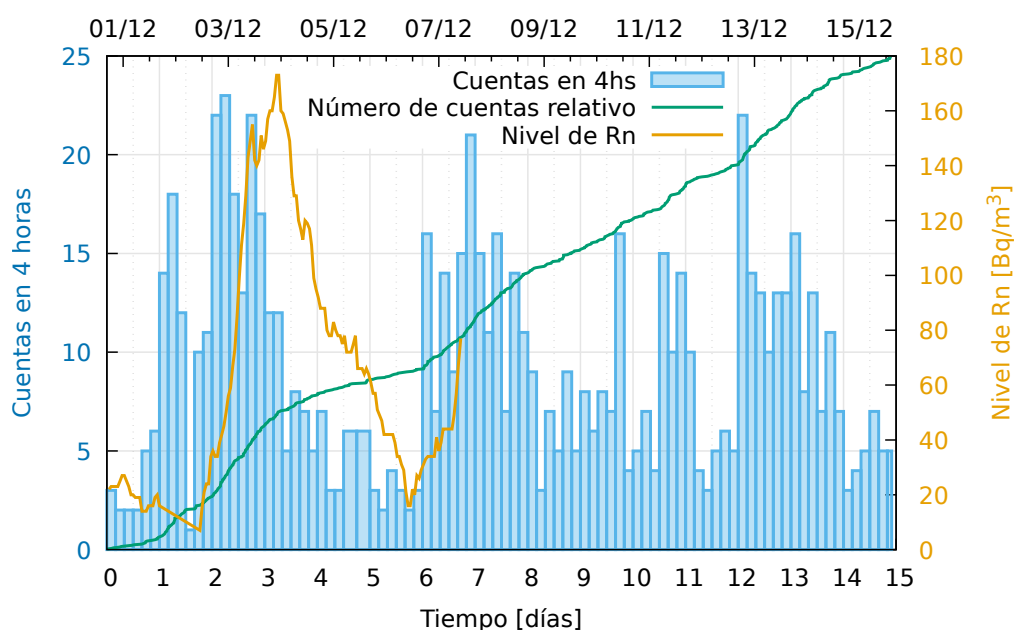


Figura 3.7: Cantidad de partículas alfa detectadas como función del tiempo. Se muestra el número de cuentas, tanto de forma acumulativa (con línea verde), como tomando una ventana de tiempo de 4 horas (en barras). Además se muestra la actividad de Rn medida con *Wave*.

Como se puede ver, existen ciertos intervalos de tiempo en el cual el sensor de *Airthings* no sincroniza correctamente los datos y estos son corrompidos. Si bien este comportamiento es concurrente en el sensor, no se encontraron patrones que permitan obtener información sobre los datos no sincronizados.

Además de los datos no sincronizados, se puede observar que el aumento de nivel de Rn según el medidor *Wave* está desfasado al aumento en el número de partículas

alfa detectado. Dicha desincronización a la hora de medir el nivel de Rn también se encontró presente durante la medición llevada a cabo en la mina de Sierra Grande (apéndice B).

3.4.2. Distribución espacial de los eventos

Al igual que para la primera medición, se procedió a analizar la distribución de los eventos sobre el sensor, tal como se muestra en la Fig. 3.8.

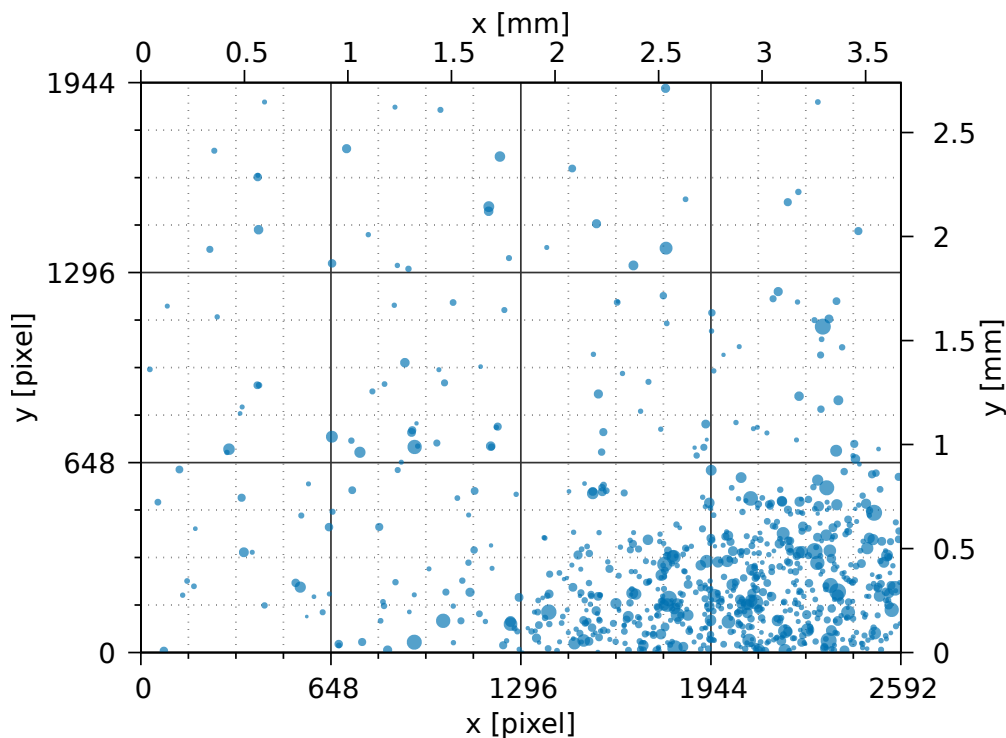


Figura 3.8: Distribución espacial de los 799 eventos correspondientes a partículas alfas detectadas sobre el sensor. El área de los círculos es proporcional al área del evento.

Al igual que en el caso anterior, se observa dicha concentración de eventos sobre una de las esquinas del sensor. Para centrar dicha distribución deberá ser necesario rediseñar el soporte impreso en 3D, con las medidas adecuadas, ya que el sensor queda fijo gracias al soporte.

3.5. Conclusiones

3.5.1. Primera medición

La primera medición comprobó el correcto funcionamiento del concentrador de iones, y permitió analizar la energía depositada en los eventos detectados. Además se observó que la concentración de iones no se encontraba centrada sobre el sensor.

3.5.2. Mediciones de medio mes

La segunda medición denota una relación entre el nivel de Rn medido por el *Wave* y el número de cuentas detectadas por el sensor. Se observó que el medidor *Wave* presenta mayor inercia el que el sensor desarrollado, con un desfase del orden de un día. Dicho comportamiento se observó también en la medición llevada a cabo en Sierra Grande.

Además, esta medición permitió distinguir con mayor precisión la forma definida por el concentrador de iones, esta asimetría se debe principalmente a un descentrado del sensor con respecto al cono concentrador. Para lograr centrarla correctamente, y con ello maximizar el número de eventos, se deberá rediseñar el sensor.

Capítulo 4

Medición de Pb con sensores CCD

*“-One goes into an experiment knowing one could fail.
-But one does not undertake an experiment knowing one has failed.”*
— Robert y Rosalind Lutece. Bioshock Infinite

4.1. Introducción y Motivación

El Instituto de Sanimetría y Evaluación Sanitaria (IMHE) estimó que, en el 2017, la exposición al plomo causó entre 708 mil y 1.4 millones de muertes, y la pérdida de aproximadamente 24 millones de años de vida saludables[14], debido a sus efectos a largo plazo en la salud[15]. Tal como se muestra en la Fig. 4.1, entre las regiones más afectadas se encuentra el norte de África, el sur de Asia, y los países caribeños.

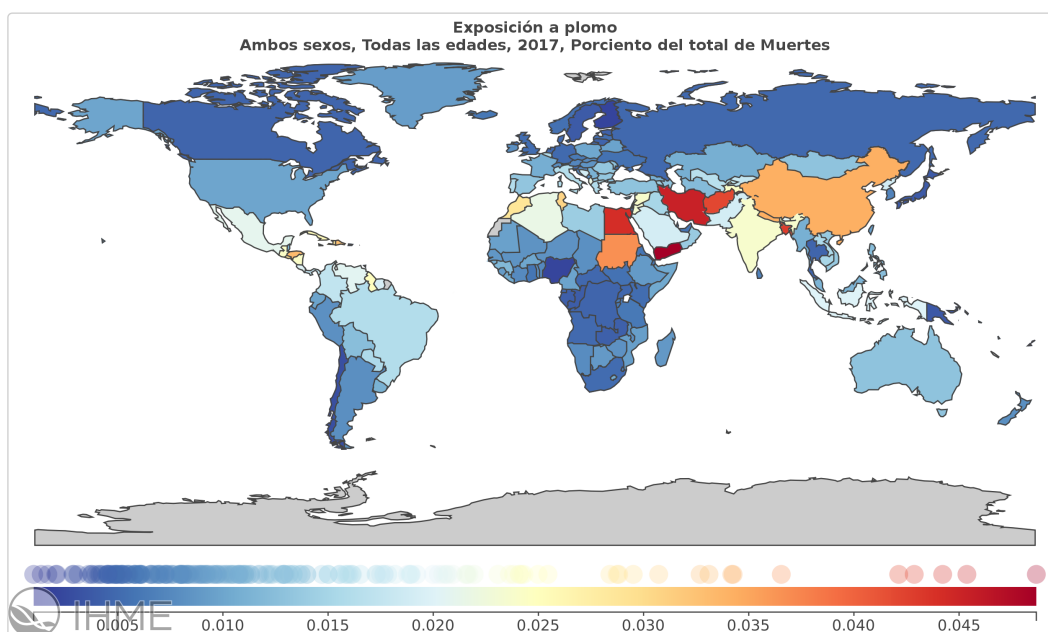


Figura 4.1: Porcentaje de muertes atribuidas a la intoxicación por Pb por países.

La Organización Mundial de la Salud ha incluido el plomo dentro de una lista de diez productos químicos causantes de graves problemas de salud pública, que exigen la intervención de los Estados Miembros para proteger la salud de los trabajadores, los niños y las mujeres en edad fecunda.

No todos los isótopos del Pb son estables y, a pesar de su inestabilidad, se encuentran presentes en la naturaleza, como es el caso del ^{210}Pb . Basándose en esto, se propone realizar mediciones de radiactividad de plomo con sensores CCD, con el fin de evaluar la posibilidad de realizar mediciones con sensores CMOS. En caso afirmativo, se podría crear dispositivos de bajo costo que sean capaz de estimar y monitorear niveles de Pb en agua, utilizando el software ya desarrollado.

4.2. Configuración Experimental

Para realizar dichas mediciones se utilizó un bloque de plomo, colocado frente del sensor CCD, en una cámara de vacío y enfriado con nitrógeno líquido. Además del Pb, se colocó una lámina de cobre tapando una de las mitades del sensor. Conociendo la energía de emisión de rayos X del Cu, es posible realizar una calibración del valor de ADC como función de la energía depositada.

En la Fig. 4.2 puede la disposición de las láminas de Cu y Pb sobre el sensor CCD.

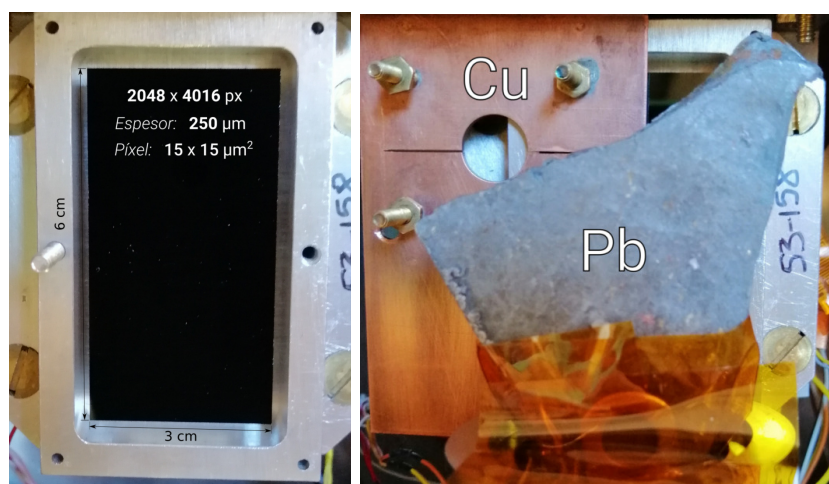


Figura 4.2: Sensor CCD Utilizado y disposición de las láminas de Cu y Pb sobre el sensor.

El sensor CCD utilizado tiene 3 cm de ancho y 6 cm de alto, presenta una resolución de 2048 columnas y 4016 filas, con un tamaño de pixel de 15 μm de lado. A diferencia de los sensores CMOS, los pixeles del CCD permiten una mayor proyección del movimiento de las partículas sobre el sensor, debido a su espesor. En este caso el espesor del sensor CCD es de 250 μm.

Se utilizaron dos tipos de configuraciones distintas, tal como lo muestran las Fig. 4.3. Una de ellas con una fuente radioactiva y otra sin la misma. La fuente radioactiva es

capaz de excitar los átomos de Pb y Cu, permitiendo que estos emitan por fluorescencia. Es por ello que se espera una señal mayor en el caso con fuente. En ambos casos se utilizó una bomba de vacío mecánica.

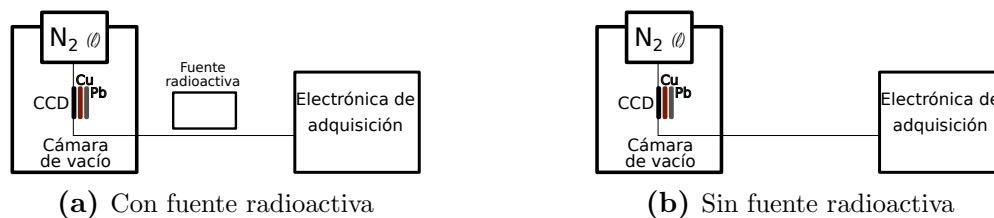


Figura 4.3: Configuraciones experimentales utilizadas

4.3. Resultados

4.3.1. Primera imagen del sensor

Como primera aproximación, se procedió a analizar una de las imágenes obtenidas. Esta imagen se obtuvo con un tiempo de exposición de 560 segundos y con la fuente radioactiva cerca. La imagen obtenida puede verse en la Fig. 4.4.

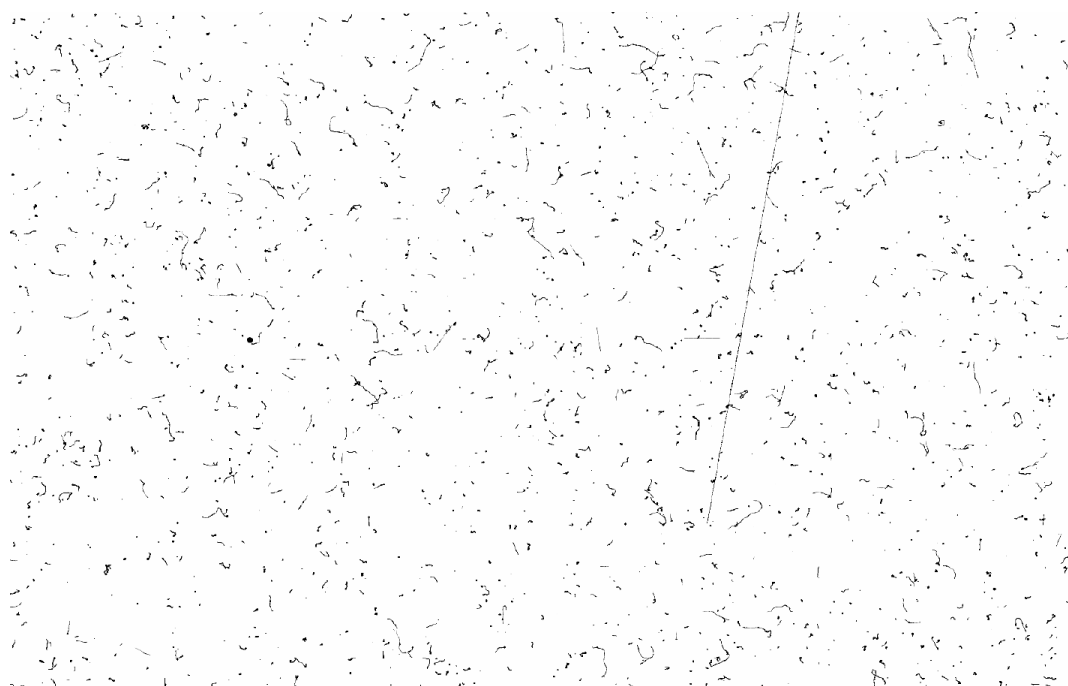


Figura 4.4: Primera imagen obtenida por el sensor CCD con un tiempo de exposición de 560 segundos, para la configuración experimental con la fuente.

En dicha imagen se pueden observar muones, electrones, partículas alfa y fotones. En lo siguiente analizaremos la distribución en el espectro energético de cada uno de los eventos. Para ello se segmentaron los eventos como se muestra en la Fig. 4.5 y se calculará la carga depositada como la suma de los valores de cada uno de los píxeles.

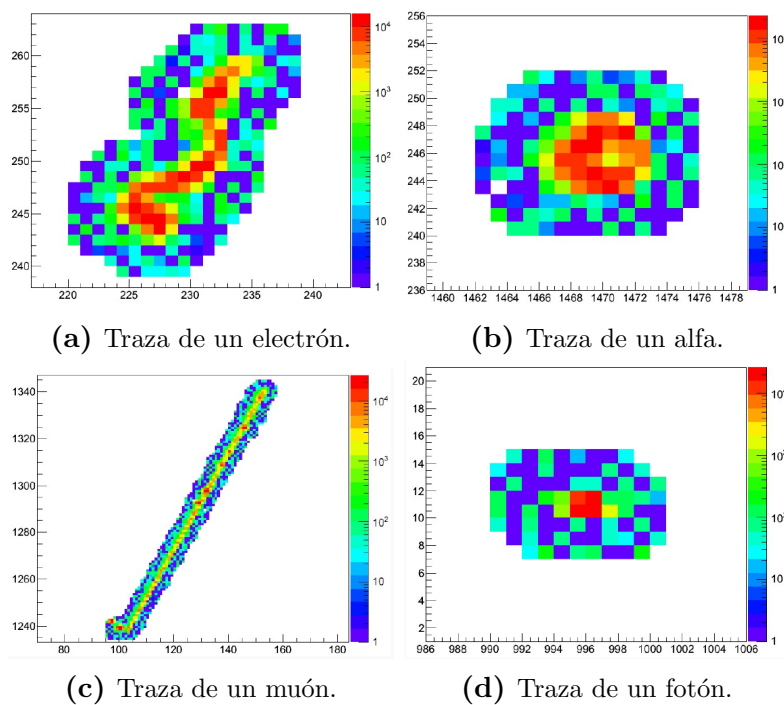


Figura 4.5: Segmentaciones de las trazas dadas por diversas partículas en el sensor CCD.

4.3.2. Medición con fuente

Se tomaron 141 imágenes con un tiempo de exposición de 560 segundos, por lo que el tiempo total de exposición fue de aproximadamente 22 horas. Para la obtención del espectro se realizó un histograma de las cargas depositadas luego de la segmentación de los eventos detectados. Con este histograma se identificaron los picos de Cu y, con ello, se pudo hacer una calibración entre el valor de ADC y la energía depositada. El resultado obtenido del espectro de energía depositada se puede observar en la Fig. 4.6.

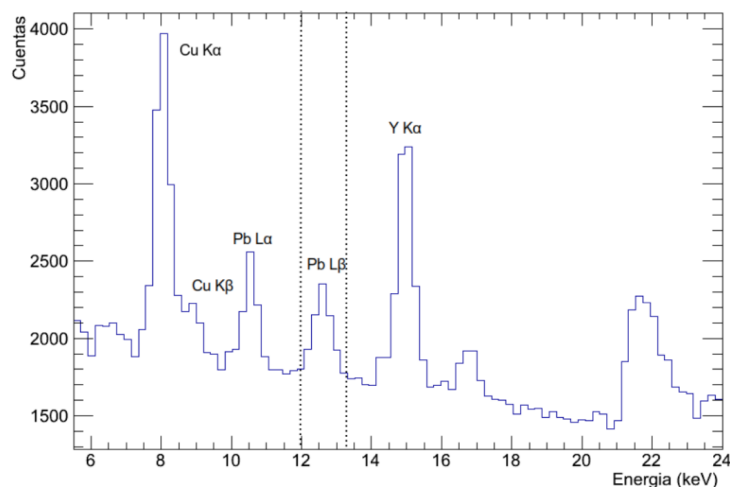


Figura 4.6: Espectro leído por el CCD entre 6 keV y 24 keV, con la configuración experimental con fuente radioactiva. Además se muestra el corte realizado para la identificación del Pb en la sección 4.3.3

4.3.3. Posición de los eventos de Pb

Una vez identificado los eventos del pico entre 12 keV y 13.5 keV se procedió a analizar la distribución espacial sobre el sensor de dichos eventos, realizando un histograma 2D, es decir, contando la cantidad de eventos con energías en dicho intervalo que caen en una sección de la imagen. El resultado obtenido se muestra en la Fig. 4.7.

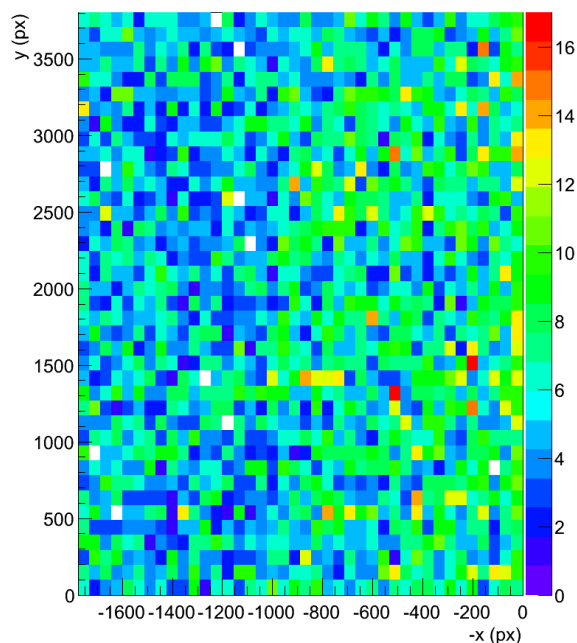


Figura 4.7: Histograma 2D de la distribución espacial de los eventos entre 12 keV y 13.5 keV sobre el sensor CCD, utilizando la configuración experimental con fuente radioactiva.

Teniendo en cuenta la configuración experimental dada en la Fig. 4.2, donde el Cu se encuentra del lado izquierdo y el Pb del lado derecho, se verifica que dichos eventos provienen de la emisión de rayos X del Pb.

4.3.4. Medición sin fuente

Utilizando un procedimiento análogo al descrito en la sección 4.3.2, pero con la configuración experimental sin fuente, se obtuvo el espectro de la Fig. 4.8. En este caso se tomaron 109 imágenes con un tiempo de exposición de 560 segundos cada una, lo que da un tiempo de exposición total de 17 horas.

En este caso se observa un decrecimiento del número de cuentas, esto se debe a la falta de la fuente radioactiva que emite partículas que provoquen fluorescencia en el Pb, además del menor tiempo de medición. Por lo que para obtener una buena estadística en el caso sin fuente, es necesario tener tiempos de medición más prolongados.

Sabiendo que los eventos de Pb provienen principalmente del lado derecho del sen-

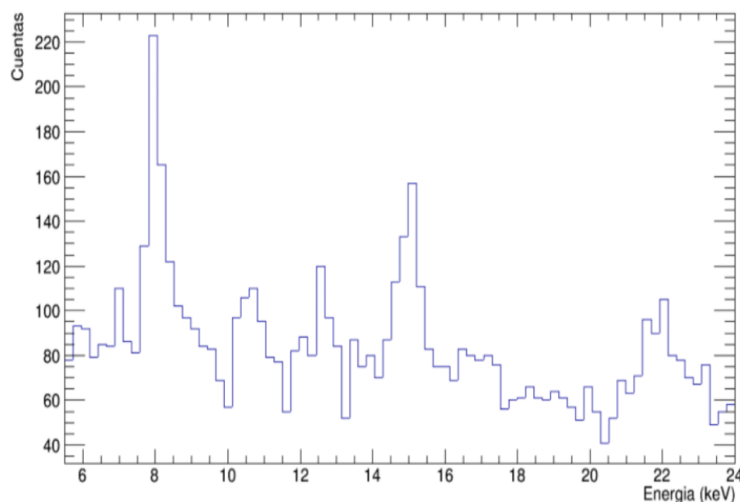


Figura 4.8: Espectro leído por el CCD entre 6 keV y 24 keV, sin la fuente radiactiva.

sor, se procedió a graficar un histograma de cargas de eventos que se encuentren en la mitad derecha del sensor. El resultado obtenido se muestra en la Fig.4.9.

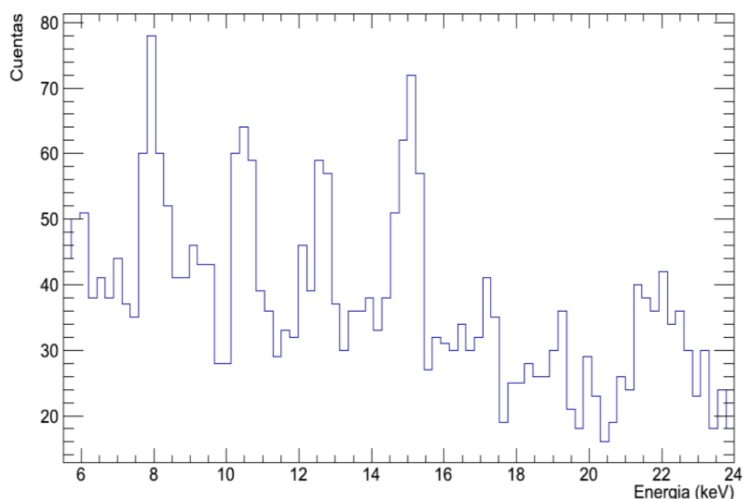


Figura 4.9: Espectro leído por el CCD entre 6 keV y 24 keV, con eventos provenientes de la mitad derecha del sensor y sin la fuente radiactiva.

En este gráfico se puede ver un aumento de la relación señal-ruido para los picos de 10.5 keV y 12.5 keV, picos que están asociados a las transiciones L_α y L_β del Pb [16].

Al integrar los picos, considerando una línea base, es posible obtener la cantidad de eventos correspondientes a cada pico, se encontraron 17 ± 7 eventos para el pico L_α y 13 ± 8 para el L_β . Normalizando en el área del sensor y el tiempo de exposición, se obtiene $0.3(1)$ eventos/ m^2/s y $0.2(1)$ eventos/ m^2/s , respectivamente. Teniendo en cuenta la superficie del sensor OV5647, descrito en la sección 2.1.1, se estiman aproximadamente del orden de 1 evento cada 100 horas para el caso del L_α . En dichas estimaciones no se tuvo en cuenta el ángulo sólido subtendido debido a la distancia del sensor al bloque de Pb.

4.4. Mediciones con sensores CMOS

Como se vio anteriormente, se pudo identificar correctamente el Pb gracias al sensor CCD. Si bien el número de eventos aumenta con la presencia de la fuente radioactiva; aún sin esta se observan picos de emisión del Pb.

4.4.1. Configuración Experimental

En base a esto, se decidió adaptar dicho experimento para realizar la medición con los sensores CMOS. Para ello se utilizó una *Raspberry Pi 1*, y el sensor OV5647 descrito en la sección 2.1.1. La configuración experimental utilizada se muestra en la Fig. 4.10



Figura 4.10: Configuración experimental para la medición de Pb con un sensor CMOS.

Tanto el sensor como el ladrillo de Pb se cubrieron con un material opaco (*blackout*) para evitar el ingreso de luz. Todo este sistema se encontraba cubierto en una caja metálica.

Utilizando el software *Radmonpi*, descrito en la sección 2.3, se generó la máscara de píxeles defectuosos correspondiente.

4.4.2. Medición sin Pb

Como parámetro de control, se expuso el sensor durante 56 horas sin presencia del ladrillo de Pb. Durante este tiempo se encontraron 5 eventos. Estos 5 eventos pueden verse en la Fig. 4.11

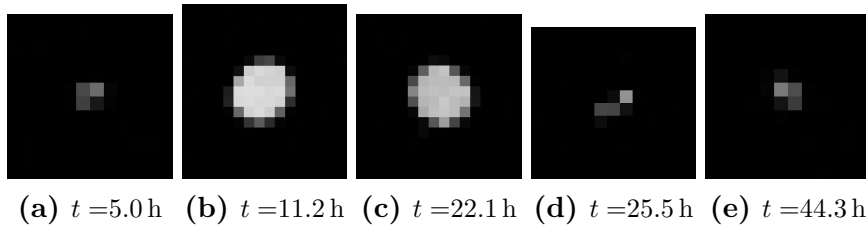


Figura 4.11: Eventos detectados en vacío, sin el ladrillo de Pb, durante 56 horas de exposición. En cada caso se indica el tiempo encontrado del evento. El valor blanco corresponde a un valor de ADC de 1023, mientras que el negro a un valor nulo.

El primer evento podría asociarse a un fotón; mientras que los dos eventos posteriores corresponden a partículas alfa. Los últimos dos eventos podrían ser asociados a electrones o muones.

4.4.3. Medición con Pb

De forma similar se expuso el sensor, esta vez apoyado sobre el ladrillo de Pb, durante 88 horas. Durante ese período de tiempo, se encontraron 4 eventos. Todos los eventos mostrados se muestran en la Fig. 4.12

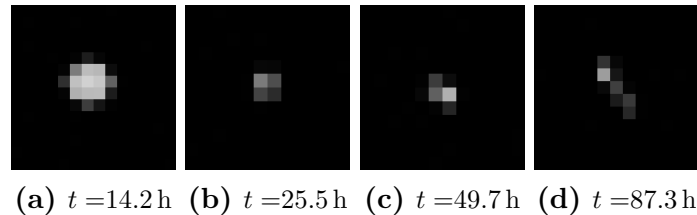


Figura 4.12: Eventos detectados con el sensor apoyado sobre el ladrillo de Pb, durante 88 horas de exposición. En cada caso se indica el tiempo de encontrado del evento. El valor blanco corresponde a un valor de ADC de 1023, mientras que el negro a un valor nulo.

En este caso el primer evento detectado corresponde a una partícula alfa, mientras que el último evento puede asignarse a un muón con un gran ángulo de incidencia.

4.5. Conclusiones

A partir de los resultados anteriores se puede observar que sí es posible detectar la presencia de un bloque de Pb con un sensor CCD, mientras que, para los tiempos de exposición dados, no es posible detectarlo con un sensor CMOS.

En ambos casos resulta inviable la medición para la detección de pocas partes por billón de Pb que determinan el umbral de toxicidad en el caso del agua, tanto para los sensores CMOS como para los sensores CCD.

En la sección 6.3 se discute la posibilidad de construcción de un monitor del nivel de Pb en agua, basándose en bacterias manipuladas genéticamente.

Capítulo 5

Reconocimiento de eventos con técnicas de Machine Learning

“Maybe, just maybe, there is no purpose in life... but if you linger a while longer in this world, you might discover something of value in it.”

— Orochimaru, Naruto

5.1. Introducción y motivación

El desarrollo de la inteligencia artificial se encuentra en su época dorada. Si bien hace un par de décadas se desarrollaron las bases teóricas para esta tecnología, no fue hasta hace poco que se desarrollaron ordenadores lo suficientemente potentes para poder utilizar algoritmos de aprendizaje artificial. Gracias al desarrollo de placas especializadas en cómputo en paralelo, se decrementaron drásticamente los tiempos de ejecución de estos algoritmos, haciendo posible su uso y masificación.

En ciertos experimentos como los llevados a cabo en CONNIE[5], para la búsqueda de neutrinos, se presentan eventos de baja intensidad que pueden ser fácilmente confundidos con ruidos de lectura. En estos casos la energía leído por los CCDs utilizados se mide en electrones (donde cada 3.7 eV equivale a un electrón) y el ancho típico de los eventos suele ser de fracciones de píxeles.

5.1.1. Usos y motivación

Existen diversos tipos de redes neuronales, y cada una de ellas es especializada para determinada tareas. Las redes recurrentes son buenas para analizar fragmentos de audios[17], los mapas de Kohonen son útiles para resolver problemas de clusterización[18], las redes de Hopfield son capaces de almacenar información accesible por

contenido[19], y las redes convolucionales son útiles para el análisis de imágenes. Es por ello que, en el presente trabajo, se utilizaron redes neuronales de este último tipo.

Las imágenes tomadas por los sensores de imágenes, ya sean CMOS o CCD, presentan un ruido de lectura. Es por ello que, al tomar una fotografía con cualquier dispositivo, se realizan diversas técnicas de procesamiento que permiten, en cierta medida, disminuir el ruido de lectura y obtener una imagen más nítida. En el caso de la detección de partículas, el ruido de lectura se traduce como una incertidumbre en la determinación de la carga depositada en el evento, y, en el peor de los casos, como un falso positivo. Muchas veces se ajustan ciertos valores umbrales de carga para filtrar eventos, este método puede ser muy eficiente cuando la carga del eventos es grande en comparación con el ruido de lectura.

Pero existen ciertos casos en los que la carga del evento es comparable con el ruido de lectura, y al aplicar cortes en carga detectada, existe un compromiso entre perder señal y quedarse con un mayor grado de pureza, o quedarse con casi toda la señal y filtrar poco ruido. Encontrar un algoritmo inteligente que permita maximizar la señal y minimizar el ruido puede llegar a ser un problema difícil de resolver.

En este sentido, las redes neuronales convolucionales están especializadas para el problema de clasificación de imágenes. Ya sea en clasificar imágenes de perros o gatos, cómo reconocer números escritos a mano[20]. En nuestro caso, es necesario desarrollar una red neuronal que nos permita separar eventos de ruido.

Uno de los principales inconvenientes, es que estas redes neuronales de aprendizaje supervisado requieren una gran cantidad de datos, que debe ser, como mínimo, mayor a la cantidad de parámetros que presenta la red. Cuidar la cantidad de parámetros libres que posee una red es útil para evitar situaciones de *overfitting*.

5.1.2. Principios de Inteligencia Artificial

El concepto de redes neuronales artificiales surge a partir de simplificaciones realizadas a los modelos matemáticos que permiten predecir el comportamiento de neuronas biológicas. El sistema de ecuaciones 5.1 representa una neurona de Hodgkin-Huxley.

$$\begin{aligned}
 C \frac{dV}{dt} &= I - g_{\text{Na}} m^3 h (V - V_{\text{Na}}) - g_{\text{K}} n^4 (V - V_{\text{K}}) - g_{\text{Cl}} (V - V_{\text{Cl}}) \\
 \frac{dm}{dt} &= (m_{\infty}(V) - m) / \tau_m(V) \\
 \frac{dh}{dt} &= (h_{\infty}(V) - h) / \tau_h(V) \\
 \frac{dn}{dt} &= (n_{\infty}(V) - n) / \tau_n(V),
 \end{aligned} \tag{5.1}$$

donde V es diferencia de potencial entre el interior y el exterior de la neurona, I es una corriente externa (variable de entrada), g_x son constantes con unidades de

conductividad, que representan la permeabilidad del ion x al paso de la membrana y V_x son los potenciales de referencia para los iones x . A su vez, se definen las funciones m , h y n , que cada una de ellas simboliza a una compuerta que puede activarse o inhibirse para el paso de iones.

Como se puede ver en el término $-g_{\text{Na}}m^3h(V - V_{\text{Na}})$ existen 3 compuertas del tipo m y una compuerta del tipo h . El pasaje de iones por estas compuertas depende del potencial de la neurona y del tiempo, tal como se muestra en la ecuación 5.1. La forma funcional de $x_\infty(V)$ y $\tau_x(V)$, pueden obtenerse experimentalmente para las tres compuertas m , h y n .

Este sistema de ecuaciones diferenciales verifica los fenómenos de *spikes* que presentan las neuronas. Los *spikes* son fluctuaciones rápidas en el potencial intermembranoso, en los cuales se dice que la neurona se encuentra activada. Además, las ecuaciones predicen la existencia de un periodo refractario, es decir un periodo en el que, ocurrido un *spike*, no puede encontrarse otro *spike*.

Esta visión de la neurona como un sistema *spike-no spike*, permite hacer una discretización en el tiempo y definir la actividad de una neurona en base a la frecuencia con la que emite estos *spikes*.

Al realizar interconexiones entre neuronas, serán estos *spikes* con los que excitarán, o inhibirán, a las siguientes neuronas.

5.1.3. El perceptrón

El modelo más simple de redes multicapa que puede emplearse es el perceptrón. Este puede ser considerado como una única neurona, que recibe los estímulos de todas las demás, tal como se muestra en la Fig. 5.1.

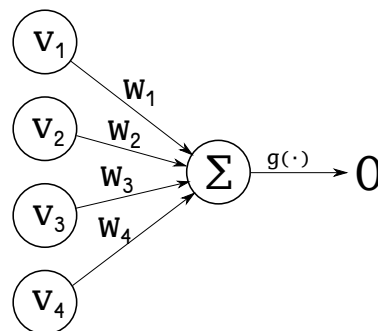


Figura 5.1: Representación esquemática de un perceptrón, con 4 neuronas de entrada y una función de activación $g(\cdot)$

Cada una de estas entradas en el perceptrón, está pesada por un cierto peso ω_i , que define cuán excitatorio o inhibitorio es el estímulo que provee dicha neurona. El valor del perceptrón estará dado por la superposición de las contribuciones de cada una de

las neuronas. Por último, es definir una función de activación monótona creciente $g(x)$, que represente el comportamiento de la neurona.

$$O = g(\vec{V} \cdot \vec{\omega}) \quad (5.2)$$

Para problemas de clasificación binaria, en donde se busca que la función de activación sean dos valores extremos (como 0 y 1, ó -1 y 1), el perceptrón es incapaz de resolver problemas que no son linealmente separables. Es decir, existirá el hiperplano $\vec{V} \cdot \vec{\omega} = 0$ que separe en dos el espacio de parámetros. Uno de estos semi-espacios tendrá un valor $\vec{V} \cdot \vec{\omega}$ positivo, y otro negativo.

5.1.4. Modelo multicapa

Ante la imposibilidad del perceptrón de resolver problemas no-linealmente separables, es que se construyó el modelo multicapa. En este caso, la entrada se propaga a distintas capas intermedias. Cada neurona intermedia actúa de manera equivalente a un perceptrón, recibiendo los estímulos de las neuronas de la capa anterior. La Fig. 5.2 representa el esquema típico de una red multicapa densa.

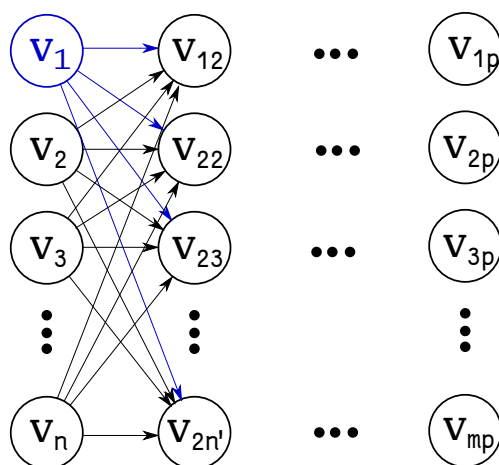


Figura 5.2: Representación esquemática de una red neuronal multicapa densa, con n entradas, p capas y m salidas. Con azul se encuentra señalados los pesos de la primera neurona a las neuronas de la segunda capa.

No existe un algoritmo o estándar para determinar la cantidad de capas, ni la cantidad de neuronas en cada capa. Al momento de entrenar una red neuronal artificial se busca que se satisfaga el problema de capacidad, es decir, que la red no sea lo suficientemente simple como para no ser capaz de aprender el patrón subyacente; y el problema de aprendizaje, es decir, que la red sea capaz de predecir correctamente cada unas de las salidas para todo el conjunto de entradas dadas en el entrenamiento, pero sin aprender el patrón subyacente, siendo incapaz de predecir correctamente un estímulo diferente a los entrenados (situación de *overfitting*).

5.1.5. Redes convolucionales

El desarrollo de las redes convolucionales está basado, desde un punto de vista biológico, en el procesamiento que realiza la corteza visual primaria. Las capas convolucionales están diseñadas para reconocer estructuras bidimensionales con un alto grado de invarianza traslacional, de escala y otras formas de distorsión. Este tipo de redes toman como peso un núcleo (matriz bidimensional) de menor tamaño, para obtener como salida el producto de convolución entre el núcleo y la entrada. También es usual, a la salida, aplicar una función de activación no lineal.

Además de ajustar los parámetros del núcleo, se especifica el número n de características a determinar, es decir, el número de núcleos a definir. El resultado final son n matrices bidimensionales, cada una obtenida con un núcleo diferente.

En general, se suelen aplicar capas de *downsampling* luego de una capa convolucional, ya sea mediante capas de máximo, valor medio o mínimo. Con esto se disminuye la cantidad de parámetros, simplificando computacionalmente las entradas a las capas siguientes, y se asegura una pequeña invarianza traslacional.

5.1.6. Algoritmo de *backpropagation* y métricas a definir

Como ya hemos establecido, las redes neuronales permiten ajustar funciones de varias variables, que pueden ser muy complejas, a partir de un cierto conjunto de datos. Para esto, presentan una gran cantidad de parámetros que deben ser ajustados convenientemente. El ajuste de estos parámetros se hace mediante el algoritmo de *backpropagation*.

El algoritmo de *backpropagation* consiste en la minimización de una función de costo, a la que llamaremos error. Para ello se presenta una de las entradas (llámese n) a la red y se calcula la salida \vec{O}_n . Para eso se calcula los valores de las neuronas capa por capa, mediante la siguiente expresión:

$$V_i = g \left(\sum_j \omega_{ij} V_j - \omega_{i0} \right) \quad (5.3)$$

donde $g(\cdot)$ es la función de activación de la neurona, V_i es el valor de la neurona i y ω_{ij} define el peso entre la conexión de la neurona j y la neurona i , ω_{i0} es el valor umbral que también representa un parámetro ajustable. Una vez obtenida la salida, se calcula el error mediante la siguiente expresión:

$$\xi_n = \frac{1}{2} \left| \vec{O}_n - \vec{\zeta}_n \right|^2 \quad (5.4)$$

donde ξ_n es el error, O_n es la salida calculada y ζ_n es la salida deseada, todas correspondientes a la entrada n . En base a esto, también es posible definir el costo energético

cuadrático medio como el promedio de los errores, en todos los casos. Esta representará la función de costo del sistema, la función a minimizar.

Una vez calculado el error, se calcula la corrección de los pesos $\Delta\omega_{ij}$, siguiendo la siguiente expresión.

$$\Delta\omega_{PQ} = \eta\delta_P V_Q \quad (5.5)$$

donde los deltas se calculan mediante la fórmula de recursión

$$\delta_p = \begin{cases} g' \left(\sum_j \omega_{ij} V_j \right) (O_p - \zeta_p) & \text{si es de la última capa} \\ g' \left(\sum_j \omega_{ij} V_j \right) \sum_p \delta_i J_{ip} & \text{si no es de la última capa} \end{cases} \quad (5.6)$$

Se define una época al aplicar las correcciones para todos las entradas de entrenamiento. La convergencia a un mínimo local está garantizada para valores pequeños de η , el mínimo absoluto se encuentra cuando todos los valores de salida son los deseados. El parámetro η actúa como constante de proporcionalidad a las correcciones, por lo que un valor pequeño de η convergerá lentamente al mínimo, y con valores grandes de η no se puede asegurar su convergencia.

Además de la función de costo, en problemas de clasificación binaria, se suele definir la precisión como el cociente entre la cantidad de aciertos, sean de una clasificación o la otra, y la cantidad total de ejemplos. En nuestro caso también definiremos precisiones por categorías, es decir, una precisión que sólo tenga ejemplos de una única categoría.

5.2. Simulación de los datos de entrada, eventos y ruidos

En general, las redes neuronales requieren una gran cantidad de datos de entrada para poder resolver el problema de aprendizaje, es decir, que ante entradas no presentadas a la red, esta prediga correctamente el valor de salida.

En caso de tener la cantidad de datos suficientes, existen diversas técnicas que permiten obtener nuevos datos de entradas a partir de otros, deformándolos ligeramente[21]. En el problema de separación de eventos de ruidos, es posible realizar simulaciones numéricas que nos permita tener una amplia cantidad de imágenes simuladas.

5.2.1. Simulación de los eventos

Se utilizaron imágenes cuadradas de 5 píxeles de lado. A cada uno de estos píxeles se asignó un valor aleatorio, siguiendo una distribución gaussiana con media nula y dispersión de 2, simulando un ruido de 2 electrones que podría presentar un CCD.

Con esta imagen de base, se consideraron E unidades (a las que, por simplicidad llamaremos electrones). Estos electrones se suman a la imagen, siguiendo una distribución gaussiana de ancho 0.6 centrada sobre el centro de la imagen, para simular las condiciones de los sensores CCDs de CONNIE. Para esto se generaron E pares de números aleatorios, con la distribución mencionada anteriormente. Sólo se utilizó la parte entera de estos números, creando así un mapa de índices. Se aumentó en 1 el valor de la imagen en cada uno de estos índices. En la Fig. 5.3 puede verse una simulación construida con el procedimiento descripto.

5.2.2. Simulación del ruido y función de validación

En general, en la detección de partículas, se espera que las imágenes dadas por los sensores tenga una línea de base fija en el tiempo. Los algoritmos más simples de detección de eventos, buscan puntos que estén por encima o por debajo de esa línea de base, con un cierto umbral fijo. Esta pre-selección de los eventos permite que no se analice la totalidad de la imagen, sino ciertas regiones de interés. Simular ruido de estas regiones de interés puede ser complicado ya que dependerá del método utilizado para la pre-selección de eventos. A continuación, se describe el procedimiento empleado para la simulación de eventos de ruido.

Se generó una imagen de prueba con el ruido gaussiano anteriormente descripto. A dicha imagen se le aplica una función booleana que permite decidir si la imagen es válida o no. La función de validación empleada fue que la suma de los 9 pixeles centrales esté comprendida entre $0,7E$ y $1,3E$. En caso de no pasar la función de validación, se genera otra imagen de prueba y se repite el procedimiento.

Para valores grandes de E , la cantidad de imágenes de prueba fallidas crece exponencialmente. Por lo que este procedimiento no es conveniente para valores de E grandes.

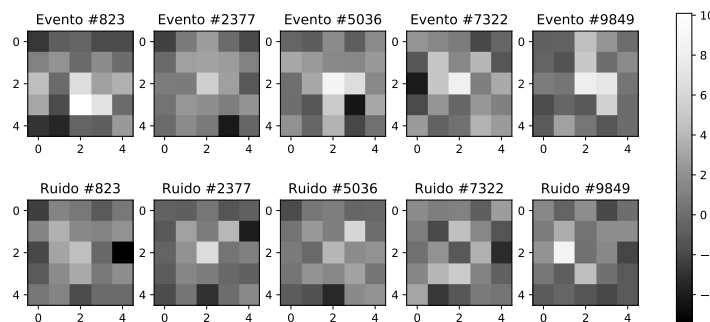


Figura 5.3: Eventos y ruidos simulados para una energía $E=27$ e. Se representan 5 de cada uno, elegidos aleatoriamente entre 10000 datos simulados. La carga central está comprendida entre 19 e y 35 e.

5.3. Descripción de la red neuronal empleada

Teniendo en cuenta que las imágenes utilizadas son monocromáticas (escalares), y el reducido tamaño de las imágenes es que se decidió utilizar una arquitectura de red simple. Esta red consiste en una primera capa convolucional bidimensional. Cuyo núcleo es de 3×3 y mapea 5 características. Una capa de alisamiento, que permite vincular la salida bidimensional con la entrada a una última capa densa de 12 neuronas. Estas 12 unidades están conectadas con una única neurona que representa la salida.

Exceptuando la última, todas las capas presentan una función de activación *relu* ($f(x) = \max(0, x)$). La última capa presenta una activación sigmoide; cuyo mapeo será 0, en caso de ser detectado como ruido, o 1, en caso de ser detectado como evento.

La red neuronal utilizada se puede resumir en la tabla 5.1.

Tabla 5.1: Descripción de la red neuronal empleada

Capa (tipo)	Formato de salida	#Parámetros	Activación
Entrada	(5, 5, 1)		
Convolucional	(3, 3, 5)	50	relu
Alisamiento	(45)	0	identidad
Densa	(12)	552	relu
Salida	(1)	13	sigmoide

Para programar dicha red se utilizó la librería *Keras* [22] con soporte en *Tensorflow*, utilizando la plataforma de *Google Colab* [23] que permite conectarse a un entorno de ejecución remoto de *Jupyter Notebook* de forma gratuita.

5.4. Resultados

5.4.1. Distribución energética de la simulación para eventos de $27 e^-$

En primer lugar, se simularon 10000 eventos de $E = 27 e$ ($\approx 100 eV$ en un CCD de CONNIE) y 10000 ruidos. La Fig. 5.3 se muestran 5 imágenes aleatorias de eventos, y 5 de ruidos, de las 10000 simulaciones que se hicieron para cada caso. En base a esto se analizó la distribución energética del conjunto de eventos y del conjunto de ruidos. El espectro energético de ambos puede observarse en la Fig. 5.4.

Se observa que la distribución de ruidos no está centrada en 27, sino alrededor el 70 % de 27, debido a la pre-selección de ruidos definida en la sección 5.2.2.

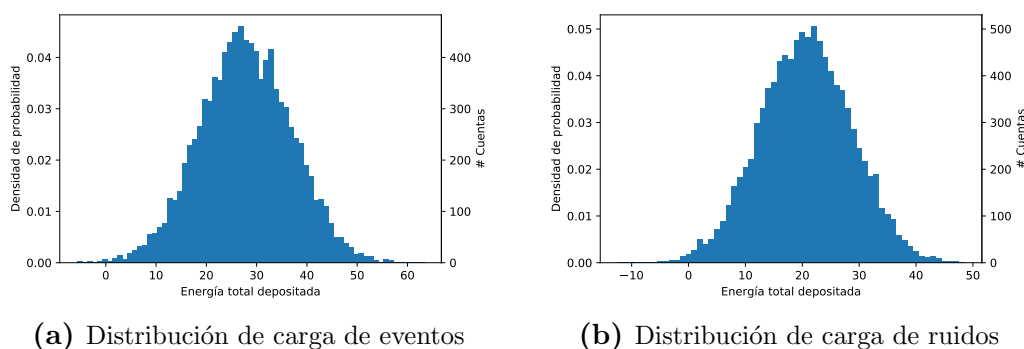


Figura 5.4: Histograma de cargas de los conjuntos simulados para $E=27 e^-$.

5.4.2. Entrenamiento de la red para eventos de $27 e^-$

Con el conjunto de los 10000 eventos y 10000 ruidos simulados en la etapa anterior, se entrenó la red neuronal descrita en la sección 5.3, utilizando el 30% de los datos como conjunto de validación. En la Fig. 5.5 se muestra la precisión del conjunto de validación y del conjunto de entrenamiento como función de la época. La precisión lograda con esta red neuronal y con el conjunto de datos dado es del $(96 \pm 1) \%$.

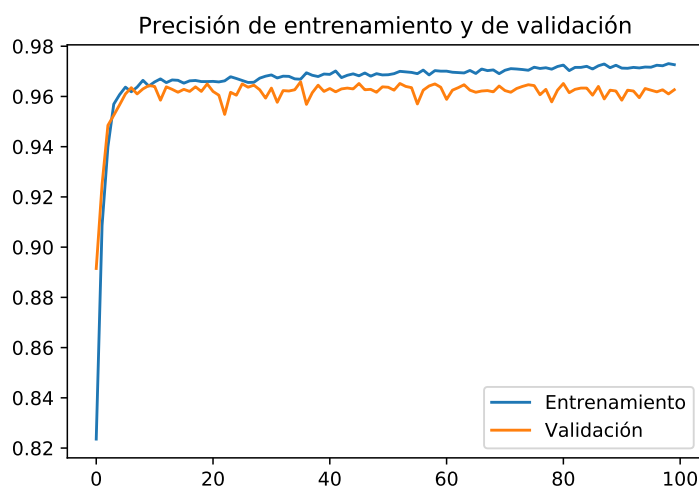


Figura 5.5: Precisión para el conjunto de entrenamiento y validación como función de la época

En base a lo anterior, se procedió a analizar los eventos y ruidos que fueron mal clasificados por la red neuronal. Para ello, en primer lugar, se graficaron 5 elementos de cada conjunto, elegidos aleatoriamente, tal como se muestra en la Fig. 5.6.

A partir de esto, se analizó la distribución de la carga total y de la carga central para los eventos y ruidos mal clasificados. La Fig. 5.7 muestra la distribución de la carga total para los eventos mal clasificados; mientras que la Fig. 5.8 muestra la de la carga central.

También se calcularon las densidades de probabilidades para cada uno de los *bines* como el cociente entre la cantidad de eventos (resp. ruidos) mal clasificados y el número

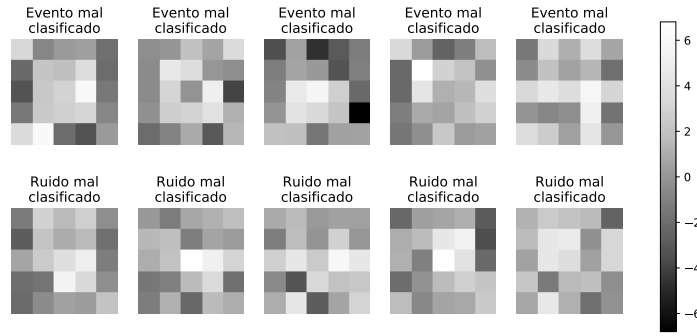


Figura 5.6: 5 eventos (resp. ruidos) tomados al azar, clasificados como ruidos (resp. eventos) por la red neuronal.

total de eventos (resp. ruidos) con esas energías.

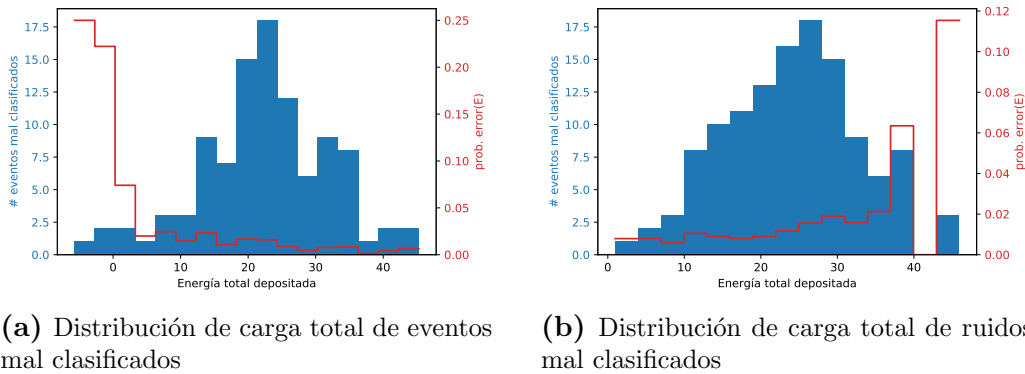


Figura 5.7: Histograma de cargas totales de las entradas mal clasificados de los conjuntos simulados para $E=27e$, con sus respectivas densidades de probabilidades.

Se observa que es más probable que, para un evento, la red neuronal disminuya su precisión cuando los eventos tienen menor energía. Respectivamente, para los ruidos, es más probable que la red neuronal los clasifique de manera incorrecta, si estos presentan una mayor carga. Esto se justifica ya que, en los datos de entrenamiento, al tener una mayor carga, es más probable que éste sea un evento.

Tanto para la carga central como para la carga total, se puede observar que ambas gráficas (para eventos y ruidos mal clasificados) presentan una forma funcional similar. Esto se justifica ya que, en caso de ser gráficas muy diferentes, la red neuronal sería capaz de minimizar la función de costo al cambiar la predicción en función de la energía. Dicho de otra forma, una manera de minimizar la función de costo es hacer que ambas gráficas sean parecidas.

5.4.3. Entrenamiento de la red para distintos valores de E

En este caso, analizaremos la precisión que puede lograr la red neuronal para distintos conjuntos de datos con distintos valores de E . Para cada uno de estos conjuntos de

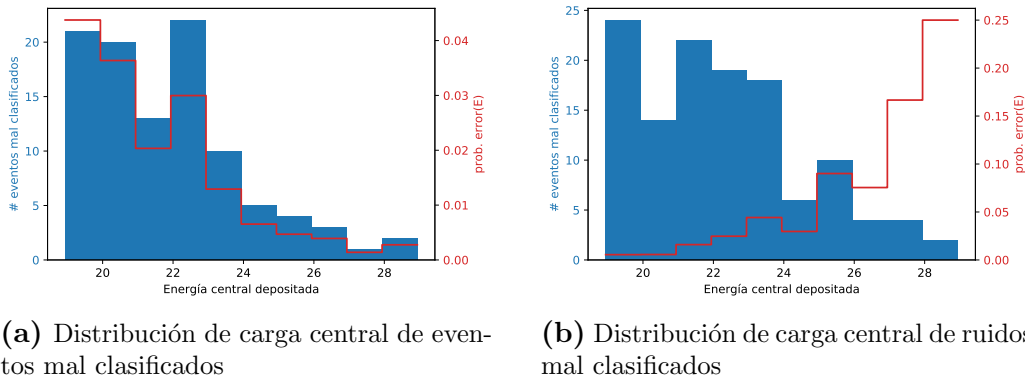


Figura 5.8: Histograma de cargas centrales de las entradas mal clasificados de los conjuntos simulados para $E=27e$, con sus respectivas densidades de probabilidades.

datos se generó una nueva red neuronal, de tal forma que la red neuronal fue entrenada con un único conjunto en cuestión, con una misma energía E . En todos los casos se consideró al conjunto de entrenamiento como el 30% de los datos de entrada. Cada conjunto de entrada poseía 10000 eventos y 10000 ruidos.

La Fig. 5.9 muestra la precisión como función de la energía. Además se representan los cocientes entre los eventos (resp. ruidos) mal clasificados y la cantidad total de eventos (resp. ruidos).

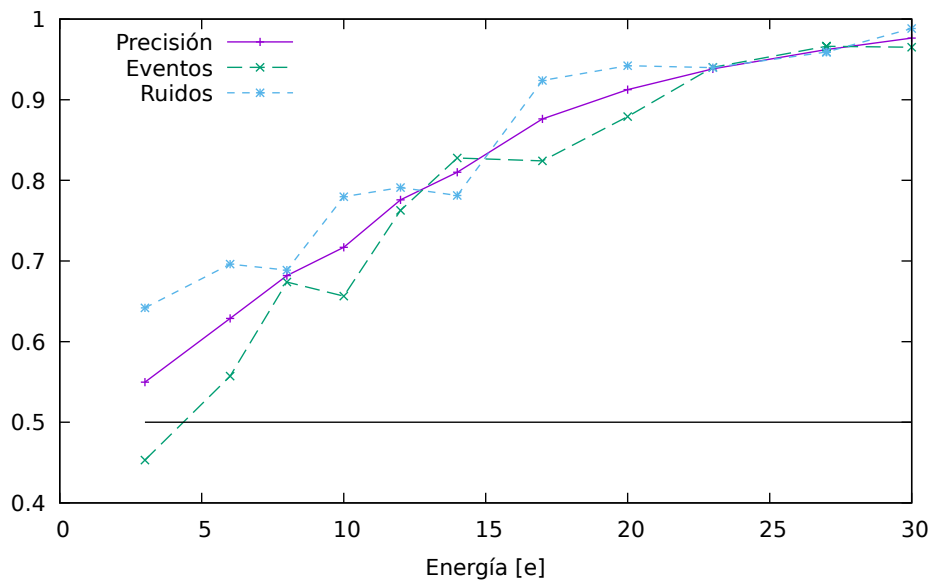


Figura 5.9: Precisión obtenida en la validación como función de la energía del conjunto de entrenamiento. Además se muestra la fracción de eventos y los ruidos de validación bien clasificados.

Se puede observar que la red neuronal presenta una precisión de más de 0.9 para la clasificación de eventos mayores a 20 e (74 eV), esto determina la capacidad máxima de separación la red neuronal dada.

5.5. Conclusiones

Se pudo observar que la red neuronal empleada resuelve satisfactoriamente el problema de clasificación de separación de eventos sobre ruidos, con una precisión mayor a 0.9 para energías mayores a $20 e^-$, cuando el ruido de lectura de cada pixel individual es de $2 e^-$ y se preseleccionan candidatos a eventos con el algoritmo descrito en la sección 5.2.

En el mismo caso, se observa que el algoritmo es más ineficiente al clasificar ruidos que presentan una energía central alta. Análogamente, cuando la energía central es baja, presenta menor precisión al clasificar eventos.

Si bien los entrenamientos de estas redes neuronales son complejos ya que se busca la convergencia mediante un proceso iterativo, evaluar el resultado predicho por una red neuronal es más simple y rápido ya que involucra operaciones matriciales, con matrices de pocos elementos. Por lo que una red especializada en detectar cierto tipo de eventos (de un ancho y energía definida), dada la pre-selección, podrá clasificarlos rápidamente.

Capítulo 6

Conclusiones, palabras finales y trabajos futuros

“You only live once, but if you do it right, once is enough”
— Mae West (1893-1980), actriz

Como se ha visto, la investigación en el área de detección de partículas con sensores de imágenes CMOS y CCD requiere múltiples metodologías. Del lado del software, se desarrolló un código en lenguaje C que permite obtener y analizar las imágenes *raw* obtenidas por el sensor, como puede verse en el capítulo 2.3. Desde el desarrollo de hardware, se fabricó un monitor del nivel de actividad de R_n , utilizando un sensor CMOS, un soporte impreso, cinta conductora y una SBC, como puede verse en el capítulo 3. Desde un punto de vista numérico, se realizaron simulaciones de eventos de baja intensidad y simulaciones de ruidos que podrían llegar a confundirse con estos eventos de baja intensidad por algunos algoritmos (sección 5.2). Con este conjunto de datos simulados se entrenó una red neuronal que permitió separar eventos de ruidos correctamente, además se estudió el poder de clasificación de la red para distintos valores de energía del conjunto de entrenamiento.

6.1. Software empleado

Se comprobó el correcto funcionamiento del código *Radmonpi* para la detección de partículas. Además, agregando optimizaciones a la hora de la generación del ejecutable se observó un mejor rendimiento. Este aumentado de rendimiento permitiría disminuir los tiempos de exposición del sensor, permitiendo que el sensor pueda estar expuestos a ambientes con menor oscuridad.

Del lado de la interfaz web, se observó que el procesamiento del lado del servidor no es conveniente en una SBC de baja potencia, y en base a ello se implementó una interfaz web modular que fue utilizada en la medición del flujo de muones en la medición

llevada a cabo en una mina de Sierra Grande, tal como se muestra en el apéndice B.

6.2. Medición de Rn

Se pudieron observar eventos de partículas alfa debido a la deposición de iones sobre el sensor, gracias al concentrador de iones desarrollado a tal fin. Se contrastaron las mediciones de la cantidad de partículas alfa detectadas con el nivel de actividad de Rn medido por el dispositivo *Wave*, desarrollado por Airthings. Se observó cierta inercia en la medición lleva a cabo por dicho medidor, esta inercia provocó un desfase en la respuesta temporal de ambas señales.

Además se observaron falsos contactos entre los recortes de cinta aluminio utilizadas para las distintas etapas de aceleración. Estos falsos contactos no permitieron el correcto funcionamiento del medidor para ser utilizado durante el experimento llevado a cabo en Sierra Grande.

Aún con un correcto contacto, se observó que la distribución de eventos sobre el sensor no estaba correctamente centrada. Esto se debe, principalmente, a que el sensor no se encontraba centrado en el soporte, debido al diseño del mismo soporte.

6.3. Medición de Pb en Agua

Para la detección de un bloque de Pb, se observó que los sensores CMOS son ineficientes, mientras que sí fue posible la identificación del Pb con un sensor CCD, con el bloque colocado frente al sensor. Aún así, se concluye que no es posible realizar la medición para la detección de cantidades de plomo en partes por billón.

Debido a que la contaminación de Pb en agua constituye una problemática de salud a nivel mundial, y de allí el interés de fabricación de un instrumento de monitoreo, es que se propone utilizar biosensores modificados genéticamente, que sean capaces de interactúan con el Pb agua para producir algún color específico [24]. Mediante esta técnica, sería posible utilizar sensores CMOS para cuantificar la intensidad del color que produzcan dichas bacterias.

6.4. Técnicas de ML aplicadas al análisis de eventos

Se observó que las redes neuronales convoluciones permiten resolver problemas de clasificación de evento-ruido. Para ello se simulaban conjuntos de datos de eventos de diferentes energías y conjuntos de ruidos pre-seleccionado como eventos, con un criterio definido en la sección 5.2.2.

La red neuronal utilizada consistió en una capa convolucional y una capa densa, debido a la simplicidad del problema a resolver. Con esta arquitectura de red neuronal se lograron precisiones mayores a 0.9, cuando la energía de los eventos es mayor que 20 e, cuando la imagen presenta un ruido de fondo de 2 e y las cargas están distribuidas con un ancho de 0,6 pixeles.

También se analizó la distribución energética de los eventos mal clasificados, observando que la red neuronal es menos eficiente para detectar eventos de baja energía o ruidos de alta energía.

Apéndice A

Utilización de las QPU

A.1. Descripción y motivación

Los procesadores integrados en la *Raspberry Pi 3b+* representan un procesador gráfico integrado *Broadcom VideoCore IV*. Estas presentan unas unidades de procesamiento vectorial que operan sobre 16 valores de 32 bits, estas unidades se denominan *Quad Processor Unit* (QPU). Teniendo 12 QPUs funcionando a 250 MHz, la *Raspberry Pi* posee un rendimiento teórico de 12 GFlops. En algunos casos las QPU pueden dar dos resultados en simultaneo, por lo que su procesamiento teórico máximo aumenta a 24 GFlops.

Para hacer uso de las QPU a la hora de analizar las imágenes tomadas, se utilizó la librería *QPULib* desarrollada por Matthew Naylor. La misma define un conjunto de clases, funciones y macros que permiten realizar cálculos en paralelo mediante el procesador gráfico. Para ello define estructuras básicas paralelizadas como *If*, *For* y *While*, y vectores almacenados en la memoria gráfica.

Mejorar el rendimiento del procesamiento permite disminuir el tiempo de exposición, permitiendo que se puedan hacer mediciones en entornos estáticos con mayor luz de fondo, o bien disminuir el valor medio del ruido de fondo presente.

A.2. Simulación

Para analizar la posible implementación de esta librería en el análisis de las imágenes se creó un vector de 5 millones de elementos, simulando una imagen obtenida por el sensor. Para simular los procesamientos aplicados a la imagen en *Radmonpise* realizó una resta entre vectores. A dicha diferencia, al tratarse de valores sin signo, se asigna el valor 0 cuando el minuendo es menor al sustraendo. Además, para contrastar, se realizaron las mismas operaciones utilizando la CPU, sin paralelizar.

A.3. Resultados y Conclusión

Para la versión paralelizada, el procesamiento sobre 5 millones de valores demoró 165 ms, mientras que la versión sin paralelizar demoró entre 130 ms y 182 ms.

Estos resultados indican que, para utilizar la *QPU* de *Raspberry Pi* se requiere una mayor optimización, probablemente utilizando las instrucciones dadas por Broadcom [25]. En caso de utilizar dichas QPU, debería ser necesaria la migración del código a C++.

Apéndice B

Mediciones del flujo de muones en Sierra Grande

“Never gonna give you up. Never gonna let you down. Never gonna run around and desert you.”

— Rick Astley (1987), músico

B.1. Introducción y motivación

El día 2 de octubre de 2019 se realizó una medición del flujo de muones en MCC Minería Sierra Grande, dichas mediciones permiten sentar las bases para la construcción de un laboratorio subterráneo en la provincia de Río Negro. Actualmente existen proyectos de construcción de laboratorios subterráneos, en los que la radiación cósmica se vea disminuida, como es el caso del Proyecto ANDES [26]. Construir un laboratorio con baja radiación cósmica no sólo resulta de interés para físicos [27], sino también para electrónicos [28], biólogos [29], geofísicos y geólogos. Es por ello que cuantificar la radiación cósmica de fondo resulta unos de los parámetros claves a la hora de construir un laboratorio subterráneo.

Además de medir el flujo de muones, se utilizó el dispositivo *Wave* para medir el nivel de actividad de Rn en la mina. El detector de Rn no fue utilizado en este caso debido a falsos contactos entre las placas aceleradora, detallados en la sección ??.

B.2. Equipo de medición y algoritmo empleado

Para realizar dicha medición se utilizaron 4 detectores centelladores, estos centelladores son capaces de producir luz visible cuando una partícula interactúa con el medio, dicha luz es leída y convertida en un pulso eléctrico medible.

Los detectores son agrupados de a pares, en donde uno de ellos será llamado *master* y otro *slave*. El dispositivo *master* cumple la función de proveer una referencia al *slave*, al proveerle un pulso cuando se detecte un evento.

Tenemos así dos pares de placas, controladas por una SBC Red Pitaya que adquieren datos en paralelo. Ambos de estos archivos serán procesados por un algoritmo desarrollado en Python.

Para la detección de muones se aprovechó la alta sensibilidad temporal de los dispositivos de medición. Detectando como evento de muones a aquellos en los que eran detectados en ambos pares de detectores. Por lo tanto, se creó un arreglo de eventos de muones, haciendo coincidencia entre los eventos detectados por los dos pares, utilizando una ventana de tiempo de $1 \mu\text{s}$.

B.3. Interfaz web desarrollada

Debido a la modularidad de la interfaz web desarrollada en *React*, se crearon tarjetas capaces de visualizar los datos obtenidos por los sensores. Desde un cálculo estimativo del flujo de muones hasta histogramas de carga. En la Fig. B.1 se puede observar la interfaz desarrollada tanto en su versión de escritorio cómo móvil.

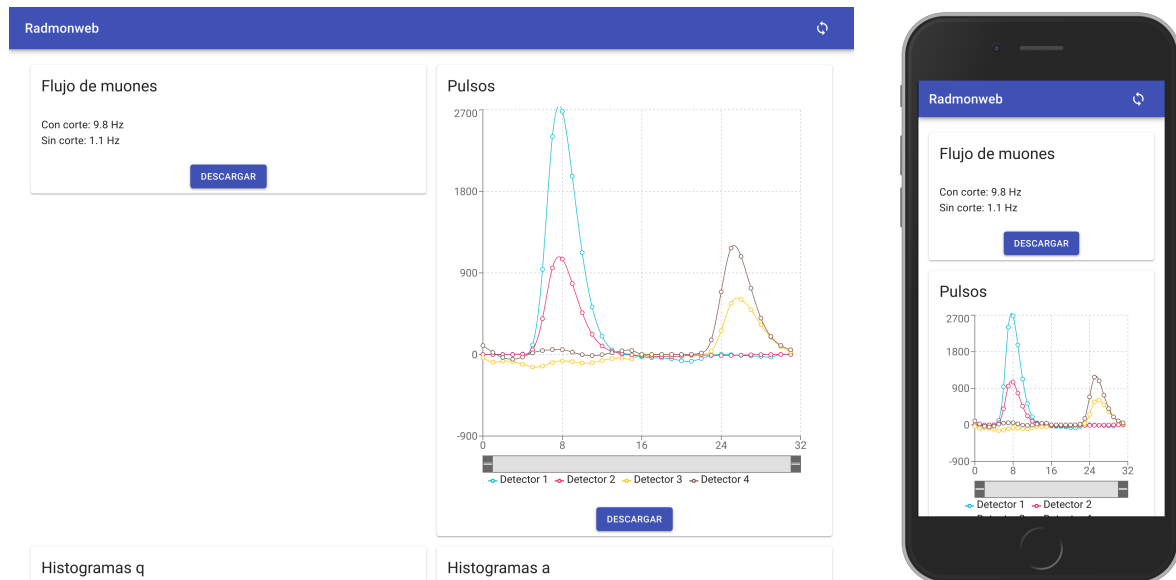


Figura B.1: Capturas de pantalla de la interfaz web utilizada durante el experimento de Sierra Grande. Se muestra tanto su versión web (izquierda), como su versión *mobile* (derecha).

B.4. Medición de la actividad de Rn con *Wave*

Se utilizó el medidor de actividad de radón desarrollado por *Airthings*, para monitorear el nivel de Rn durante las, aproximadamente, 6 horas de medición en la mina.

Para contrastar el cambio de actividad, se decidió mantener prendido y funcionando el dispositivo en todo momento (antes y después de la medición bajo tierra). La Fig. B.2 muestra la actividad de Rn dada por el sensor durante esa semana.

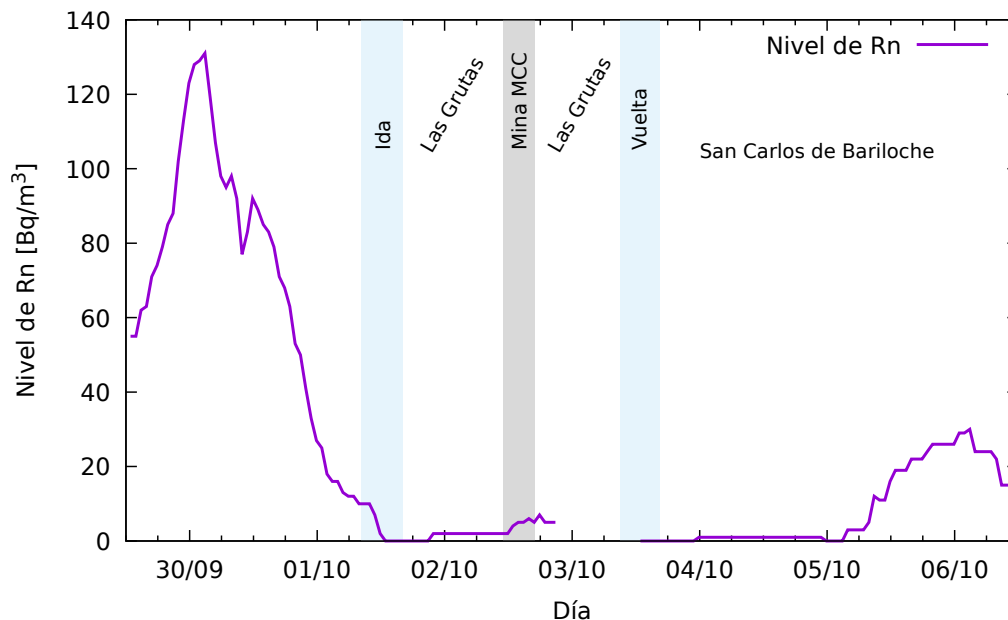


Figura B.2: Actividad de radón medida por el *Wave* durante la semana del experimento de Sierra Grande. También se detallan los intervalos de viaje entre Las Grutas y San Carlos de Bariloche y el intervalo de dicho experimento desarrollado en la mina.

Se puede observar que existe cierta región en la cual la información no se sincronizó correctamente. Además, tal como sugiere el fabricante, es necesario mediciones del orden de semanas para obtener una estadística significativa para el nivel de actividad de radón. Se esperan realizar futuras mediciones tanto con el medidor desarrollado y con el medidor disponible comercialmente.

Bibliografía

- [1] Galimberti, C. L., *et al.* A low cost environmental ionizing radiation detector based on COTS CMOS image sensors. En: 2018 IEEE Biennial Congress of Argentina (ARGENCON). IEEE, 2018. URL <https://doi.org/10.1109/argencon.2018.8645967>. vii, 9
- [2] Whiteson, D., Mulhearn, M., Shimmin, C., Cranmer, K., Brodie, K., Burns, D. Searching for ultra-high energy cosmic rays with smartphones. *Astroparticle Physics*, **79**, 1–9, jun 2016. URL <https://doi.org/10.1016/j.astropartphys.2016.02.002>. 2
- [3] Perez, M., *et al.* Implementation of an ionizing radiation detector based on a FPGA-controlled COTS CMOS image sensor. En: 2017 XVII Workshop on Information Processing and Control (RPIC). IEEE, 2017. URL <https://doi.org/10.23919/rpic.2017.8214376>. 2
- [4] The DAMIC Collaboration. Damic: a novel dark matter experiment, 2013. 2
- [5] The CONNIE Collaboration. The connie experiment. *Journal of Physics: Conference Series*, **761** (1), 2016. URL <http://stacks.iop.org/1742-6596/761/i=1/a=012057>. 2, 33
- [6] Groom, D. E. A. Temperature dependence of mean number of eh pairs per ev of x-ray energy deposit, 2009. 3
- [7] MIPI, Mobile Industry Processor Interface. MIPI Alliance Specification for Camera Serial Interface 2 (CSI-2), 2009. Draft Version 1.01.00. 3
- [8] Ghosh, S., Marshall, I., Freitas, A. Autonomously detecting the defective pixels in an imaging sensor array using a robust statistical technique. En: S. P. Farnand, F. Gaykema (eds.) Image Quality and System Performance V. SPIE, 2008. URL <https://doi.org/10.1117/12.765147>. 4
- [9] Omnivision. OV5647, color CMOS QSXGA (5 megapixel) image sensor with OMNIBSI technology[®], 2009. Rev. 1. 7

- [10] Sony. IMX219PQ, 2016. Ver. 1.0. 7
- [11] Griffin, R. H., Tarr, N. G. Optical image sensors and their application in radon detection. *En*: P. Cheben, J. Schmid, C. Boudoux, L. R. Chen, A. Delâge, S. Janz, R. Kashyap, D. J. Lockwood, H.-P. Loock, Z. Mi (eds.) *Photonics North 2013*. SPIE, 2013. URL <https://doi.org/10.1117/12.2033635>. 16
- [12] Ibanez, L. Raspberry Pi Camera to Microscope Adapter . <https://www.thingiverse.com/thing:214466>, 2013. 17
- [13] Santry, D., Werner, R. Stopping power measurements of c, al, si, ti, ni, ag, au and mylar using radioactive alpha sources. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, **1** (1), 13–15, jan 1984. URL [https://doi.org/10.1016/0168-583x\(84\)90471-3](https://doi.org/10.1016/0168-583x(84)90471-3). 21
- [14] Institute for Health Metrics and Evaluation (IHME). GBD Compare. Seattle, WA: IHME, University of Washington, 2015. Disponible en <http://vizhub.healthdata.org/gbd-compare>. Accedido en noviembre del 2018. 25
- [15] Demayo, A., *et al.* Toxic effects of lead and lead compounds on human health, aquatic life, wildlife plants, and livestock. *C R C Critical Reviews in Environmental Control*, **12** (4), 257–305, oct 1982. URL <https://doi.org/10.1080/10643388209381698>. 25
- [16] Thompson, A., *et al.* X-ray data booklet, 2009. URL <https://xdb.lbl.gov/xdb-new.pdf>, table 1-2 Photon energies, in electron volts, of principal K-, L-, and M-shell emission lines. 30
- [17] Phan, H., *et al.* Audio scene classification with deep recurrent neural networks, 2017. 33
- [18] Kiang, M. Y. Extending the kohonen self-organizing map networks for clustering analysis. *Computational Statistics & Data Analysis*, **38** (2), 161–180, dic. 2001. URL [https://doi.org/10.1016/s0167-9473\(01\)00040-8](https://doi.org/10.1016/s0167-9473(01)00040-8). 33
- [19] Hopfield, J. Hopfield network. *Scholarpedia*, **2** (5), 1977, 2007. URL <https://doi.org/10.4249/scholarpedia.1977>. 34
- [20] Deng, L. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, **29** (6), 141–142, nov. 2012. URL <https://doi.org/10.1109/msp.2012.2211477>. 34
- [21] Kloster, M. A. Robustez de las redes neuronales profundas para clasificar imágenes médicas, 2019. 38

-
- [22] Chollet, F., *et al.* Keras. <https://keras.io>, 2015. 40
- [23] Team, G. Google colab. <https://colab.research.google.com/>. 40
- [24] Nadra, A. D., *et al.* Design and evaluation of an incoherent feed-forward loop for an arsenic biosensor based on standard iGEM parts. *Synthetic Biology*, **2** (1), ene. 2017. URL <https://doi.org/10.1093/synbio/ysx006>. 46
- [25] Broadcom. Videocode[®] IV 3D, Architecture Reference Guide, 2013. Rev. VideoCoreIV-AG100-R. 50
- [26] Bertou, X. The ANDES underground laboratory. *The European Physical Journal Plus*, **127** (9), sep. 2012. URL <https://doi.org/10.1140/epjp/i2012-12104-1>. 51
- [27] Colaboration, T. D. Long-baseline neutrino facility (LBNF) and deep underground neutrino experiment (DUNE) conceptual design report, 2016. 51
- [28] Tylka, A., *et al.* CREME96: a revision of the cosmic ray effects on micro-electronics code. *IEEE Transactions on Nuclear Science*, **44** (6), 2150–2160, 1997. URL <https://doi.org/10.1109/23.659030>. 51
- [29] Lampe, N., *et al.* Background study of absorbed dose in biological experiments at the modane underground laboratory. *EPJ Web of Conferences*, **124**, 00006, 2016. URL <https://doi.org/10.1051/epjconf/201612400006>. 51

Agradecimientos

“The flow of time is always cruel, its speed seems different for each person, but no one can change it. A thing that does not change with time is a memory of younger days.”

— Sheik (1998), The Legend of Zelda, Ocarina of Time

El autor desea agradecer a su familia por respetar y alentar la decisión de desarrollarse en el ámbito científico-tecnológico.

A sus compañeros de estudio, por fomentar el trabajo colaborativo y por hacer una mejor experiencia la vida en el Instituto; en especial a Rodrigo López, su compañero de cuarto, y a Andrés Ziemecki, su compañero de Dota.

A los profesores del secundario Merceditas de San Martín de Cesap y a cada uno de los profesores y divulgadores, que motivaron al autor a seguir la carrera de física. Desde el profesor Américo Poblete, impulsando las Olimpíadas de Matemática, hasta Arcadi García, con sus videos de divulgación científica en su canal de Youtube.

También desea agradecer especialmente a Florencia Tejada, por ayudar al autor a seguir adelante a pesar de las adversidades, por tener la habilidad de transformar momentos malos en buenos, y momentos buenos en mejores.

