La Mecatrónica en México, Enero 2022, Vol. 11, No. 1, páginas 34 – 47
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

Wasserstein GAN con penalización de gradiente para generación de sonidos ambientales

González Huerta Rodrigo(✉), Ramos Arreguín Juan Manuel, Takacs Andras.

Facultad de Ingeniería, Universidad Autónoma de Querétaro, México
✉rghbecker2908@hotmail.com

Resumen

Las redes generativas antagónicas son una clase de modelo de aprendizaje profundo cuya arquitectura consta generalmente de dos redes neuronales, un generador y un discriminador. Ambos modelos se entrenan simultáneamente con el objetivo de que el generador aprenda a sintetizar instancias de datos que en el mejor de los casos difícilmente pueden distinguirse de los datos originales. En este trabajo se presenta la implementación de una clase de red generativa antagónica llamada WGAN-GP y su aplicación como generadora de sonidos ambientales. Para evaluar el modelo, se sintetizaron archivos de audio de dos clases distintas de sonidos (“crying baby” y “chainsaw”) y se analizó la influencia de esta acción en una tarea de clasificación utilizando una red neuronal recurrente. El aumento de datos mejoró la exactitud del clasificador en 2.24%. Los resultados obtenidos muestran que el enfoque propuesto puede mejorar el desempeño de los sistemas de reconocimiento de sonidos ambientales.

Palabras clave: Sonidos ambientales, red generativa antagónica, aumento de datos.

Abstract

Generative Adversarial Networks are a class of deep learning model whose architecture generally consists of two neural networks, a generator and a discriminator. Both models are trained simultaneously with the purpose that the generator learns to synthesize data instances that in the best case can hardly be distinguished from the original data. In this work, the implementation of a class of Generative Adversarial Network called WGAN-GP and its application as a generator of environmental sounds is presented. To evaluate the model, audio files of two different classes of sounds (“crying baby” and “chainsaw”) were synthesized and the influence of this action on a classification task was analyzed using a recurrent neural network. The data augmentation improved the accuracy of the classifier by 2.24%. The results obtained show that the proposed approach can improve the performance of environmental sound recognition systems.

Keywords: Environmental sounds, generative adversarial network, data augmentation.

La Mecatrónica en México, Enero 2022, Vol. 11, No. 1, páginas 34 – 47
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

1. Introducción

El éxito que puede tener incorporar técnicas de aprendizaje automático en sistemas electrónicos depende en gran medida de la cantidad de información disponible, principalmente para abarcar la alta variabilidad que exhiben ciertos tipos de datos. En algunos casos, ciertas bases de datos que podrían ser consideradas grandes no permiten entrenar un modelo lo suficiente para que exhiba el comportamiento deseado durante la realización de la tarea para la que ha sido diseñado.

Las bases de datos de sonidos ambientales (aquellos que no son lenguaje hablado o música) no son la excepción. Prácticamente cualquier clase de sonido es susceptible a presentar cierto grado de variabilidad cuando se producen y detectan en condiciones no controladas; algunos factores que influyen en la manera en como un sonido es detectado son la distancia entre la fuente y el detector, las características del equipo con el que se está llevando a cabo la detección y las condiciones del entorno donde ocurre el sonido, por ejemplo, si ocurre en un entorno aislado o con ruido de fondo. Dependiendo de la manera como son producidos y la manera como son detectados, podría concluirse que los sonidos difícilmente pueden replicarse exactamente. Sin embargo, tener en cuenta las variaciones posibles en los datos provenientes de una misma clase de sonido es un detalle importante que deberían incorporar los sistemas que pretendan ser de utilidad en la vida cotidiana.

Existen maneras de atacar este inconveniente. El aumento de datos es una técnica que se ocupa de este problema mediante la generación de datos sintéticos a partir de las muestras disponibles de bases de datos. Algunos métodos de aumento de datos que funcionan bien en otros dominios, por ejemplo en el procesamiento de imágenes, se han adaptado bien al dominio del sonido. Un ejemplo de esto es la adición de ruido aleatorio. Otras transformaciones básicas incluyen el cambio de tono, la compresión de rango dinámico y el estiramiento del tiempo. El problema con las técnicas de aumento de datos “clásicas” es que su aplicación está restringida al problema que se está intentado resolver.

La idea básica detrás de estas técnicas es que al aplicar una transformación a los datos, se apliquen de tal manera que el significado semántico de las etiquetas asociadas a los datos no cambie. Las ventajas de aplicar alguna técnica de aumento de datos son las siguientes:

- Aumenta el tamaño de los datos disponibles para entrenar un modelo.
- Hace que el modelo sea más robusto ante variaciones de los datos.
- Disminuye el sobreajuste del modelo (overfitting).

En cuanto a las aplicaciones que pueden beneficiarse de la incorporación de algún método de aumento de datos, se encuentran los sistemas de reconocimiento automático de sonidos. A través de estos sistemas es posible detectar y reconocer fuentes de contaminación acústica en una ciudad, mejorar sistemas de vigilancia de tal manera que permita reconocer eventos particulares como disparos, ruptura de cristales o ladridos de perros. Otro tipo de aplicaciones con potencial incluyen la generación de respuestas más rápidas en caso de accidentes, por ejemplo caídas, accidentes automovilísticos o explosiones. Las personas que de alguna manera dependen de algún tipo de asistencia, como son los ancianos o las personas con ciertos tipos de discapacidades, como sordera o ceguera, también podrían beneficiarse de esta tecnología.

Si bien el desarrollo de técnicas de aumento de datos no es algo novedoso, hasta hace poco se comenzaron a buscar soluciones a través de la implementación de modelos de aprendizaje profundo. Se ha demostrado que las redes generativas antagónicas, que son sistemas que utilizan simultáneamente dos modelos de red neuronal, tienen un buen desempeño para atacar el problema de escasez de datos. Sin embargo, a pesar de su utilidad que ya ha sido demostrada en aplicaciones para el área de procesamiento de imágenes, el uso de esta tecnología para sintetizar sonido se encuentra en una etapa temprana. Donahue et al. (2019) fueron los primeros que propusieron el uso

de redes generativas antagónicas para sintetizar sonidos utilizando archivos de audio crudos (WaveGAN) e imágenes de espectrogramas (SpecGAN) [1].

En este artículo se muestran los resultados de implementar un tipo de red generativa antagónica llamada WGAN-GP (Wasserstein Generative Adversarial Network with Gradient Penalty), basada en el modelo WaveGAN. Adicionalmente, en la sección de resultados se analiza la influencia de la GAN en una tarea de clasificación de sonidos ambientales. Como clasificador se utilizó una red neuronal recurrente, mientras que el rendimiento del método fue evaluado utilizando una base de datos de acceso público.

2. Marco Teórico

2.1 Redes Generativas Antagónicas

Las redes generativas antagónicas, conocidas en inglés como GANs (Generative Adversarial Networks), son una clase de arquitectura de red neuronal utilizada para sintetizar datos. Los datos pueden ser imágenes de rostros humanos u obras de arte, cadenas de texto, audio, como música, sonidos ambientales o lenguaje hablado. También se han utilizado para sintetizar moléculas con propiedades farmacológicas específicas. Es una tecnología que se ha popularizado ampliamente desde que fue introducida en 2014 por Ian J. Goodfellow [2].

Una red generativa antagónica está compuesta de dos componentes principales: un modelo generador y un modelo discriminador, usualmente redes neuronales profundas. El objetivo del modelo generador es capturar la distribución de los datos de entrenamiento y generar nuevas instancias a partir de estos. Por otro lado, la función del discriminador consiste en determinar si las muestras que se le presentan provienen de los datos originales o si fueron producidas por la red generadora. Ambos modelos se entrenan simultáneamente de forma "adversarial" de tal manera que para entrenar correctamente al generador se debe maximizar la probabilidad de que el discriminador cometa un error, mientras que se intenta maximizar la probabilidad de que el discriminador asigne correctamente la etiqueta correspondiente a las muestras que se le presentan. La dinámica de esta competencia impulsa que ambos modelos mejoren su desempeño progresivamente.

Como se mencionó, una red generativa antagónica puede aprender a replicar la distribución de probabilidad de un conjunto de datos y sintetizar nuevas instancias de estos. El proceso consiste en generar un vector **z** de valores aleatorios obtenido de otra distribución conocida, por ejemplo, una distribución normal o una distribución uniforme $p_z(\mathbf{z})$. Este vector latente es mapeado a la distribución de los datos observados a través de la red neuronal generadora $G(\theta_g, \mathbf{z})$. La utilización de una red neuronal tiene como ventaja que no es necesario conocer explícitamente la forma dicha distribución. También se define una red discriminadora $D(\theta_d, \mathbf{x})$ cuya salida $D(\mathbf{x})$ representa la probabilidad de que **x** sea una muestra sintetizada por el generador [2].

En cuanto al entrenamiento de la red generativa antagónica original propuesta en [2], el entrenamiento del discriminador consiste en maximizar la probabilidad de asignar la etiqueta correcta a las muestras que se le presentan, mientras que el generador debe minimizar la expresión $\log(1 - D(G(\mathbf{z})))$. Esta dinámica de entrenamiento puede representarse mediante la ecuación (1):

$$\min_G \max_D \quad V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log\left(1 - D\left(G(z)\right)\right)] \qquad (1)$$

Idealmente, la red se detiene llegado el punto en que los datos generados se vuelven indistinguibles de los datos originales. Sin embargo, el problema con las GANs "clásicas" es que adolecen de inestabilidad durante el entrenamiento de las dos redes. Ambas redes actualizan sus funciones de pérdida de forma simultánea e independiente y esto trae como consecuencia que no

existe ninguna garantía de convergencia. Además, el desempeño del sistema es muy sensible a la elección de los hiperparámetros, lo cual puede provocar fácilmente la desestabilización del sistema. Controlar el desempeño del discriminador mediante este planteamiento resulta particularmente difícil. Originalmente, como función de pérdida del discriminador se utilizó la divergencia de Jensen-Shannon para determinar la distancia entre la distribución del modelo generador y la distribución objetivo [2]. Sin embargo, la divergencia de Jensen-Shannon no siempre es continua con respecto a los parámetros del generador, lo cual conduce al fenómeno conocido como desvanecimiento de gradiente.

Para remediar algunas de estas deficiencias, se ha analizado el comportamiento de las redes generativas antagónicas utilizando distintas funciones de pérdida. Una de estas es la función de pérdida de Wasserstein, basada en una métrica conocida como distancia de Wasserstein. Esta función fue diseñada con el objetivo de evitar el desvanecimiento de gradiente y brindar más estabilidad a los dos modelos durante el entrenamiento [3].

2.2 Wasserstein GAN

Para mejorar el funcionamiento de la red generativa antagónica, se propuso cambiar la función de pérdida por una que tuviera la propiedad de ser continua y diferenciable, proporcionando un gradiente lineal incluso cuando el discriminador estuviera bien entrenado. Una función de pérdida que logra esto es la pérdida de Wasserstein.

La pérdida de Wasserstein fue propuesta en 2017 como una alternativa a la pérdida de Jensen-Shannon [3]. Dicha métrica calcula la distancia entre dos distribuciones de probabilidad en términos del costo de convertir una distribución en la otra.

El beneficio de la pérdida de Wasserstein es que proporciona un gradiente útil en casi todas partes, permitiendo el entrenamiento continuo de los modelos. Sin embargo, para garantizar que esta medida sea válida, se requiere garantizar que el discriminador cumpla con la propiedad de continuidad 1-Lipschitz [3]. Para lograr esto, se sugirió constreñir los pesos del discriminador [3]. Sin embargo, en el mismo trabajo se concluye que esta estrategia no es suficientemente sofisticada para garantizar la convergencia del modelo. Como alternativa, se propuso reemplazar el método de constricción de pesos con una penalización de gradiente que hace cumplir la misma condición [4]. Esto permite un entrenamiento más estable y requiere muy poco ajuste de los hiperparámetros de la red.

Es importante mencionar que la implementación de la función de pérdida de Wasserstein cambia la noción del discriminador a la de crítico, pues con este cambio la salida del modelo pasa de representar la probabilidad de que una muestra sea sintetizada por el generador, a representar la calidad de la muestra generada.

2.3 WaveGAN

WaveGAN es el primer modelo de una red generativa antagónica que fue utilizada para sintetizar archivos de audio [1]. Su arquitectura se basa en el modelo DCGAN (Deep Convolutional Generative Adversarial Networks) que popularizó el uso de las GANs para sintetizar imágenes [5]. La característica principal de la DCGAN tiene que ver con la utilización de la operación de convolución transpuesta, la cual genera mapas de características de baja resolución a partir de un vector latente de baja dimensionalidad; esto con el objetivo de construir una imagen de alta resolución.

Sin embargo, la WaveGAN modifica la operación de convolución transpuesta de la DCGAN, bajando la dimensionalidad de los filtros, de dos a una sola dimensión. Además, utiliza filtros más largos; en lugar de filtros cuadrados de tamaño (5,5) utiliza filtros unidimensionales de longitud 25. El

"stride" o paso del filtro también crece en una de las dos dimensiones y se reduce en la otra; pasa de ser de tamaño (2,2) a (4,1) [1].

Estos cambios dan como resultado que la WaveGAN tenga el mismo número de parámetros y el mismo número de valores de salida que una DCGAN (64x64 = 4096). Sin embargo, se agrega una capa de convolución transpuesta adicional para que el modelo arroje muestras de tamaño 16384, equivalente a un poco más de un segundo de duración a una tasa de muestreo de 16 kHz [1].

3. Materiales y base de datos

La red generativa antagónica fue implementada utilizando el lenguaje de programación Python y la distribución Anaconda, en conjunto con las librerías Tensorflow, Keras, Numpy, Librosa y Matplotlib. El equipo en el cual se desarrolló el programa es una computadora Asus Tuf Gaming A15 con un procesador Ryzen 5 (4000 series), 24 GB de memoria RAM y tarjeta gráfica NVIDIA GeForce GTX 1660Ti.

La base de datos que se utilizó para realizar pruebas es la ESC-10 [6]. Esta colección de archivos de audio está conformada por 400 archivos que poseen las siguientes características:

- Duración de 5 segundos.,
- Frecuencia de muestreo de 44.1 kHz
- Un solo canal (mono)
- Formato de compresión Ogg Vorbis a 192 kbit/s.

.
Las clases en las que está dividida la base de datos son 10 (40 archivos por clase):

- Sonidos transitorios: estornudos, ladridos de perro y reloj.
- Sonidos con alto contenido armónico: llanto de bebés, gallo cacareando.
- Sonidos cuasi-estructurados: lluvia, olas de mar, fuego, helicóptero, sierra eléctrica.

4. Metodología

4.1 Preprocesamiento

Dado que el número de archivos disponibles por clase es bajo y que se utiliza una sola clase por cada entrenamiento, es muy probable que la red presente un comportamiento de sobreajuste, lo que resultará en un rendimiento deficiente. Por esta razón, se optó por dividir cada archivo en cinco archivos de 1 segundo. De esta forma se pasa de 40 archivos por clase a 200.

Una vez que se tiene la base de datos ampliada se debe seleccionar alguna de las clases con la que se planea experimentar; para este trabajo se utilizaron las clases “baby crying” y “chainsaw”. Para reducir el costo computacional durante el entrenamiento se aplicó una tasa de muestreo de 16384 muestras por segundo. Este valor está basado en [5] y se utiliza como la dimensión de entrada al modelo discriminador. La forma del arreglo que se obtiene es de (200, 16384, 1).

4.2 Arquitectura

La arquitectura de la red generativa antagónica se basa en el modelo WaveGAN [5]. Un cambio realizado respecto al modelo original, consistió en utilizar funciones de activación LeakyReLu en el discriminador en lugar de funciones de activación ReLu, después de cada capa convolutiva. Este

cambio se hizo con el objetivo de preservar los atributos negativos de las señales de audio. La única excepción es la función de activación de la capa de salida, la cual es lineal. Esta función se utiliza para obtener un valor que represente una puntuación en lugar de una probabilidad, como en el caso de la DCGAN, la cual utiliza una función sigmoide en su capa de salida.

Por otro lado, se implementaron capas de perturbación de fase (phase shuffle) como se indica en [1], con el objetivo de eliminar artefactos de ruido que se presentan al hacer uso de las capas de convolución transpuestas y que facilitan la tarea del discriminador. Esta técnica consiste en perturbar de manear aleatoria la fase de cada mapa de características antes de ingresar a la siguiente capa. La estructura del modelo discriminador se muestra en la figura 1.

En cuanto al modelo generador, su arquitectura está conformada de tal forma que a la salida entrega un vector de tamaño 16384, equivalente a un segundo de audio a una tasa de muestreo de 16 kHz. Por otro lado, la entrada a esta red es un vector latente de dimensiones (100,1) obtenida de una distribución de probabilidad normal en el rango (-1,1). Las capas que conforman el modelo generador se muestran en la figura 2. La función de activación de salida que fue utilizada es la función tangente hiperbólica.

Para ambos modelos se utilizó el algoritmo de optimización Adam con valores $\beta1 = 0.5$, $\beta2 = 0.9$ y un learning rate de 0.0001. Tanto el generador como el discriminador utilizan funciones de pérdida basadas en la de Wasserstein. Además, se utilizó una penalización de gradiente con el valor de lambda = 10.

4.3 Entrenamiento

La dinámica del entrenamiento de la red generativa antagónica consiste en actualizar cinco veces los pesos del discriminador por una actualización de los pesos del generador. Una iteración del entrenamiento del discriminador debe seguir los siguientes pasos:

- Se genera un conjunto de imágenes falsas (sintetizadas por el generador).
- Se pasa un batch (lote) de imágenes reales al discriminador.
- Se pasa el conjunto de imágenes sintetizadas al discriminador.
- Se calcula la penalización.
- Se calcula la pérdida de Wasserstein del discriminador.
- Se calcula el gradiente de la función de pérdida con respecto a los pesos de la red.
- Se actualizan los pesos.

Para el generador se sigue algo similar:

- Se genera un batch de imágenes falsas.
- Se “alimenta” el discriminador con el batch de imágenes falsas.
- Se calcula la pérdida utilizando la función de pérdida del generador (diferente de la del discriminador).
- Se calcula el gradiente de la función de pérdida con respecto a los pesos de la red.
- Se actualizan los pesos.

La dinámica del sistema se ilustra en la figura 3.

La Mecatrónica en México, Enero 2022, Vol. 11, No. 1, páginas 34 – 47
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

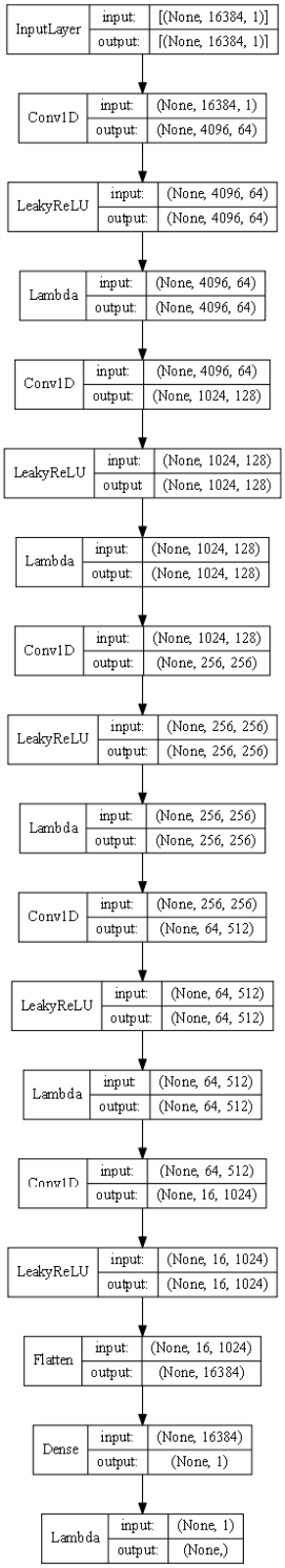

Figura 1: Arquitectura del modelo discriminador

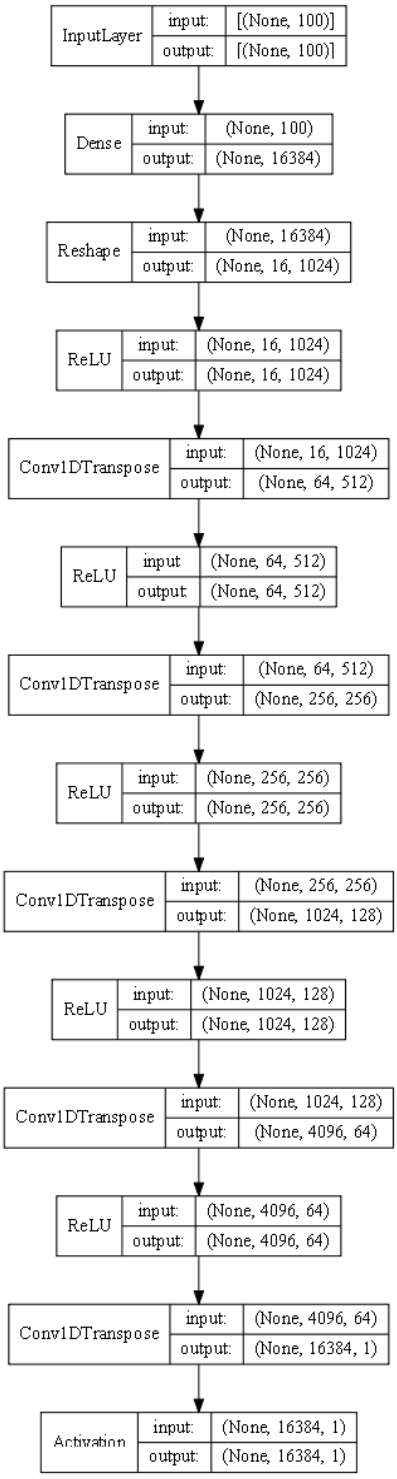

Figura 2: Arquitectura del modelo generador

La Mecatrónica en México, Enero 2022, Vol. 11, No. 1, páginas 34 – 47
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

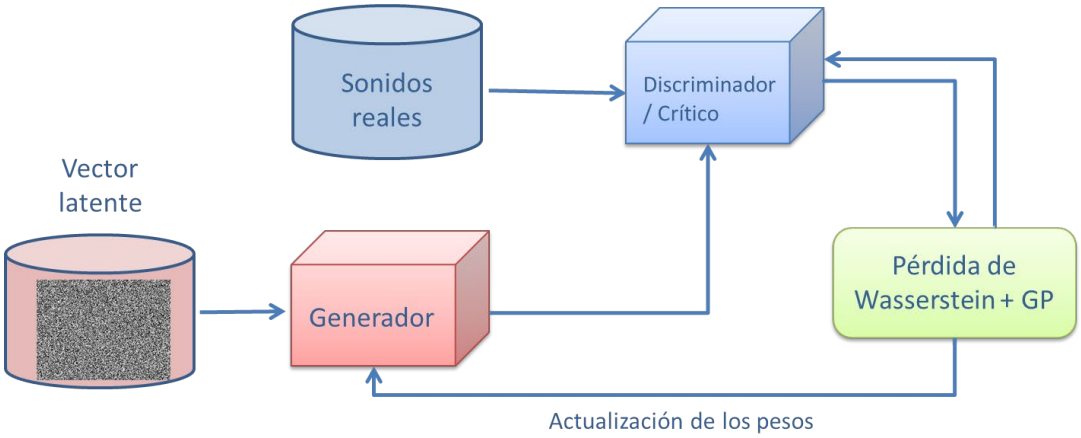

Figura 3: Diagrama de flujo de la dinámica de una Wasserstein con penalización de gradiente.

5. Resultados

5.1 Generación de sonidos ambientales

En esta sección se presentan los resultados de dos pruebas que se realizaron utilizando dos de las diez clases de la base de datos ESC-10. Las clases que se utilizaron fueron “crying_baby” y “chainsaw”. El número de épocas para cada prueba fue de 2500 (14 horas aproximadamente) con un tamaño de batch de 64. Los parámetros utilizados se muestran en la tabla 1. Es importante mencionar que aunque el entrenamiento tarda varias horas, una vez que se cuenta con el modelo generador entrenado, sintetizar un lote de archivos de audio se puede realizar en cuestión de segundos.

Tabla 1. Hiperparámetros de la WGAN-GP

Nombre	Valor
Tamaño del batch	64
Épocas	2500
WGAN-GP (λ)	10
Actualizaciones de D por una de G	5
Optimizador	Adam: α=0.0001, β_1=0.5, β_2=0.9
Perturbación de fase	2
Tamaño del vector latente	100

Para la clase “crying_baby” se pudo observar el comportamiento de la figura 4. La primera y la segunda gráfica contienen los valores de pérdida del generador y del discriminador, respectivamente, mientras que la tercera gráfica muestra los valores de la penalización.

La Mecatrónica en México, Enero 2022, Vol. 11, No. 1, páginas 34 – 47
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

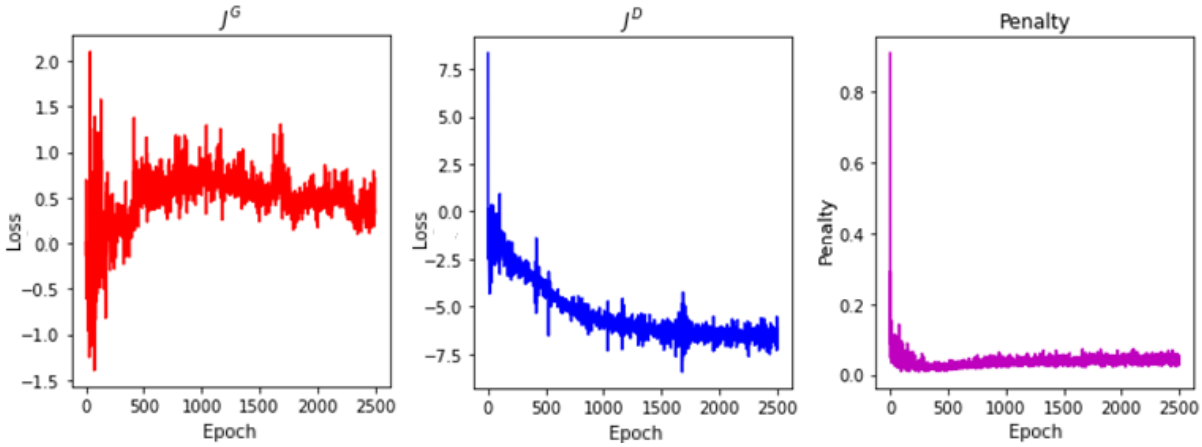

Figura 4: Comportamiento del entrenamiento de la GAN utilizando la clase "baby_crying".

Como se puede apreciar la pérdida del generador oscila entre cero y uno, mientras que la del discriminador comienza cercana a cero y disminuye conforme aumentaron las épocas hasta un valor cercano a -6. En cuanto a los sonidos generados, se puede apreciar claramente el llanto de bebés a partir de la época 1000, aproximadamente. La calidad de los sonidos aumenta conforme aumentan las épocas, aunque nunca llegan a ser “perfectos”, en el sentido de que estén libres de ruido. Aún en la época 2500 se percibe fácilmente un ruido similar a interferencia. Sin embargo, esto no impide que se pueda percibir claramente el llanto de los infantes.

Para la clase “chainsaw” se ejecutó el programa utilizando los mismos parámetros. Las gráficas obtenidas se muestran a continuación:

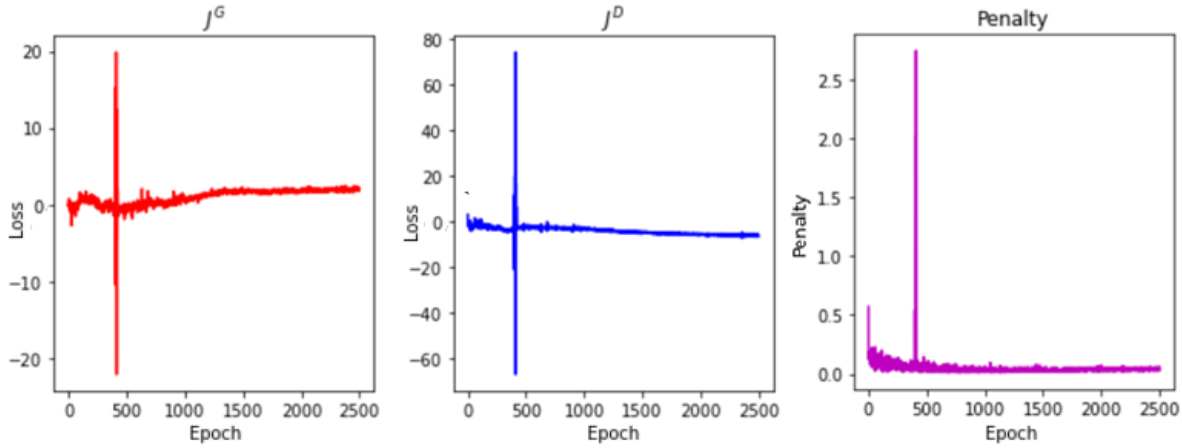

Figura 5: Comportamiento del entrenamiento de la GAN utilizando la clase "chainsaw".

La convergencia del discriminador es muy similar a la de la prueba anterior. Sin embargo, en esta prueba se presentó un pico en una de las épocas que al parecer se debe a una desestabilización del sistema que es corregido inmediatamente después. En este caso también se logra identificar la clase con la que se entrenó a la red al escuchar los archivos generados. Sin embargo, por tratarse de un aparato mecánico, resultó más difícil evaluar la calidad de los sonidos.

La Mecatrónica en México, Enero 2022, Vol. 11, No. 1, páginas 34 – 47
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

Como pruebas adicionales se modificaron algunos de los hiperparámetros de la red y se volvió a ejecutar el código para la clase “crying_baby”. La primera prueba que se realizó fue aumentando el valor del learning rate, de 0.0001 a 0.0002. Al igual que en los experimentos anteriores, se pudo observar que la pérdida del discriminador converge a un valor cercano a -6. También se pudo apreciar que el sistema comenzó a oscilar pronunciadamente a partir de una época cercana a la 1500. La calidad de los sonidos es parecida a la de la prueba anterior, aunque esto es un juicio meramente subjetivo. Las gráficas obtenidas se muestran a continuación:

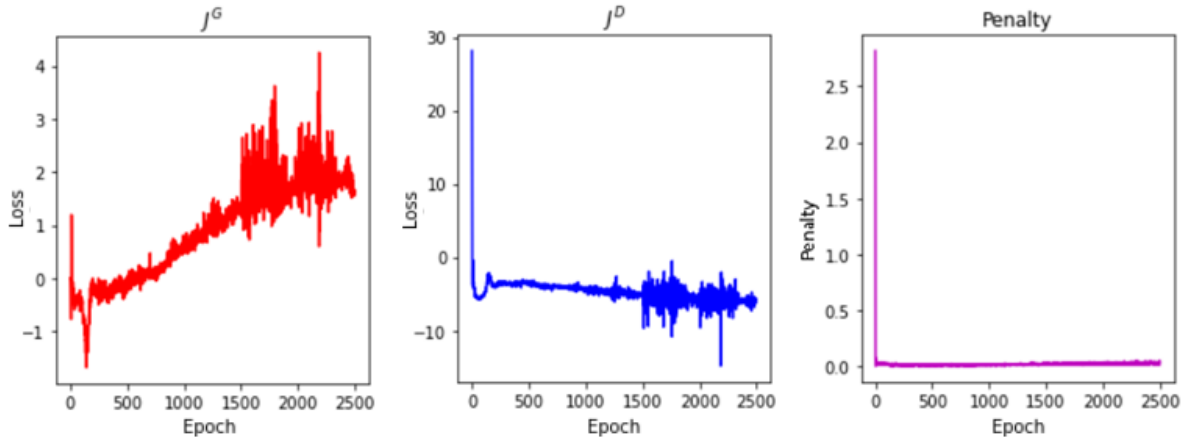

Figura 6: Comportamiento del entrenamiento de la GAN utilizando la clase "chainsaw" con learning rate de 0.0002.

También se realizó una prueba cambiando el algoritmo de optimización Adam por el algoritmo RMSprop (Root Mean Square Propagation) (figura 7). Los resultados obtenidos fueron muy similares a los de las pruebas anteriores. Es importante observar que en todas las pruebas se obtuvieron valores pequeños de la penalización, lo cual es un comportamiento deseado.

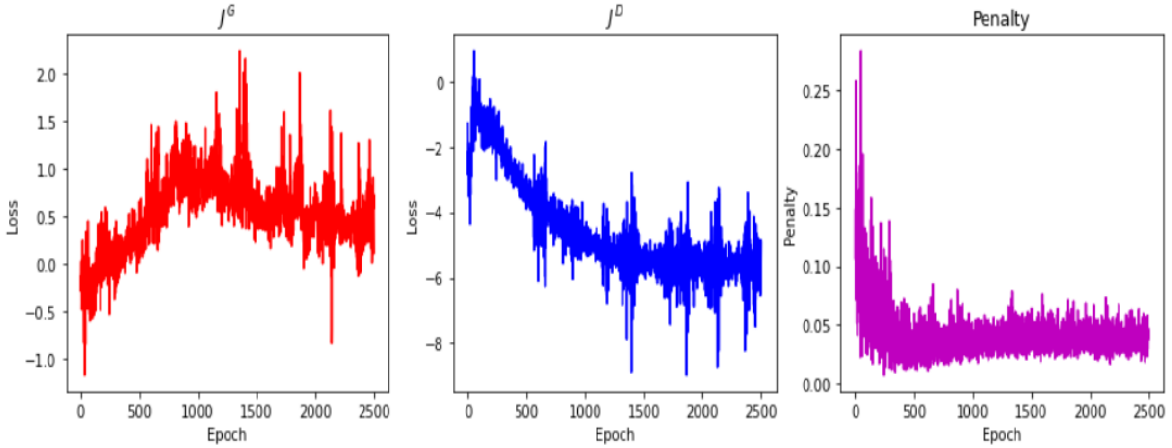

Figura 7: Comportamiento del entrenamiento de la GAN para la clase "crying_baby" utilizando el algoritmo de optimización RMSprop con un learning rate de 0.0001.

La Mecatrónica en México, Enero 2022, Vol. 11, No. 1, páginas 34 – 47
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

5.2 Evaluación

Para evaluar el funcionamiento de la red generativa se analizó la influencia que tienen los sonidos generados al agregarlos a los archivos originales y utilizar los nuevos conjuntos de datos en una tarea de clasificación. Como clasificador se implementó una red neuronal recurrente con entropía cruzada categórica como función de pérdida, Adam como algoritmo de optimización y un parámetro de aprendizaje de 0.001. La arquitectura de la red neuronal recurrente se presenta en la figura 8.

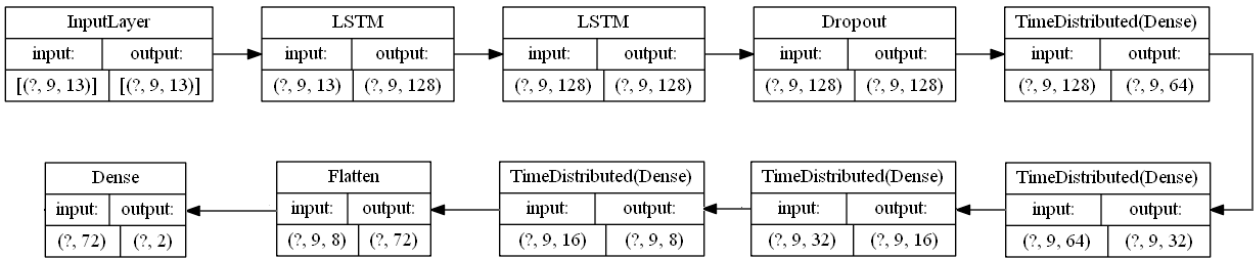

Figura 8: Arquitectura de la red neuronal recurrente utilizada para realizar la clasificación de las dos clases "aumentadas".

Originalmente los dos conjuntos que fueron incrementados (“baby_crying” y “chainsaw”) contenían 40 archivos cada uno. Después del entrenamiento de la GAN con cada una de estas clases, se seleccionaron de manera aleatoria 40 de los archivos generados por cada clase y se agregaron a las carpetas donde se encuentran los archivos originales. Esta estrategia para incrementar el tamaño de la base de datos no garantiza que todos los archivos sean representativos de la clase a la que pertenecen. Siguiendo estos pasos, cada clase pasó de tener 40 archivos a 80.

El siguiente paso fue establecer la metodología a seguir para el preprocesamiento de los archivos, previo a la clasificación:

1. Limpieza de los archivos a través de la eliminación de los silencios. El umbral de la máscara fue de 0.0001.
2. Creación de un nuevo conjunto de datos, formado por extractos de los archivos de audio de la base de datos (aumentada o sin aumentar). Para definir el número de extractos se usa (2).

$$n = \frac{\sum_{i=1}^{2} \mu_i}{0.1} \tag{2}$$

   Siendo μ_i la duración promedio de los archivos de audio de cada clase y 0.1 la duración de los extractos, en este caso una décima parte de segundo. Para la obtención de un extracto se selecciona una clase aleatoriamente y después se toma uno de los archivos de esa clase, también de manera aleatoria. Después se posiciona al azar una ventana y se toma un número de muestras equivalentes a una décima parte de segundo. El número de extractos varía dependiendo el tiempo removido en el primer punto (eliminación de silencios), pero es de aproximadamente 4500 para la base de datos aumentada.
3. Obtención de los coeficientes cepstrales en la frecuencia Mel (MFCCs) para cada extracto. Estos son los datos que ingresan al clasificador.
4. Implementación de un modelo de validación K-Fold Cross-Validation (K = 5) para la red neuronal recurrente (número de épocas = 100, tamaño de batch = 32). Se divide el conjunto de extractos en cinco particiones (folds) y en cada iteración de las K iteraciones se utilizan cuatro de estas particiones como conjunto de entrenamiento y la partición restante como conjunto de prueba. En cada iteración se alterna la partición que hace de conjunto de prueba.

La Mecatrónica en México, Enero 2022, Vol. 11, No. 1, páginas 34 – 47
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

El preprocesamiento y el entrenamiento del clasificador fueron realizados para las siguientes configuraciones de los archivos de audio:

1.- Las dos clases sin aumento de datos.
2.- Las dos clases con aumento de datos.
3.- Las dos clases con aumento de datos pero con los archivos generados intercambiados (baby_crying + chainsaw_augmented , chainsaw + baby_crying_augmented).

La última prueba se realiza con el objetivo de intentar confundir a la red, ya que si los sonidos sintetizados tienen una calidad aceptable, se espera que al intercambiarlos la red no sea capaz de hallar una manera de distinguir ambas clases y por lo tanto su decisión sería similar a la de una moneda tirada al aire. Los resultados de correr el experimento propuesto se muestran en las figuras 9, 10 y 11.

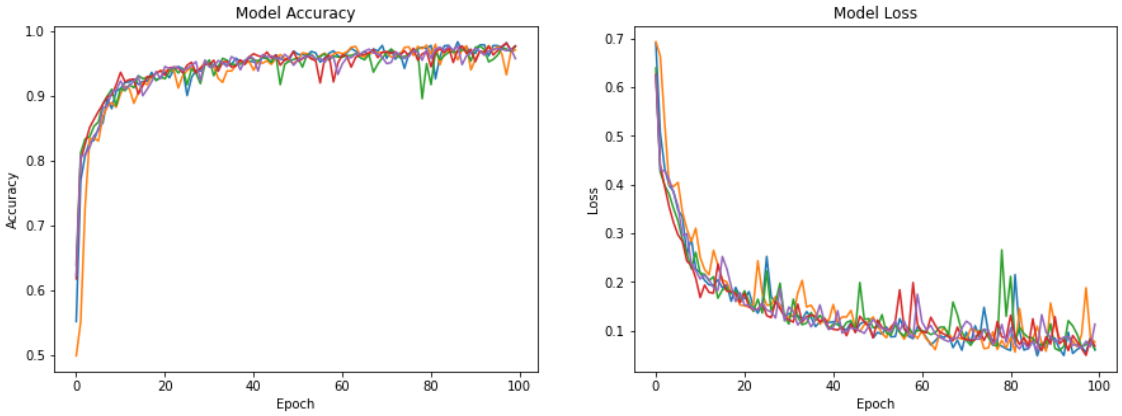

Figura 9: Validación del clasificador utilizando el conjunto sin aumento de datos. Cada línea representa una iteración del modelo K-Fold Cross-Validation. (a) Exactitud. (b) Pérdida.

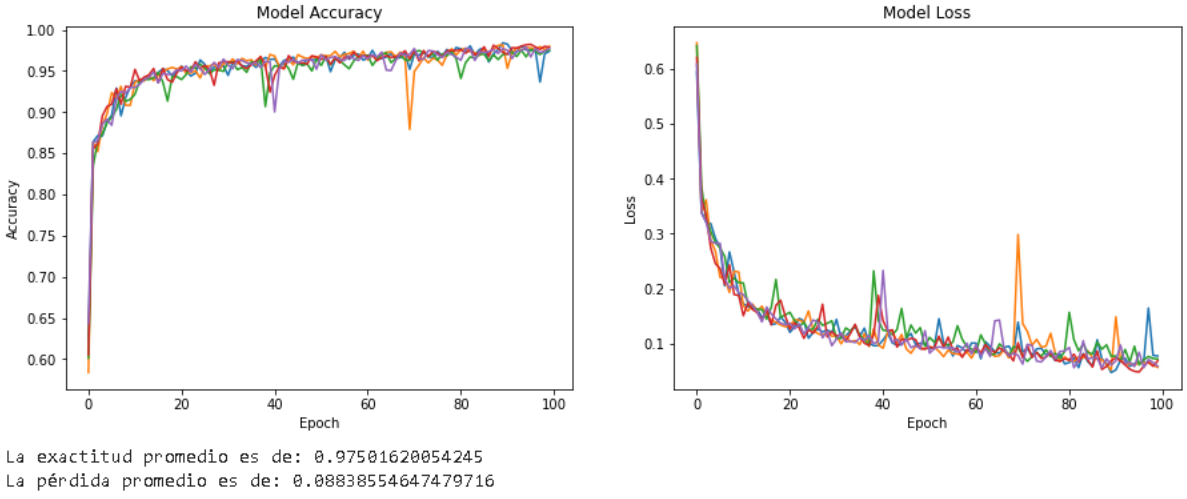

Figura 10: Validación del clasificador utilizando el conjunto con aumento de datos. Cada línea representa una iteración del modelo K-Fold Cross-Validation. (a) Exactitud. (b) Pérdida.

La Mecatrónica en México, Enero 2022, Vol. 11, No. 1, páginas 34 – 47
Disponible en línea en www.mecamex.net/revistas/LMEM
ISSN: 2448-7031, Asociación Mexicana de Mecatrónica A.C

Para el primer experimento (figura 9), “clasificación sin aumento de datos”, se obtuvo una exactitud promedio de 95.26% y una pérdida promedio de 0.12. De las gráficas de pérdida y de exactitud se puede apreciar un comportamiento bastante estable para cada una de las particiones. En el siguiente experimento (figura 10), “clasificación con aumento de datos” se obtuvo una mejora en la exactitud promedio, 97.5%. Una diferencia de 2.24%. Por otro lado, la pérdida en este experimento también fue menor respecto al valor obtenido en el anterior, 0.088.

En cuanto al tercer experimento (figura 11), “clasificación con los archivos generados intercambiados”, se obtuvo un valor de exactitud y de pérdida que refleja el comportamiento esperado y que fue explicado en el párrafo anterior. Debido a que ambos conjuntos poseen sonidos de ambas clases, uno los sonidos originales de “baby_crying” y los sonidos sintetizados de “chainsaw” y el otro los sonidos originales de “chainsaw” y los sonidos sintetizados de “baby_crying”, se esperaba un valor de exactitud cercano al 50%.

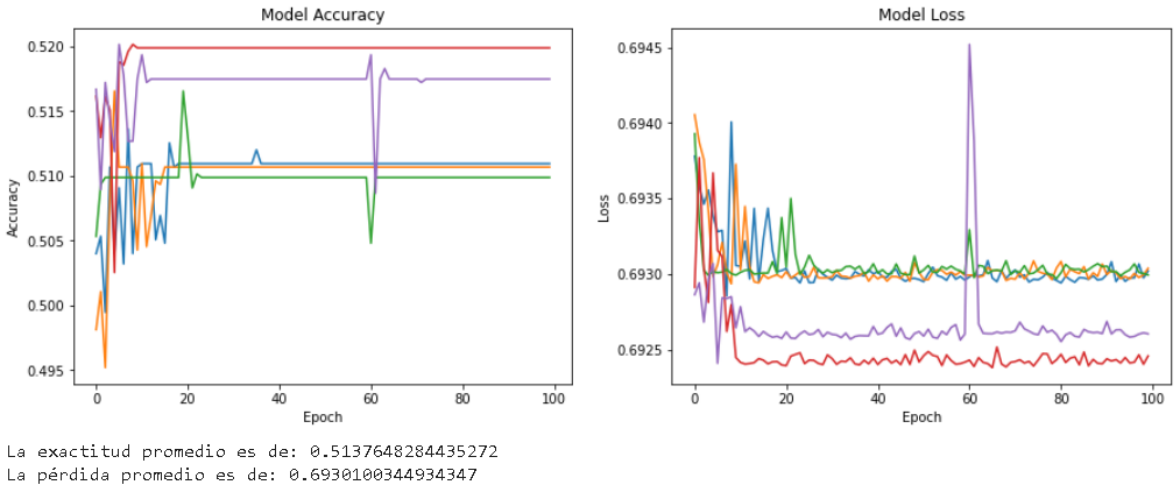

Figura 11: Validación del clasificador utilizando el conjunto con aumento de datos pero intercambiados. (a) exactitud de cada partición de los datos. (b) pérdida para cada partición.

6. Conclusiones

En este trabajo se implementó un modelo de red generativa antagónica conocida como WGAN-GP, basado en el modelo WaveGAN, el cual fue diseñado con el propósito de sintetizar archivos de audio. El modelo fue entrenado en varias ocasiones con dos clases distintas (“chainsaw” y “baby_crying”). Posteriormente se evaluó la influencia de los datos generados a través de una tarea de clasificación. Para validar el desempeño del clasificador se utilizó un modelo de validación K-Fold Cross-Validation (K = 5).

Los resultados obtenidos en la tarea de clasificación sugieren que los datos sintetizados no afectan el desempeño del clasificador, pero tampoco mejoraron la exactitud considerablemente, tan solo un 2.24%. Adicionalmente, se intentó confundir a la red mezclando los archivos originales con los sintetizados intercambiando las clases. El bajo desempeño de la red indica que, efectivamente, la red no es capaz de distinguir una clase de otra. Los tres experimentos parecen apuntar a que los datos sintetizados poseen cierta calidad; sin embargo, es necesario implementar un algoritmo a partir del cual se pueda obtener una evaluación cuantitativa de la calidad de los archivos generados, como por ejemplo Inception Score. Esto es parte de otro trabajo que se está desarrollando actualmente.

Referencias

[1] Donahue, C., McAuley, J., & Miller, S. (2019). Adversarial Audio Synthesis. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, New York, NY, USA.OpenReview.net.
[2] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014). Generative Adversarial Networks (cite arxiv:1406.2661).
[3] Arjovsky, M., Chintala, S. & Bottou, L. (2017). Wasserstein GAN (cite arxiv:1701.07875)
[4] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A.C. (2017). Improved Training of Wasserstein GANs. *NIPS*.
[5] Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. CoRR, abs/1511.06434.
[6] Piczak, K. J. (2015). ESC: Dataset for Environmental Sound Classification. In Proceedings of the 23rd ACM international conference on Multimedia (MM '15). Association for Computing Machinery, New York, NY, USA, 1015-1018. 10.1145/2733373.2806390.

Biografía de Autores

González Huerta Rodrigo. Es Licenciado en Tecnología por la Universidad Nacional Autónoma de México. Actualmente cursa la Maestría en Ciencias en Inteligencia Artificial en la Universidad Autónoma de Querétaro, México. Su investigación está orientada al procesamiento de sonidos ambientales y su clasificación mediante la utilización de distintos modelos de aprendizaje profundo.

Ramos Arreguín Juan Manuel. Tiene Doctorado en Ciencias con línea terminal en Mecatrónica, en el Programa Interinstitucional de Posgrado del Centro en Ingeniería y Desarrollo Industrial, en Querétaro, Qro. Pertenece al Sistema Nacional de Investigadores en nivel I, y es profesor de tiempo completo en la Universidad Autónoma de Querétaro.

Takacs Andras. Es Doctor en Ingeniería por la Universidad Autónoma de Querétaro. Maestro en Ciencias Computacionales (University College London, Inglaterra). Actualmente es Profesor de Ciencias de la Computación en la Facultad de Ingeniería da la Universidad Autónoma de Querétaro, México. Sus intereses de investigación incluyen percepción visual robusta, reconocimiento de lugares en condiciones cambiantes, aprendizaje automático, SLAM y estimación probabilística.