

# Sistema de detección de minas GPR y Machine Learning

Grupo Desminado

February 2022

## 1 Introducción

La detección de minas explosivas es un reto complejo debido a que los métodos usuales están enfocados en detectar metales mediante inducción electromagnética. Sin embargo, las minas modernas poseen poco contenido metálico. Adicionalmente, las características electromagnéticas del suelos suelen no ser homogéneas [1]. En este contexto, los sistemas de Ground Penetrating Radar (GPR) cobran importancia para la detección de objetos de plástico enterrados en el suelo. El sistema GPR captura las reflexiones electromagnéticas de las discontinuidades del suelo (objetos) permitiendo detectar objetos que no están hechos de metal o ser utilizado como complemento a los métodos de detección de metales.

En este documento se presenta una síntesis de la implementación de un sistema de detección de minas a partir del artículo [2] utilizando Dictionary Learning para obtener Sparse Representations de las señales capturadas (A-Scans) por un sistema de GPR del proyecto de Desminado Humanitario. También se entrena un modelo de Convolutional Neural Network 1D (CNN-1D), equivalente al utilizado en [2] para realizar comparación. Finalmente, se entrena un modelo de CNN-2D, basado en la utilización de B-Scan en dos canales espaciales (eje X y eje Y) para entregar resultados equivalentes a los modelos anteriores.

Sparse Representation consiste en representar vectores como una combinación lineal de pocos elementos que permiten reconstruir la información original. El uso de Sparse Representation ha sido efectivo para tareas de reconstrucción de señales y clasificación de audio e imágenes [3]. Las CNN buscan obtener información espacial y local de los datos haciendo uso de la operación de convolución, por lo que se han probado eficientes para procesar datos como señales e imágenes, para tareas de Machine Learning.

## 2 Implementación

### 2.1 Mediciones de GPR

El sistema de medición está compuesto por un Analizador de Redes Vectoriales (VNA), Anritsu MS2026C (Fig. 1), conectado a dos antenas PowerLog UWB con un ancho de banda de 600 MHz - 6 GHz (Fig. 2 ), conectadas a cada uno de los puertos del VNA.

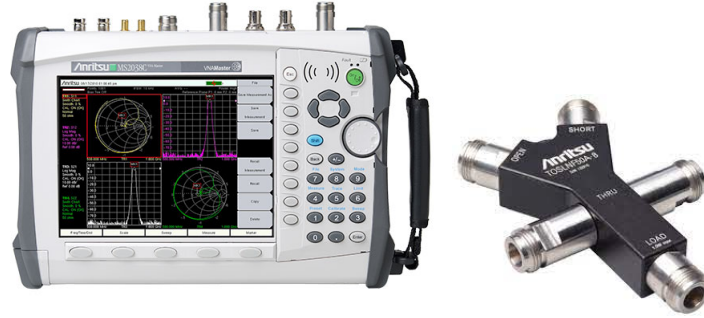


Figure 1: VNA Anritsu MS2026C con Kit de calibración TOSLNF50A-8

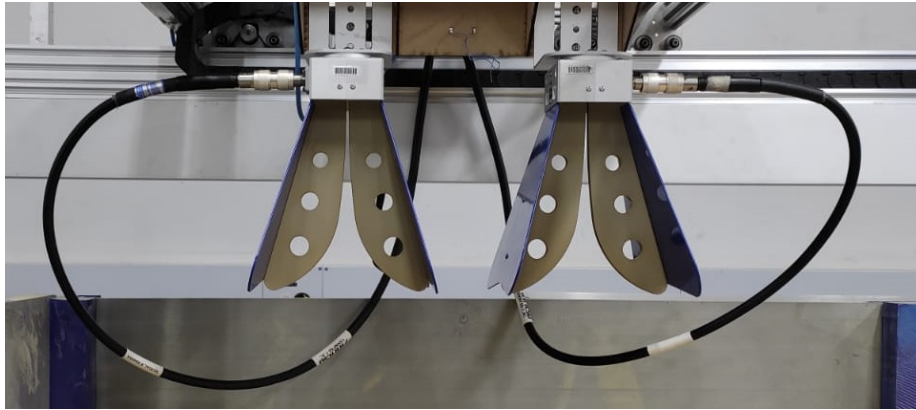


Figure 2: Antenas PowerLog del GPR

Para la toma de datos, el GPR es ubicado en puntos X-Y para realizar las mediciones (Fig. 3). En cada punto X-Y se capturan los parámetros  $S_{21}$  para un rango de frecuencias entre 600 MHz y 6 GHz, obteniendo así un A-Scan en el dominio de la frecuencia. El GPR, luego es desplazado por el sistema de posicionamiento para cubrir la totalidad del campo.

Para cada A-scans en el dominio de la frecuencia se puede utilizar la transformada inversa de Fourier y obtenerlos en el dominio del tiempo, que son usualmente más útiles para su análisis.

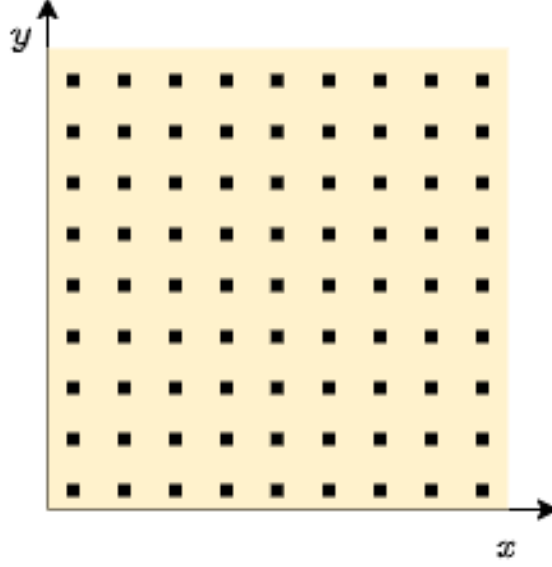


Figure 3: Puntos de medición del GPR.

Al juntar varios A-scans en el dominio del tiempo tomados a lo largo de una línea recta en el eje  $x$  o el eje  $y$  se obtiene un B-Scan. En un B-scan los puntos de la superficie producen hipérbolas en el plano  $x - t$  debido a la diferencia en los tiempos de llegada de las ondas electromagnéticas a ese punto en particular. Siendo la ubicación del punto el ápice de la hipérbola. Adicional a la información temporal se tiene la información de las amplitudes de las ondas. Las discontinuidades en el suelo (objetos) producen una reflexión de las ondas electromagnéticas que llevan a hipérbolas más notorias, por lo que la detección de objetos se puede llevar a cabo buscando las hipérbolas.

## 2.2 Sparse Representation

Sparse Representation, consiste representar una señal como una combinación lineal de pocos coeficientes. Considerando la ecuación (1). Dada una señal  $\mathbf{y}$  y un diccionario  $\mathbf{D}$  se busca una representación dispersa  $\mathbf{x}$ .

$$\mathbf{y} = \mathbf{D}\mathbf{x} \quad (1)$$

Donde:

- El vector  $\mathbf{y} \in \mathbb{R}^m$  es la señal.
- La matriz  $\mathbf{D} \in \mathbb{R}^{m \times k}$  es llamada diccionario y sus columnas átomos, es decir, tiene  $k$  átomos, cumpliendo  $k > m$ .

- El vector  $\mathbf{x} \in \mathbb{R}^k$  es la representación dispersa de la señal.

En el caso en que  $k > M$  la solución de  $\mathbf{x}$  no está determinada (hay infinitas soluciones), por que lo se puede buscar una solución dispersa (con pocos elementos diferentes de cero). Para ello se añaden restricciones sobre  $\mathbf{x}$  (que sea disperso) y se resuelve la ecuación (1) como un problema de optimización.

En Orthogonal Matching Pursuit (OMP), el problema de optimización es formulado como un problema dual de las ecuaciones (2) y (3):

$$\min_x \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \text{ sujeto a } \|\mathbf{x}\|_0 < S \quad (2)$$

$$\min_x \|\mathbf{x}\|_0 \text{ sujeto a } \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 < \delta \quad (3)$$

Donde  $S$  es el número de dispersión (número máximo de entradas diferentes de 0 en  $\mathbf{x}$ ) y  $\delta$ , es el parámetro de residuo.

Para encontrar una representación dispersa la librería de python, scikit-learn [4], provee varios algoritmos, entre ellos, Batch-OMP.

### 2.3 Dictionary Learning

El objetivo de Dictionary Learning, es hallar un diccionario  $\mathbf{D}$ , que permita obtener una representación dispersa  $\mathbf{X} \in \mathbb{R}^{K \times L}$  para un conjunto de datos de entrenamiento  $\mathbf{Y} \in \mathbb{R}^{M \times L}$ , donde  $L$  es el número de señales de entrenamiento, como se presenta en la ecuación (4).

$$\mathbf{Y} = \mathbf{D}\mathbf{X} \quad (4)$$

De forma que el problema de optimización para hallar tanto  $\mathbf{D}$  como  $\mathbf{X}$  se plantea en términos del número de dispersión de cada representación dispersa o de su valor de residuo como se observa en las ecuaciones (5) y (6):

$$\begin{aligned} \min_{\mathbf{D}, \mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F \\ \text{sujeto a } \|\mathbf{x}_i\|_0 < S \forall i | 1 \leq i \leq L \end{aligned} \quad (5)$$

$$\begin{aligned} \min_{\mathbf{D}, \mathbf{X}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F \\ \text{sujeto a } \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_0 < \epsilon \forall i | 1 \leq i \leq L \end{aligned} \quad (6)$$

Debido a que tanto  $\mathbf{D}$  como  $\mathbf{X}$  son desconocidos, se suele resolver el problema en dos pasos iterativos repetidos hasta cumplir un criterio de convergencia o número fijo de iteraciones:

1. Representación dispersa: Hallar  $\mathbf{X}$  dado  $\mathbf{D}$ , se pueden utilizar los métodos mencionados anteriormente.
2. Actualización del diccionario: Hallar  $\mathbf{D}$  dado  $\mathbf{X}$ . Se puede implementar mediante descomposición de valores singulares (SVD) o descenso del gradiente.

Existen distintos métodos para obtener el diccionario entre ellos: K-SVD el cual se basa en el enfoque anterior utilizando descomposición de valores singulares y OMP [5]. Online Dictionary Learning (ODL) [3]. Los autores de [2], en particular, proponen un algoritmo Drop-Off Mini-Batch Online Dictionary Learning (DOMINODL).

El algoritmo K-SVD utiliza dos pasos, la descomposición de valores singulares y OMP, ambos se encuentran implementados en skit-learn. Adicionalmente, ODL se encuentra directamente implementado en la misma librería de python. Se resalta que el parámetro más importante para Dictionary Learning, utilizando el algoritmo ODL, es el número de átomos  $K$ .

## 2.4 Support Vector Machine

El problema de clasificación binaria consiste en dado un vector de características  $\mathbf{x}$ , el clasificador entrega como resultado una clase  $y \in \{-1, 1\}$ .

Support Vector Machine es un algoritmo de clasificación basado en kernels, que busca la separación de máximo margen entre dos conjuntos de datos en un espacio dimensional más alto del que se encuentran en originalmente, mediante un mapeo  $\phi(\mathbf{x})$ .

A continuación se presenta una explicación intuitiva basada en [4]:

Dados  $n$  vectores de características  $\mathbf{x}_i$  y sus respectivas clases  $y_i \in \{-1, 1\}$  para  $i = 1, \dots, n$ . El problema de máximo margen entre los conjuntos de datos se plantea en forma primal como se presenta en la ecuación (7).

$$\begin{aligned} \min_{\mathbf{w}, b, \zeta} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^n \zeta_i \\ \text{sujeto a } & y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \zeta_i \\ & \zeta_i \geq 0 \\ & i = 1, \dots, n \end{aligned} \tag{7}$$

El hiperplano de separación en el espacio dado por la función  $\phi(\mathbf{x})$  está dado por la ecuación  $\mathbf{w}^T \phi(\mathbf{x}) + b = 0$  (como se presenta en la Fig. 4 para el caso de dos dimensiones), de forma que la predicción para un vector  $\mathbf{x}$  está dada por  $\text{sign}((\mathbf{w}^T \phi(\mathbf{x}) + b))$ . En este sentido la primera restricción indica que la clasificación para los vectores de entrenamiento sea correcta con unas variables de holgura  $\zeta_i$  (relajan la restricción).

El primer término de la función objetivo busca maximizar el margen (minimizando la norma de  $\mathbf{w}$ , véase 4) mientras que el segundo busca que las variables de holgura sean las menores posibles junto ponderado por una variable de regularización  $C$ .

El problema dual de (7) se presenta en la ecuación (8):

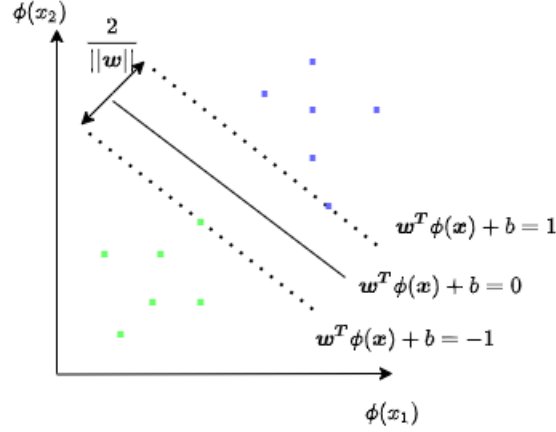


Figure 4: Separación de máximo margen

$$\begin{aligned}
& \min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\
& \text{sujeto a } y^T \alpha = 0 \\
& 0 \leq \alpha_i \leq C \\
& i = 1, \dots, n
\end{aligned} \tag{8}$$

Donde  $Q \in \mathbb{R}^{n \times n}$  es la matriz de gram definida como  $Q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$  siendo  $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_i)$ , el kernel. Las variables  $\alpha$  son las variables duales y  $e \in \mathbb{R}^n$  es un vector cuyas todas sus entradas son 1. Al resolver el problema de optimización, los vectores  $\mathbf{x}$  cuyo valor de  $\alpha$  sea diferente de 0 se conocen como vectores de soporte ya que la clasificación depende de estos, que está dada por la ecuación (9):

$$\sum_{i=0}^n y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{9}$$

Para resolver el problema dual (8) no es necesario conocer la función de mapeo  $\phi(\mathbf{x})$ , solo se requiere de un kernel, el cual es una medida de similitud entre dos vectores ( $\mathbf{x}$  y  $\mathbf{x}'$ ). El caso más simple de un kernel es un kernel lineal (donde el mapeo corresponde es la identidad), el cual se calcula mediante el producto punto entre los vectores.

Algunos kernels comunes son [6]:

- Lineal:  $K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$
- Polinomial:  $K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle^d$  siendo  $d$  el grado del polinomio.
- Gaussiano:  $K(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2)$



Figure 5: Detección de bordes. Tomada de [7]

La solución de este problema se puede realizar mediante algoritmo cómo Sequential Minimal Optimization. La librería skit-learn posee un clasificador de Support Vector Machine, el cual incluye varios tipos de kernels. En particular, se utilizará el kernel gaussiano. Los parámetros más importantes para ajustar con la variable de regularización  $C$ , junto con  $\gamma = 1/\sigma^2$ , en la ecuación del kernel gaussiano.

## 2.5 Convolutional Neural Networks

Las redes convolucionales (CNN) son un tipo de redes neuronales las cuales incluyen una capa convolucional, que como su nombre lo indica realizan una operación matemática conocida como convolución [7]. Sus aplicaciones se encuentran principalmente en el área de procesamiento de señales de una dimensión como pueden ser electrocardiogramas o audios; de dos dimensiones como pueden ser las imágenes.

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \quad (10)$$

La convolución se define como se presenta en (10) donde  $x$  es la entrada y  $w$  es el kernel (no confundir con el kernel de SVM). La convolución con un kernel adecuado logra extraer características de la señales. Por ejemplo, los bordes de una imagen como se presenta en la Fig. 5. En este sentido el propósito de una red convolucional es aprender el kernel que logre extraer las características relevantes de una señal, de tal manera que se logre una tarea como clasificación o regresión.

En una capa convolucional se hace uso de tres pasos principales:

- Convolución: Transforma la señal mediante la operación de convolución. Típicamente se realizan varias operación de convolución de manera paralela los que se conocen como filtros.
- Activación: Aplican una función no lineal sobre la salida del filtro. Entre las activaciones más comunes se encuentran ReLU ( $f(x) = \max(0, x)$ ), la

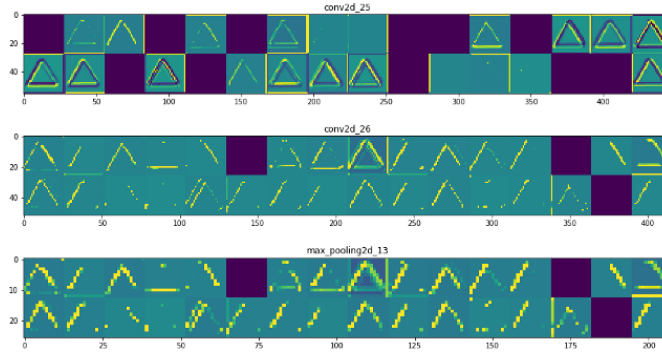


Figure 6: Salida de las primeras capas recibiendo de entrada un triángulo. Tomado de [10]

tangente hiperbólica y la función sigmoide. La utilización de funciones no lineales permite que el modelo se pueda ajustar a comportamientos no lineales, que son usualmente de interés.

- Pooling: La función de esta capa es reportar una característica de resumen, por ejemplo el promedio o el máximo de cierto número de entradas. La operación es útil debido a que reduce el número de entradas de la siguiente capa y también permite añadir la característica de invariancia a pequeñas traslaciones de la característica de interés [7]. Por ejemplo, en una tarea de clasificación de imágenes, es importante la presencia del objeto de interés más no su ubicación dentro de la imagen.

En [8] se utilizó la primera red convolucional para clasificar dígitos escritos a mano. En la página web [9] presenta una visualización de las capas intermedias para la misma tarea de clasificación. En [10] también se visualizan las salidas de las capas intermedias para la clasificación de figuras geométricas (cuadrado, círculo y triángulo).

En la Fig. 6 se observa la salida de las tres primeras capas: dos capas convolucionales con 32 filtros cada una y activación ReLU con una capa de MaxPooling. Como se puede observar los filtros de la primeras capas mantienen información de los lados del triángulo y forma del triángulo. En la Fig. 7 se presenta la salida de la última capa convolucional de la red. Como se puede observar, el número de puntos ha disminuido por los repetitivos procesos de MaxPooling y como los puntos activados son menos interpretables a simple vista. En este sentido, las etapas convolucionales son vistas o interpretadas como un paso de extracción de características de datos crudos para posteriormente realizar una clasificación con estas características.

El sitio web [11] presenta de manera interactiva los pasos de una red convolucional simple para la clasificación de imágenes. La red consiste en cuatro capas convolucionales en las que se presentan el efecto para una imagen en particular a lo largo de toda la red.



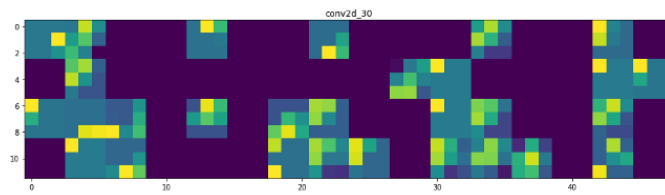


Figure 7: Salida de la última capa convolucional. Tomado de [10]

### 3 Detección de minas

#### 3.1 Escenarios de entrenamiento

Con el sistema GPR del laboratorio, se tomaron muestras en distintos escenarios para entrenar y evaluar el desempeño del modelo. La configuración que se hizo para el barrido del GPR (con el Software **GPR-V2**) se encuentra resumida en la tabla 1. Estos parámetros fueron los que se utilizaron en todas las mediciones efectuadas y documentadas en el presente artículo.

Rango de frecuencias	600 MHz - 6 GHz
Número de puntos por medición	501
Trayectoria	Grilla
Límites espaciales X-Y	[0-700] mm
Número de puntos X-Y	141
Orientación de las antenas	Eje X

Table 1: Configuración del barrido para los experimentos.

La configuración de objetos en el escenario está descrita por la altura de la antena, las posiciones X-Y de los objetos junto con sus profundidades respecto a la superficie de la arena, y su geometría. Esto se observa en las Figuras. 8 y 9, que muestra las vistas frontal y superior del campo de pruebas y las medidas relevantes.

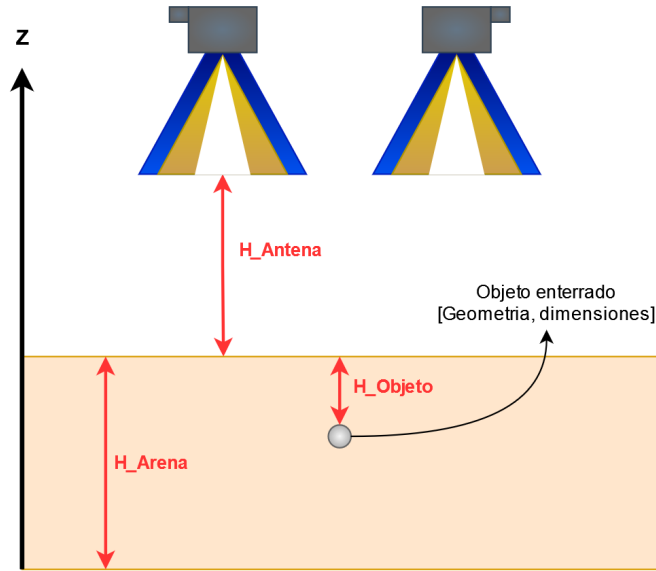


Figure 8: Vista frontal del campo de pruebas

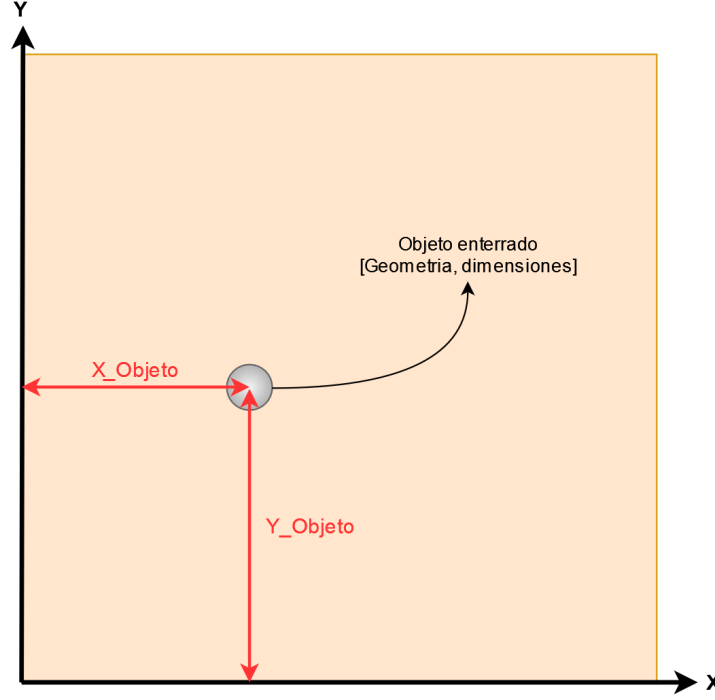


Figure 9: Vista superior del campo de pruebas

Es importante señalar que los procedimientos de Dictionary Learning, Sparse Representation y Support Vector Machine, requieren que todas las señales posean un número igual de puntos, lo que se traduce en la necesidad de que todas las mediciones se deben realizar en el mismo rango de frecuencias y número de puntos. En este caso se utilizó un rango de frecuencias de 600 MHz - 6 GHz de las cuales se tomaron 501 mediciones.

Por otra parte, el sistema no tiene limitaciones en términos de la trayectoria, límites espaciales o número de puntos en X-Y (A-scan), ya que el sistema trabaja como muestra un A-scan. Sin embargo, para realizar gráficas de mapas como el mostrado en la Fig. 10, sí es necesario realizar mediciones con un espaciamiento en X y Y igual (independiente para cada eje).

### 3.1.1 Escenario 2: Esfera metálica

En este escenario hay una esfera metálica de 7 cm de diámetro enterrada a 1 cm de profundidad en las coordenadas X-Y (35 cm, 35 cm). Los detalles se observan en la Tabla. 2 Así el resultado esperado por el clasificador se presenta en la Fig. 10.

La Fig. 11 presenta el B-scan a lo largo del eje  $y$ , estando ubicado en  $x = 0.35m$ , es decir las antenas logran pasar justo sobre la esfera. La Fig. 11a

Table 2: Configuración del barrido para los experimentos.

$H_{antena}$	600 mm
$H_{arena}$	260 mm
$H_{objeto}$	10 mm
$X_{objeto}$	350 mm
$Y_{objeto}$	350 mm
Objeto	[Esfera, 70mm de diámetro]

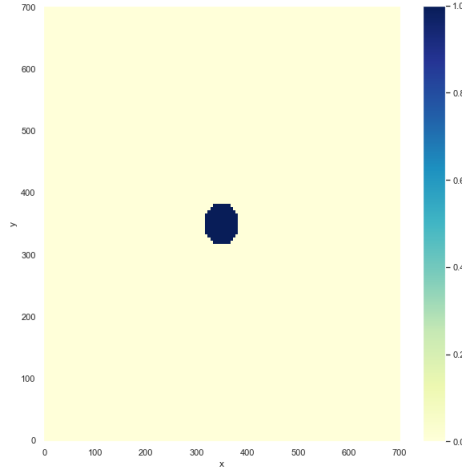


Figure 10: Resultado esperado por el sistema de detección para los escenarios: 2, 3 y 6.

presenta el B-Scan sin ningún procesamiento. Las Fig. 11b y Fig. 11c presentan el resultado de dos técnicas para la remoción del fondo, de forma que sea más notoria el efecto del objeto enterrado. En la Fig. 11b se sustrae el promedio a lo largo del eje  $y$  para cada valor de  $t$ , mientras que en Fig. 11c se sustrae el B-Scan de este escenario con el del escenario 1.

Como se puede observar los efectos más notorios son el acople entre antenas, la superficie de la arena y el fondo de la caja. En la Fig. 11a los efectos son tan notorios que no se logra observar ninguna hipérbola entre la superficie y el fondo, de manera tenue se observan hipérbolas cuyos ápices estarían ubicados por fuera del B-Scan (valores de  $y$  negativos).

En la primera técnica de remoción de fondos (Fig. 11b) se destacan más las hipérbolas ubicadas al lado izquierdo. También se observa que el acople entre antenas es eliminado. Aunque esta técnica no logra eliminar por completo el efecto de la superficie y el fondo si hace más perceptible el efecto de la esfera en el centro cercano a la superficie.

En la segunda técnica, en el cual se sustraen los escenarios (Fig. 11c) ni el acople entre antenas, la superficie y el fondo son eliminados por completo. Sin

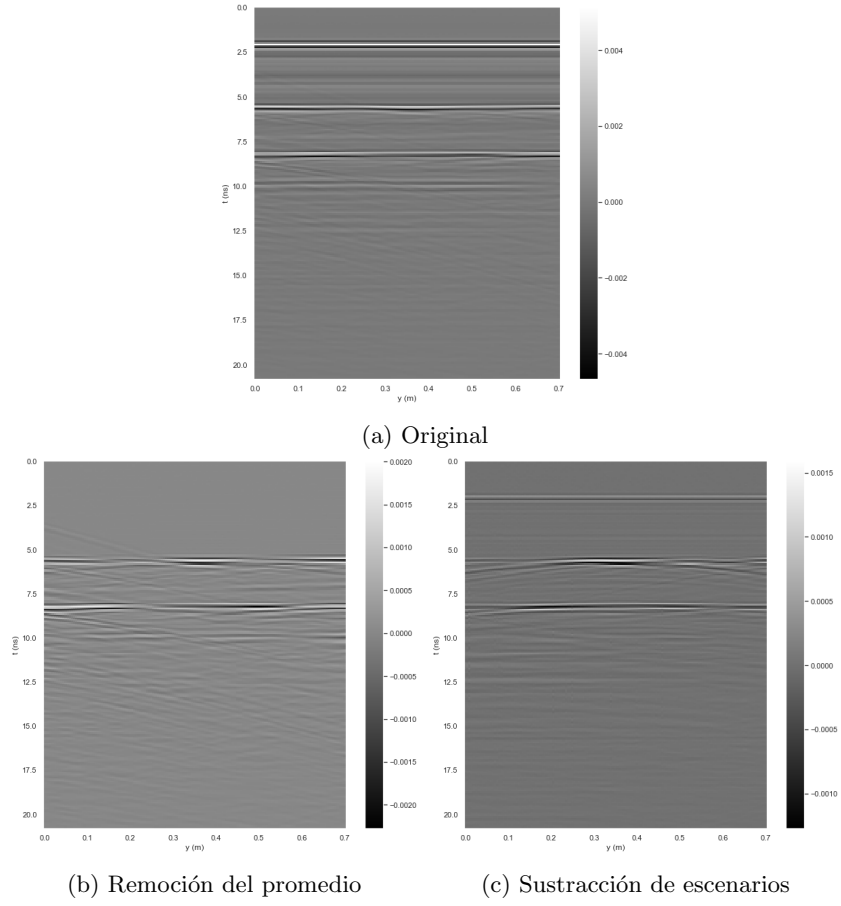


Figure 11: B-Scan Escenario 2.  $x = 0.35m$

embargo, las hipérbolas del lado izquierdo desaparecen de la imagen y es clara la hipérbola con ápice en el centro cercano a la superficie por lo que se asume que corresponde al efecto de la esfera.

### 3.1.2 Escenario 3: Esfera metálica

Este escenario es equivalente al escenario 2, con la excepción que la esfera se encuentra enterrada a una profundidad de 11 cm. Los detalles se observan en la Tabla. 3 Debido a que el sistema de detección no entrega información respecto a la profundidad de los objetos, el resultado esperado y con el que se realiza el entrenamiento es igual al mostrado en la Fig. 10.

Table 3: Configuración del barrido para los experimentos.

$H_{antena}$	600 mm
$H_{arena}$	260 mm
$H_{objeto}$	110 mm
$X_{objeto}$	350 mm
$Y_{objeto}$	350 mm
Objeto	[Esfera, 70mm de diámetro]

La Fig. 12 presenta el B-scan original y procesado con las dos técnicas de remoción de fondo presentadas para el escenario anterior. De manera general, se observan los mismos resultados. Sin embargo, en la Fig. 12c se observan al menos tres hipérbolas entre la línea de la superficie y el fondo por lo que no es evidente a cual corresponde la esfera enterrada.

### 3.1.3 Escenario 5: Mina

Este escenario es tiene una mina plástica de geometría cilíndrica con un diámetro de 7 cm y una altura de 20 cm. La mina se encuentra enterrada en el centro del campo a una profundidad de 2 cm. Los detalles del escenario se presentan en la Tabla. 4. Debido a ello la respuesta esperada es la misma que la de los escenarios anteriores (Fig. 10).

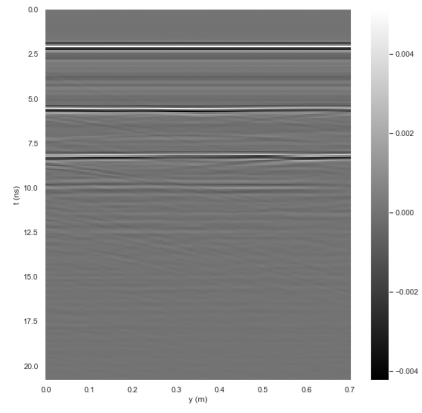
Table 4: Configuración del barrido para los experimentos.

$H_{antena}$	420 mm
$H_{arena}$	440 mm
$H_{objeto}$	20 mm
$X_{objeto}$	350 mm
$Y_{objeto}$	350 mm
Objeto	[Mina, 70mm de diámetro, 200 mm de alto]

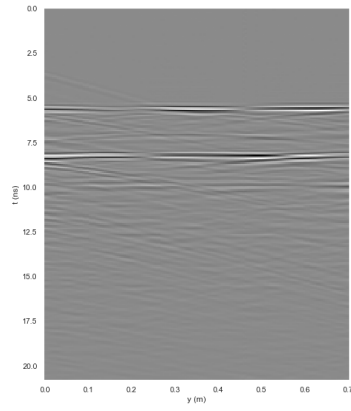
La Fig. 13 presenta el B-Scan de este escenario a lo largo del eje  $y$  con la coordenada  $x$  fija en  $0.35m$ . Se puede observar que la hipérbola formada por la mina (Fig. 13b y 13c) es poco notoria independientemente de la técnica de remoción del fondo.

### 3.1.4 Escenario 6: Placa metálica

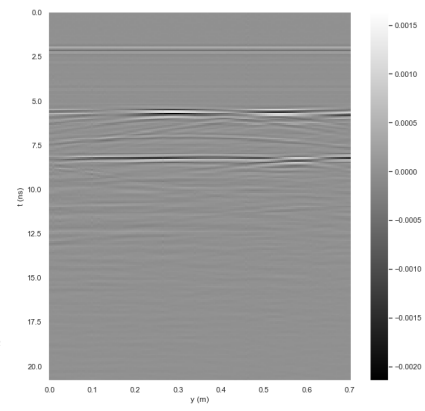
El último escenario tiene una placa metálica cuadrada enterrada de lado de 25 cm, cuyo centro se encuentra en las coordenadas X-Y (35 cm, 35 cm). La placa se encuentra enterrada a una profundidad de X cm. Los detalles se presentan en la Tabla. X. El resultado esperado para este escenario se presenta en la Fig. 14.



(a) Original

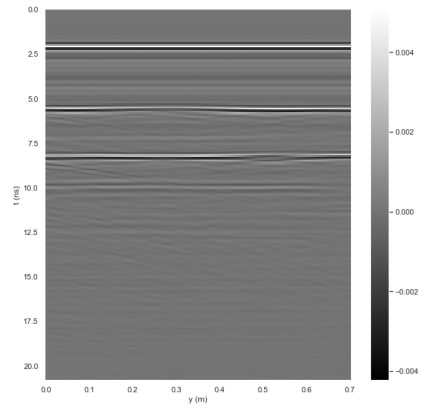


(b) Remoción del promedio

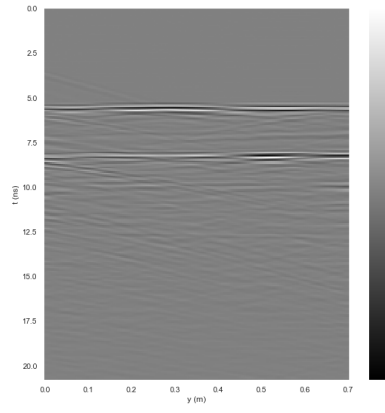


(c) Sustracción de escenarios

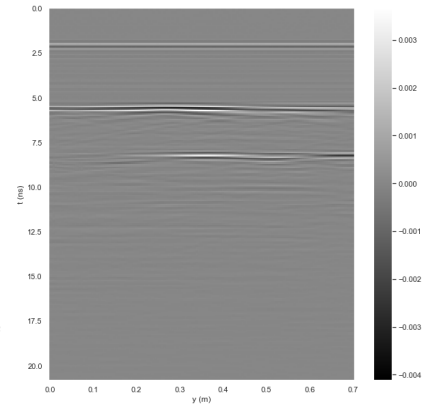
Figure 12: B-Scan Escenario 3.  $x = 0.35m$



(a) Original



(b) Remoción del promedio



(c) Sustracción de escenarios

Figure 13: B-Scan Escenario 5.  $x = 0.35m$



Table 5: Configuración del barrido para los experimentos.

$H_{antena}$	600 mm
$H_{arena}$	260 mm
$H_{objeto}$	20 mm
$X_{objeto}$	350 mm
$Y_{objeto}$	350 mm
Objeto	[Mina, 70mm de diámetro, 200 mm de alto]

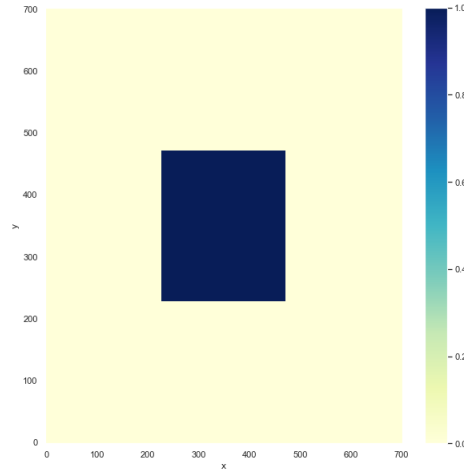


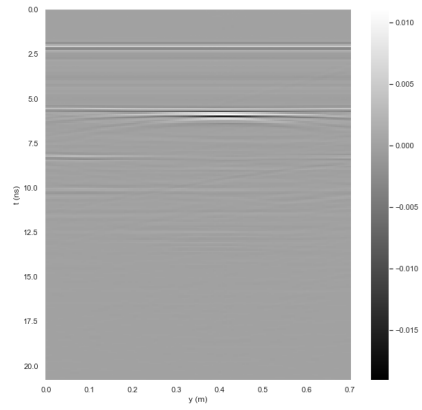
Figure 14: Resultado esperado por el sistema de detección para el escenario 6

La Fig. 15 presenta el B-Scan original de este escenario y los resultados de las técnicas de remoción de fondo. A diferencia de los escenarios presentados anteriormente, en este escenario el efecto de la placa metálica es tan notorio que se visualiza claramente en el B-scan original (Fig. 15a) sin necesidad de utilizar las técnicas de remoción de fondo.

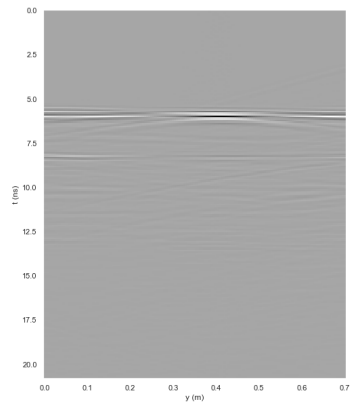
### 3.2 Metodología

Por claridad se adoptan las siguientes convenciones para esta sección:

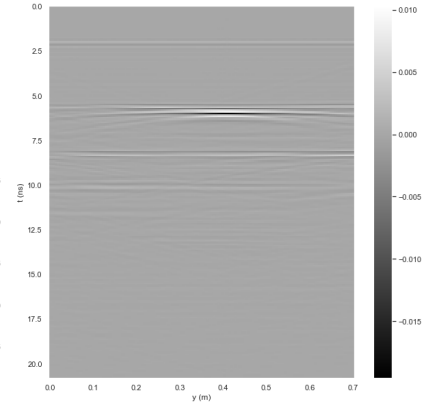
- $\mathbf{X} \in \mathbb{R}^{m \times n}$  es la matriz formada por un conjunto de A-scans en el dominio del tiempo  $\{\mathbf{x}_1, \dots, \mathbf{x}_i, \dots, \mathbf{x}_m\}$ . Siendo  $m$  el número de A-scans y  $n$  el número de puntos por A-scan. En otras palabras cada A-scan es una fila.
- $\mathbf{X}' \in \mathbb{R}^{m \times k}$  es la matriz de representaciones dispersas de  $\mathbf{X}$ , siendo  $k$  el número de átomos del diccionario  $\mathbf{D}$ .
- $\mathbf{y} \in \mathbb{R}^m$  son las clases a las que corresponden cada señal  $\{0, 1\}$ . El valor 0 indicando que no hay presencia de objetos y el 1 su presencia.



(a) Original



(b) Remoción del promedio



(c) Sustracción de escenarios

Figure 15: B-Scan Escenario 6.  $x = 0.35m$

Para el entrenamiento de el sistema de detección se realizan los siguientes pasos:

1. Clasificar cada A-Scan ( $\mathbf{x}_i$ ) de acuerdo si en sus coordenadas X-Y hay o no una mina enterrada ( $y_i$ ), obteniendo el vector  $\mathbf{y}$ .
2. Realizar Dictionary Learning con un conjunto de entrenamiento (A-Scans) ( $\mathbf{X}_{Train}$ ) para la obtención de  $\mathbf{D}$  y  $\mathbf{X}'_{Train}$
3. Entrenar un clasificador basado en Support Vector Machine, utilizando la representación dispersa de las señales de entrenamiento  $\mathbf{X}'_{Train}$ .

Para la detección de minas de un nuevo escenario se siguen los siguientes pasos:

1. Dados los A-scan del escenario ( $\mathbf{X}$ ) Obtener la representación dispersa ( $\mathbf{X}'$ ) utilizando el diccionario hallado anteriormente.
2. Utilizar  $\mathbf{X}'$  para clasificar los A-Scan con la Support Vector Machine entrenada anteriormente.
3. Ordenar los resultados  $\mathbf{y}$  en una matriz que represente el campo.

Los anteriores procedimientos se resumen en la Fig. 16 Es importante tener en cuenta el orden de las A-scans en las matrices de señales  $\mathbf{X}$  y las clases  $\mathbf{y}$  de forma que se conozca la posición X-Y de cada A-scans. De esta forma, el resultado final es fácilmente interpretado como un mapa en el plano X-Y que indica la presencia de que exista una mina en cada coordenada.

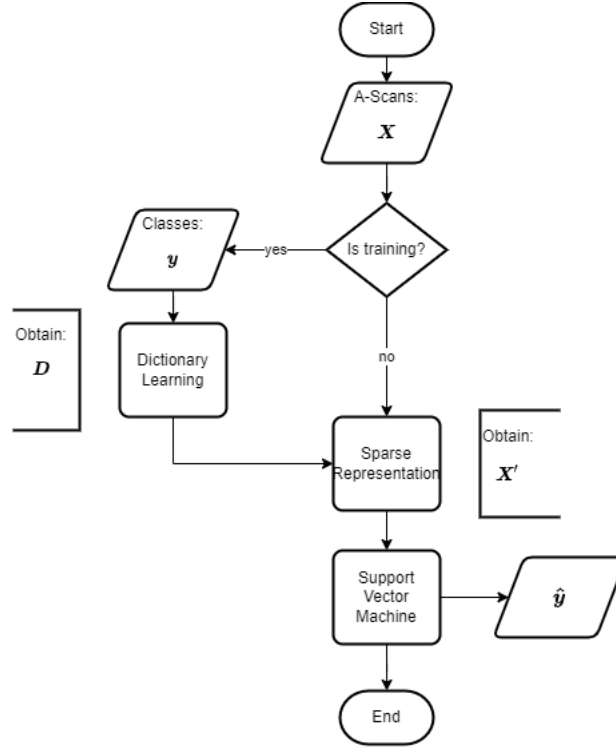


Figure 16: Flujo del sistema de detección de minas

Es importante mencionar que el clasificador, Support Vector Machine, no entrega de manera nativa una probabilidad, en cambio hace uso de validación cruzada K-fold con  $K=5$  para obtener un estimado de la misma [4].

### 3.3 Selección de parámetros

Recordando lo mencionado en X, el parámetro más importante es el número de átomos en el diccionario,  $K$ . Por el lado de el clasificador  $SVM$ , los parámetros son  $C$  y  $\gamma$ .

En este sentido debido a qué el paso de mayor tiempo computacional y siguiendo los resultados de [2], mayor  $K$  reduce el coeficiente de variación, lo que indica una mejor reconstrucción entre la señal y la estimada por la representación dispersa. En este sentido y teniendo en cuenta que mayor  $K$  resultará también en mayor complejidad computacional para el clasificador, se definió este parámetro como  $K = 600$ .

Por parte de SVM, se realizó una búsqueda entre los parámetros  $C$  y  $\gamma$  en grilla, buscando maximizar el recall, el cual indica el número de A-scans clasificados como presencia de objetos sobre el número de A-scans que efectivamente sus coordenadas X-Y tienen objetos. Notese que no se tienen en cuenta las

clasificaciones de la arena, ya que es de mayor interés reconocer la ubicación de los objetos costando un mayor número de falsos positivos.

La Fig. 17 muestra el recall para cada combinación de parámetros  $C$  y  $\gamma$ , existen valores faltantes debido a que no lograron ser ajustados. Como se presenta en la gráfica la mejor combinación de parámetros con un recall promedio de 0.53 (promedio validación cruzada K-fold con K de 3) corresponde a la combinación  $C = 1000$  y  $\gamma = 0.01$ .

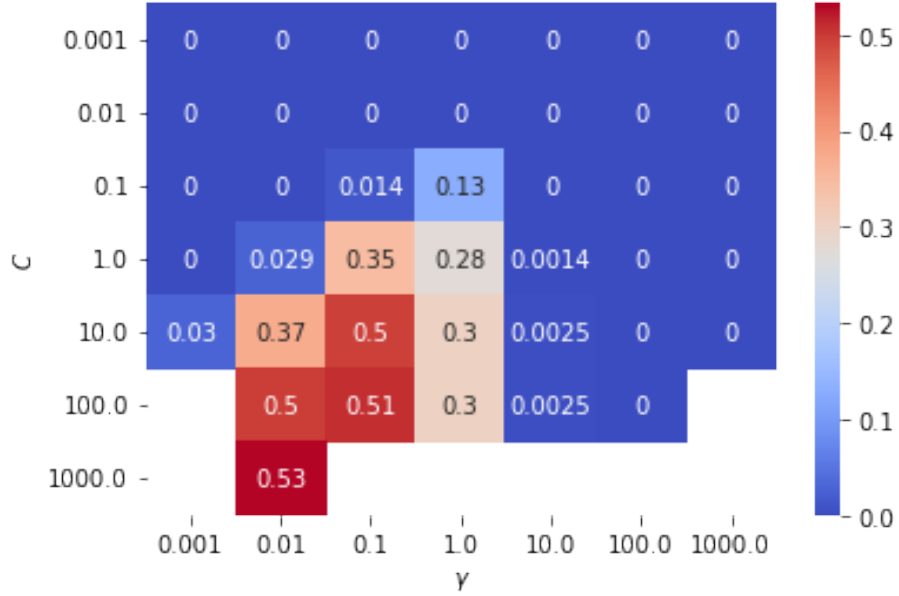


Figure 17: Recall para diferentes parámetros de SVM

### 3.4 CNN 1D

Siguiendo la comparación entre el modelo DL + SVM con CNN de [2], se utilizó una red convolucional con arquitectura equivalente, ajustando la capa las dimensiones para que correspondieran al número de muestras en las señales de entrada y la capa de salida para que corresponda una única clase. La Fig 18 muestra la arquitectura de la red convolucional: dos capas convolucionales y una capa completamente conectada con una neurona de salida. Mayor detalle acerca de la arquitectura de la red se puede observar en la Tabla 6.

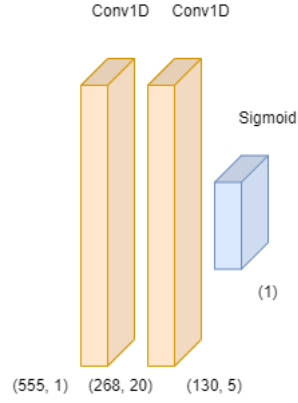


Figure 18: Arquitectura de la red

Table 6: Arquitectura de la red

Capa	Activation	Filters	Kernel Size	Strides	Output Shape
Conv1D	ReLU	20	20	2	(268, 20)
Conv1D	ReLU	5	10	2	(130, 5)
Flatten	-	-	-	-	650
Dense	Sigmoid	-	-	-	1

### 3.5 CNN 2D

Se propone un detección a partir de B-Scan utilizando redes neuronales. En este orden de ideas el problema de Machine Learning es más evidente ya que corresponde a un problema de detección de imágenes, es decir, identificar la presencia de una hipérbola en el B-Scan. Sin embargo, esta aproximación tiene como problema que no identificaría de manera precisa la ubicación de la mina ya que el B-Scan comprende toda una línea recta. Para solventar este problema se propone utilizar un modelo cuya unidad de datos tiene dos canales: un B-Scan a lo largo del eje X y otro a lo largo del eje Y como se muestra en la , de esta manera cada punto X-Y queda identificado de manera única y la tarea de aprendizaje se vuelve más compleja ya que más que identificar la presencia de hipérbolas en ambos B-Scans se requiere identificar la presencia del ápice de la hipérbola, de forma que la ubicación espacial es más detallada. El uso de canales es usual en el procesamiento de imágenes y de audio. Por ejemplo, en imágenes a color se pueden utilizar tres canales RGB (Rojo, Verde y Azul) y en audio se tienen los canales L-R (Izquierda - Derecha). En

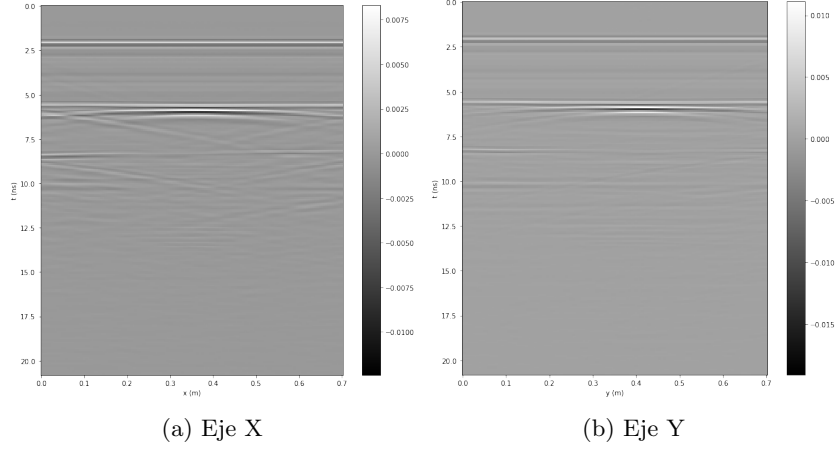


Figure 19: B-Scans Escenario 6.  $x = 0.35m$   $y = 0.35m$

La arquitectura de la red se presenta en la Tabla 7. Como se puede notar la entrada consta de un tensor de dimensión (Número de A-Scans, Número de puntos por A-Scan, Número de Canales). También se puede observar la arquitectura de manera resumida en la Fig. 20.

Table 7: Arquitectura de la red

Capa	Activation	Filters	Kernel/Pool Size	Strides	Output Shape
BatchNormalization	-	-	-	-	(141, 555, 2)
Conv2D	ReLU	16	(3, 3)	1	(139, 553, 16)
MaxPooling2D	-	-	(2, 2)	-	(46, 184, 16)
Conv2D	ReLU	32	(7, 7)	2	(20, 89, 32)
MaxPooling2D	-	-	(2, 2)	-	(10, 44, 32)
Flatten	-	-	-	-	14080
Dense	ReLU	-	-	-	64
Dropout(0.5)	-	-	-	-	64
Dense	Sigmoid	-	-	-	1

Es importante resaltar que la aproximación por canales implica un elevado consumo de memoria principal. Ya que para un solo escenario se obtiene un tensor de (Número de puntos, Número de A-Scans, Número de puntos por A-Scan, Número de canales), es decir, (141\*141, 141, 555, 2). Debido a que el escenario no es posible almacenarlo en memoria principal se adoptó una estrategia de entrenamiento por batches de 141 puntos por batch, tanto para entrenar como para predecir, para ello se creo un generador de datos que alimenta al modelo unicamente con batches de (141, 141, 555, 2), que corresponden a los puntos de un eje.

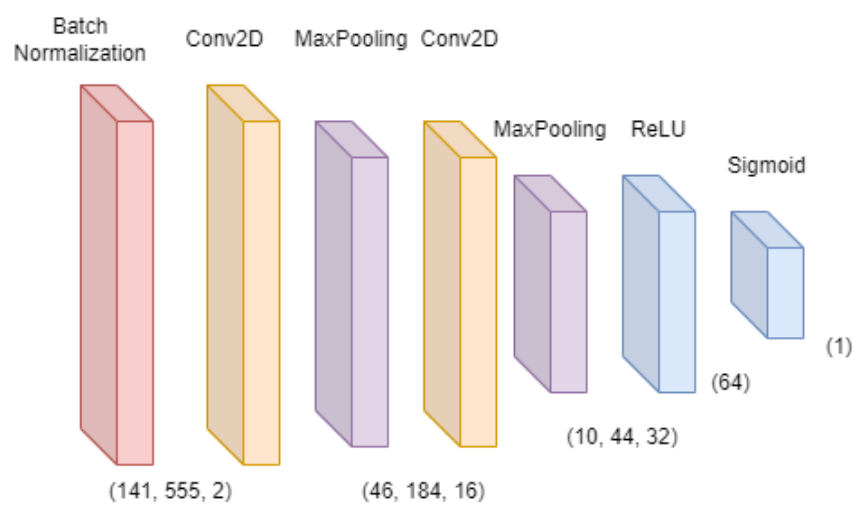


Figure 20: Arquitectura CNN 2D



## 4 Resultados

Para la evaluación del sistema se utilizaron escenarios que no fueron utilizados para el entrenamiento del modelo. A continuación se presentan los escenarios y los respectivos resultados dados por el clasificador:

### 4.1 Escenario 4: Solo arena

En este escenario no hay ningún objeto enterrado. La Fig. 21 muestra el resultado esperado y el obtenido por el clasificador. En la imagen se ve ruido, y una agrupación aproximadamente en las coordenadas (45 cm, 45 cm).

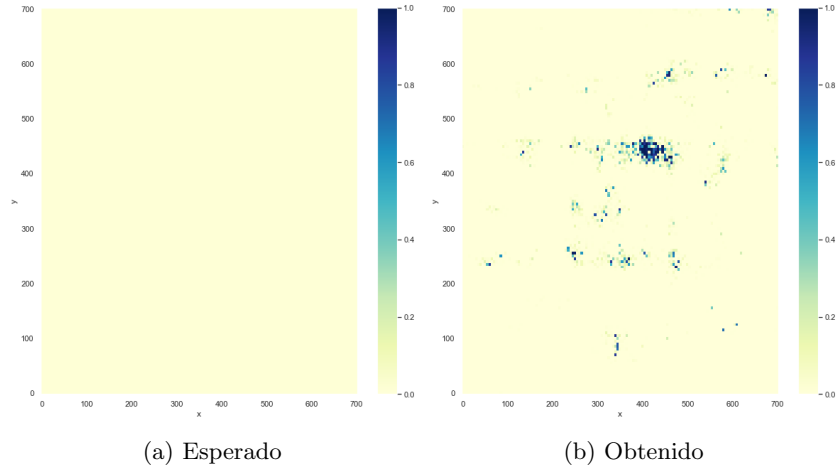


Figure 21: Resultados escenario 4 - Dictionary Learning + SVM

Para el caso de CNN de una dimensión se observan pequeñas agrupaciones en el borde superior derecho y en el borde central derecho de la caja.

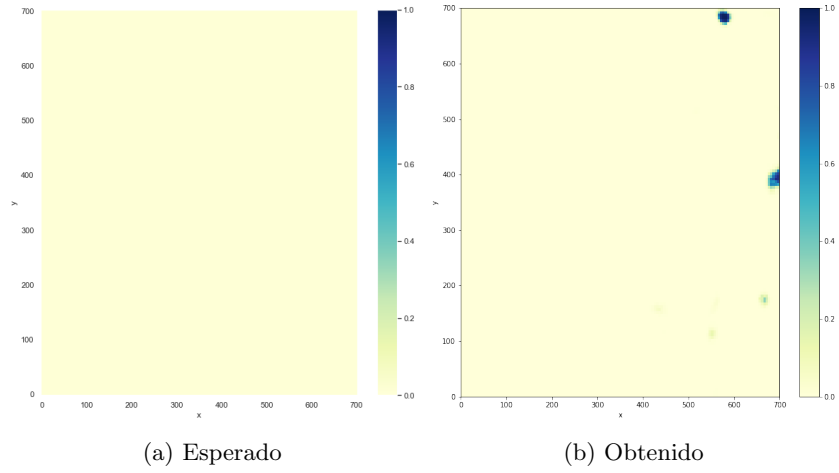


Figure 22: Resultados escenario 4 - CNN

Utilizando el modelo de CNN en dos dimensiones propuesto no se observa ningún punto con probabilidad de objetos enterrados.

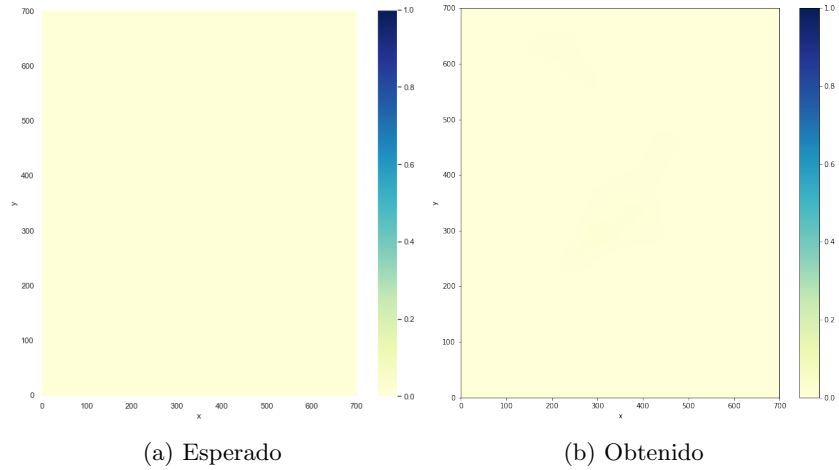


Figure 23: Resultados escenario 4 - CNN 2D

## 4.2 Escenario 7: Mina y esfera

Este escenario tiene enterrados dos objetos: En el centro del campo (35 cm, 35 cm) se encuentra una mina plástica como la mostrada en el escenario 6. Mientras que una esfera metálica de 7 cm de diámetro se encuentra en las coordenadas (45cm, 25 cm). La Fig. 24 presenta el resultado esperado por el clasificador de acuerdo con la ubicación en X-Y de los objetos, junto con los

resultados obtenidos al utilizar el sistema de detección. En la figura se observa claramente la presencia de la mina, no hay ningún punto que de indicios de la presencia de la esfera metálica.

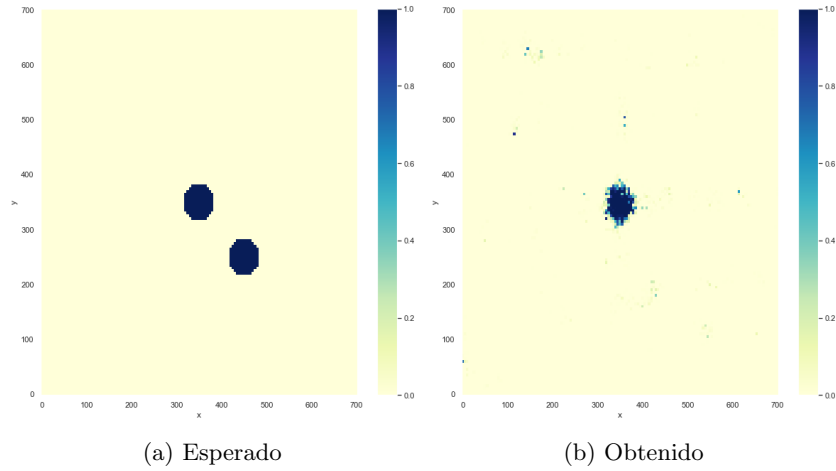


Figure 24: Resultados escenario 7

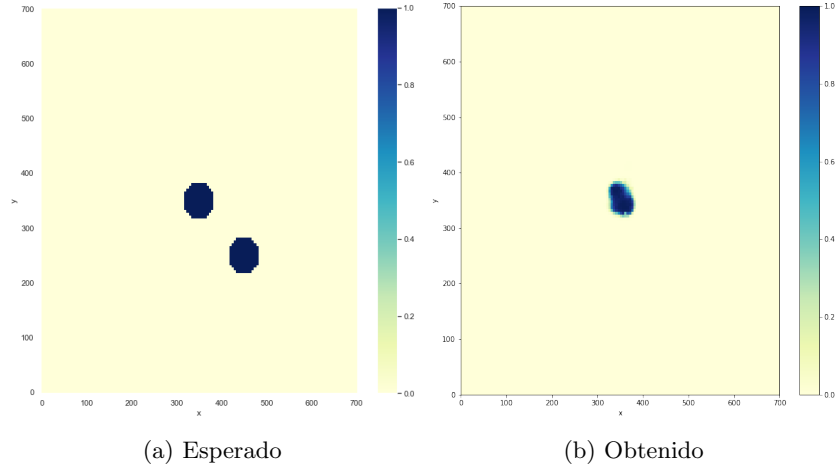


Figure 25: Resultados escenario 7 - CNN

El resultado para el escenario 7 del modelo de CNN 2D propuesto (Fig. 26), al igual que los dos anteriores, no permite observar la esfera ubicada lejos del centro del campo y existe menor definición de la forma del objeto encontrado, no obstante se nota mayor gradualidad en la probabilidad siendo esta menor hacia los bordes del objeto y mayor en el centro del mismo.

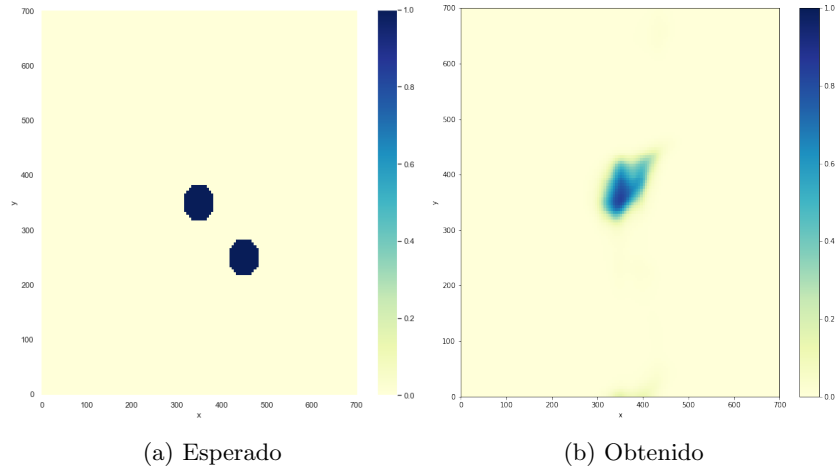


Figure 26: Resultados escenario 7 - CNN 2D

### 4.3 Escenario 8: Placa y esfera

En este escenario hay una placa metálica cuyo centro está ubicado en las coordenadas (35 cm, 35 cm) y tiene 25 cm de lado. También se encuentra enterrada una esfera metálica de 7 cm de diámetro en las coordenadas (20 cm, 50 cm). El resultado esperado y el obtenido mediante el clasificador se observa en la Fig. 27. Como se puede observar se distingue la placa metálica con precisión incluso en su forma y dimensiones, sin embargo, no hay puntos que indiquen la presencia de la esfera.

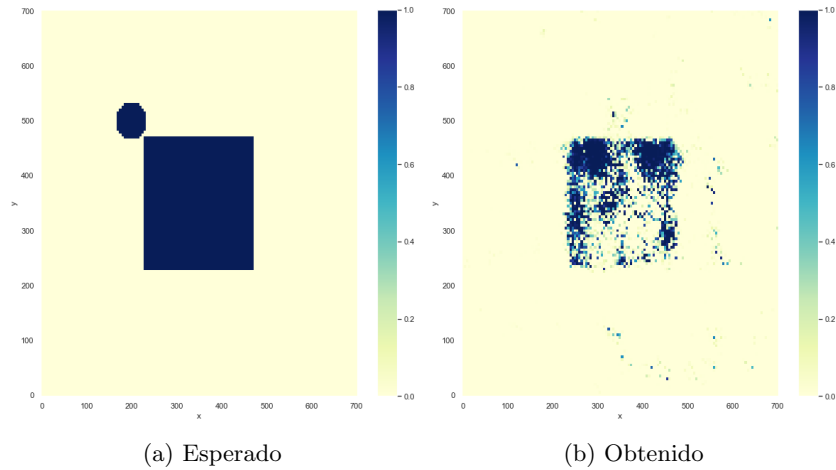


Figure 27: Resultados escenario 8

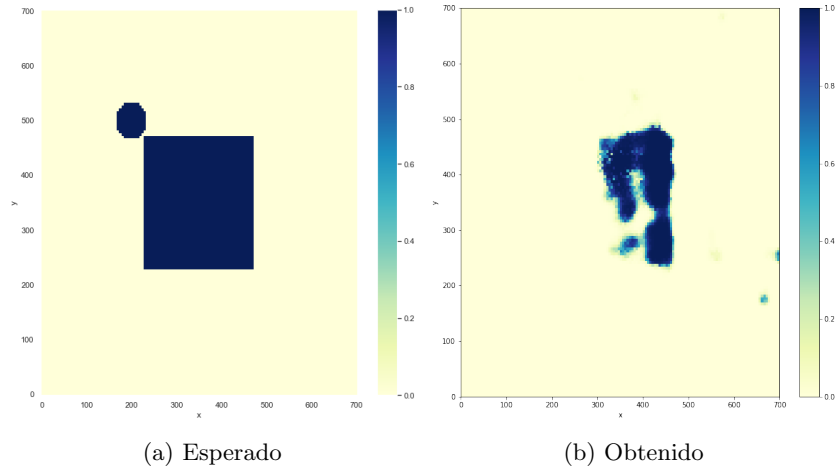


Figure 28: Resultados escenario 8 - CNN

El modelo propuesto CNN 2D presenta en el escenario 8 la mejor definición de forma y tamaño, adicionalmente se observa también la gradualidad de detección siendo menor la probabilidad cercano a los bordes de la placa metálica. No obstante al igual que los otros modelos no se evidencia la esfera que se encuentra al lado de la placa metálica.

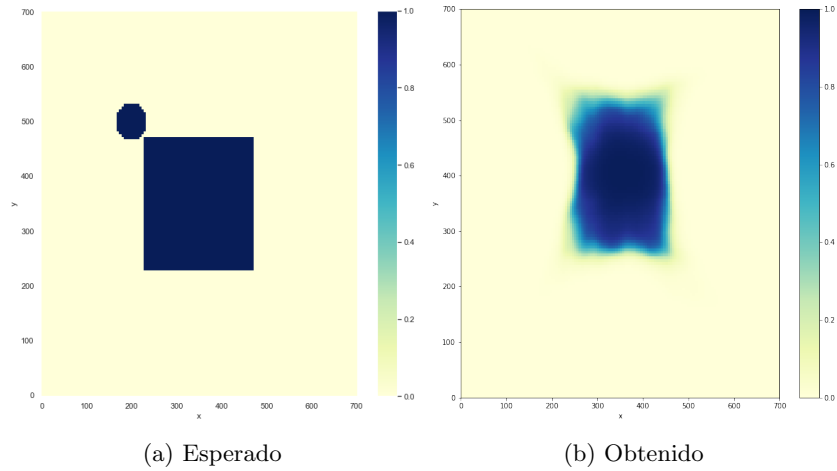


Figure 29: Resultados escenario 8 - CNN 2D

## 5 Discusión

En primer lugar, respecto a los B-Scans mostrados para los escenarios de entrenamiento se evidencia que el acople entre antenas, la superficie y el fondo son los efectos de mayor efecto visible en los B-scans. En este orden de ideas solo la placa metálica (Fig. 15) tienen una respuesta muy pronunciada para ser visible sin aplicar técnicas de remoción de escenario. Se puede observar en todos los escenarios que hay hipérbolas cuyo ápice parece corresponder a un objeto que se encuentra fuera del campo como podría ser la pared de la caja las cuales solo logran ser removidas mediante la técnica de sustracción de escenarios. Cabe resaltar que aunque a nivel general se obtuvieron mejores resultados aplicando esta técnica es poco práctica su realización.

Por otra parte, en todos los escenarios se observa que gran parte del A-scan no es útil para la detección de objetos ya que el fondo termina en aproximadamente 8 ns mientras que la señal va hasta los 46.8ns (en las figuras se presenta solo la mitad). Por lo que subir la frecuencia baja podría resultar en menor almacenamiento, mayor velocidad de toma de datos sin perder información.

De acuerdo con los resultados obtenidos se observa que tanto en los resultados del escenario 7 (Fig. 24), como en los resultados del escenario 8 (Fig. 27), la esfera metálica ubicada por fuera del centro no es detectada. Respecto a ello vale la pena comentar que aunque el modelo no recibe ninguna información explícita de la posición X-Y de los A-scans, debido a las paredes de las cajas y el patrón de radiación de las antenas, en los A-scans pueden estar presente reflexiones de estas. De forma que la falta de objetos cercanos a las paredes de la caja, en los escenarios de entrenamiento, pudo evitar que el modelo obtuviera información que le permitiera detectar objetos fuera del centro.

Respecto a la propuesta basada en detección de B-Scan de dos canales espaciales, se observa una mejora importante en cuanto a la gradualidad de la detección, mostrando de manera más precisa la forma de los objetos. Ello se observa principalmente en el escenario 8 al comparar la Fig. 29 con las Fig. 27 y 28. Lo que sugiere que hacer uso de información en dos dimensiones (parcialmente tres) da mayor información y mejora el resultado respecto a la detección utilizando únicamente A-Scans. Sin embargo, también conlleva a mayor necesidad de recursos

También es preciso mencionar que ninguno de los escenarios de entrenamiento posee más de un objeto enterrado, en este sentido, las señales recolectadas poseen interacciones más simples que los que se podrían dar en un campo real, lo que evita que los modelos aprendan particularidades de las señales ante objetos enterrados.

Adicionalmente, la presencia de pocos objetos enterrados en los escenarios de entrenamiento implica que el número de A-scans de clase 1 (hay objeto), sea muy bajo, resultando en un problema de clasificación desbalanceado, lo que resulta en un bajo desempeño de clasificación para la clase minoritaria. Por ello resulta importante la selección del recall como métrica de selección del modelo. Se puede observar que este indicador aún en el mejor modelo fue bajo (53%).

Respecto al escenario 8 (Fig. 27) vale la pena mencionar que la respuesta

de la placa metálica es bastante pronunciada, de forma que también se puede pensar que debido a su fuerte reflexión puede ocultar las reflexiones de la esfera metálica que también se encontraba enterrada en el campo de prueba, de forma que puede ser otra razón por la cual no es detectada en los mapas entregados por el sistema de detección. Esto refuerza la necesidad de utilizar escenarios más complejos para el entrenamiento de modelo, es decir, aumentar el número de objetos enterrados, mayor diversidad en la profundidad de los objetos y altura de la antena.

Otro punto a resaltar es que fueron detectados dos objetos de materiales con propiedades electromagnéticas diferentes: plástico y metal, en los escenarios 7 y 8 respectivamente.

## 6 Conclusiones

El sistema de detección muestra tener utilidad para la detección de objetos enterrados, lo que es relevante para el desminado de explosivos antipersonales. Como ventajas posee que requiere de pocos datos para ser entrenado. También se resalta la precisión de la información que puede entregar en términos de la ubicación y forma de los objetos.

Respecto a sus debilidades se propone incrementar el número de escenarios de entrenamiento, buscando escenarios más complejos en distintos términos: número, ubicaciones y profundidad de objetos. De igual forma, se pueden buscar condiciones del suelo que resulten menos homogéneas, por ejemplo distintas condiciones de humedad.

Por otra parte, el modelo se podría beneficiar de mayor especificidad en términos de los objetos a buscar, por ejemplo, que solo sea entrenado con minas plásticas. También el alcance del sistema de detección se podría limitar de manera explícita en términos de la profundidad máxima de los objetos enterrados de acuerdo con las características electromagnéticas del suelo.



## References

- [1] S. Lameri, F. Lombardi, P. Bestagini, M. Lualdi, and S. Tubaro, “Landmine detection from GPR data using convolutional neural networks,” *25th European Signal Processing Conference, EUSIPCO 2017*, vol. 2017-January, pp. 508–512, 10 2017.
- [2] F. Giovanneschi, K. V. Mishra, M. A. Gonzalez-Huici, Y. C. Eldar, and J. H. Ender, “Dictionary learning for adaptive GPR landmine classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, pp. 10036–10055, 12 2019.
- [3] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, ICML '09, (New York, New York, USA), pp. 1–8, ACM Press, 2009.
- [4] “scikit-learn: machine learning in Python — scikit-learn 1.0.1 documentation.”
- [5] R. Rubinstein, M. Zibulevsky, and M. Elad, “Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit,”
- [6] B. Schölkopf and A. J. Smola, “Learning with kernels: support vector machines, regularization, optimization, and beyond,” 2002.
- [7] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning / Ian Goodfellow, Yoshua Bengio, and Aaron Courville*. 2016.
- [8] Y. L. Cun, I. Guyon, L. D. Jackel, D. Henderson, B. Boser, R. E. Howard, J. S. Denker, W. Hubbard, and H. P. Graf, “Handwritten Digit Recognition: Applications of Neural Network Chips and Automatic Learning,” *IEEE Communications Magazine*, vol. 27, no. 11, pp. 41–46, 1989.
- [9] A. Chakravarthy, “Visualizing Intermediate Activations of a CNN trained on the MNIST Dataset — by Arnav Chakravarthy — Towards Data Science,” 6 2019.
- [10] G. Pierobon, “Visualizing intermediate activation in Convolutional Neural Networks with Keras — by Gabriel Pierobon — Towards Data Science,” 11 2018.
- [11] J. Wang, R. Turko, O. Shaikh, H. Park, N. Das, F. Hohman, M. Kahng, and P. Chau, “CNN Explainer.”

## A Implementación en Python

Para la implementación del algoritmo DL-SVM se hace uso de la librería *sklearn*. Para acelerar el entrenamiento se utilizó también, *sklearnex* (para procesadores Intel). El archivo *demining\_ml.ipynb* contiene el Jupyter Notebook con la carga de escenarios y sus etiquetado.

El paso de Dictionary Learning se realiza con *MiniBatchDictionaryLearning* y *DictionaryLearning*, mientras que el de clasificación (SVM) con *SVC*. Estos dos pasos se unen utilizando *Pipeline*.

Para el algoritmo de CNN-1D se hace uso de la librería Keras utilizando como backend TensorFlow. El archivo *GPR - CNN.ipynb* contiene el Jupyter Notebook, en el cual se leen los escenarios y se realiza el etiquetado, para este se realiza una reorganización de los A-Scans.

Para el algoritmo de CNN-2D también se hace uso de la librería Keras. El El archivo *GPR - CNN B-Scan.ipynb* contiene el Jupyter Notebook con esta implementación. Carga los archivos y realiza el etiquetado. En este caso para alimentar el algoritmo bien sea en entrenamiento o predicción se hace uso de un generador de datos en batches *DataGenerator* que entrega batches de 141 muestras.

En todos los archivos existe una función *plot\_mine\_map()* que recibe los ejes y el mapa con las probabilidades de que en un punto  $x$ ,  $y$  haya un objeto enterrado.