



**INSTITUCIÓN UNIVERSITARIA  
PASCUAL BRAVO**

**Proyecto Final**

**Informe de Extracción de Características de Audio para Reconocimiento de Sonidos  
de Emergencia**

**Reconocimiento De Patrones**

**Realizado Por**

**Juan José Gañan Monsalve**

**Docente**

**PhD. Juan Carlos Briñez De León**

**Institución Universitaria Pascual Bravo**

**Faculta De Ingeniería**

**Ingeniería De Software**

**Medellín**

**2024**

# Informe de Extracción de Características de Audio para Reconocimiento de Sonidos de Emergencia

**Link Notebook:** <https://colab.research.google.com/drive/168yRQjr6QnIj1urWi5ERy4lai2-HqnkP?usp=sharing>

## 1. Introducción

El reconocimiento de sonidos de emergencia desempeña un papel crucial en la seguridad pública, permitiendo una respuesta rápida y efectiva a situaciones críticas en entornos urbanos. En este informe, se presenta un análisis detallado del proceso de extracción de características de archivos de audio para la clasificación de sonidos de emergencia, abordando su importancia, metodología y resultados obtenidos.

## 2. Objetivo

El objetivo principal de este estudio es extraer características relevantes de archivos de audio que representen sonidos de emergencia, como sirenas de ambulancias y camiones de bomberos, con el fin de desarrollar un sistema de reconocimiento de sonidos de emergencia preciso y confiable.

## 3. Metodología

3.1. Recopilación de Datos: Se recopilaron archivos de la plataforma Kaggle y grabaciones específicas obtenidas en entornos urbanos.

Fuente: <https://www.kaggle.com/code/aryashah2k/emergency-vehicle-siren-sound-classification/input>

3.2. Preprocesamiento de Datos: Los archivos de audio fueron preprocesados para eliminar ruidos no deseados, normalizar niveles de volumen y estandarizar la duración de las muestras, asegurando así la consistencia y calidad de los datos.

```
1 #Normalizamos los datos
2 scaler = StandardScaler()
3 X_train_normalized = scaler.fit_transform(X_train)
4 X_test_normalized = scaler.transform(X_test)
```

3.3. Extracción de Características: Se aplicaron técnicas de procesamiento de señales para extraer características relevantes de los archivos de audio. Esto incluye el cálculo de estadísticas temporales (varianza, desviación estándar, amplitud RMS) y características espectrales (centroides espectrales, ancho de banda espectral, contraste espectral, rolloff espectral).

```
# Extracción de características
Varianza = np.var(Señal_New)
Desviacion = np.std(Señal_New)
rms_amplitude = np.sqrt(np.mean(np.square(Señal_New)))
zero_crossings = np.where(np.diff(np.sign(Señal_New)))[0]
zcr = len(zero_crossings)
skewness = skew(Señal_New)
# Nuevos descriptores agregados
spectral_centroid = np.mean(librosa.feature.spectral_centroid(y=Señal_New, sr=sr))
spectral_bandwidth = np.mean(librosa.feature.spectral_bandwidth(y=Señal_New, sr=sr))
spectral_contrast = np.mean(librosa.feature.spectral_contrast(y=Señal_New, sr=sr))
spectral_rolloff = np.mean(librosa.feature.spectral_rolloff(y=Señal_New, sr=sr))
mfccs = np.mean(librosa.feature.mfcc(y=Señal_New, sr=sr), axis=1)[0] # Solo el primer coeficiente MFCC
```

3.4. Selección de Características: Se realizó un análisis exploratorio de las características extraídas para identificar aquellas con mayor capacidad discriminativa entre las clases de sonidos de emergencia.

#### 4. Características Extraídas

- Varianza
- Desviación estándar
- Amplitud RMS
- Cruces por cero
- Centroides espectrales
- Ancho de banda espectral
- Contraste espectral
- Rolloff espectral

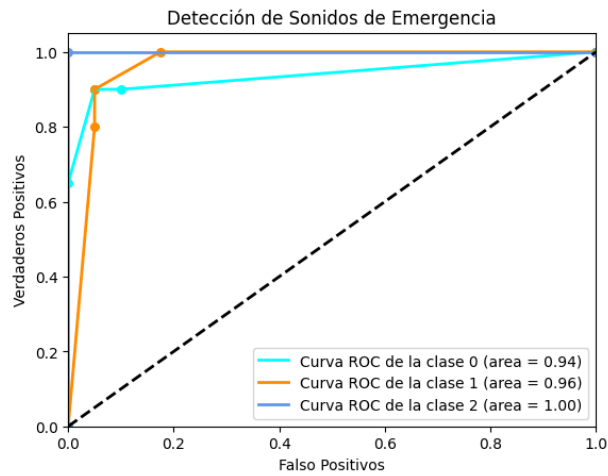
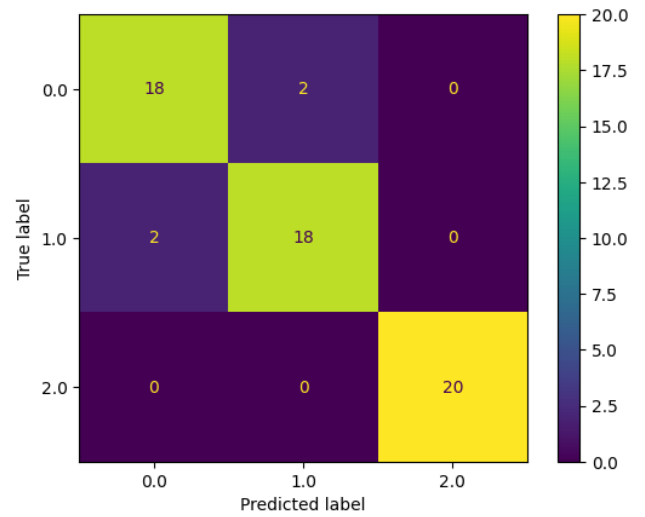
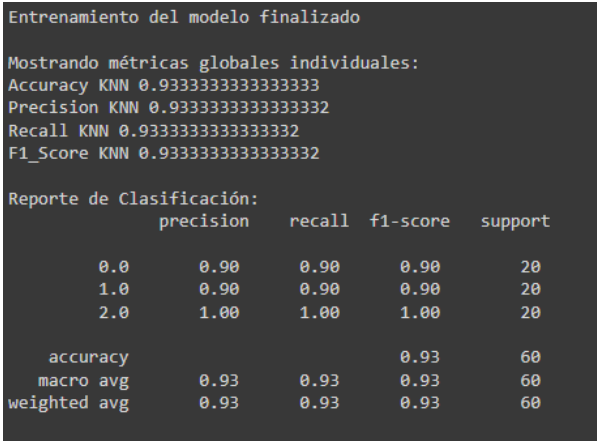
#### Feature extraction

##### Spectral features

chroma_stft ("[, y, sr, S, norm, n_fft, ...])	Compute a chromagram from a waveform or power spectrogram.
chroma_cqt ("[, y, sr, C, hop_length, fmin, ...])	Constant-Q chromagram
chroma_cens ("[, y, sr, C, hop_length, fmin, ...])	Compute the chroma variant "Chroma Energy Normalized" (CENS)
chroma_vqt ("[, y, sr, V, hop_length, fmin, ...])	Variable-Q chromagram
mel_spectrogram ("[, y, sr, S, n_fft, ...])	Compute a mel-scaled spectrogram.
mfcc ("[, y, sr, S, n_mfcc, dct_type, norm, ...])	Mel-frequency cepstral coefficients (MFCCs)
rms ("[, y, S, frame_length, hop_length, ...])	Compute root-mean-square (RMS) value for each frame, either from the audio samples or from a spectrogram
spectral_centroid ("[, y, sr, S, n_fft, ...])	Compute the spectral centroid.
spectral_bandwidth ("[, y, sr, S, n_fft, ...])	Compute p'th-order spectral bandwidth.
spectral_contrast ("[, y, sr, S, n_fft, ...])	Compute spectral contrast
spectral_flatness ("[, y, S, n_fft, ...])	Compute spectral flatness
spectral_rolloff ("[, y, sr, S, n_fft, ...])	Compute roll-off frequency.
poly_features ("[, y, sr, S, n_fft, ...])	Get coefficients of fitting an nth-order polynomial to the columns of a spectrogram.
tonnetz ("[, y, sr, chroma])	Compute the tonal centroid features (tonnetz)
zero_crossing_rate (y, "[, frame_length, ...])	Compute the zero-crossing rate of an audio time series.

5. Resultados

Los resultados preliminares muestran que las características extraídas son capaces de capturar eficazmente las propiedades distintivas de los sonidos de emergencia, permitiendo una discriminación precisa entre clases. El análisis detallado de cada característica revela su contribución individual al proceso de clasificación.



6. Conclusiones

El proceso de extracción de características de archivos de audio para el reconocimiento de sonidos de emergencia es fundamental para el desarrollo de sistemas robustos y efectivos. Los resultados obtenidos destacan la importancia de seleccionar y analizar cuidadosamente las características más relevantes, lo que puede mejorar significativamente el rendimiento del sistema de reconocimiento de sonidos de emergencia.