

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



**“IDENTIFICACIÓN DE ANOMALÍAS EN COLMENAS DE
ABEJAS MEDIANTE EL ANÁLISIS DE SEÑALES
BIOLÓGICAS”**

CRISTOPHER LUIS JIMÉNEZ MEZA

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA**

Profesor Guía: Dr. JOSÉ LINO CONTRERAS VÉLIZ
Profesora Correferente: Dra. KAREN DEL PILAR YAÑEZ ARELLANO

Marzo - 2023

DEDICATORIA

A mi hermana mayor Ingrid María José Jiménez Meza. Una gran persona y gran hermana mayor a quien quiero y aprecio mucho.

AGRADECIMIENTOS

- A mi hermana mayor quien siempre está conmigo, me acompaña y me da motivación para seguir adelante.
- A mi hermana menor quien me brinda su ayuda en aquellos momentos en los que la necesito.
- A mis padres por apoyarme a lo largo de la vida y en este proceso de la redacción del presente trabajo.
- A dios por darme fuerza y motivación para seguir adelante con este proyecto.
- A la doctora Karin Jury por toda la ayuda que me ha brindado en estos dos años.
- A todas las personas que me han estado apoyando en estos dos últimos años.
- A mi profesor guía Dr. José Lino Contreras Véliz por guiarme en el desarrollo de la memoria, leer el presente trabajo, e ir dándome ideas para mejorar los resultados obtenidos en este proyecto.
- A mi profesora correferente Dra. Karen del Pilar Yañez Arellano por proporcionar material y acceso a las colmenas de la UTFSM.

RESUMEN

Las abejas son parte importante del ecosistema, su disminución no solo afecta la biodiversidad, sino que también impacta actividades humanas como la agricultura, generando repercusiones económicas. Cada año la población de abejas va descendiendo a nivel mundial, por diferentes anomalías como parásitos, pérdidas de hábitats, desastres naturales, entre otros. Para colaborar en el ámbito de la detección de anomalías dentro de las colmenas de abejas, en esta memoria se desarrolló una aplicación web cuyo propósito es detectar la ausencia de abeja reina dentro de la colmena. Para llevar esto a cabo, se realizaron experimentos en los cuales se entrenaron redes neuronales convolucionales usando audios de colmenas con y sin presencia de abeja reina. Estos fueron caracterizados mediante espectrogramas logarítmicos, espectrogramas de Mel, y coeficientes cepstrales en la frecuencia de Mel extraídos desde la serie de tiempo de la amplitud y del espectrograma de Mel. Estos últimos generaron los mejores resultados, permitiendo que las redes neuronales se entrenaran más rápido, consumiendo menos memoria y dando mejores resultados en las métricas examinadas. Se espera que los resultados obtenidos a lo largo del presente trabajo puedan servir como un posible punto de referencia para la detección de otras anomalías como la presencia de Varroa.

Palabras Clave— apicultura, bioseñales, reconocimiento de patrones de audio, anomalías en colmenas, colmenas sin abeja reina.

ABSTRACT

Bees are an important part of the ecosystem, their decline not only affects biodiversity, but also impacts human activities such as agriculture, generating economic repercussions. Every year the bee population is decreasing worldwide, due to different anomalies such as parasites, habitat loss, natural disasters, among others. To collaborate in the field of anomaly detection within bee hives, in this report a web application was developed whose purpose is to detect the absence of queen bees within the hive. To carry this out, experiments were carried out in which convolutional neural networks were trained using audio from hives with and without the presence of a queen bee. These were characterized by logarithmic spectrograms, Mel spectrograms, and cepstral coefficients at the Mel frequency extracted from the amplitude time series and the Mel spectrogram. The latter generated the best results, allowing the neural networks to be trained faster, consuming less memory and giving better results in the examined metrics. It is expected that the results obtained throughout this work can serve as a possible point of reference for the detection of other anomalies such as the presence of Varroa.

Keywords— beekeeping, biosignals, audio pattern recognition, anomalies in hives, hives without a queen bee.

GLOSARIO

BW: *Bandwidth* o ancho de banda.

CCD: *Colony Collapse Disorder* o síndrome de colapso de colonias.

CNN: *Convolutional Neural Network* o red neuronal convolucional.

DFT: *Discrete Fourier Transform* o transformada discreta de Fourier.

FFT: *Fast Fourier Transform* o transformada rápida de Fourier.

HHT: *Hilbert-Huang Transform* o transformada de Hilbert Huang.

LDA: *Linear Discriminant Analysis* o análisis de discriminante lineal.

MFCCs: *Mel Frequency Cepstral Coefficients* o coeficientes cepstrales en la frecuencia de Mel.

MVC: *Model View Controller* o Modelo vista controlador.

PCA: *Principal Component Analysis* o análisis de componentes principales.

PF: *Peak Frequency* o frecuencia pico.

RVF: *Root Variance frequency* o Frecuencia de variación.

SAG: Servicio Agrícola y Ganadero del gobierno de Chile.

SC: *Spectral Centroid* o centroide espectral.

SVD: *Singular Value Decomposition* o descomposición en valores singulares.

SVM: *Support Vector Machine* o máquina de soporte vectorial.

UTFSM: Universidad Técnica Federico Santa María.

VM: *Virtual Machine* o Máquina virtual.

ÍNDICE DE CONTENIDOS

RESUMEN	4
ABSTRACT	4
GLOSARIO	5
ÍNDICE DE FIGURAS	9
ÍNDICE DE TABLAS	15
INTRODUCCIÓN	16
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	17
1.1 Macro problema	17
1.2 Colmena huérfana	18
1.2.1 Soluciones actuales	20
1.3 Varroa	22
1.3.1 Situación actual	25
1.3.2 Soluciones actuales	25
1.4 Participantes del problema	29
1.5 Impacto inicial de solucionar el problema	30
1.6 Objetivo	30
1.6.1 Objetivo general	30
1.6.2 Objetivos específicos	30
CAPÍTULO 2: MARCO CONCEPTUAL	31
2.1 Transformada discreta de Fourier (DFT)	31
2.2 Transformada rápida de Fourier (FFT)	32
2.3 Espectrograma	32
2.4 Escala de Mel	33
2.5 Espectrograma de Mel	34
2.6 Coeficientes cepstrales en la escala de Mel (MFCCs)	34
2.7 Red neuronal convolucional	35
2.7.1 Capa de entrada	35
2.7.2 Capa de convolución	36
2.7.3 Capa de pooling	38
2.7.4 Capa de aplanado	39

2.7.5 Capa clasificadora	39
2.7.6 Capa de salida	40
2.8 Estado del arte	41
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	47
3.1 Entornos de trabajo	47
3.1.1 Extracción de características y modelo de reconocimiento de patrones	47
3.1.2 Desarrollo	48
3.1.3 Producción	49
3.2 Modelo de reconocimiento de patrones de audio	49
3.2.1 Obtención de audios	51
3.2.2 Etiquetado	51
3.2.3 Generación de datasets	51
3.2.4 Extracción de características	51
3.2.5 Selección de características	52
3.2.6 Enfoque utilizado para generar el modelo de reconocimiento de patrones de audio	53
3.2.7 Arquitectura de la red	53
3.2.8 Hiperparámetros	55
3.2.9 Entrenamiento	55
3.3 Arquitectura del software	56
3.3.1 Herramientas utilizadas	57
3.3.2 Backend	58
3.3.3 Frontend	58
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	59
4.1 Análisis de las características extraídas	60
4.1.1 Espectrograma logarítmico vs Espectrograma de Mel	59
4.1.2 Comparación entre Espectrograma de Mel y MFCCs	60
4.1.3 Comparación entre MFCCs extraídos de la serie de tiempo de la amplitud y del espectrograma de Mel	62
4.2 Análisis del desempeño del clasificador	64
4.2.1 Clasificación usando espectrograma logarítmico	65
4.2.2 Clasificación usando espectrograma de Mel	66
4.2.3 Clasificación usando MFCCs extraídos desde la serie de tiempo de la amplitud	69

4.2.4 Clasificación usando MFCCs extraídos desde el espectrograma de Mel	76
4.2.5 ¿Cómo afecta la duración de la grabación en los resultados de la clasificación?	84
4.2.6 Uso de 13, 20 o 40 MFCCs	85
4.2.7 Comparación entre los diferentes enfoques	85
4.3 Pruebas en terreno	86
4.3.1 Contexto	86
4.3.2 Obtención de los audios	87
4.3.3 Resultados preliminares	88
4.3.4 Reentrenamiento de la red	88
4.3.5 Resultados finales	89
CAPÍTULO 5: CONCLUSIONES	91
5.1 Sobre los resultados de los diferentes enfoques	91
5.1.1 Enfoque basado en MFCCs	91
5.1.2 Enfoque basado en espectrograma de Mel	92
5.1.3 Enfoque basado en espectrograma logarítmico	92
5.2 Acerca del análisis visual de las caracterizaciones	92
5.2.1 Sobre el espectrograma de Mel	92
5.2.2 Sobre los MFCCs extraídos del espectrograma de Mel	93
5.2.3 Sobre los MFCCs extraídos desde el espectrograma de Mel y desde la serie de tiempo de la amplitud	93
5.3 Otras conclusiones	93
5.4 Trabajo futuro y posibles mejoras	94
REFERENCIAS BIBLIOGRÁFICAS	96

ÍNDICE DE FIGURAS

Figura 1: Ciclo de vida de las castas de abejas que componen la colmena. Fuente: http://indesol.gob.mx/cedoc/pdf/III.%20Desarrollo%20Social/Cr%C3%ADa%20de%20Animales/Manual%20de%20Buenas%20Practicas%20Ap%C3%ADcolas.pdf	18
Figura 2: Marco de una colmena zanganera. Fuente: https://www.latiendadelapicultor.com/blog/que-hacer-con-una-colmena-zanganera/	19
Figura 3: Partes de una colmena Langstroth. Fuente: https://www.wikiwand.com/es/Colmena_Langstroth	20
Figura 4: Ciclo de reproducción de la Varroa. Fuente: https://cooperativapampero.coop/conociendo-sobre-varroa-destructor-y-la-llegada-del-invierno/	24
Figura 5: Proceso del método del azúcar glass. Fuente: Elaboración propia	26
Figura 6: Procedimiento del método del alcohol (o jabón). Fuente: https://inta.gob.ar/documentos/apicultura-tras-la-cosecha-control-de-varroa	26
Figura 7: Agregar gas dentro del frasco con abejas. Fuente: https://www.youtube.com/watch?v=jgH6GA40SrY	27
Figura 8: Aplicación del método rasca la cría. Fuente: https://www.latiendadelapicultor.com/blog/rasca-la-cria-varroa/	28
Figura 9: Aplicación de DFT. Fuente: https://rufianenlared.com/fft/	31
Figura 10: Representación bidimensional del espectrograma. Fuente: https://pybonacci.org/2012/09/29/transformada-de-fourier-discreta-en-python-con-sci-py/	32
Figura 11: Representación tridimensional del espectrograma. Fuente: https://www.promax.es/esp/noticias/158/Espectrograma/	33
Figura 12: Escala de Mel. Fuente: https://repositorio.uchile.cl/bitstream/handle/2250/181887/Deconvolucion-en-audio-utilizando-modelos-basados-en-Machine-Deep-Learning.pdf?sequence=1	33
Figura 13: Ejemplo de espectrograma de Mel. Fuente: https://guillermoacharui.com/portfolio/biometriadevoz/	34
Figura 14: Proceso para obtener los MFCCs. Fuente: https://www.revistaespacios.com/a20v41n32/a20v41n32p16.pdf	34
Figura 15: Arquitectura de una CNN clasificadora. Fuente: https://datasmarts.net/es/redes-neuronales-convolucionales-en-profundidad	35
Figura 16: Ejemplo de CNN, entrada denotada por el rectángulo azul. Fuente: https://www.rcs.cic.ipn.mx/2019_148_7/Clasificador%20de%20plantas%20medicinales%20por%20medio%20de%20Deep%20Learning.pdf	36
Figura 17: Ejemplo de CNN, convolución denotada por el rectángulo azul. Fuente:	

https://www.rcs.cic.ipn.mx/2019_148_7/Clasificador%20de%20plantas%20medicinales%20por%20medio%20de%20Deep%20Learning.pdf	36
Figura 18: Ejemplo de convolución. Fuente: https://bootcampai.medium.com/redes-neuronales-convolucionales-5e0ce960caf8	38
Figura 19: Ejemplo de pooling. Fuente: https://www.diegocalvo.es/red-neuronal-convolucional/	38
Figura 20: Ejemplo de CNN, pooling denotada por el rectángulo azul. Fuente: https://www.rcs.cic.ipn.mx/2019_148_7/Clasificador%20de%20plantas%20medicinales%20por%20medio%20de%20Deep%20Learning.pdf	39
Figura 21: Aplanado (Flatten) arquitectura CNN, denotada por el rectángulo azul. Fuente: https://www.researchgate.net/publication/339207158_Optimizacion_Distribuida_de_Redes_Convolucionales_para_la_Clasificacion_de_Imagenes	39
Figura 22: Ejemplo de CNN, clasificación denotada por el rectángulo azul. Fuente: https://www.rcs.cic.ipn.mx/2019_148_7/Clasificador%20de%20plantas%20medicinales%20por%20medio%20de%20Deep%20Learning.pdf	40
Figura 23: Ejemplo de CNN, salida denotada por el rectángulo azul. Fuente: https://www.rcs.cic.ipn.mx/2019_148_7/Clasificador%20de%20plantas%20medicinales%20por%20medio%20de%20Deep%20Learning.pdf	40
Figura 24: Imagen de la actividad de la abeja <i>Apis Mellifera Ligustica</i> italiana registrada durante 3 días seguidos, etiquetado manualmente. Fuente: (Ferrari et al., 2008, 75).	41
Figura 25: Espectrograma del evento de enjambre. Fuente: (Ferrari et al., 2008, 75).	42
Figura 26: Diagrama del proceso para generar el modelo de reconocimiento de patrones. Fuente: Elaboración propia	50
Figura 27: Modelo para procesar 20 y 40 MFCCs. Fuente: Elaboración propia	54
Figura 28: Modelo para procesar 13, 20 y 40 MFCCs. Fuente: Elaboración propia	54
Figura 29: Diagrama general del sistema. Fuente: Elaboración propia	56
Figura 30: Arquitectura del sistema. Fuente: Elaboración propia.	56
Figura 31: Espectrograma logarítmico vs Espectrograma de Mel, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena con abeja reina. Aparición de frecuencias alrededor 4096 y 8192 [Hz]. Fuente: Elaboración propia	59
Figura 32: Espectrograma logarítmico vs Espectrograma de Mel, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena con abeja reina. Sin frecuencias alrededor de 4096 y 8192 [Hz]. Fuente: Elaboración propia	60
Figura 33: Espectrograma logarítmico vs Espectrograma de Mel, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena sin abeja reina. Fuerte presencia de frecuencias alrededor de 4096 y 8192 [Hz]. Fuente: Elaboración propia	60

Figura 34: Espectrograma de Mel vs 13 MFCCs, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena con abeja reina. Fuente: Elaboración propia 61

Figura 35: Espectrograma de Mel vs 13 MFCCs, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena sin abeja reina. Fuente: Elaboración propia 62

Figura 36: Serie de tiempo de la amplitud y 13 MFCCs, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena con abeja reina. Fuente: Elaboración propia 63

Figura 37: Espectrograma de Mel (obtenido desde la serie de tiempo de la imagen anterior) y 13 MFCCs, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena con abeja reina. Fuente: Elaboración propia 63

Figura 38: Serie de tiempo de la amplitud y 13 MFCCs, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena sin abeja reina. Fuente: Elaboración propia 64

Figura 39: Espectrograma de Mel (obtenido desde la serie de tiempo de la imagen anterior) y 13 MFCCs, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena sin abeja reina. Fuente: Elaboración propia 64

Figura 40: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 4 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 20). Fuente: Elaboración propia 65

Figura 41: Matriz de confusión generada por el segundo modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 3 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 13). Fuente: Elaboración propia 65

Figura 42: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, dataset 1 [s], 4 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 20). Fuente: Elaboración propia 66

Figura 43: Matriz de confusión generada por el segundo modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, dataset 1 [s], 3 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 13). Fuente: Elaboración propia 66

Figura 44: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, dataset 2 [s], 4 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 20). Fuente: Elaboración propia 67

Figura 45: Matriz de confusión generada por el segundo modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, dataset 2 [s], 3 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 13). Fuente: Elaboración propia 67

Figura 46: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, dataset 3 [s], 4 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 20). Fuente: Elaboración propia 68

- Figura 47: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, *dataset 3* [s], 4 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 13). Fuente: Elaboración propia 68
- Figura 48: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 13 MFCCs, *dataset 1* [s], 3 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 13). Fuente: Elaboración propia 69
- Figura 49: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 1* [s], 3 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 13). Fuente: Elaboración propia 70
- Figura 50: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 1* [s], 4 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 20). Fuente: Elaboración propia 70
- Figura 51: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset 1* [s], 3 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 13). Fuente: Elaboración propia 71
- Figura 52: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset 1* [s], 4 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 20). Fuente: Elaboración propia 71
- Figura 53: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 13 MFCCs, *dataset 2* [s], 3 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 13). Fuente: Elaboración propia 72
- Figura 54: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 2* [s], 3 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 13). Fuente: Elaboración propia 72
- Figura 55: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 2* [s], 4 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 20). Fuente: Elaboración propia 73
- Figura 56: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset 2* [s], 3 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 13). Fuente: Elaboración propia 73
- Figura 57: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset 2* [s], 4 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 20). Fuente: Elaboración propia 74
- Figura 58: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 13 MFCCs, *dataset 3* [s], 3 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 13). Fuente: Elaboración propia 74
- Figura 59: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 3* [s], 3 capas *Conv2D + MaxPooling2D + Dropout*. (mfcc 13). Fuente: Elaboración propia 75

- Figura 60: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset* 3 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20). Fuente: Elaboración propia 75
- Figura 61: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset* 3 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia 76
- Figura 62: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset* 3 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20). Fuente: Elaboración propia 76
- Figura 63: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 13 MFCCs, *dataset* 1 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia 77
- Figura 64: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset* 1 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia 77
- Figura 65: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset* 1 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20). Fuente: Elaboración propia 78
- Figura 66: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset* 1 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia 78
- Figura 67: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset* 1 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20). Fuente: Elaboración propia 79
- Figura 68: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 13 MFCCs, *dataset* 2 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia 79
- Figura 69: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset* 2 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia 80
- Figura 70: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset* 2 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20). Fuente: Elaboración propia 80
- Figura 71: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset* 2 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia 81
- Figura 72: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset* 2 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20). Fuente: Elaboración propia 81

- Figura 73: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 13 MFCCs, *dataset* 3 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia 82
- Figura 74: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset* 3 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia 82
- Figura 75: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset* 3 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20). Fuente: Elaboración propia 83
- Figura 76: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset* 3 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia 83
- Figura 77: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset* 3 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20). Fuente: Elaboración propia 84
- Figura 78: Instalación de micrófono dentro de la colmena extra, realizada por el Dr. José Lino Contreras Véliz y la Dra. Karen del Pilar Yañez Arellano. Fuente: Elaboración propia. 87
- Figura 79: CNN probada con audios del día 25/11/2022. Fuente: Elaboración propia. 88
- Figura 80: Matriz de confusión obtenida por la “CNN modelo final”. Probada con los audios del conjunto de prueba. Fuente: Elaboración propia. 89
- Figura 81: Matriz de confusión obtenida por la “CNN modelo final”. Probada con audios del día 02/12/2022. Fuente: Elaboración propia. 90
- Figura 82: Matriz de confusión obtenida por la “CNN modelo final”. Probada con audios del día 09/12/2022. Fuente: Elaboración propia. 90

ÍNDICE DE TABLAS

Tabla 1: Reducción de la productividad de miel (colmena / año). Fuente: (Rico et al., 2013).	24
Tabla 2: Traducción del resumen de las investigaciones asociadas al procesamiento del audio emitido por las abejas. Fuente (Terenzi et al., 2020, 168).	44

INTRODUCCIÓN

La polinización consiste en el transporte de polen desde la parte masculina de una flor, a la parte femenina de otra. Este importante fenómeno para la preservación de la biodiversidad se puede producir de dos formas: por interferencia del viento o el agua conocido como polinización abiótica o por la intervención de algún animal denominado polinización biótica. Según Innovatione AgroFood Design (2019) dentro del 90% de la polinización que se produce en todo el mundo un 67% es producido por insectos. Dentro de los insectos polinizadores más comunes que se pueden encontrar, están las abejas. Su población es importante para el estilo de vida que llevamos los humanos, ya que según Innovatione AgroFood Design (2019) 87% de los alimentos que consumimos actualmente son polinizados por ellas, además su labor también tiene repercusiones en actividades humanas como la agricultura, la apicultura y la textilería, ya que su polinización afecta el tamaño de los frutos, son generadoras de las materias primas obtenidas de la apicultura y afectan la generación de plantas textiles como el algodón.

Actualmente la población de abejas se va reduciendo cada año a nivel mundial, esto debido a problemas tales como parásitos, desastres naturales, contaminación, ausencia de abeja reina, entre otras. Por dicho motivo, para ayudar a la detección de anomalías dentro de las colmenas de abejas, a lo largo de la presente memoria se ha desarrollado una aplicación web con el fin de poder detectar la ausencia de la abeja reina dentro de la colmena a tiempo, con el fin de poder reducir la probabilidad de que se produzca el fenómeno de la “colmena zanganera”, la cual a grandes rasgos consiste en que las abejas obreras comienzan a colocar huevos de los cuales solo nacerán zánganos, lo que eventualmente producirá el colapso de la colmena si no se trata a tiempo, en secciones posteriores se presentará este fenómeno con más detalle. Para llevar esto a cabo, se realizaron experimentos en los cuales se entrenaron dos modelos de redes neuronales convolucionales con audios provenientes de colmenas con y sin abeja reina, los cuales fueron caracterizados mediante espectrogramas logarítmicos, espectrogramas de Mel, 13, 20 y 40 coeficientes cepstrales en la frecuencia de Mel extraídos desde la serie de tiempo de la amplitud y del espectrograma de Mel, además durante los experimentos también se hizo variar la duración de las muestras de audio generando *datasets* con audios de duración 1, 2 y 3 segundos. Las redes neuronales fueron entrenadas con las matrices de densidad de energía obtenidas de las caracterizaciones mencionadas previamente. Finalmente, al terminar de realizar la etapa de experimentación, se tomó una de las redes neuronales entrenadas que proporcionaba buenos resultados y se procedió a construir la aplicación que sería utilizada por el usuario para poder interactuar con esta red neuronal, la cual consiste en un servidor de vistas que conforma el *frontend* y una API que conforma el *backend* la cual almacena y utiliza la red neuronal al momento de consultar el estado de la colmena.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1 Macro problema

La disminución de las abejas trae consecuencias como disminución de la polinización que a su vez produce pérdidas económicas en la agricultura debido a la mala formación de frutos, pequeños tamaños y mengua de la producción de alimentos, decremento de la biodiversidad que también es generado por los problemas de la polinización, reducción de la elaboración de miel, polen, jalea real, propóleo y otros productos derivados de las abejas afectando la rentabilidad de los apicultores, disminución en la generación de fibras textiles como el algodón y el lino, también se reduce la cantidad de semillas producidas, cabe tener en consideración que el 87% de los alimentos que consumimos actualmente dependen de la polinización producida por la abejas Innovatione AgroFood Design (2019), entre otras.

Cada año, la población de abejas va descendiendo a nivel mundial. Según (Manzano, n.d.) entre un 20-35% de las abejas europeas desaparecen al año, en Estados Unidos esa cifra llega a ser del 50%. (Stancich, 2020) menciona que la cantidad de colonias de abejas melíferas en Estados Unidos bajó de 5.9 millones en la década de 1940 a 1.9 millones en 1996. En Italia, Austria, Francia, Alemania, Países Bajos, Suecia y Hungría tuvieron un descenso entre 15-30% en un periodo comprendido desde 1985 hasta 2005. Según la noticia “Pérdidas de colmenas en Latinoamérica. Resultado: 2016/2017” obtenida desde <https://solatina.org/perdidas-2016-2017/> publicada en SoLatInA, datos obtenidos de productores ubicados en países latinoamericanos, muestran que del 1 de octubre de 2016 al 1 de octubre de 2017 se perdieron 56.1% de las colmenas de abejas melíferas en Chile, 12.6% en Ecuador y Perú, además Brasil tuvo una pérdida de 41.0% de abejas nativas sin aguijón.

Algunos factores que influyen en esta situación son mencionados por (Manzano, n.d.) tales como la plantación de monocultivos, uso de pesticidas, estrés de las abejas, pérdida de la diversidad genética, parásitos y enfermedades propagadas por los apicultores. Otros factores que contribuyen a este hecho son contaminación, pérdida de hábitats, desastres naturales (incendios forestales, inundaciones, entre otros), cambio climático y CCD.

A continuación, se abordarán dos de los problemas que afectan a las abejas, la colmena huérfana y la Varroa.

1.2 Colmena huérfana

Antes de tratar el tema de la colmena huérfana, es necesario entender su organización interna y el rol que cada tipo de abeja desempeña dentro de ella. Según se menciona en (La tienda del apicultor, n.d.), la colmena de abejas está formada por tres castas: reina, obrera y zángano. A continuación se describirán sus funciones en la colonia.

- **Reina (hembra).** Puede colocar huevos fecundados y no fecundados): Sus funciones incluyen poner huevos (los huevos fecundados generan abejas obreras, en cambio los no fecundados producen zánganos) y regular las actividades de la colonia. También emite feromonas que evita el desarrollo de los óvulos de las abejas obreras, con el fin de que éstas últimas no coloquen huevos.
- **Obrera (hembra):** Realizan diferentes labores dependiendo de su edad. Entre las labores realizadas están limpieza (puede ser de las celdas para la puesta de la reina o evacuación de residuos y cadáveres de la colmena), nodrizas (cuidado de las crías), constructoras, almacenadoras, ventiladoras (encargándose de regular el microclima dentro de la colonia), guardianas y pecoreadoras (encargadas de recolección de elementos como néctar, polen, propóleo y agua). Generalmente no coloca huevos (a menos que no haya abeja reina o larvas de abeja reina), como éstas abejas no se aparean, sólo pueden colocar huevos no fecundados.
- **Zángano (macho):** Se encarga de fecundar a la abeja reina durante los vuelos de fecundación. Cuando se aparea con la reina, deja sus órganos en los órganos sexuales de la reina, luego cae al suelo y muere al poco tiempo.

En la siguiente imagen, se puede apreciar el ciclo de vida de cada una de las castas que componen la colmena.

DURACIÓN DE CICLO DE VIDA (días)			
ESTADO	REINA	OBRERA	ZANGANO
Huevo	3	3	3
Larva	5.5	6.5	6.5
Prepupa	1	1	1
Pupa	6-7	9.5	13.5
Adulto	15-16	19-20	24

Figura 1: Ciclo de vida de las castas de abejas que componen la colmena. Fuente: <http://indesol.gob.mx/cedoc/pdf/III.%20Desarrollo%20Social/Cr%C3%ADa%20de%20Animales/Manual%20de%20Buenas%20Practicas%20Ap%C3%ADcolas.pdf>

Una colmena huérfana es aquella que no posee abeja reina, esto se puede deber a que la reina pudo haber sufrido alguna clase de accidente o problema en su vuelo nupcial, el apicultor realizó alguna revisión de la colmena generando la muerte de la reina, u otros

motivos. Tal como señala (Espin, n.d.) la falta de abeja reina puede producir que las obreras se transformen en ponedoras (al no estar presentes las feromonas que emite la abeja reina o las larvas de abeja reina, éstas pueden empezar a desarrollar sus ovarios y comienzan a colocar huevos con el fin de que la colmena sobreviva), en un periodo de 15 a 30 días (Coordinación Nacional de Ganadería, n.d., 23) éstas comenzarán a colocar huevos no fecundados generando que la colmena se llene de huevos de zánganos, produciéndose el fenómeno de “colmena zanganera”. Si no se detecta a tiempo el problema de la colmena huérfana, la población dentro de ella se verá seriamente afectada, en esa situación se deberá examinar si se pueden salvar las abejas restantes para reforzar otras colmenas. En aproximadamente dos meses desde que la colonia se ha transformado en zanganera, esta se pierde completamente (Coordinación Nacional de Ganadería, n.d., 23).



Figura 2: Marco de una colmena zanganera. Fuente: <https://www.latiendadelapicultor.com/blog/que-hacer-con-una-colmena-zanganera/>

1.2.1 Soluciones actuales

Detección de colmenas huérfanas

- **Revisión externa de la colmena:** En la revisión externa, se revisa el exterior de la colmena sin abrirla. Si se detectan abejas nerviosas desorientadas corriendo por la tabla de vuelo, esto puede indicar que la colmena está huérfana (La tienda del apicultor, n.d.). También hay apicultores experimentados que pueden diferenciar el sonido que emite una colmena con abeja reina y una sin abeja reina.

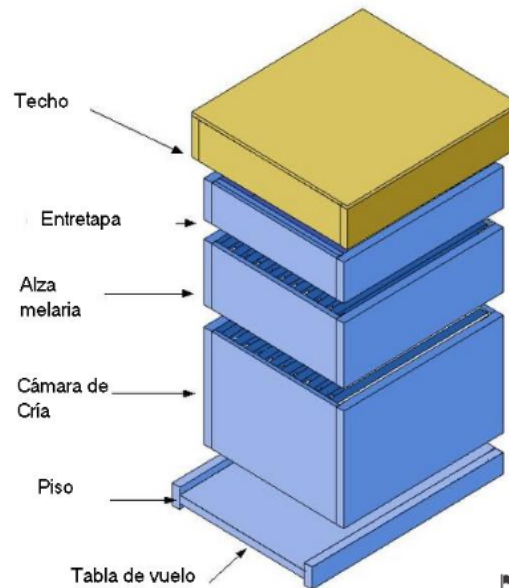


Figura 3: Partes de una colmena *Langstroth*. Fuente: https://www.wikiwand.com/es/Colmena_Langstroth

- **Revisión interna de la colmena:** En este caso se destapa la colmena y se revisa el interior para ver lo que sucede dentro. Si la colmena ha estado mucho tiempo sin abeja reina (un periodo de 15 a 30 días, en los que comenzarán a aparecer obreras ponedoras) se podrían identificar síntomas como: solo hay crías de zánganos, muchos huevos en la misma celda, celdas de obreras operculadas con elevación, huevos fuera de la base de la celda, los que corresponden a una colmena zanganera (La tienda del apicultor, n.d.).

Herramientas basadas en *software* para la detección de colmena huérfana

- **ColmenApp:** Esta herramienta se basa en técnicas de procesamiento de audio e inteligencia artificial. Según el Departamento de ingeniería informática de la Universidad de Santiago de Chile (2020) a través del procesamiento del zumbido emitido por la colmena busca detectar anomalías en ella. (Neumann, 2020) señala que esta herramienta detecta cuatro estados del panal: sano, que la colmena no tenga abeja reina, que la colmena tenga alta presencia de Varroa, que la colmena tenga presencia de Varroa y no tenga abeja reina. Hasta el 28 de septiembre del 2020, la aplicación estaba en un estado avanzado de la implementación, no se ha podido encontrar material reciente sobre el estado de la aplicación.
- **Sistema de monitoreo del sonido emitido por una colmena de abejas como herramienta para estudios comportamentales:** Este sistema fue desarrollado por Oscar Alejandro Amado Sarmiento en una tesis de magister, consiste en sensores ubicados en las colmenas que capturan y procesan los zumbidos emitidos por la colmena, estos son enviados a un servidor web que se encarga de almacenar la información e identifica su estado con el fin de detectar si tiene o no abeja reina. Esto se logra con algoritmo de reconocimiento de patrones (Amado Sarmiento, 2013), fue probado en el Centro Agropecuario Marengo de la Universidad Nacional de Colombia. No se ha obtenido información sobre el estado actual del proyecto.
- **B-Keep y B-Swarm:** Estos son sensores que permiten a apicultores profesionales y aficionados monitorear a distancia las colmenas de abejas, mediante una aplicación para *smartphones* (cada una tiene su propia aplicación), estas aplicaciones envían notificaciones cada 1 hora y en caso de alguna alerta envían notificaciones especiales, tienen herramientas de geolocalización para saber específicamente la ubicación de la colmena y puede generar seguimiento estadístico de la temperatura y humedad de la colmena. Según (hostabee, 2020) se puede obtener información sobre la temperatura interna, temperatura externa, humedad entre otras. Si bien esta herramienta no está diseñada específicamente para la detección de colmenas con o sin abejas reinas, el poder detectar ciertas variaciones en determinados atributos asociados a la colmena, puede dar señales a los apicultores sobre anomalías (tanto beneficiosas como perjudiciales), que estén ocurriendo dentro de ella, de esta manera pueden tomar las respectivas medidas para evitar mayores daños.

1.3 Varroa

El más destacable de los parásitos que atacan a las abejas, es la Varroa. Este ácaro proviene de Asia y se encuentra presente en la mayor parte del mundo, siendo la principal preocupación de los apicultores, pues ataca tanto a abejas adultas como a crías, causando desastres en las colmenas. Si bien existen ciertas abejas que pueden soportar su ataque, como la *Apis cerana* otras especies como la *Apis mellifera* no pueden hacerlo, debido a que no cuentan con mecanismos de defensas naturales para protegerse de él, por lo tanto estas últimas dependen totalmente del apicultor para combatir la infestación.

Actualmente no es posible eliminar completamente su presencia de las colmenas, puesto que se encuentra principalmente dentro de las celdas¹ operculadas² donde los tratamientos no son muy efectivos (también ha generado resistencia a alguno de ellos). La Varroa puede llegar a los panales aferrándose a las abejas recolectoras que vuelven de su viaje, por recolectoras infestadas perdidas provenientes de otras colmenas, por el pillaje³, u otras formas, de esta manera puede infestar o reinfestar colmenas sanas. Cuando este parásito entra en la colonia, busca celdas con larvas para esconderse justo antes de que sean operculadas, luego de la operculación comenzará a reproducirse y a alimentarse del tejido corporal graso de la abeja en formación, provocando: deformaciones, pérdida de peso, debilitamiento del sistema inmunológico, contagio de enfermedades transmitidas por la misma Varroa y en el peor de los casos la muerte. Cuando la cría salga de la celda, las Varroas hembras se adherirán a ella, alimentándose del tejido graso, continuando el ciclo de la propagación.

Como no es posible eliminar completamente su presencia de la colmena, una de las actividades realizadas es revisar el nivel de la invasión del panal (tasa de infestación) y dependiendo del resultado obtenido se procede a ejecutar determinadas acciones para reducir su población, (Pérez, 2006, 8) indica que si la tasa de infestación medida con el método del lavado con alcohol (o jabón) es mayor al 5% es necesario tratar la colonia, al igual que con el método de cortar un cuadro de cría cuando es mayor al 10%, (Araos, 2021) menciona que si la presencia de Varroa supera el 3% es necesario combatirla, en cambio (La tienda del apicultor, n.d.) expone que con una tasa menor a 5% el tratamiento puede postergarse hasta 2 meses, si está entre 5 y 20% no puede demorarse más de un mes y cuando sobrepasa el 20% es urgente.

¹ Recipientes hexagonales que se encuentran dentro del panal, creados por las abejas a partir de cera. Algunas de sus funciones son guardar polen, miel y crías, dependiendo de la función a desempeñar, es el tamaño que tienen.

² Para este contexto, hace referencia a las celdas que están cerradas o al momento en el cual las abejas van a cerrar la celda.

³ Invasión por parte de abejas de una colonia a otra, con el fin de robar sustancias dulces, generalmente miel. Provocado por factores como la falta de néctar en las flores, mal clima, pocas reservas de alimentos en las colmenas, entre otras.

La detección tardía de este ácaro, generada por factores como su tamaño, demora en la detección de síntomas visuales y/o la aparición de ellos, incluso la poca experiencia del apicultor tratando con este problema, puede provocar grandes daños entre las abejas de la colmena, si no recibe tratamiento se pierde completamente en un periodo que va de dos a tres años, (Eberhardt, 2006, 11-12) indica que una colmena de abejas europeas infestadas con Varroa, sin la intervención de un apicultor tiene una probabilidad de mortalidad entre 10 y 15% en el primer año, 20 y 30% durante el segundo y cercana al 100% en el tercero. También cabe mencionar que otro problema de la presencia de Varroa en la colmena, es que dicho parásito es un vector de enfermedades⁴, (Achury & Smith, 2020) indica que puede transmitir 6 virus: virus de la parálisis aguda de las abejas, virus de las celdas negras de la reina, virus del ala deformada, virus de la parálisis aguda Israeli, virus de Cachemira y virus *Sacbrood* o cría ensacada. (Araos, 2021) menciona que este ácaro transmite enfermedades que pueden reducir la producción de miel de las abejas entre un 20 y 80%, algo que queda en evidencia cuando se observa la diferencia de producción de miel que presenta una colmena infestada y otra sana.

Tabla 1: Reducción de la productividad de miel (colmena / año). Fuente: (Rico et al., 2013).

	Apis mellifera ligustica	Apis mellifera carnica	Apis mellifera iberica
Normal	67 [kg]	44 [kg]	29 [kg]
Infectada	54 [kg]	24 [kg]	19 [kg]

Toda la información mencionada en esta sección, indica que si la presencia de Varroa se dispara dentro de una colmena y el apicultor no se percata de ello, esto traerá una serie de complicaciones. Resumiendo lo explicado previamente, entre los problemas de salud que la Varroa trae a las abejas se pueden encontrar: deformaciones, enfermedades y muerte, trayendo consigo problemas económicos al apicultor entre los cuales estan: costos asociados a tratar tardíamente la Varroa, reducción en la producción de productos generados por las abejas, pérdidas de colmenas, costos asociados a obtener nuevas abejas, además del efecto que tiene en la polinización que como fue mencionado anteriormente también repercute en la agricultura. Es importante señalar que este parásito con el tiempo ha desarrollado resistencia a ciertos tipos de tratamientos como los piretroides, el amitraz y otros, además la mayor parte de su población se encuentra en las celdillas operculadas donde el apicultor no puede verlas. Todo ello vuelve su detección tardía un serio problema. A continuación se presenta la figura 4, en la cual se puede

⁴ Un vector de enfermedades es un organismo vivo, que transmite enfermedades a otros organismos vivos, algunos ejemplos de ello son las pulgas, garrapatas y mosquitos.

apreciar de mejor manera el ciclo de reproducción de la Varroa.

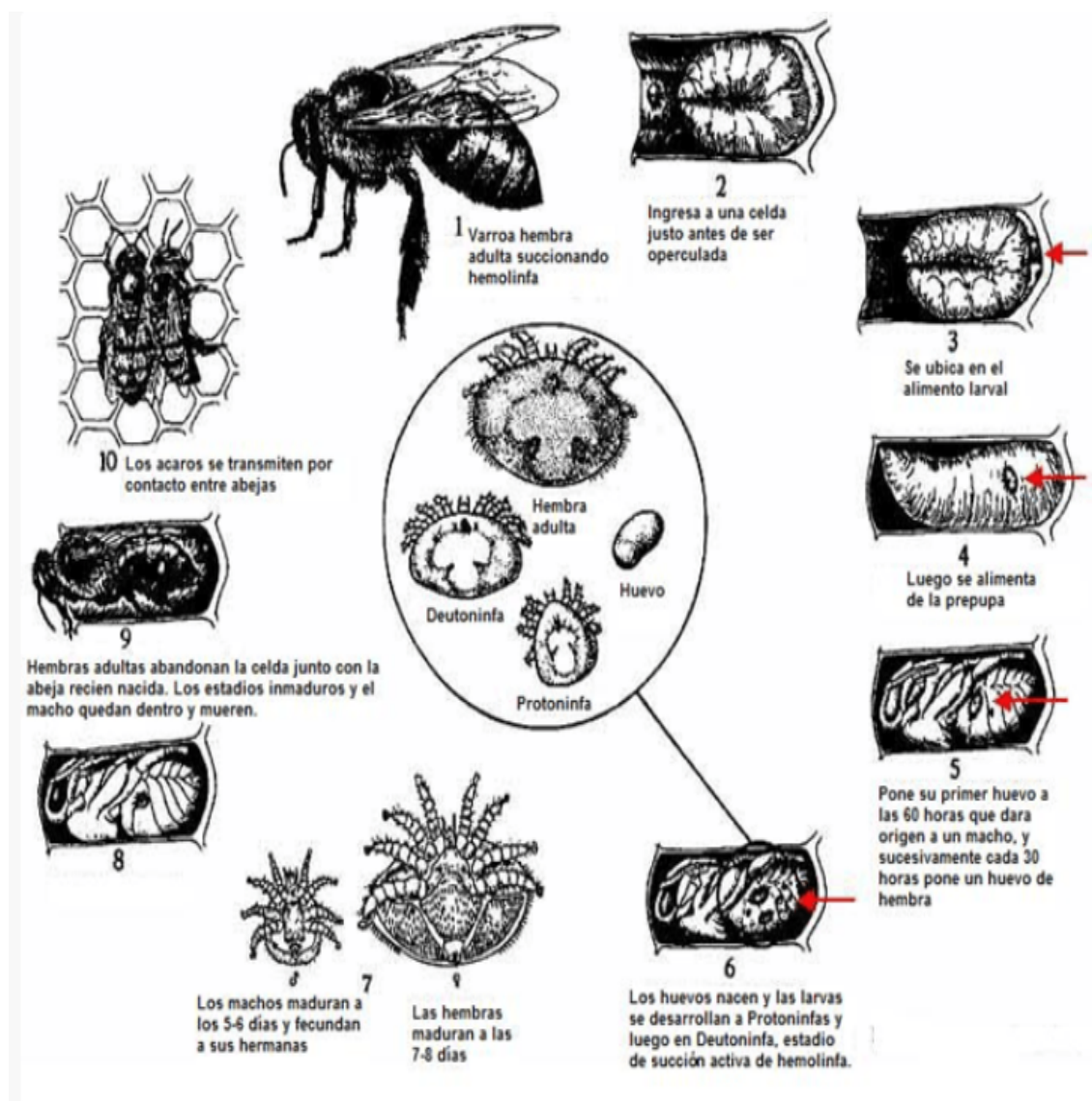


Figura 4: Ciclo de reproducción de la Varroa. Fuente:
<https://cooperativapampero.coop/conociendo-sobre-varroa-destructor-y-la-llegada-del-invierno/>

1.3.1 Situación actual

Según el boletín apícola N° 5 de mayo 2020 emitido por el SAG, de las cuatro enfermedades endémicas que se encuentran en Chile y que afectan a las abejas (Varroasis, Acarapisosis, Loque Europea y Loque Americana) la Varroasis⁵ es la principal, afectando en promedio a 55.57% de los apiarios⁶ inspeccionados en los años 2015 a 2019, también se menciona que la Varroasis al igual que la Acarapisosis y Loque Europea, se encuentra en todas las regiones del país exceptuando la Isla de Pascua.

En el informe de resultados del programa de vigilancia 2019-2020 sobre las pérdidas de colonias de abejas en España, se indica que en otoño de 2019 se encontró la presencia de *Varroa destructor* en un 62.0% de las colonias y 94.6% de los apiarios investigados, 43.9% de los apiarios presentó niveles nulos o muy leves de infestación⁷ en cambio un 27.7% tuvo niveles entre moderados a muy graves⁸, el resto cae en el rango leve⁹. En primavera de 2020 la presencia del ácaro disminuyó tanto en apiarios como en colmenas, siendo encontrado en un 90.8% y 56.8% de ellos respectivamente, además los niveles de infestación nulos y muy leves fueron de 49.0%, 15.3% para moderados a muy graves y el resto cae en el rango leve.

1.3.2 Soluciones actuales

Recordando que debido a que no es posible eliminar completamente la presencia de *Varroa* dentro de las colmenas, lo que se suele hacer es emplear algún método de detección para luego aplicar algún tipo de tratamiento dependiendo del nivel de infestación. Entre los métodos ocupados para su detección, se encuentran:

Detección de la *Varroa*

- **Azúcar glass:** Se colocan entre 250-300 abejas en un frasco, se tapa con tela mosquitera y se vierten dos cucharadas de azúcar glass por sus agujeros, se agita el frasco suavemente, posteriormente durante un minuto se vierte el azúcar del tarro por medio de los agujeros, al final se liberan a las abejas para que vuelvan a la colmena y se cuentan las Varroas. En la figura 5 se muestran algunos pasos a realizar, para llevar a cabo este método.

⁵ Varroasis o Varroosis, es una parasitosis externa producida por el parásito *Varroa*.

⁶ Es el sitio donde se encuentra el conjunto de colmenas que pertenecen a un apicultor.

⁷ Niveles de infestación menores al 1%.

⁸ Infestación que va en los rangos: moderado 5-10%, grave 10-20% y muy grave mayor a 20%.

⁹ Este caso se encuentra entre 1-5%.



Figura 5: Proceso del método del azúcar glass. Fuente: Elaboración propia

- **Lavado con alcohol (o jabón):** Consiste en sumergir una muestra de abejas en un frasco con agua y alcohol (o jabón), agitarlo y luego pasar su contenido por un filtro para separar las abejas de las Varroas, finalmente se procede a contarlas para determinar el nivel de infestación. Cabe destacar que al terminar este proceso, las abejas muestreadas terminarán muertas, es por ello que no se debe incluir a la reina. A continuación se presenta la figura 6, que contiene una descripción de todos los pasos que se deben realizar para completar exitosamente el lavado.



Figura 6: Procedimiento del método del alcohol (o jabón). Fuente: <https://inta.gob.ar/documentos/apicultura-tras-la-cosecha-control-de-varroa>

- **Aplicación de gas CO₂¹⁰:** Se coloca una muestra de abejas en un frasco y se tapa, luego se abre un poco y se agrega gas CO₂ durante 5-6 segundos hasta que las abejas dejen de volar, se vuelve a tapar el frasco y se deja reposar aproximadamente unos 10 segundos hasta que las abejas queden adormecidas, luego se agita suavemente para que las varroas (también adormecidas por el gas) se desprendan, se libera a las abejas para que vuelvan a la colmena y se cuentan las Varroas. En la figura 7 se puede ver como el apicultor abre levemente el frasco que contiene la muestra de abejas, para ingresar el gas mediante un artefacto específico.



Figura 7: Agregar gas dentro del frasco con abejas. Fuente: <https://www.youtube.com/watch?v=jgH6GA40SrY>

¹⁰ Esta notación hace referencia al gas dióxido de carbono.

Eliminación de Varroa

El siguiente método presentado, tiene la finalidad de combatir la Varroa en lugar de detectarla.

- **Rasca la cría:** Con este método se busca eliminar las crías que se encuentran en las celdillas operculadas (en esas celdillas es donde se encuentra la mayor concentración de Varroa, además los tratamientos son poco eficaces. Es por ello que quienes los desarrollan, indican que los tratamientos son más efectivos en períodos con poca cría.) para aprovechar el instinto de limpieza de las abejas. Se sacan los panales de la colmena y con un peine desoperculador se rompen todas las celdillas operculadas destruyendo a la cría, posteriormente se devuelven los panales a la colmena. Esto provoca que las abejas limpien los restos de larva, cera y eliminen las Varroas que se encontraban dentro, luego de ello, se procede a aplicar algún tratamiento para eliminar las Varroas que se encuentran en las abejas adultas y las que posiblemente quedan en las celdillas rotas. La figura 8 muestra cómo dos personas aplican el método rasca la cría en los panales de una colmena de abejas, usando los cepillos desoperculadores.



Figura 8: Aplicación del método rasca la cría. Fuente: <https://www.latiendadelapicultor.com/blog/rasca-la-cria-varroa/>

Herramientas basadas en *software* para la detección de Varroa

Además de los métodos presentados anteriormente, también existen herramientas basadas en *software*¹¹, diseñadas con el fin de detectar su presencia. A continuación se mencionarán dos de ellas:

- **BeeScanning:** Esta aplicación se basa en técnicas de procesamiento de imágenes e inteligencia artificial. Según (Lagerman, n.d.) mediante las imágenes de abejas vivas que se encuentran en los marcos de crías, puede detectar la presencia de Varroa y calcular la tasa de infestación con una precisión del 80%.
- **ColmenApp:** Tal como se señaló en el punto 1.2.2, esta aplicación basada en técnicas de procesamiento de audio e inteligencia artificial, también puede detectar la presencia de Varroa en una colmena, a través del zumbido que emite.

1.4 Participantes del problema

Los principales afectados por el problema de las colmenas huérfanas, falta de detección temprana de la presencia del parásito Varroa y otras enfermedades asociadas a las abejas son:

- **Apicultor:** Las abejas que están a su cargo pueden terminar sumamente dañadas. Para el caso de las colmenas huérfanas, no disponer de nuevas abejas obreras, generará problemas a nivel general dentro de la colmena, produciendo en el peor de los casos su pérdida total. Con respecto al problema de la Varroa, esto conlleva un costo económico asociado a tratamientos para combatir la plaga detectada tardíamente y los costos asociados a la reposición de las abejas que mueren a causa de la infestación, además el deterioro en la salud de la colmena. Los problemas anteriores también traerán una reducción en los productos generados a partir de las abejas, que a su vez afecta la rentabilidad.
- **Agricultor:** Debido a que la actividad apícola y agrícola van de la mano, esta se ve seriamente afectada cuando la actividad polinizadora se ve reducida. La falta de polinización en los cultivos agrícolas, van a producir deformaciones en los frutos y pequeños tamaños, reducirá la producción de los alimentos que generan las plantas al igual que sus semillas, todo esto se traduce en pérdidas económicas en ese sector.

Otras entidades que podrían estar interesadas en la presente problemática son: investigadores de abejas, organizaciones como Fundación Amigos de las Abejas u ONG Plan Bee, industria de la miel entre otras.

¹¹ Según el diccionario de la Real Academia Española (RAE), el software es un conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

1.5 Impacto inicial de solucionar el problema

Como se mencionó en la sección 1.2, la colmena huérfana se verá afectada a nivel general (no existirá una abeja reina que regule las actividades que ocurren dentro de la colonia), además la falta de nuevas obreras terminará afectando la producción de productos derivados de las abejas y afectará la salud de la colmena, generando su colapso, produciendo en el peor de los casos su pérdida completa. Con respecto a los daños que provoca el parásito Varroa (mencionados en la sección 1.3) se encuentran: infestación y reinfestación de colmenas sanas, aparición de enfermedades provocadas por los virus transmitidos, reducción de la población de abejas dentro de la colmena, lo que conlleva a la disminución de la producción de miel, jalea real, propoleo, polen, cera entre otros productos producidos por las abejas y a la reducción de la polinización que a su vez afecta a la agricultura, además los propios tratamientos que son utilizados para controlar su población pueden contaminar los productos producidos por las abejas. Es por ello que generar una solución que permita detectar anomalías en colmenas de abejas de forma temprana, permitiría disminuir los daños señalados anteriormente, además de reducir los costos asociados a tratamientos tardíos, la reposición de las abejas muertas y el número de colmenas perdidas.

1.6 Objetivo

1.6.1 Objetivo general

Diseñar y programar un sistema informático basado en técnicas de reconocimiento de patrones, para detectar de forma temprana anomalías en el estado de las colmenas de abejas.

1.6.2 Objetivos específicos

1. Encontrar relaciones entre los problemas que afectan a las abejas y sus señales emitidas.
2. Diseñar un sistema enfocado en la detección de colmenas con y sin abeja reina, para dispositivos móviles, web o de otro tipo.
3. Implementar una solución basada en reconocimiento de patrones en zumbidos de colmenas, para determinar la existencia o ausencia de abeja reina en ellas.
4. Validar con datos reales la solución creada, para verificar su desempeño.

CAPÍTULO 2: MARCO CONCEPTUAL

Esta sección tiene como principal objetivo exponer al lector los conceptos más relevantes que serán usados de manera recurrente a lo largo del presente trabajo. Estos serán de suma importancia ya que conforman la médula del enfoque utilizado a lo largo de la investigación en la cuál se basa el presente documento.

2.1 Transformada discreta de Fourier (DFT)

La DFT es una transformación matemática discreta, que toma una función en el dominio del tiempo y la transforma en otra cuyo dominio es la frecuencia (Gómez, 2017). Con ella se puede por ejemplo tomar una grabación de un zumbido de una colonia de abejas representada por la amplitud en función del tiempo y obtener mediante DFT la amplitud en función de la frecuencia, con el fin de extraer información. La figura 9 muestra un ejemplo del uso de DFT.

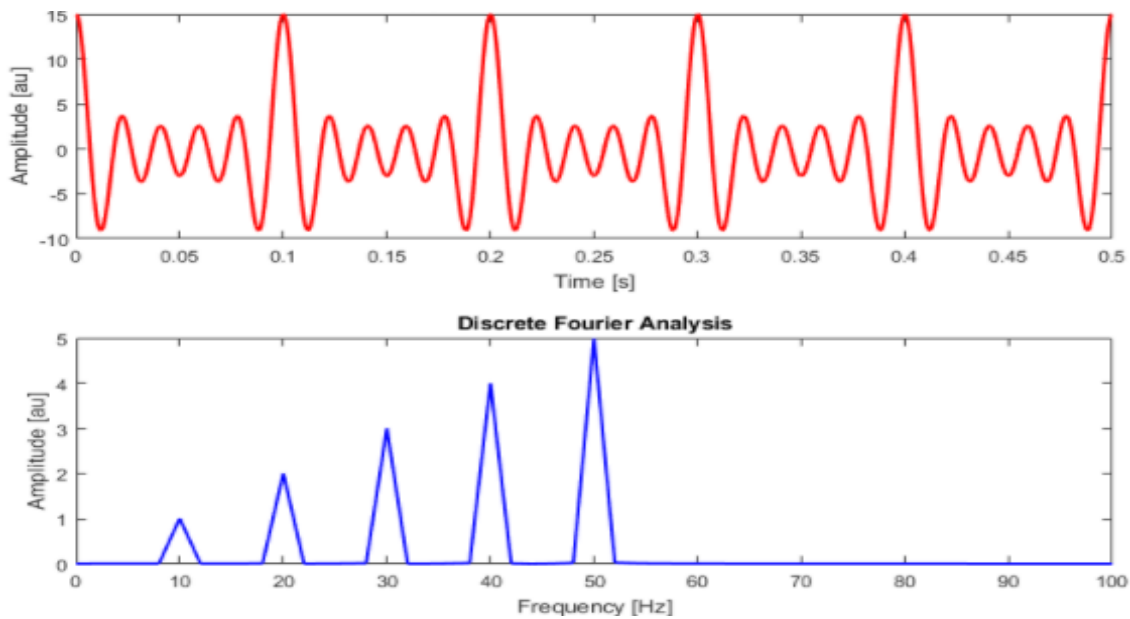


Figura 9: Aplicación de DFT. Fuente: <https://rufianenlared.com/fft/>

DFT es ocupada en áreas como el procesamiento de audio, para analizar las señales desde el punto de vista de la frecuencia, tal como se señaló en 2.1 se puede usar para generar espectrogramas.

2.2 Transformada rápida de Fourier (FFT)

FFT es una familia de algoritmos que buscan calcular la DFT de forma eficiente, algunos de ellos lo hacen en $O(n \log(n))$. El algoritmo más común de esta familia es Cooley-Tukey, cuyos creadores fueron James William Cooley y John Wilder Tukey (Lee, n.d.). Es ocupada en algunas áreas como procesamiento de imágenes y audio.

2.3 Espectrograma

El espectrograma es una representación visual de la variación de la energía contenida en cada frecuencia que compone una señal a través del tiempo (Orellana, 2021). Esta se calcula aplicando DFT a la señal, por lo general se usa FFT que es una familia de algoritmos que permite calcular de forma eficiente DFT (Paz et al., 2020, 35). Entre sus representaciones está la forma bidimensional donde el eje X corresponde al tiempo y el eje Y a las frecuencias que componen el sonido, los colores muestran su energía, también es posible encontrar su representación tridimensional donde el eje X e Y coinciden con el caso bidimensional y el eje Z es su energía.

Algunas de sus aplicaciones son: extracción de características, visualización de señales eléctricas y de comunicaciones, análisis acústico de la voz, entre otras. A continuación se presentan las figuras 10 y 11 que muestran las dos representaciones de espectrogramas mencionadas anteriormente.

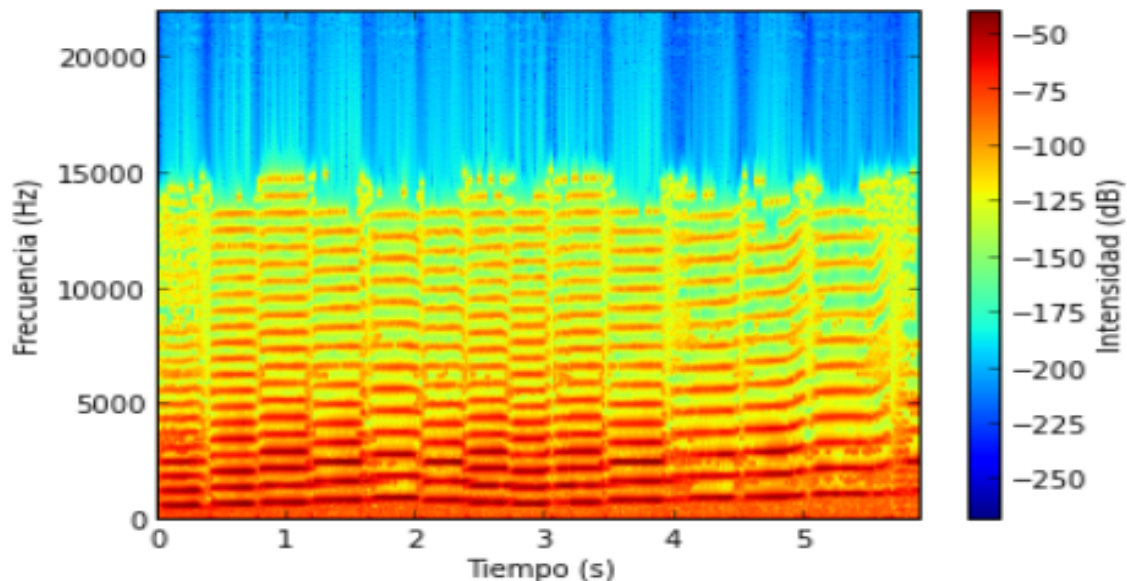


Figura 10: Representación bidimensional del espectrograma. Fuente: <https://pybonacci.org/2012/09/29/transformada-de-fourier-discreta-en-python-con-scipy/>

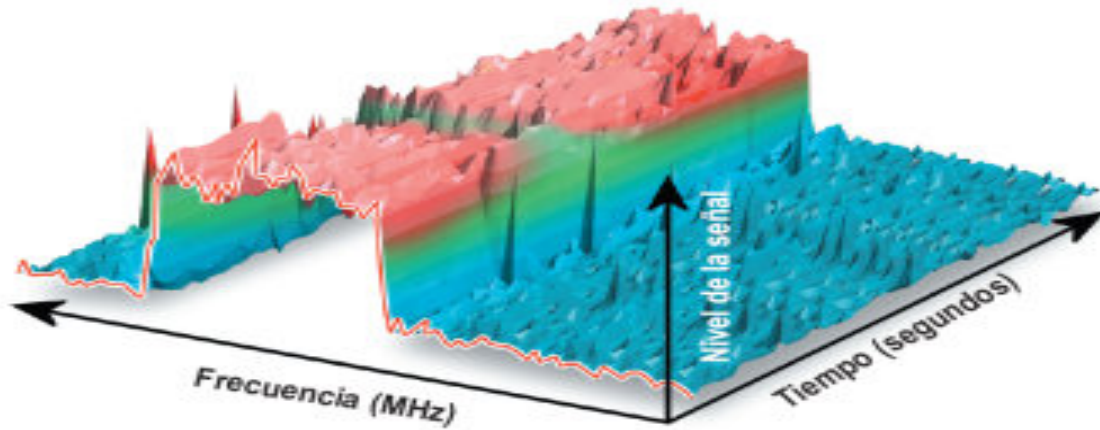


Figura 11: Representación tridimensional del espectrograma. Fuente: <https://www.promax.es/esp/noticias/158/Espectrograma/>

2.4 Escala de Mel

La escala de Mel fue propuesta por Stevens, Volkman y Newman en 1937, esta se basa en la forma en la cual los humanos perciben el sonido. Fue construida a partir de tests psicoacústicos (López Monfort, 2015). Según señala (Ruiz Vargas, 2018, 32), por debajo de los 1000 Hz los intervalos de frecuencias se perciben de forma lineal, en cambio por encima de los 1000 Hz los intervalos de frecuencias se perciben de forma logarítmica. Esto se debe a que las personas son mejores percibiendo diferencias en bajas frecuencias.

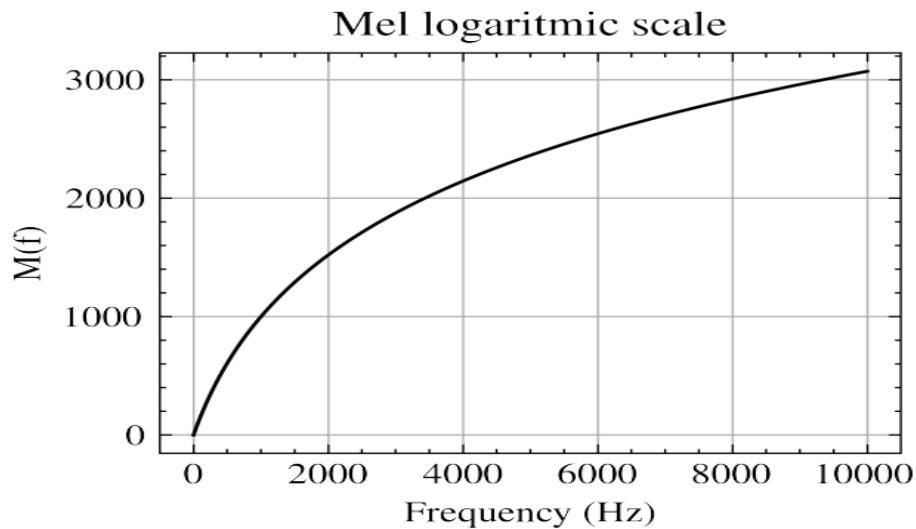


Figura 12: Escala de Mel. Fuente: <https://repositorio.uchile.cl/bitstream/handle/2250/181887/Deconvolucion-en-audio-utilizando-modelos-basados-en-Machine-Deep-Learning.pdf?sequence=1>

2.5 Espectrograma de Mel

El espectrograma de Mel, es un espectrograma al cual se le ha aplicado la escala de Mel al eje de las frecuencias, mostrando una representación visual de las frecuencias más acorde a como serían interpretadas por el oído humano (Acha Ruiz, n.d.). Donde en el eje X se encuentra el tiempo, en el eje Y está la frecuencia transformada a escala de Mel y la intensidad del sonido se mide en decibeles.

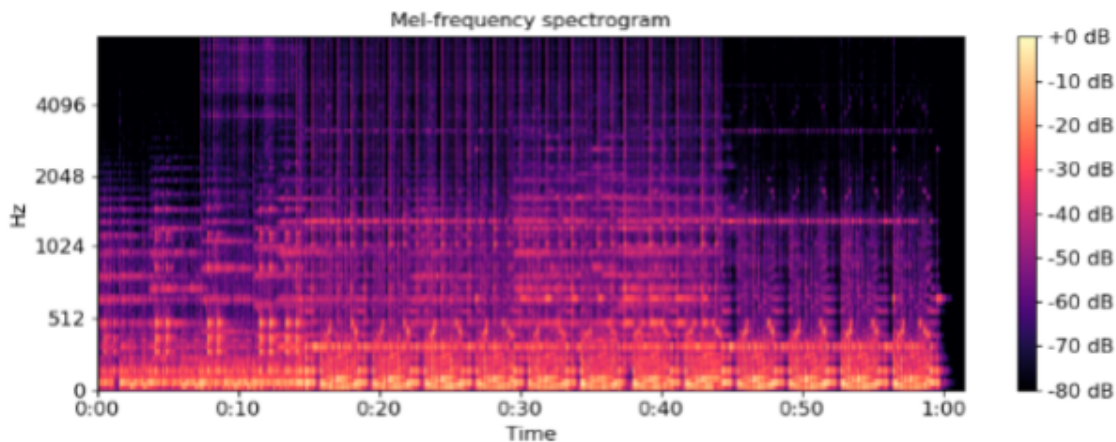


Figura 13: Ejemplo de espectrograma de Mel. Fuente: <https://guillermoacharuz.com/portfolio/biometriadevoz/>

2.6 Coeficientes cepstrales en la escala de Mel (MFCCs)

Según (Rincón Llorente, 2007, 45), los MFCCs son coeficientes para la representación del habla basados en la percepción auditiva humana. Esto último se debe a que en el proceso de su obtención se usa la escala Mel. El siguiente diagrama muestra el proceso que se debe realizar para obtener los MFCCs.

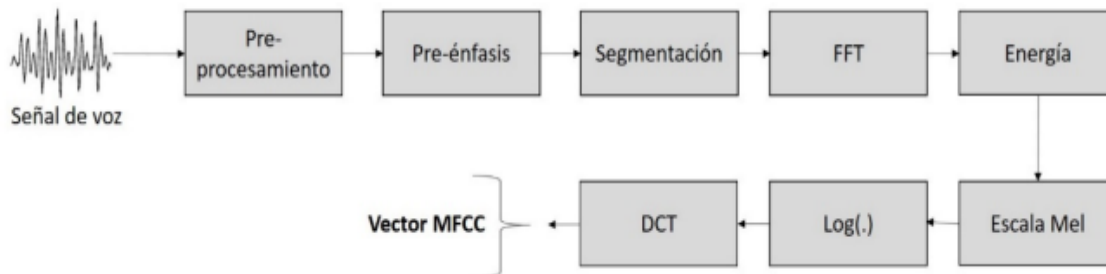


Figura 14: Proceso para obtener los MFCCs. Fuente: <https://www.revistaespacios.com/a20v41n32/a20v41n32p16.pdf>

Con los pasos que van desde el pre-procesamiento hasta la energía se obtiene un espectrograma, al aplicar la escala de Mel se obtiene un espectrograma de Mel, luego se aplica un logaritmo para transformar la amplitud a decibeles y al aplicar finalmente la

transformada discreta del coseno (DCT) se obtienen los coeficientes cepstrales de Mel.

Estos coeficientes se han utilizado en sistemas de reconocimiento de habla, reconocimiento de locutores, análisis de música entre otras.

2.7 Red neuronal convolucional

Las *Convolutional Neural Networks* o CNNs por sus siglas en inglés, son un tipo especial de redes neuronales que cuentan con capas¹² de convolución y *pooling*, además asumen que los datos que procesarán tendrán forma matricial aunque esto puede variar, tal como señala (García, 2019, 17). Tienen un amplio uso en el área de la visión artificial siendo algunos la clasificación de imágenes, detección de objetos dentro de imágenes, reconocimiento facial entre otros. También se ocupan para el procesamiento de voz. La figura 15 presenta la estructura clásica de una CNN que realiza tareas de clasificación.

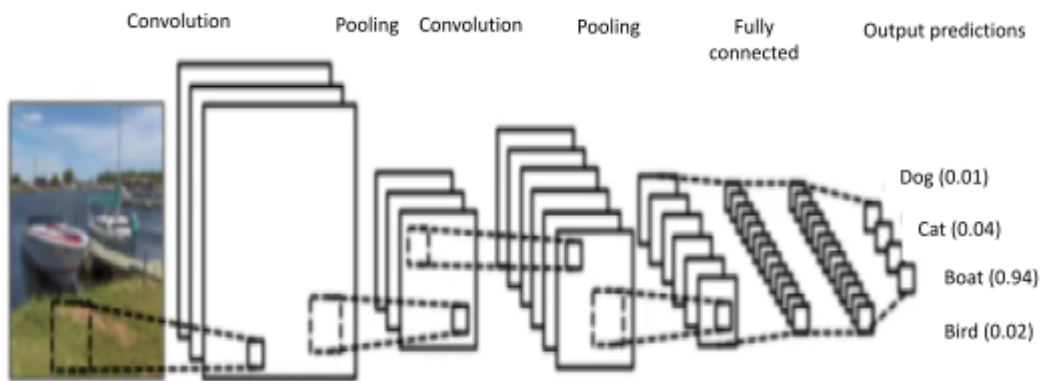


Figura 15: Arquitectura de una CNN clasificadora. Fuente:
<https://datasmarts.net/es/redes-neuronales-convolucionales-en-profundidad>

En general, este tipo de CNN está compuesta por una capa de entrada, varias de convolución (*convolution*) intercaladas con *pooling* y finalmente se usan las capas clasificadoras (*fully connected*) que se encargarán de discriminar entre las diferentes clases (*output predictions*) entregando los resultados a la capa de salida. A continuación se explicará en qué consiste cada una.

2.7.1 Capa de entrada

Recibe los datos que debe procesar la CNN, por lo general estos tienen forma matricial aunque puede variar, tal como señala (García, 2019, 17). La figura 16 muestra el sitio donde está ubicada.

¹² En el área de redes neuronales, una capa es un conjunto de neuronas. Una red neuronal está compuesta por un grupo de ellas, donde los datos recibidos provienen de la capa anterior y los datos procesados van a la posterior.

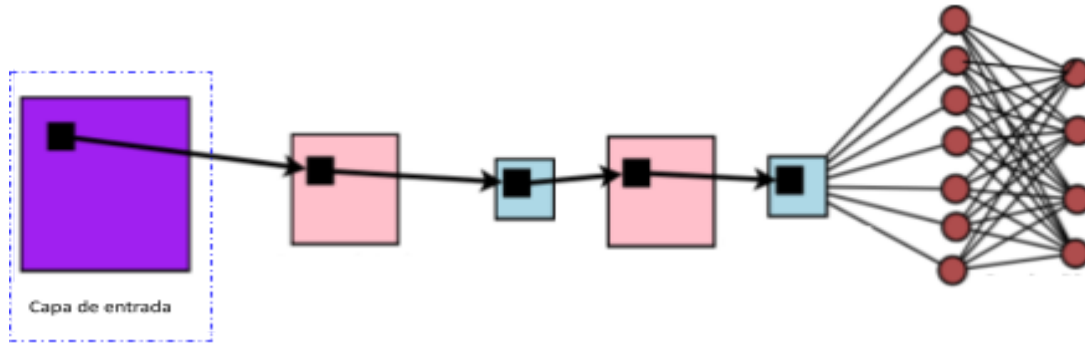


Figura 16: Ejemplo de CNN, entrada denotada por el rectángulo azul. Fuente:

https://www.rcs.cic.ipn.mx/2019_148_7/Clasificador%20de%20plantas%20medicinales%20por%20medio%20de%20Deep%20Learning.pdf

En el área de visión artificial, la capa de entrada por lo general tendrá forma de matriz (para imágenes en escala de grises) o cubo (para imágenes a color). Los datos recibidos por ella, son entregados a la capa de convolución.

2.7.2 Capa de convolución

Son aquellas que utilizan la operación matemática convolución para procesar los datos que reciben. La convolución toma dos funciones y las transforma en una tercera, en el caso de las CNNs, usa los datos provenientes de la capa anterior y una matriz denominada filtro o *kernel*, generando como resultado un mapa de características¹³, tal como señala (García, 2019, 18-19).

Estas van ubicadas luego de la capa de entrada o de *pooling*, tal como se puede apreciar en la figura 17.

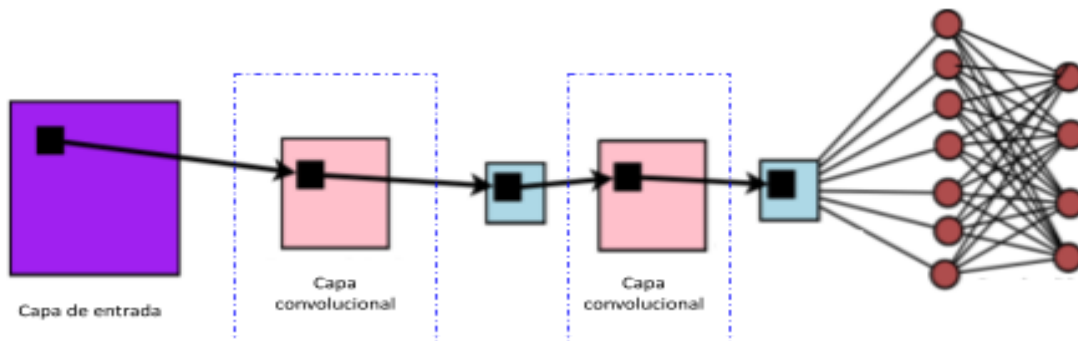
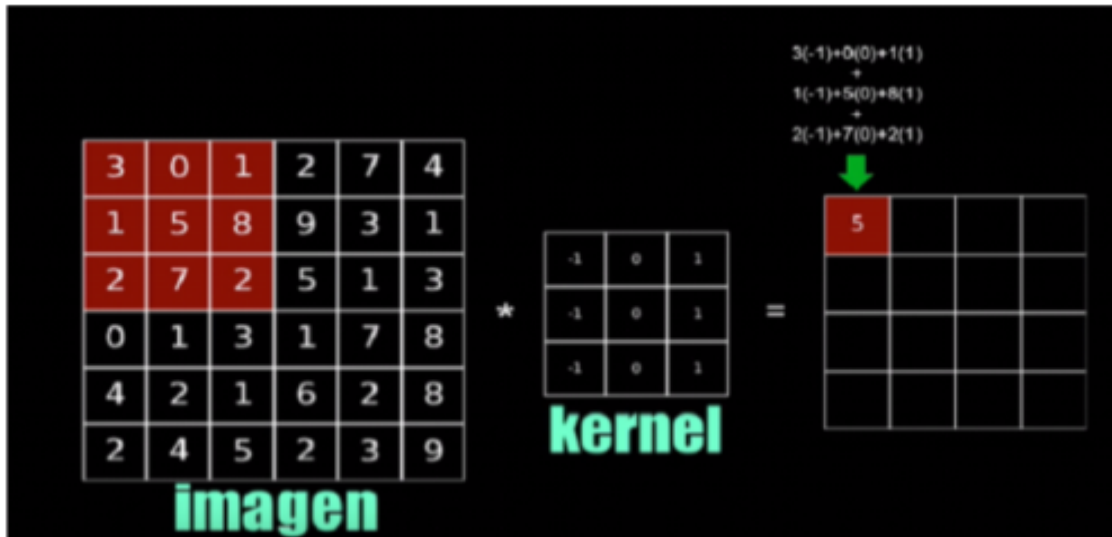


Figura 17: Ejemplo de CNN, convolución denotada por el rectángulo azul. Fuente:

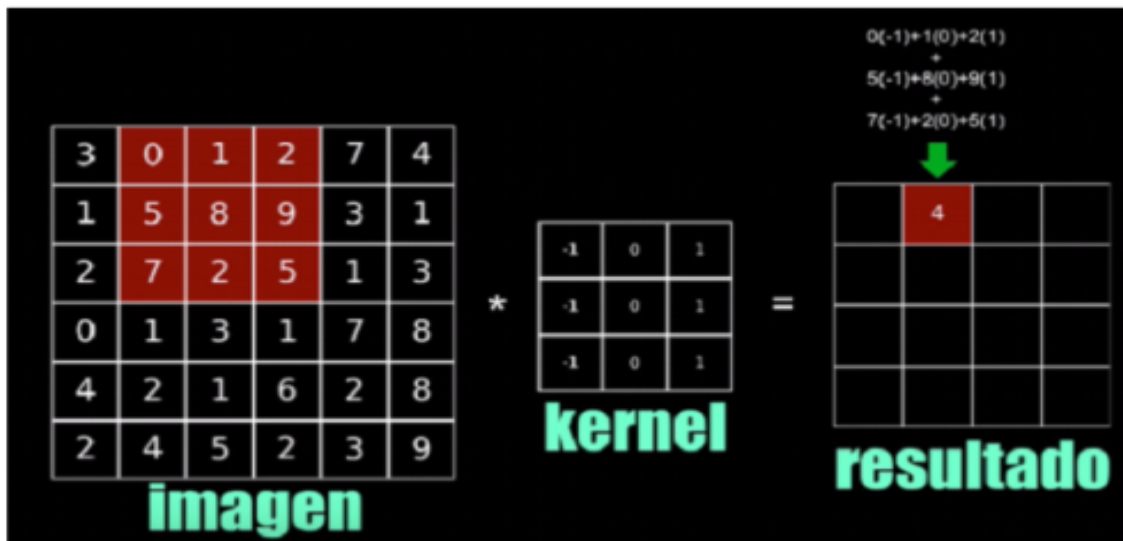
https://www.rcs.cic.ipn.mx/2019_148_7/Clasificador%20de%20plantas%20medicinales%20por%20medio%20de%20Deep%20Learning.pdf

¹³ Hace referencia a la matriz obtenida en la capa de convolución, luego de pasar el filtro o kernel por los datos de entrada.

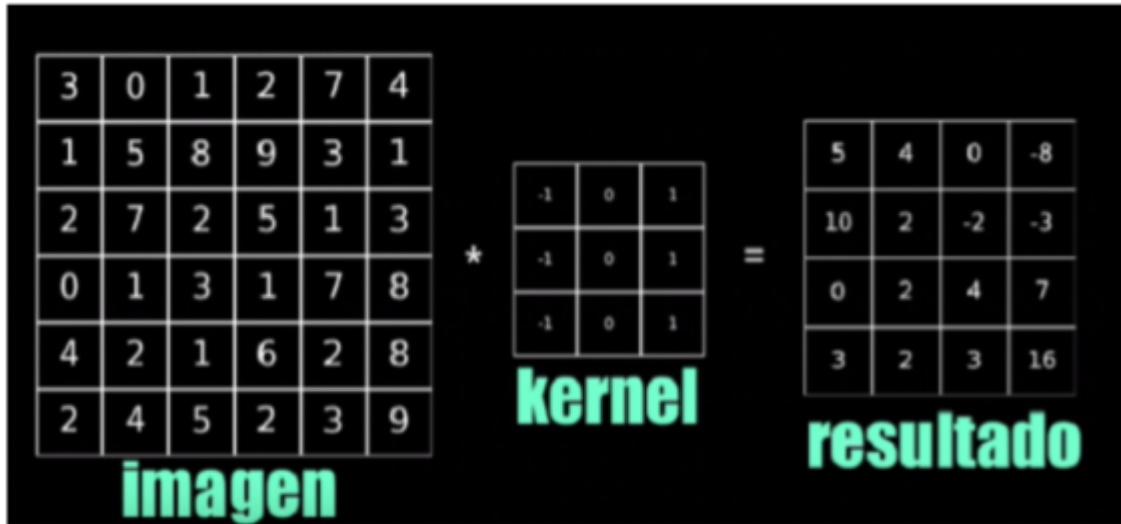
El procedimiento que realizará la convolución dentro de la CNN será tomar un grupo de elementos provenientes de la capa anterior y calculará el producto punto entre ellos y el filtro, obteniendo como resultado un valor que será parte de la matriz convolucionada, luego se tomará otro grupo de elementos mediante el desplazamiento horizontal del filtro y se volverá a hacer el procedimiento hasta que se haya procesado toda la entrada. La figura 18 muestra un ejemplo del proceso, para un mejor entendimiento de el, la página <https://programmerclick.com/article/49251777146/> proporciona el ejemplo 6 en el cual se puede apreciar mediante el uso de animaciones, el cálculo de los coeficientes de la matriz convolucionada.



(a) Producto punto entre el primer grupo de píxeles y kernel de tamaño 3x3.



(b) Se desplaza el kernel un píxel y se vuelve a realizar el proceso anterior.



(c) Se repite el proceso hasta pasar el filtro por toda la imagen.

Figura 18: Ejemplo de convolución. Fuente: <https://bootcampai.medium.com/redes-neuronales-convolucionales-5e0ce960caf8>

2.7.3 Capa de pooling

Su finalidad es reducir el volumen de los datos extrayendo los píxeles más representativos, con el fin de reducir la cantidad de neuronas de la CNN. (García, 2019, 20) menciona dos tipos de *pooling*: *max-pooling* que busca obtener los valores máximos de los elementos provenientes de la capa anterior en determinadas regiones y *average-pooling* similar al anterior pero funciona con el promedio en lugar del máximo. La figura 19, muestra un ejemplo de *max-pooling*.

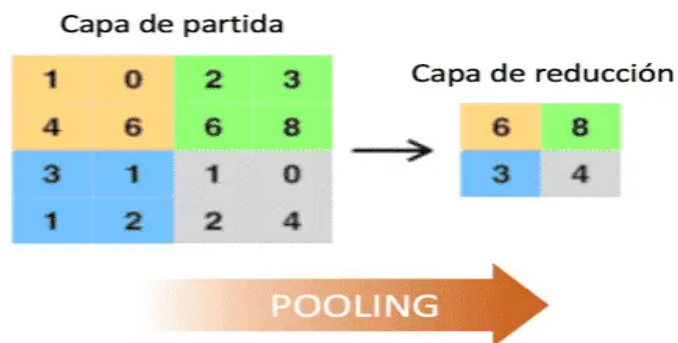


Figura 19: Ejemplo de *pooling*. Fuente: <https://www.diegocalvo.es/red-neuronal-convolucional/>

Reducir la cantidad de neuronas utilizadas disminuye el tiempo de procesamiento de la red, además permite bajar el sobreajuste del modelo. Suele ir luego de una capa de convolución, tal como se puede apreciar en la figura 20.

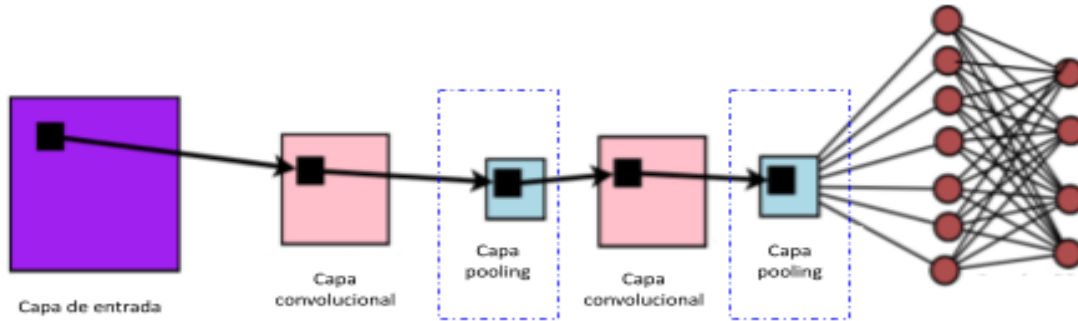


Figura 20: Ejemplo de CNN, *pooling* denotada por el rectángulo azul. Fuente: https://www.rcs.cic.ipn.mx/2019_148_7/Clasificador%20de%20plantas%20medicinales%20por%20medio%20de%20Deep%20Learning.pdf

2.7.4 Capa de aplanado

Recibe datos en forma matricial y los transforma a una versión vectorial, para que la capa de clasificación pueda procesar los datos provenientes del *pooling*. La figura 21 muestra su ubicación, está entre el último *pooling* y la clasificación (*fully connected*).

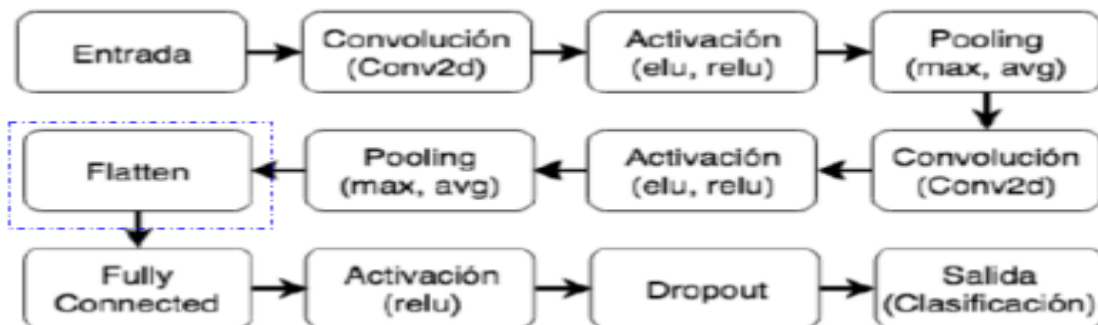


Figura 21: Aplanado (*Flatten*) arquitectura CNN, denotada por el rectángulo azul. Fuente: https://www.researchgate.net/publication/339207158_Optimizacion_Distribuida_de_Rednes_Convolucionales_para_la_Clasificacion_de_Imagenes

2.7.5 Capa clasificadora

Viene después de la capa de aplanado, y tiene como finalidad aprender a clasificar los elementos entregados a la CNN. Compuesta por una o varias capas de neuronas totalmente conectadas a la capa siguiente, usada para hacer un clasificador, tal como lo menciona (Quezada, 2020, 11). Generalmente, suele terminar con una capa que usa la función *softmax*, esta función permite generar un vector de probabilidades donde cada elemento indica la probabilidad de que la salida pertenezca a una determinada clase. La figura 22 muestra su posición dentro de la CNN.

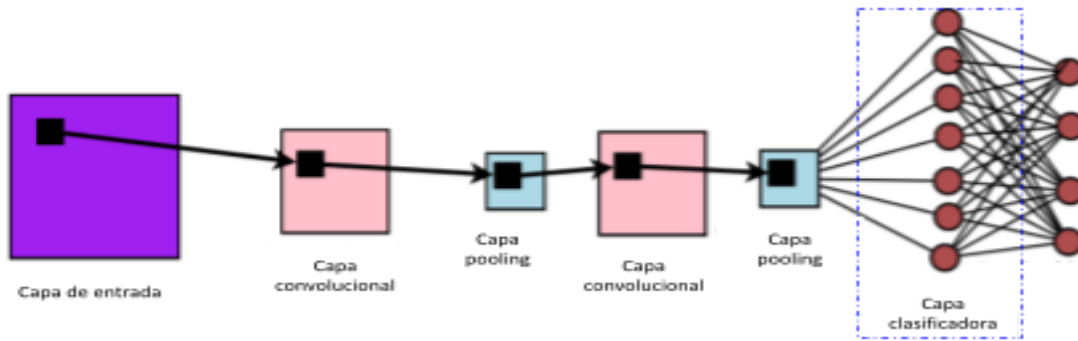


Figura 22: Ejemplo de CNN, clasificación denotada por el rectángulo azul. Fuente: https://www.rcs.cic.ipn.mx/2019_148_7/Clasificador%20de%20plantas%20medicinales%20por%20medio%20de%20Deep%20Learning.pdf

2.7.6 Capa de salida

Última capa de la CNN, está conectada con la de clasificación y entrega los resultados del procesamiento que realiza la CNN sobre los datos de entrada. Tal como señala (Quezada, 2020, 11) en el caso de tareas asociadas a clasificar, la cantidad de neuronas que debe tener es igual a la cantidad de clases involucradas en el problema que a su vez es el mismo número de neuronas que tiene la capa final del clasificador. La figura 23 muestra su ubicación dentro de la CNN.

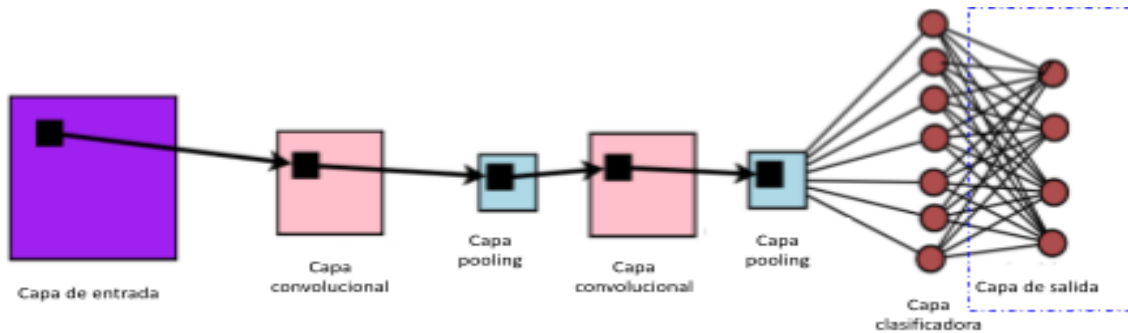


Figura 23: Ejemplo de CNN, salida denotada por el rectángulo azul. Fuente: https://www.rcs.cic.ipn.mx/2019_148_7/Clasificador%20de%20plantas%20medicinales%20por%20medio%20de%20Deep%20Learning.pdf

2.8 Estado del arte

Entre los estudios que se dedican al monitoreo de la salud de la colmena mediante el procesamiento del zumbido de la misma, se encuentran la detección de la ausencia de la abeja reina, el enjambre, la presencia de Varroa, entre otros. A continuación se presentarán algunas de las investigaciones examinadas.

A. (Ferrari et al., 2008, 72-77) usan audio obtenido de abejas para identificar el periodo de enjambre¹⁴ de la abeja *Apis Mellifera Ligustica* italiana. Captura los datos de sonido, humedad y temperatura del panal y analiza el sonido mediante espectrograma, con el fin de investigar los cambios en el zumbido y la relación que tiene con los cambios en la temperatura y humedad. Se obtuvo que en los momentos de enjambre la frecuencia de la colmena aumenta, pasa de estar en el rango de 100 - 300 Hz (actividad normal) a 500 - 600 Hz (enjambre) y la temperatura baja de 35 C° a 33 C°. La figura 24 muestra la grabación de los sonidos emitidos por la colmena, donde la franja verde corresponde a la frecuencia registrada en el evento de enjambre, la franja azul es la frecuencia de la actividad nocturna y la roja corresponde a la actividad diurna, la figura 25 presenta el espectrograma obtenido a partir del evento de enjambre.

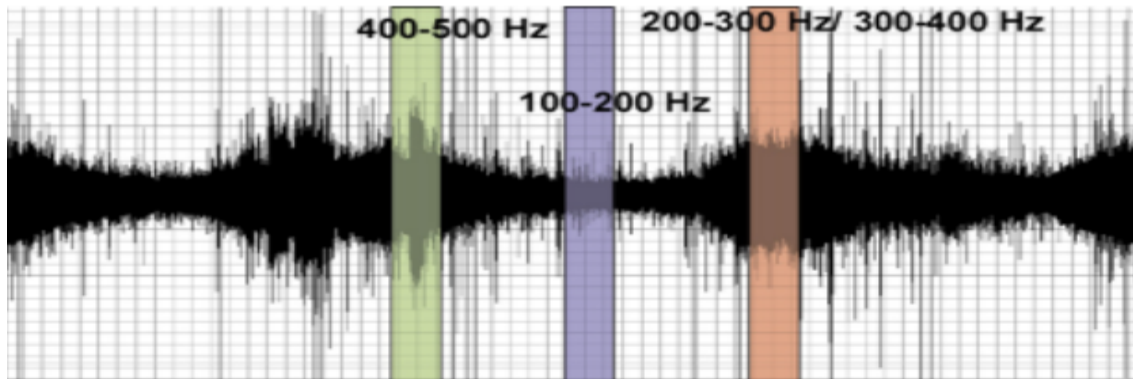


Figura 24: Imagen de la actividad de la abeja *Apis Mellifera Ligustica* italiana registrada durante 3 días seguidos, etiquetado manualmente. Fuente: (Ferrari et al., 2008, 75).

¹⁴ Hace referencia al proceso en el cual las abejas abandonan una colmena, para establecerse en otro sitio y formar una nueva.

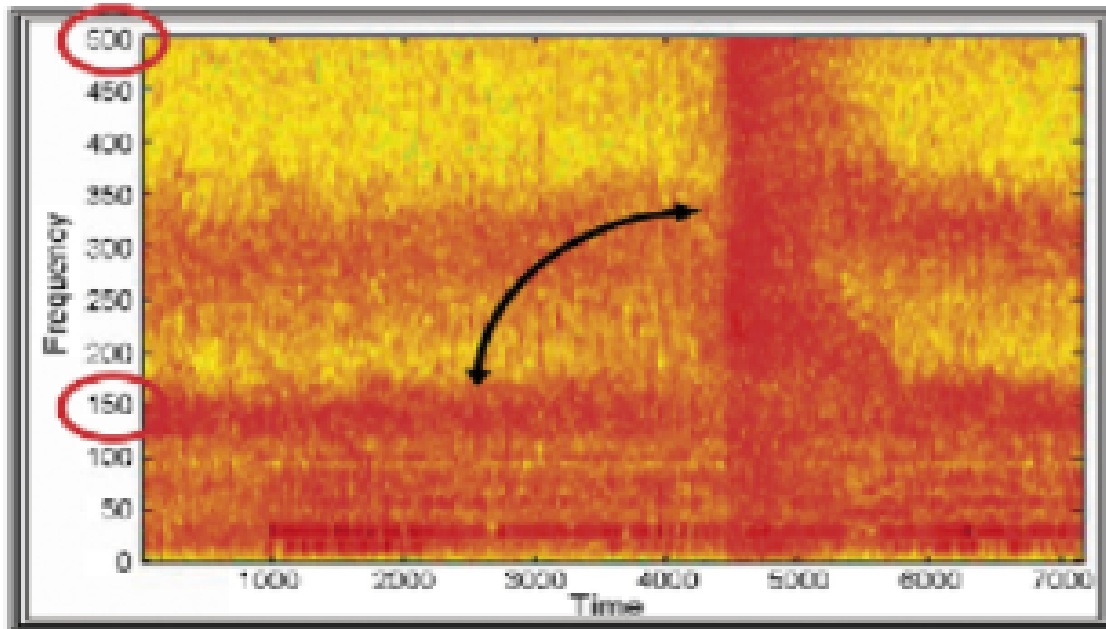


Figura 25: Espectrograma del evento de enjambre. Fuente: (Ferrari et al., 2008, 75).

Se puede apreciar cómo se produce un salto en la frecuencia cuando la colmena pasa del estado normal (150 Hz aprox) al evento de enjambre (500 Hz aprox).

B. (Qandour et al., 2014, 204-209) propusieron un sistema que indicaba si una colmena estaba o no enferma de Varroa, mediante el procesamiento de su zumbido. Se utilizó un *dataset* con grabaciones de audio de colonias sanas e infestadas. Fue ocupada (según la arquitectura mostrada en la figura dos de ese artículo) FFT y se extrajeron cuatro características (conjunto de datos inicial): PF, BW, SC y RVF. Se realizaron cuatro experimentos, en los dos primeros se aplicó PCA para reducir las características a dos y se usaron para entrenar dos modelos de clasificación, una SVM y otra basada en LDA, los otros dos experimentos eran similares a los anteriores solo que no se utilizó PCA, en su lugar se entregó el conjunto de datos inicial a los modelos. Usar todas las características permitió a ambos obtener una alta precisión al momento de identificar el estado de la colmena, a costa de aumento en el tiempo de cálculo, se menciona que SVM y LDA generaron porcentajes de precisión de predicciones mejores que los porcentajes basados en probabilidad.

C. (Qandour et al., 2014, 204-209) continuación del trabajo presentado en **B.** Se realizaron cuatro experimentos, se entrenó SVM y LDA con las cuatro características usadas en el trabajo anterior, luego en cada uno de los siguientes tres se aplicó PCA para reducir la dimensionalidad de los datos, por lo que se entrenaron los modelos con los tres, dos y una característica más importantes. La precisión obtenida al momento de clasificar el estado de las abejas (sanas o con el parásito) fue: 70% diferenciando los zumbidos mediante SC, 75% usando SC y PF, 79% mediante SC, PF y RVF y finalmente 81% al utilizar todos los datos, al igual que aumentó la precisión en este último caso, también lo hizo el

tiempo de cálculo.

D. (Robles et al., 2017, 89-98) determinaron la ausencia de la abeja reina dentro de una colonia usando una metodología que aplica MFCCs y modelo de regresión logística para la clasificación. MFCC fue utilizado en la extracción de características, luego de su cálculo se generaron los siguientes descriptores estadísticos por cada coeficiente: media, media truncada, asimetría estadística, coeficiente de variación inverso, curtosis, desviación estándar, mediana, varianza, coeficiente de varianza y cuantiles (2.5, 25, 50, 75, 97.5). Posteriormente para la selección se utilizó SVD para obtener el mejor estadístico que pudiera discriminar la ausencia de reina, los resultados mostraron que uno solo era suficiente para realizar buenas predicciones, por lo que se eligió la media, además no todos los coeficientes eran necesarios, con *Random Forest* se pudo identificar que los dos más importantes eran el 4 y 10. Finalmente se entrenó el modelo con los datos. Como resultado se obtuvo una buena clasificación mediante el uso de las medias de los coeficientes 4 y 10.

E. (Nolasco et al., 2018) para determinar la ausencia de la abeja reina, utilizaron un enfoque basado en dos pasos, el primero consistió en el uso de MFCCs, espectrograma de Mel y HHT para la extracción de características, luego en el segundo paso correspondiente a la clasificación se emplearon SVMs y CNNs. Los modelos fueron probados con diferentes conjuntos de características, para SVM se utilizaron los 5 siguientes: 1. uso de 20 coeficientes de Mel (SVM_MFCCs20), 2. modificación en la frecuencia máxima del espectrograma generado por HHT generando 20 bandas de frecuencias (SVM_HHTdwns20), 3. combinación de los dos anteriores (SVM_MFCCs20_HHTdwns20), 4. cada espectrograma de Mel de una grabación se promedia a lo largo del tiempo generando un vector de 120 dimensiones, reduciéndola a 20 bandas (SVM_MEL120dwns20), 5. similar a la anterior pero se usa un espectrograma log-Mel (SVM_LOG_MEL120dwns20). En el caso de CNN, los conjuntos utilizados fueron 3: 1. uso de 20 coeficientes de Mel (CNN_MFCCs20), 2. los datos de entrada son obtenidos desde espectrograma de Mel con 120 bandas de frecuencia (CNN_MEL120), 3. similar a lo anterior pero se usa un espectrograma log-Mel (CNN_LOG_MEL120). SVM obtuvo los mejores resultados cuando utilizó la combinación HHT y MFCCs para discriminar los colmenares en cambio CNN lo hizo procesando MFCCs, cabe destacar que SVM generó un mejor resultado que CNN.

F. (Harding, 2021) indica que quizás el problema de la Varroa podría ser tratado desde otro punto de vista, mediante la hipótesis “las abejas buscan sitios con vibración de 250 Hz para establecerse”. Se explica que las abejas en el bosque vibran a una frecuencia de 190 - 250 Hz, por lo que colocar los panales en lugares que tengan una frecuencia de 250 Hz (generado por ríos subterráneos, en la explicación lo mencionan como línea de fluido o línea) podría modificar el estado de salud del colmenar. Se presentaron dos casos de estudio, en el primero se toman dos colmenas A y B (con Varroa), A se coloca sobre una

línea, luego de seis u ocho semanas presentaría muy poca presencia de Varroa en cambio B aún tendrá mucha y estará más debilitada, al cambiarlas de posiciones, A comenzará a cargarse del parásito y B la disminuirá. En el segundo caso de estudio se toman dos colmenares C y D (con Varroa), C es colocada sobre una línea (dentro de seis u ocho semanas pasaría lo mismo que con A y B), pasado tres meses se intercambian las reinas de los colmenares, con el supuesto de que el estado de C y D cambiarían, pero estos se mantuvieron. De lo anterior se concluyó que un factor de importancia que influye en la salud del colmenar es la frecuencia que envuelve el sector en el cual es colocado.

G. A continuación se muestra la tabla que resume los enfoques utilizados en el estudio de los zumbidos de las colmenas de abejas presentados por (Terenzi et al., 2020, 168).

Tabla 2: Traducción del resumen de las investigaciones asociadas al procesamiento del audio emitido por las abejas. Fuente (Terenzi et al., 2020, 168).

Enfoque	Descripción	Aplicación
Espectrogramas.	El sonido es grabado y analizado usando espectrogramas, para buscar cambios en el contenido armónico de la señal.	Análisis del baile de las abejas, análisis del <i>piping</i> y <i>tooting</i> . Detección de enjambre. Medir la reacción de las abejas ante un ataque de avispa.
Síntesis de sonido basado en tono.	Se coloca un parlante dentro de la colmena, se generan diferentes tonos con diferentes amplitudes y frecuencias. Las reacciones de las abejas son monitoreadas.	Encontrar la frecuencia en la cuál las abejas cesan sus movimientos. Reproducir el armónico generado por la abeja reina para estimular el enjambre.
Monitoreo de la amplitud.	La amplitud y la envolvente de la señal grabada es usada para detectar diferentes comportamientos.	Cambios de amplitud en diferentes estaciones y condiciones. Medir el nivel de presión de sonido durante el baile de la abeja. Detección de enjambre.
Síntesis de sonidos de abejas.	Primero se graban los sonidos emitidos por las abejas, luego se analizan y sintetizan computacionalmente. Luego los sonidos sintetizados son reproducidos dentro de la colonia y sus reacciones son monitorizadas.	Medir la respuesta de las abejas obreras frente a los sonidos sintetizados de la abeja reina.

Análisis de ruido.	Los sonidos grabados dentro de la colmena son considerados un ruido con características estadísticas específicas. Algunos indicadores estadísticos son extraídos de los sonidos, cambios en los indicadores estadísticos son asociados a comportamientos específicos de la colmena.	Detección y predicción de enjambre.
Análisis de indicadores estadísticos.	De los sonidos grabados se extraen la frecuencia pico, centroide espectral, ancho de banda, frecuencia de variación. PCA es usado para reducir la dimensionalidad de los indicadores y finalmente se usa SVM y LDA para clasificar la señal.	Detectar la presencia de la Varroa destructor dentro de la colonia.
Detección de grito.	Acelerómetros de precisión dentro de las colmenas son usados para grabar la vibración de las abejas. Los espectrogramas de vibraciones están correlacionados con una señal de pulso para detectar señales pulsadas. Luego LDA y PCA son usados para aislar señales de grito.	Medir la variación de las señales de grito durante diferentes estaciones y ubicaciones geográficas.
Detección de sonidos de abejas.	Los sonidos son adquiridos dentro de la entrada de la colmena. Los espectrogramas de los sonidos grabados son clasificados usando diferentes algoritmos, tales como: CNNs, regresión logística, SVM, K-NN, uno vs. el resto y bosques aleatorios.	Distinguir el sonido de las abejas del ruido de fondo y de los chillidos de los grillos.
Análisis de sonido mediante codificación predictiva lineal (LPC).	Los sonidos adquiridos dentro de la colmena son analizados usando LPC como algoritmo de extracción de características. El	Detección de presencia de abeja reina.

	algoritmo T-SNE es usado para reducir la dimensionalidad y finalmente SVM es usado para clasificar la señal.	
Análisis mediante HHT y MFCC.	Los audios grabados dentro de las colmenas son analizados usando MFCCs y HHT como características. CNNs y SVM son aplicados para clasificar las señales.	Detección de abeja reina. Detección de enjambre.
Análisis MFCC.	MFCCs son estimados de las señales grabadas, regularización <i>Lasso</i> es usada para reducir la dimensionalidad y finalmente algoritmos de regresión logística son usados para la clasificación.	Detección de abeja reina.
Análisis <i>Wavelet</i> .	La transformada de <i>Wavelet</i> es aplicada a las señales grabadas para analizar el sonido y detectar diferentes comportamientos.	Detección de abeja reina. Detección de enjambre.
Análisis MFCC y LPC.	MFCC y LPC son usados como características, HMM y GMM son usados como clasificadores.	Detección de enjambre.
FFT multidimensional.	Los espectrogramas de dos y tres dimensiones son generados a partir de los sonidos grabados usando acelerómetros colocados dentro de las colonias. Una función discriminante es usada para clasificar las señales y detectar eventos específicos usando dos algoritmos diferentes.	Detección y predicción de enjambre.

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN

Esta sección tiene como principal objetivo describir la metodología usada para generar el modelo de reconocimiento de patrones de audio, utilizado en la clasificación del estado de salud de la colmena de abejas, que incluye tareas tales como: recopilación de audios, generación del *dataset*, extracción de características, selección de características y otras. Además se explicará la arquitectura de software utilizada para que el usuario pueda interactuar con el algoritmo en cuestión.

3.1 Entornos de trabajo

A continuación se describirán las principales herramientas usadas en los diferentes entornos usados para generar y desplegar el sistema.

3.1.1 Extracción de características y modelo de reconocimiento de patrones

Para las labores asociadas a la extracción de características y la obtención del modelo de reconocimiento de patrones, se trabajó ejecutando código *Python* en *Google Colaboratory* con *Google Computer Engine*, esto se hizo para contar con un entorno de trabajo cómodo de usar y para aprovechar recursos como las GPUs proporcionadas por dicho producto. A continuación se dan más detalles sobre dichas herramientas.

Google Computer Engine

Como señala (*Preguntas Frecuentes De Google Compute Engine | Documentación De Compute Engine*, 2022), *Google Computer Engine* es un producto de infraestructura como servicio que ofrece VMs basadas en sistemas operativos *Linux* y *Windows*, opciones de almacenamiento local y duradero, y una *API REST* simple para la configuración y control.

(McLuckie, 2012) indica que entre los beneficios entregados por *Google Computer Engine* están escala, desempeño y valor (50% más de poder de cálculo, en comparación de otros proveedores, por la misma suma de dinero). Entre las capacidades ofrecidas se encuentran cómputo (inicio de VMs bajo demanda), almacenamiento, red y herramientas para configurar y controlar las VMs.

Cabe tener en cuenta que hay un plan gratuito que se puede consultar en el siguiente enlace <https://cloud.google.com/free/docs/gcp-free-tier/?hl=es-419#compute>, este incluye: 1 instancia de *VM e2-micro* no interrumpible por mes (ubicada en ciertas regiones de Estados Unidos), disco persistente estándar de 30 GB por mes, 5 GB de almacenamiento de instantáneas por mes (ubicada en ciertas regiones), salida de red de 1 GB por mes desde Norteamérica hacia todas las regiones de destino (excepto China y

Australia). También existen planes de pago, cuyo costo se calcula según el uso del producto, esto se puede calcular a partir de una hoja de costos que se puede encontrar en el siguiente enlace <https://cloud.google.com/compute/all-pricing?hl=es-419>.

Google Colaboratory

Como señala (*Google Colab*, n.d.), *Google Colaboratory* (abreviado como *Google Colab*) es un producto de *Google Research*. Este permite al usuario escribir y ejecutar código arbitrario de Python en el navegador. En general este producto es usado para tareas como aprendizaje automático, análisis de datos y educación. Permite usar recursos sin coste adicional, como por ejemplo GPU.

Hay un plan gratuito de *Google Colaboratory* que cuenta con 12.68 GB de memoria RAM, 78.19 GB de disco, además permite la opción de usar GPU, aunque esta no está disponible todo el tiempo en este plan. También existen planes de paga, estos son *Colab Pro* y *Colab Pro +*, el primero incluye GPU más rápida, más memoria y tiempos de ejecución más largos, en cambio el segundo plan ofrece ejecución en segundo plano, GPU más rápidas, aún más memoria y tiempos de ejecución todavía más largos. Se puede consultar el siguiente enlace para ver con mayor detalle los planes de paga mencionados anteriormente <https://colab.research.google.com/signup>.

3.1.2 Desarrollo

Para desarrollar el sistema se utilizaron contenedores docker en los cuales se instalaron todas las herramientas utilizadas por los componentes del sistema (esto se hizo para aislar los componentes del sistema de la máquina donde se estaba ejecutando), de esta forma se podía probar su funcionamiento a medida que estos eran desarrollados. Los contenedores eran desplegados de forma automática por medio de la herramienta docker-compose. A continuación se dan más detalles sobre esas herramientas.

Docker

(Docker Inc., n.d.) la documentación de *docker* indica que es una plataforma abierta para desarrollar, enviar y ejecutar aplicaciones. Permite separar las aplicaciones de la infraestructura, también se basa en conceptos de microservicios. Usa elementos como imágenes, contenedores y volúmenes que son usados para desplegar aplicaciones.

Docker-compose

(Docker Inc., n.d.) herramienta que permite definir y ejecutar aplicaciones que contienen varios contenedores *docker*, esto se hace por medio de un archivo YAML en el cual se define la estructura de los componentes del sistema, de esta forma se puede automatizar el despliegue de una aplicación que usa varios contenedores *docker*.

3.1.3 Producción

Para poder desplegar el sistema en la nube, se utilizaron dos servicios de *hosting* que proporcionan planes gratuitos, estos son *alwaysdata* y *PythonAnywhere*. *Alwaysdata* fue utilizado para hospedar todo lo asociado a la interfaz con la cual el usuario debe interactuar y *PythonAnywhere* fue utilizado para hospedar la *API* que contiene toda la lógica asociada al procesamiento de audio (la arquitectura del sistema se detalla en la sección 3.3). A continuación se darán más detalles sobre estos servicios.

Alwaysdata

(Alwaysdata, n.d.) Empresa francesa que proporciona servicios de *hosting*, ofrece soporte para sitios basados en *Python*, *Node.js*, *PHP*, *Ruby*, entre otros. También proporciona bases de datos administradas, copias de seguridad automáticas, dominios y correos ilimitados, sitios y aplicaciones ilimitados, servicios y tareas programadas. Ofrece distintos tipos de planes, el plan gratuito entrega 100 MB de disco para alojar los archivos del sitio web y tráfico limitado

PythonAnywhere

(Fernandez, 2016) Servicio de *hosting* que ejecuta sólo aplicaciones basadas en *Python*. Contiene administración de bases de datos, entornos virtuales, trae varios módulos de *Python* instalados de forma predeterminada, proporciona consolas *bash* o *Python*. Proporciona un plan gratuito para aplicaciones con bajo tráfico, este plan entrega 512 MB de disco, 100 segundos de CPU, permite tener abierta hasta dos consolas para interactuar con la máquina de *hosting*, permite tener solo una aplicación web desplegada entre otros beneficios.

3.2 Modelo de reconocimiento de patrones de audio

Para obtener un modelo de reconocimiento de patrones de audio que pueda reconocer el estado de salud de una colmena de abejas, se ha propuesto el uso de CNNs en combinación de distintas caracterizaciones del zumbido de la colmena. Para ello se han realizado 38 experimentos, en los cuales se han utilizado *datasets* que contienen audios asociados a colmenas con y sin abejas reinas, se han generado tres *datasets* cuyos audios tienen duración 1, 2 y 3 segundos (esto se ha hecho con la intención de examinar cómo la duración del zumbido afecta la identificación del estado de salud de la colmena) con 1667, 1635 y 1606 muestras respectivamente. Las características ocupadas para entrenar las CNNs son espectrograma logarítmico, espectrograma de Mel, 13, 20 y 40 MFCCs extraídos desde la serie de tiempo de la amplitud y del espectrograma de Mel. El entorno en el cuál se realizaron los experimentos fue *google colaboratory*, los recursos del entorno fueron 12.69 GB de RAM, 78.19 GB de disco y GPU del *backend* de *Google Computer Engine* que utiliza *Python* 3. Para la extracción de características se utilizó la librería *librosa* y para el

caso del modelo basado en CNNs, se ocupó la librería *keras*. En los experimentos realizados se probaron dos modelos basados en CNNs, el primero tiene 3 capas de convolución (con lo cual se pueden procesar 13, 20 y 40 MFCCs), el segundo tiene 4 capas de convolución (los que procesan 20 y 40 MFCCs). A continuación, se detalla un poco más los experimentos realizados.

1. *Dataset 1* [s]: Para este caso se probaron ambos modelos, usando espectrograma logarítmico, espectrograma de Mel, 20 y 40 MFCCs extraídos desde la serie de tiempo de la amplitud y desde el espectrograma de Mel. El primer modelo (3 capas de convolución) es el único con el que se probó los 13 MFCCs.

2. *Dataset 2* [s]: En este caso se realizó el mismo experimento descrito en el punto 1, pero no se utilizaron los espectrogramas logarítmicos, debido a que ocupa una cantidad mayor de memoria RAM de la que se disponía.

3. *Dataset 3* [s]: El experimento realizado fue el mismo que el expuesto en el punto anterior.

El siguiente diagrama muestra el proceso realizado para obtener el modelo de reconocimiento de patrones.

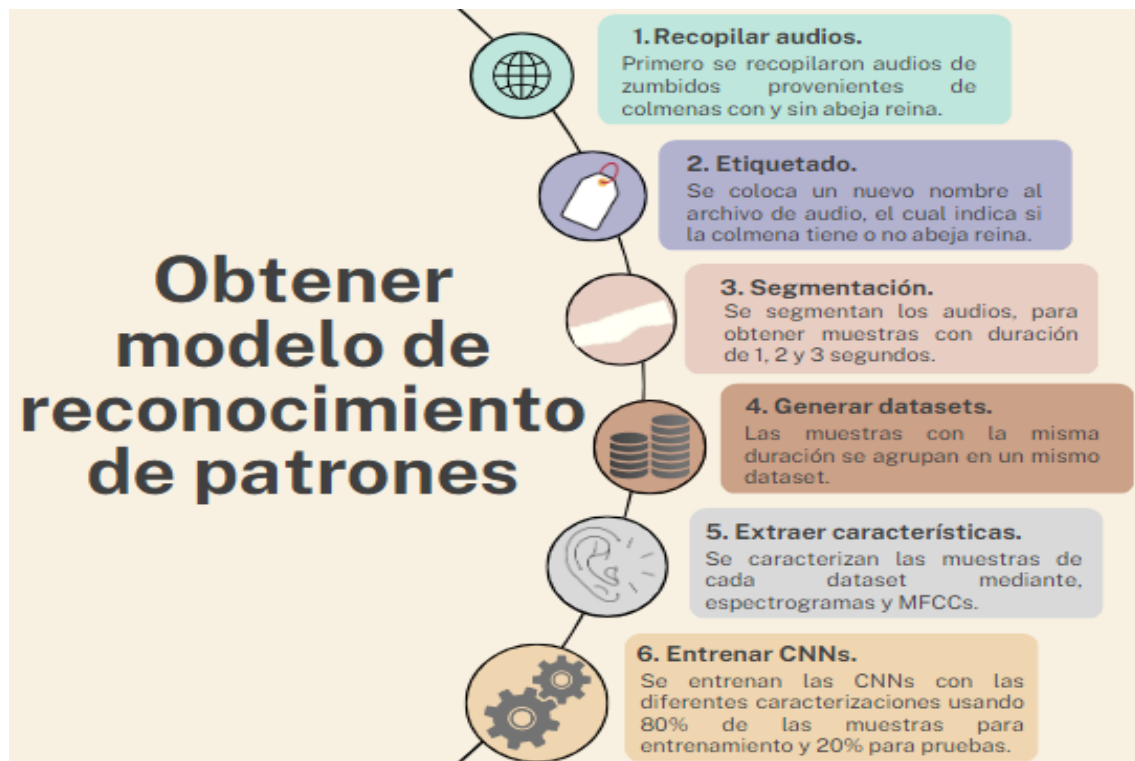


Figura 26: Diagrama del proceso para generar el modelo de reconocimiento de patrones. Fuente: Elaboración propia.

Cabe señalar que en los experimentos realizados, todas las CNNs fueron entrenadas con matrices de densidad de energía extraídas en la etapa de caracterización del audio de la colmena. Se hace énfasis en esto, ya que en algunos documentos asociados al entrenamiento de CNNs con espectrogramas, se ocupaban imágenes de espectrogramas en lugar de matrices de densidad de energía.

3.2.1 Obtención de audios

Para este escenario, se han recopilado 20 grabaciones provenientes de la comunidad *hiveeyes*, los cuales fueron obtenidos de 11 colmenas ubicadas en una ciudad de Berlín cerca de una calle con poco tráfico, las abejas de dichas colmenas pueden ser *apis mellifera cárnicas* o un híbrido entre *apis melliferas cárnicas* y abejas *buckfast*. El audio con duración mínima es de 32 [s] y la máxima tiene 90 [s]. 5 de ellos corresponden a colmenas sin abeja reina, el resto si tiene su presencia. La extensión de los archivos de audio es .mp3 y su tasa de muestreo es de 44100 muestras por segundo, lo que permite capturar frecuencias de hasta 22050 [Hz]. La mayoría de las grabaciones fueron recopiladas de la siguiente url <https://community.hiveeyes.org/t/sound-samples-and-basic-analysis-hive-with-queen-vs-queenless/399>.

3.2.2 Etiquetado

Se cambia el nombre original del audio por uno que representa su tipo de contenido, este nuevo nombre tiene el formato **#-estado**, donde **#** es un identificador único que no contiene el elemento - en su nombre, - es un símbolo separador que separa el identificador del estado de la colmena y **estado** es el estado en el que se encuentra la colmena, **SA** para colmena sin abeja reina y **CA** para colmena con abeja reina.

3.2.3 Generación de *datasets*

Luego de la recopilación de los audios, estos fueron segmentados en muestras de 1, 2 y 3 segundos. Las muestras con la misma duración fueron agrupadas para formar cada uno de los *datasets*.

3.2.4 Extracción de características

Para realizar este proceso, se utilizó principalmente el paquete *librosa* de *Python*, el cuál provee variadas funcionalidades que permiten procesar audio, además los resultados fueron graficados mediante el paquete *matplotlib* el cuál provee distintas funciones que permiten generar diferentes tipos de gráficos.

Para cada muestra de los *datasets* utilizados, se extrajeron las siguientes características, la mayor parte mencionada en la sección 3.2: serie de tiempo de la

amplitud, espectrograma logarítmico, espectrograma de Mel; 13, 20 y 40 MFCCs, extraídos desde la serie de tiempo de la amplitud y 13, 20 y 40 MFCCs, extraídos desde el espectrograma de Mel.

Esto fue realizado con el fin de examinar visualmente cuál de todas ellas aporta mayor información para poder clasificar correctamente el estado de salud de la colmena.

La cantidad de coeficientes extraídos fueron determinados por las siguientes razones:

1. En general, en varios sitios webs se explica que 13 coeficientes cepstrales en la frecuencia de Mel son suficientes para caracterizar de buena manera la voz humana. Quizás también sean suficientes para caracterizar de buena forma el zumbido emitido por una colmena de abejas.
2. (Nolasco et al., 2018), entre los experimentos realizados se utilizaron 20 coeficientes, con lo cual obtuvieron buenos resultados. Por este motivo podría ser que trabajar con 20 en lugar de 13, fuera una mejor opción al momento de trabajar con audios de abejas.
3. Los puntos 1 y 2, indican que 13 y 20 coeficientes podrían dar buenos resultados, como se explicará en la sección 3.2.6 se tomará una red neuronal convolucional de la web y se usará para procesar MFCCs, dicha red trabaja con 40 MFCCs, por lo que sería interesante ver si al tomar una mayor cantidad de coeficientes cepstrales, la clasificación realizada por el modelo de reconocimiento de patrones de audio mejora o empeora en comparación con los resultados que podrían ofrecer 1 y 2.

3.2.5 Selección de características

Es de esperar que no todas las características que se extraigan de los audios generen los mismos resultados, algunas podrán ser “mejores” o “peores”. Un buen punto de referencia viene siendo el oído humano, ya que un apicultor con un oído bien entrenado es capaz de identificar si una colmena de abejas tiene o no presencia de abeja reina con tan solo escuchar los zumbidos que esta emite. Es por ello que seleccionar las características que se asemejan a como el oído humano trabaja, es una buena idea para comenzar a generar un modelo de clasificación basado en reconocimiento de audio.

National Institute on Deafness and Other Communication Disorders (2022) a groso modo, cuando las ondas de sonido viajan por el oído, este vibra. Cuando estas vibraciones llegan a la cóclea, las células ciliadas que están cerca de la parte ancha de la cóclea detectan los sonidos de alta frecuencia, en cambio las que se ubican cerca del medio, detectan los sonidos de baja frecuencia. Por lo tanto sería conveniente trabajar desde el punto de vista de las frecuencias (cabe mencionar que tiene más sentido para este trabajo, ya que se quiere tratar de dotar al modelo con características similares a las del oído humano), por este motivo se descarta la serie de tiempo de la amplitud, de esta manera

los posibles candidatos se reducen a los espectrogramas y MFCCs. Si bien, los candidatos trabajan en el ámbito de la frecuencia, también se debe considerar que el oído humano no las percibe en una escala lineal, sino que son percibidas en una escala más bien logarítmica, como por ejemplo la escala de Mel. Por ese motivo se puede esperar que el espectrograma logarítmico (cuyas frecuencias están representadas en escala logarítmica, pero su energía no ha sido transformada a una escala similar a la percibida por el oído humano) no tenga tan buenos resultados como el uso del espectrograma de Mel y MFCCs, de esta manera las características más adecuadas para caracterizar los audios de las abejas serían el espectrograma de Mel y los MFCCs, en el experimento se incluye también el espectrograma logarítmico, para verificar la suposición realizada.

3.2.6 Enfoque utilizado para generar el modelo de reconocimiento de patrones de audio

Para este trabajo, la idea inicial era hacer una red neuronal convolucional desde cero, pero debido a algunas complicaciones presentadas al momento de entrenar la red, se decidió cambiar de enfoque. El nuevo enfoque consistía en tomar desde la web una arquitectura de red neuronal convolucional que ya haya sido creada para clasificar espectrogramas y entrenarla con algún conjunto grande de datos (de preferencia con los que el desarrollador haya probado la red) para realizar aprendizaje por transferencia, de esta manera dicha red habría sido entrenada con una cantidad de muestras mayores a las que se habían obtenido en 3.2.1, pero un problema que surgió es que los modelos encontrados trabajaban clasificando imágenes de espectrogramas lo cual no se ajustaba a las matrices que contenían la distribución de la densidad de la energía que se habían extraído usando *librosa*. Por esta razón se modificó el enfoque anterior y se buscó una red convolucional que procesara MFCCs en forma de matrices de datos (como los obtenidos por *librosa.feature.mfcc*) de esta manera también podría utilizarse la red para procesar espectrogramas. Afortunadamente fue posible encontrar un modelo que presentaba buenos resultados al momento de clasificar MFCCs, el código se puede encontrar en el <https://gist.github.com/mikesmales>, dicho modelo fue probado con parte del *dataset Urbansound8K*. Inicialmente se aplicaría aprendizaje por transferencia a la red, pero luego de entrenarla y probarla con las muestras recopiladas en 3.2.1 y obtener resultados bastante buenos al predecir el conjunto de prueba, eso no fue necesario.

3.2.7 Arquitectura de la red

Para procesar las características descritas en 3.2.5, se utilizaron dos modelos de redes neuronales convolucionales, la primera es la red mencionada en 3.2.6 y la segunda es una adaptación de ella, a continuación se presentan los códigos de ambas:

```

#utilizado para procesar las características 20 y 40 mfcc, no se puede con 13
#ya que última capa no lo permite
def crear_modelo_20mfcc(n_filas, n_columnas, n_canal, n_etiquetas):
    return Sequential([
        Conv2D(filters=16, kernel_size=2,
               input_shape=(n_filas, n_columnas, n_canal),
               padding='same', activation='relu'),
        MaxPooling2D(pool_size=2),
        Dropout(0.2),
        Conv2D(filters=32, kernel_size=2, padding='same',
               activation='relu'),
        MaxPooling2D(pool_size=2),
        Dropout(0.2),
        Conv2D(filters=64, kernel_size=2, padding='same',
               activation='relu'),
        MaxPooling2D(pool_size=2),
        Dropout(0.2),
        Conv2D(filters=128, kernel_size=2, padding='same',
               activation='relu'),
        MaxPooling2D(pool_size=2),
        Dropout(0.2),
        GlobalAveragePooling2D(),
        Dense(n_etiquetas, activation='softmax')
    ])

```

Figura 27: Modelo para procesar 20 y 40 MFCCs. Fuente: Elaboración propia.

```

##utilizado para procesar las características 13, 20 y 40 mfcc.
def crear_modelo_13mfcc(n_filas, n_columnas, n_canal, n_etiquetas):
    return Sequential([
        Conv2D(filters=16, kernel_size=2,
               input_shape=(n_filas, n_columnas, n_canal),
               padding='same', activation='relu'),
        MaxPooling2D(pool_size=2),
        Dropout(0.2),
        Conv2D(filters=32, kernel_size=2, padding='same',
               activation='relu'),
        MaxPooling2D(pool_size=2),
        Dropout(0.2),
        Conv2D(filters=64, kernel_size=2, padding='same',
               activation='relu'),
        MaxPooling2D(pool_size=2),
        Dropout(0.2),
        GlobalAveragePooling2D(),
        Dense(n_etiquetas, activation='softmax')
    ])

```

Figura 28: Modelo para procesar 13, 20 y 40 MFCCs. Fuente: Elaboración propia.

Si bien la primera arquitectura fue diseñada para procesar 40 MFCCs, también fue posible entrenarla con 20 coeficientes cepstrales, pero no con 13, debido a las últimas capas de *Conv2D*, *MaxPooling2D* y *Dropout*. Es por ello que para entrenar la red con los 13 MFCCs fue necesario crear un nuevo modelo sin esas últimas capas. Ambas redes también

fueron utilizadas para procesar los espectrogramas logarítmicos y de Mel.

3.2.8 Hiperparámetros

Ambos modelos comparten la mayoría de los hiperparámetros, la única diferencia se da dentro de la arquitectura de cada red. Los hiperparámetros compartidos por ambas son:

1. Función de activación: *Relu* para las capas ocultas y *softmax* para la capa de salida.
2. Función de pérdida: *Categorical cross entropy*.
3. Métricas: *Accuracy*.
4. Optimizador: *Adam*.
5. Número de épocas: 72.
6. Tamaño del lote: 32.
7. *Dropout rate*: 0.2.

Los hiperparámetros que no son compartidos por ambos modelos.

1. Neuronas por capa primer modelo: 80 (1° capa), 2080 (2° capa), 8256 (3° capa), 32896 (4° capa) y 258 (capa de salida).

2. Neuronas por capa segundo modelo: 80 (1° capa), 2080 (2° capa), 8256 (3° capa) y 130 (capa de salida).

3.2.9 Entrenamiento

Para realizar el entrenamiento de las redes neuronales, se dividió el *dataset* en 80% para el conjunto de entrenamiento y 20% para pruebas. Tal como se mencionó en 3.2.8 se entrenaron ambas redes neuronales usando 72 épocas, los datos fueron entregados en grupos de 32 elementos y se ha mantenido la métrica *accuracy* (usada por el autor del modelo) para medir los aciertos de la clasificación, también cabe recordar que se ocupó la GPU proporcionada por *google colaboratory*. Una vez terminado el entrenamiento de las CNNs, se procedió a generar las matrices de confusión de cada uno.

3.3 Arquitectura del software

Una vez entrenada y probada la CNN, se procedió a construir los componentes del sistema que permitirían alojarla e interactuar con ella. A continuación se muestra el diagrama de casos de uso, el cual muestra cómo el usuario va a interactuar con el sistema para ver si la colmena tiene o no presencia de abeja reina.

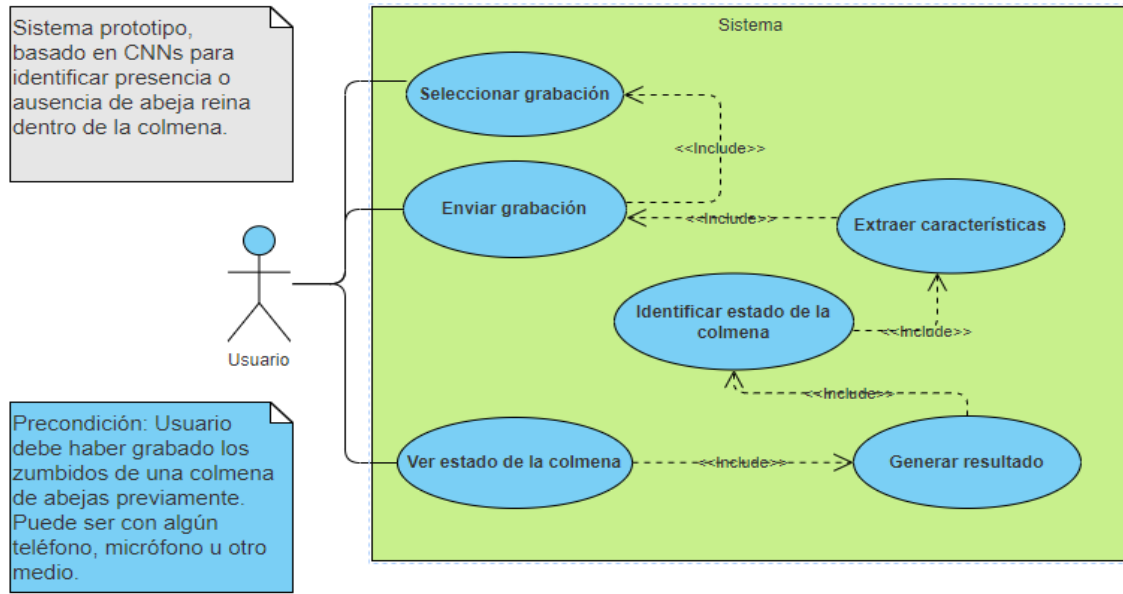


Figura 29: Diagrama general del sistema. Fuente: Elaboración propia.

El sistema está conformado por una *API REST* que contiene la CNN y un servidor de vistas que se encarga de la parte asociada a la interfaz de usuario. Se utilizó *docker* para crear contenedores que ejecutan el *backend* (*API REST*) y *frontend* (servidor de vistas) a partir de archivos *Dockerfile* multietapas (para el *backend* y *frontend*), que contienen instrucciones para poder trabajar en entornos de desarrollo y producción (por si en algún momento se tuviera que desplegar la aplicación en modo producción con *docker*), cabe destacar que los contenedores *docker* fueron utilizados solo en el entorno de desarrollo. También se usó *docker-compose* para poder automatizar el despliegue del sistema en el entorno de desarrollo.

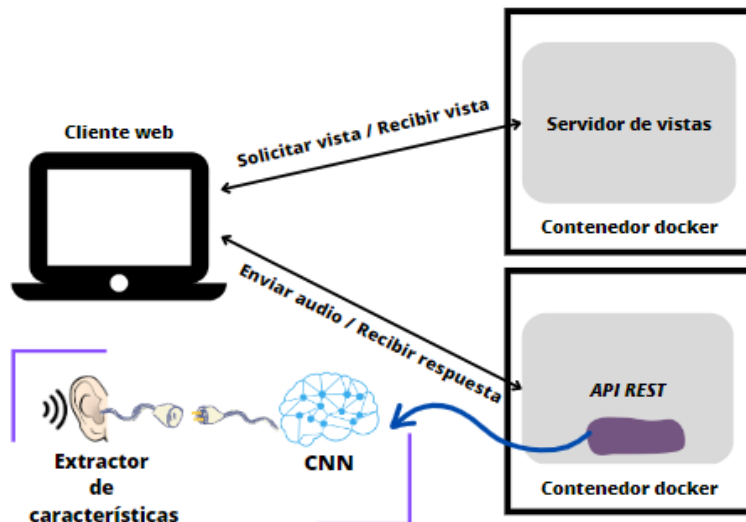


Figura 30: Arquitectura del sistema. Fuente: Elaboración propia.

Al momento de llevar el sistema a la nube (entorno de producción) los contenedores docker mencionados anteriormente no se pudieron utilizar, ya que el uso de docker para *alwaysdata* (este *hosting* se usó para hospedar el servidor de vistas en la nube) está asociado a planes de pago y para el caso de *PythonAnywhere* (este *hosting* se usó para hospedar la *API REST* en la nube) no está disponible, en su lugar se dejó que esos servicios se encargaran de administrar los componentes del sistema con sus políticas.

A nivel de repositorios, se usó *gitlab* para respaldar el código del sistema en la nube, el *frontend* y *backend* están divididos en dos repositorios independientes y existe un tercer repositorio pensado principalmente para facilitar el despliegue de la aplicación en modo desarrollo, en el cual se almacenan los archivos *docker-compose* (encargados de automatizar el despliegue de aplicaciones con múltiples contenedores *docker*) y se usa *git-submodule* para incluir los dos repositorios mencionados anteriormente dentro de este último. A continuación se detallarán algunas de las herramientas utilizadas y se dará más detalles sobre el *backend* y *frontend*.

3.3.1 Herramientas utilizadas

- *Django*: Framework web para *Python* que utiliza el patrón de diseño MVC. Dispone de un servidor de desarrollo con recarga en caliente, que permite recargarlo de manera automática a medida que se codifica código. Según (Django Software Foundation, n.d.) algunas de las ventajas que ofrece este *framework* son: fácil de usar, seguro, escalable.
- *Vue*: Según (You, n.d.) es un *framework* web para *javascript* basado en componentes que permite desarrollar interfaces de usuario. Permite hacer desde aplicaciones de una sola página hasta aplicaciones web progresivas, al igual que *Django*, también cuenta con un servidor de desarrollo integrado con recarga en caliente.
- *Bootstrap*: (Team Bootstrap, n.d.) framework para desarrollo frontend. Algunas de las ventajas que ofrece son sistema de grilla, componentes prediseñados, elementos responsive, complementos de javascript entre otros.
- *Git-submodule*: (*Git - Submodules*, n.d.) herramienta de *git* que permite entre algunas cosas usar un repositorio como una subcarpeta de otro repositorio. Para el presente trabajo se ha usado para colocar los proyectos de *frontend* y *backend* como subcarpetas de otro proyecto, el cuál fue pensado para facilitar el proceso de desarrollo de la aplicación.

3.3.2 Backend

El *backend* está compuesto por una *API REST*, la cuál se encarga de procesar las solicitudes que contienen los audios enviados desde el cliente y retorna una respuesta que

consiste en un objeto formado por el nombre del audio, la anomalía detectada (presencia o ausencia de abeja reina) y los porcentajes de probabilidad de que dicha anomalía esté presente o no dentro de la colmena. Para llevar esto a cabo, la *API* usa una CNN que procesa 13 coeficientes cepstrales en la frecuencia de mel extraídos desde la serie de tiempo de la amplitud del audio, la cuál es cargada dentro de la misma *API*. Su implementación se realizó mediante el *framework Django*, las razones por la cual se ha usado este *framework* es su fácil uso y el hecho de que está basado en *Python*, estos dos factores fueron importantes, debido a que toda la parte experimental consistente en extraer las características y entrenar la CNN fueron realizadas en *Python*, se ha utilizado la versión 3.10 para trabajar con librerías como *librosa* y *keras*. Para poder procesar los audios, la *API* hace uso de las librerías *libsndfile1-dev* (permite a *librosa* trabajar con archivos *.wav*) y *ffmpeg* (permite a *librosa* procesar otros formatos como *.mp3*, mediante la interfaz de comandos), en modo desarrollo la *API* se ejecuta en contenedores *docker* y se sirve usando el servidor de desarrollo proporcionado por *Django*, para el entorno de producción, la *API* se sirve usando *gunicorn* desde *PythonAnywhere*.

3.3.3 Frontend

Para el *frontend* se desarrolló un servidor de vistas implementado en *Vue*, este se encarga de servir al cliente web las vistas con las que el usuario de la aplicación se encargará de interactuar. Se ha usado un diseño gratuito de la página *startbootstrap* como base, el cual es *responsive* (lo que permite que las vistas se adapten al tamaño del dispositivo) y utiliza el *framework bootstrap*, dicho diseño ha recibido algunas modificaciones menores para ser adaptado a los propósitos del presente trabajo, se han incluido librerías como *axios* (encargada de realizar el trabajo de red entre el cliente y la *API*) y *sweetalert2* (para desplegar mensajes informativos y los resultados de procesar el audio que contiene el zumbido de la colmena mediante ventanas emergentes). Al igual que en el *backend*, se utilizó *docker* para ejecutar el servidor de vistas dentro de contenedores para el entorno de desarrollo y es servido por el servidor de desarrollo proporcionado por *Vue*, en la etapa de producción se delega la labor de servir el *frontend* a *alwaysdata*.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

Esta sección pretende mostrar y analizar los resultados obtenidos tras probar el modelo de clasificación y la arquitectura del software en cuestión.

4.1 Análisis de las características extraídas

Esta sección pretende comparar las características extraídas de los *datasets* utilizados (enfocado en el *dataset* con audios de 2 segundos), con el fin de mostrar qué tan diferenciadoras son cada una de ellas al momento de representar los estados de la colmena de abejas.

4.1.1 Espectrograma logarítmico vs Espectrograma de Mel

Según los resultados obtenidos, luego de extraer estas características de las muestras, se pudo observar que el espectrograma de Mel representa de una manera más diferenciadora los estados de salud de la colmena en comparación con el espectrograma logarítmico, esto se puede apreciar en los siguientes resultados.

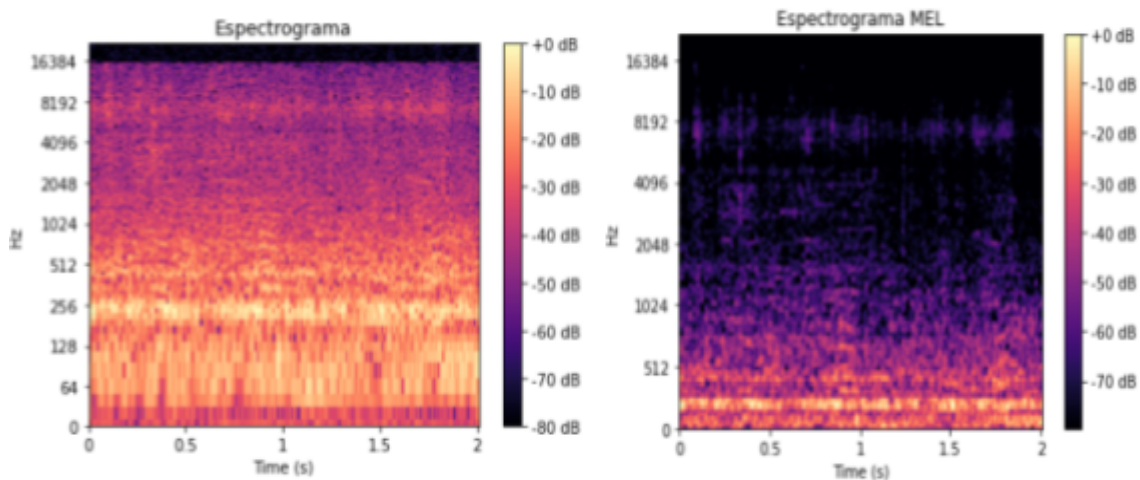


Figura 31: Espectrograma logarítmico vs Espectrograma de Mel, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena con abeja reina. Aparición de frecuencias alrededor 4096 y 8192 [Hz]. Fuente: Elaboración propia.

Como se puede apreciar, visualmente se ve mucho más complejo el análisis del espectrograma logarítmico en comparación con el espectrograma de Mel. En general, al ver los resultados de este último se puede obtener como patrón que al no estar presentes o en poca medida las frecuencias alrededor de los rangos 4096 y 8192 [Hz], la colmena podría estar en el estado “con abeja reina” (esto pasa en gran parte de las muestras) en cambio una fuerte presencia de esas frecuencias podrían clasificar a la colmena como “sin abeja reina”, tal como se muestra a continuación.

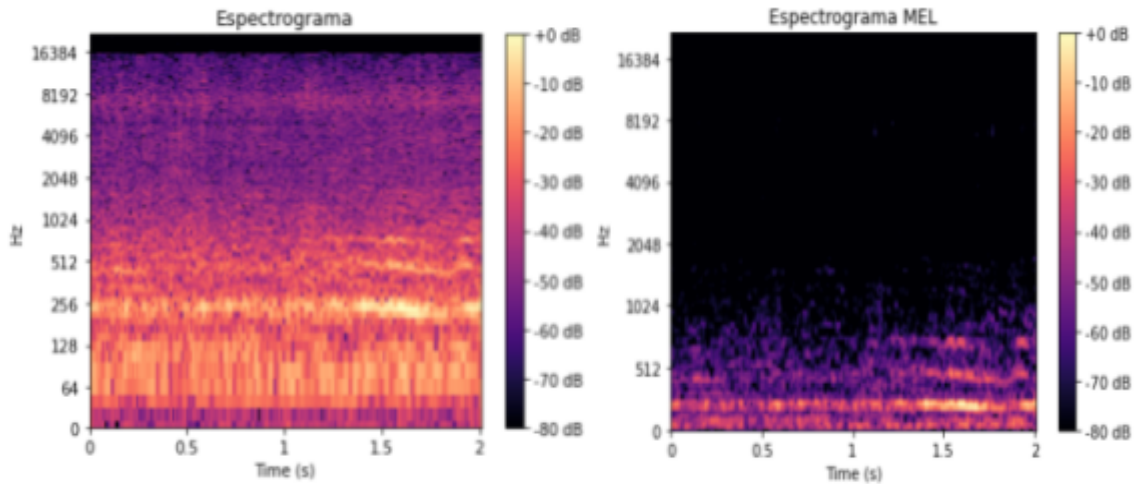


Figura 32: Espectrograma logarítmico vs Espectrograma de Mel, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena con abeja reina. Sin frecuencias alrededor de 4096 y 8192 [Hz]. Fuente: Elaboración propia.

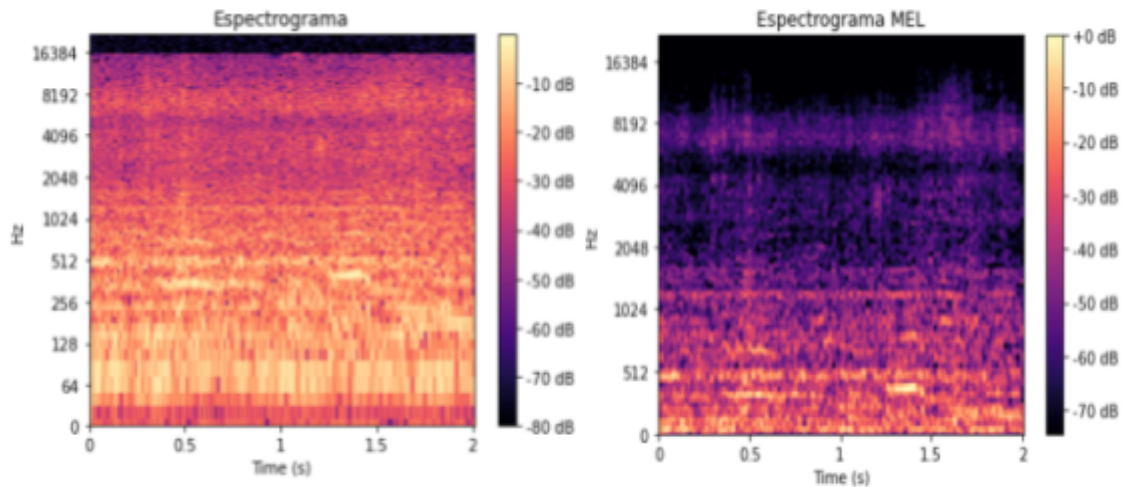


Figura 33: Espectrograma logarítmico vs Espectrograma de Mel, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena sin abeja reina. Fuerte presencia de frecuencias alrededor de 4096 y 8192 [Hz]. Fuente: Elaboración propia.

Es razonable pensar que el espectrograma de Mel puede ser más fácil de analizar que el espectrograma logarítmico, ya que el primero fue transformado a una escala más acorde a como el oído humano percibe los sonidos. Esto puede mostrar que es una buena idea usar características basadas en la percepción del oído humano para generar modelos de reconocimiento de audio que generen “buenas” predicciones.

4.1.2 Comparación entre Espectrograma de Mel y MFCCs

Si bien en la sección 4.1.1 se pudo ver que el espectrograma de Mel es mucho más sencillo de analizar visualmente que el espectrograma logarítmico y al parecer revela que

fuertes presencias de las frecuencias que se encuentran alrededor de los rangos 4096 y 8192 [hz] podrían ser utilizadas para determinar que la colmena no tiene abeja reina, también existe otra herramienta que es bastante usada en el campo de procesamiento de audio que permite caracterizarlo mediante la extracción de ciertos coeficientes, esta herramienta se llama MFCC. Al extraerlos de las muestras y hacer la comparación con los espectrogramas de Mel se puede ver que las primeras son mucho más diferenciadoras que las segundas.

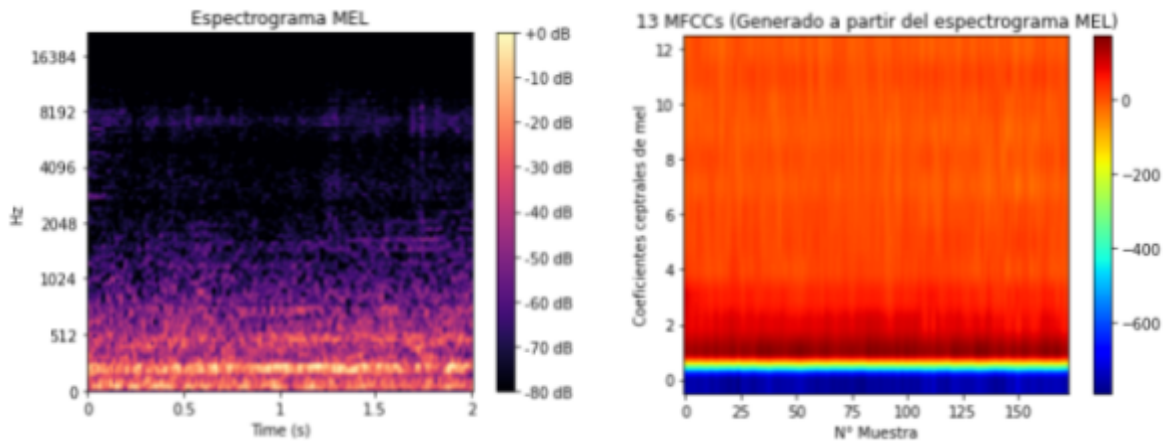


Figura 34: Espectrograma de Mel vs 13 MFCCs, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena con abeja reina. Fuente: Elaboración propia.

En esta imagen, se puede ver que al extraer los 13 MFCCs del espectrograma de Mel, en general se tiende a ver una imagen anaranjada o roja, donde no se remarca con claridad ningún coeficiente. Esto se da en los zumbidos que provienen de colmenas que tienen abejas reinas. En el caso de aquellas colmenas que no tienen abeja reina, en general se puede ver una línea amarilla mejor definida sobre el 4° coeficiente, además se vislumbran líneas menos marcadas sobre los coeficientes 7° y 9°, un ejemplo de ello se puede ver en la siguiente imagen.

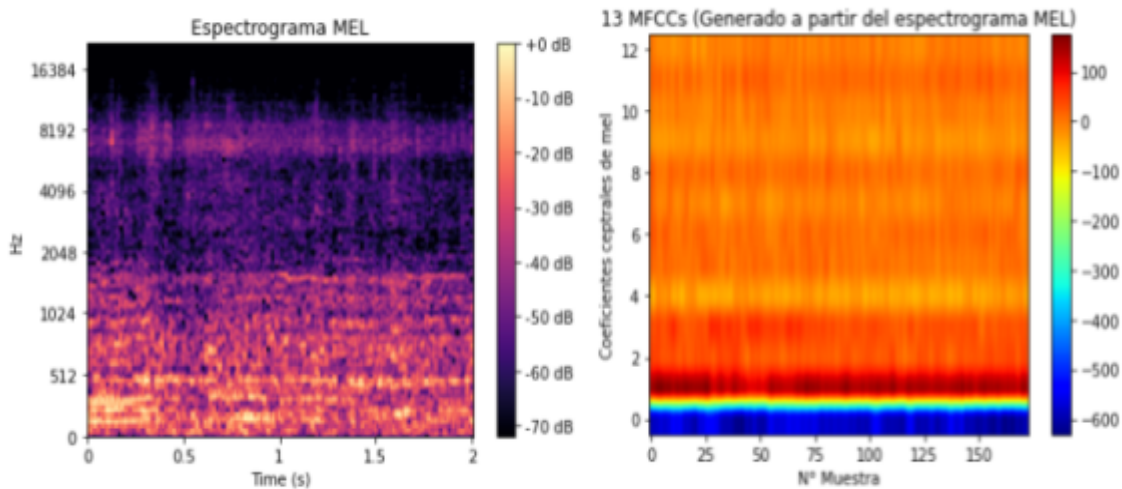


Figura 35: Espectrograma de Mel vs 13 MFCCs, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena sin abeja reina. Fuente: Elaboración propia.

El análisis realizado anteriormente, revela que en general los MFCCs pueden caracterizar de mejor manera si una colmena tiene o no una abeja reina. En algunos de los casos donde es más difícil evidenciar la existencia de abeja reina mediante el espectrograma de Mel, resulta ser más sencillo al utilizar los MFCCs extraídos de dicho espectrograma. Además para el caso de MFCCs, basta con conocer una pequeña cantidad de coeficientes (más o menos 13 coeficientes son usados frecuentemente para caracterizar el audio), en comparación con la gran cantidad de frecuencias presentadas por el espectrograma de Mel.

4.1.3 Comparación entre MFCCs extraídos de la serie de tiempo de la amplitud y del espectrograma de Mel

Al comparar los MFCCs extraídos desde la serie de tiempo de la amplitud con los del espectrograma de Mel, resulta más complejo (al menos visualmente) diferenciar el estado de salud de la colmena al comparar los primeros con los segundos, esto se puede observar en los siguientes resultados.

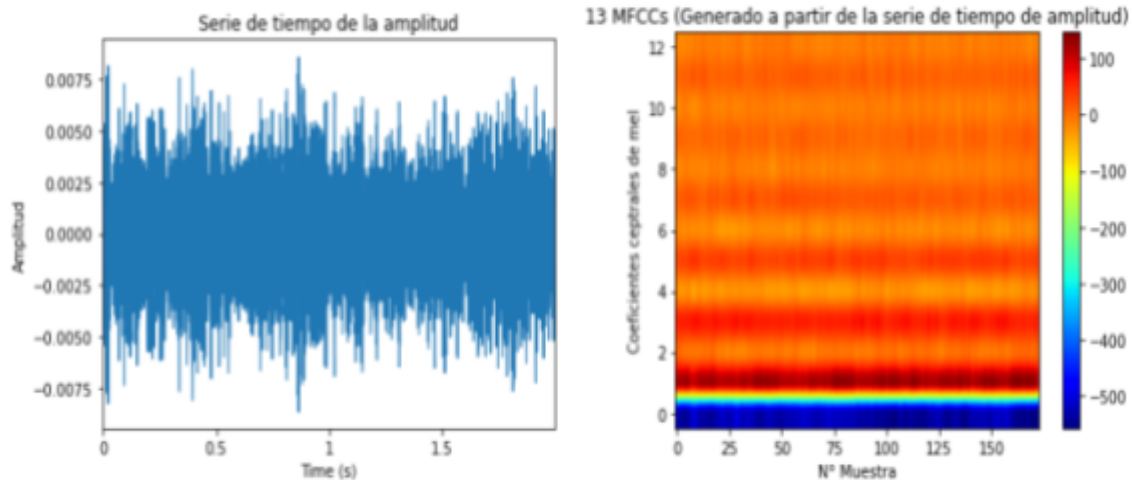


Figura 36: Serie de tiempo de la amplitud y 13 MFCCs, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena con abeja reina. Fuente: Elaboración propia.

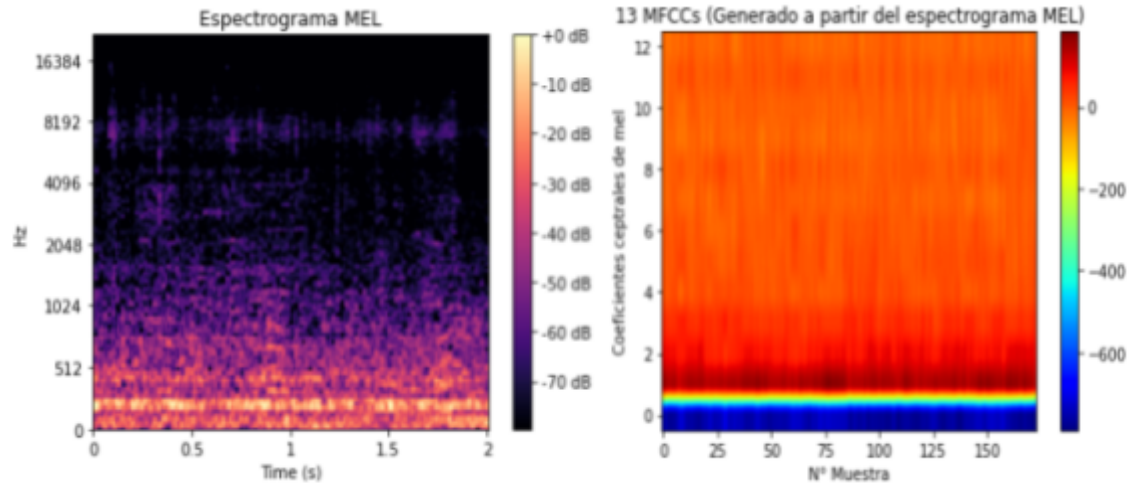


Figura 37: Espectrograma de Mel (obtenido desde la serie de tiempo de la imagen anterior) y 13 MFCCs, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena con abeja reina. Fuente: Elaboración propia.

Como se puede apreciar en las dos imágenes anteriores, en la primera aparecen marcados varios coeficientes no así en la segunda, ya que esta última no tiene coeficientes bien marcados. En general, se pueden ver varios coeficientes cepstrales al extraerlos desde la serie de tiempo de la amplitud, algo que no ocurre al usar el espectrograma de Mel. Además, este aumento en la cantidad de coeficientes, hace que sea más complejo el análisis de la clasificación (al menos para el ojo humano), aunque puede ser que una CNN pueda encontrar un patrón significativo para aprender a diferenciarlos. A continuación se presentan los resultados que fueron obtenidos de grabaciones de colmenas sin abeja reina, para que el lector pueda observar las diferencias entre los MFCCs obtenidos desde el espectrograma de Mel y los de la amplitud de mejor manera.

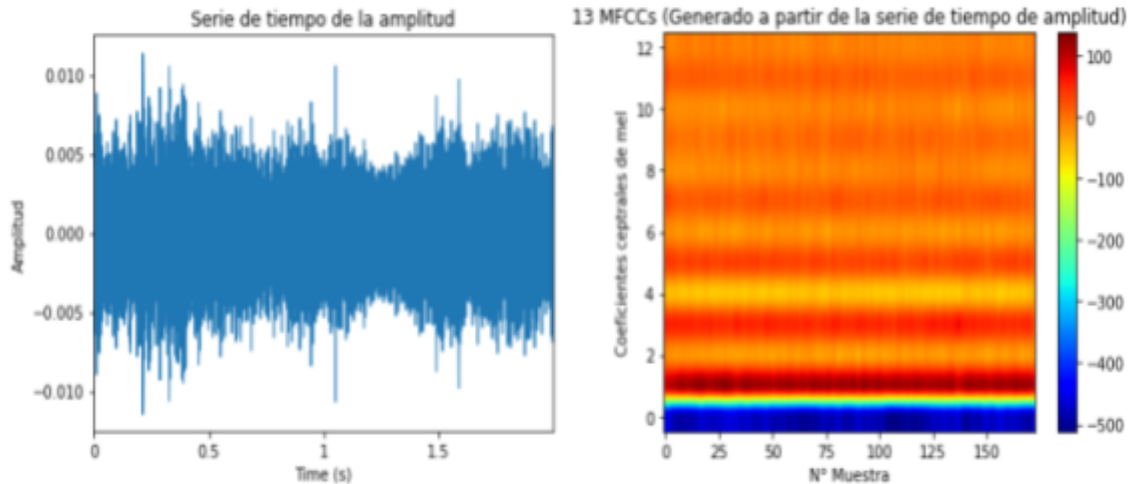


Figura 38: Serie de tiempo de la amplitud y 13 MFCCs, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena sin abeja reina. Fuente: Elaboración propia.

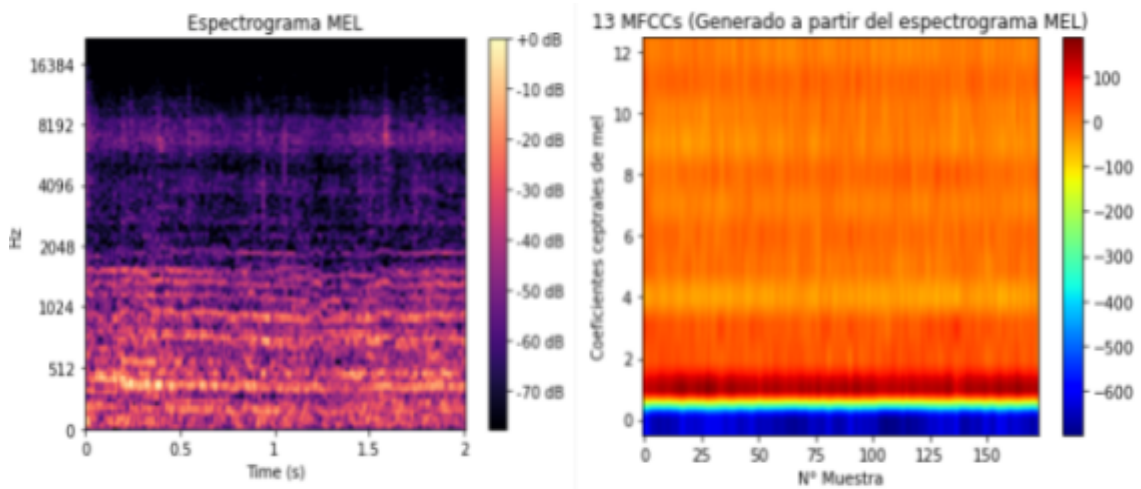


Figura 39: Espectrograma de Mel (obtenido desde la serie de tiempo de la imagen anterior) y 13 MFCCs, usando muestra de audio de 2 segundos, correspondiente a zumbidos de una colmena sin abeja reina. Fuente: Elaboración propia.

Al parecer, la forma en que librosa procesa la serie de tiempo de la amplitud y el espectrograma de Mel para extraer los MFCCs, producen las apariciones de nuevos coeficientes. Pero en ambos casos, el 4° coeficiente resalta sobre los demás en los casos donde la colmena no tiene abeja reina.

4.2 Análisis del desempeño del clasificador

A continuación se presentarán los resultados asociados al proceso de entrenamiento del modelo de reconocimiento de patrones de audio, estos resultados corresponden a las matrices de confusión generadas por cada modelo.

4.2.1 Clasificación usando espectrograma logarítmico

Para este caso, solo se pudo utilizar el *dataset* con audios de zumbidos de colmenas de duración 1 [s], esto debido a limitaciones con la memoria RAM. Fueron utilizadas ambas CNNs, la primera demoró 4:23 [min] en ser entrenada y la segunda 3:47 [min]. A continuación se presentan los resultados de ambos modelos.



Figura 40: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20). Fuente: Elaboración propia.



Figura 41: Matriz de confusión generada por el segundo modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia.

En general, se puede apreciar que ambos modelos reconocen bastante bien los zumbidos asociados a colmenas con y sin abeja reina. El primer modelo presenta una *precision* de 94.68% y *recall* de 83.96 % con respecto al estado “sin abeja reina”, para el estado “con abeja reina” las métricas son *precision* 92.92 % y *recall* 97.81%. En cambio el segundo modelo presenta una *precision* de 75% y *recall* de 82.08% para el estado “sin abeja reina”, cuando la colmena presenta el estado “con abeja reina”, las métricas son *precision* 91.28 % y *recall* 87.28 %. Para este experimento, los mejores resultados fueron

obtenidos por el primer modelo descrito en 3.2.7.

4.2.2 Clasificación usando espectrograma de Mel

En este caso, se pudieron utilizar ambos modelos de clasificación con los *datasets* con zumbidos de colmenas con duración de 1, 2 y 3 segundos. Los tiempos de entrenamiento de los modelos están en los rangos de 38 [s] - 1:30 [min] aproximadamente, esto varía dependiendo del *dataset* utilizado y la arquitectura de la CNN. A continuación se presentan los resultados obtenidos por ambos modelos, utilizando los tres *datasets* mencionados anteriormente.

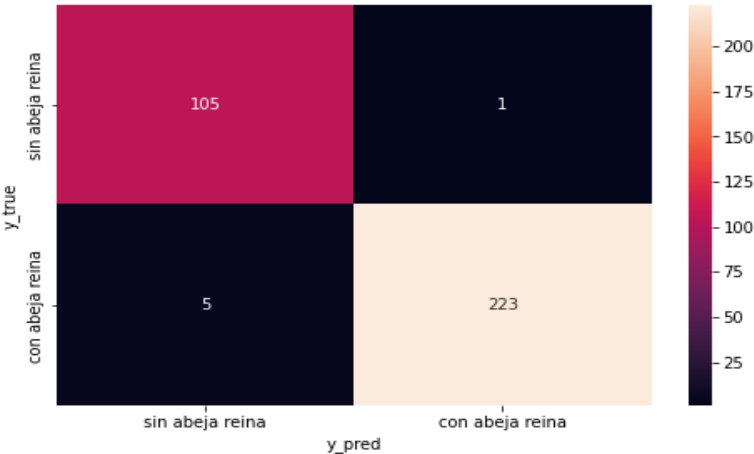


Figura 42: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, *dataset* 1 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20). Fuente: Elaboración propia.



Figura 43: Matriz de confusión generada por el segundo modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, *dataset* 1 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia.

Para este caso, usando el *dataset* con zumbidos de colmenas de 1 [s], se ha podido

obtener un mejor resultado en la clasificación del estado de salud de la colmena en comparación con el espectrograma logarítmico, cabe señalar que los pesos iniciales de la CNN afectan bastante los resultados del entrenamiento de las CNNs basadas en espectrogramas logarítmicos y espectrogramas de Mel, por lo que en ciertas ocasiones será necesario entrenarlas algunas veces hasta obtener los resultados deseados. Con respecto a las métricas, el primer modelo obtuvo una *precision* de 95.45% y *recall* de 99.06% para el estado “sin abeja reina”, para el segundo estado los resultados fueron *precision* 99.55% y *recall* 97.81%. En cambio el segundo modelo logró una *precision* de 82.68% y *recall* de 99.06% para el estado “sin abeja reina”, las métricas del segundo estado son *precision* de 99.52% y *recall* de 90.35%. En este experimento, el primer modelo obtuvo mejores resultados.

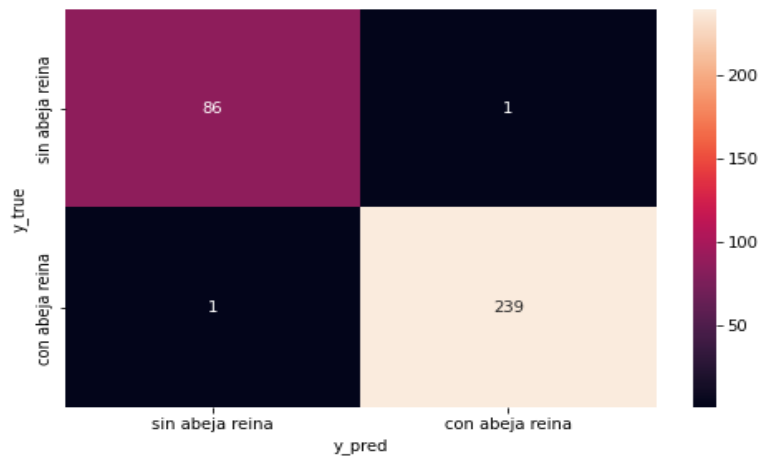


Figura 44: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, *dataset* 2 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20). Fuente: Elaboración propia.

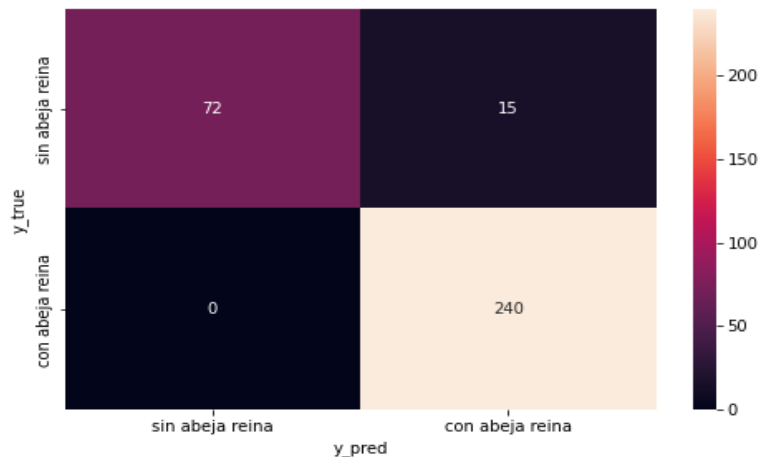


Figura 45: Matriz de confusión generada por el segundo modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, *dataset* 2 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia.

Con respecto al *dataset* con audios que tienen una duración de 2 [s], se puede ver que el primer modelo identifica mejor las colmenas sin abeja reina en lugar del segundo. Las métricas presentadas por el primer modelo son *precision* 98.85% y *recall* 98.85% para el estado “sin abeja reina” y para el estado “con abeja reina” se obtuvo *precision* 99.58% y *recall* de 99.58%. Con respecto al segundo modelo, las métricas obtenidas fueron *precision* 100% y *recall* 82.76% para el estado “sin abeja reina”, para el segundo estado se obtuvo *precision* 94.12% y *recall* de 100%. Para este caso, el primer modelo tuvo mejores resultados.



Figura 46: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, *dataset* 3 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20). Fuente: Elaboración propia.

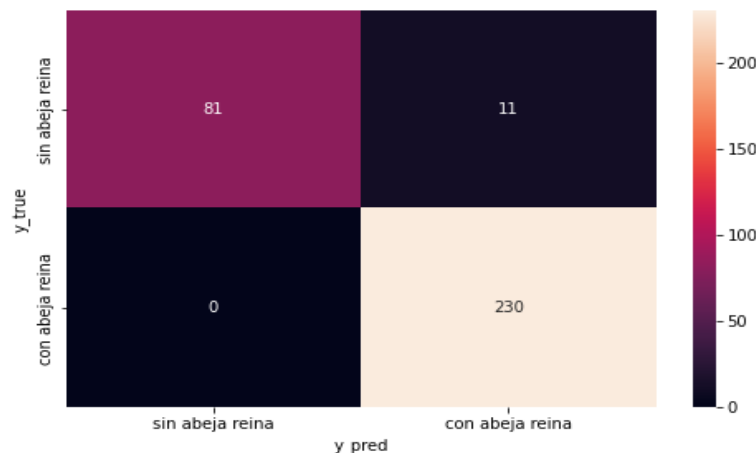


Figura 47: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, *dataset* 3 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13). Fuente: Elaboración propia.

Finalmente, para el caso del *dataset* con zumbidos cuya duración es de 3 [s] se puede ver que en general los modelos de redes neuronales están identificando bastante

los estados de salud de la colmena. Las métricas del primer modelo son *precision* 93.88% y *recall* 100% para el estado “sin abeja reina” y *precision* 100% y *recall* 97.39 para el segundo estado. Con respecto al segundo modelo, se obtuvo *precision* 100% y *recall* 88.04% para el estado “sin abeja reina” y para el segundo estado los resultados fueron *precision* 95.44% y *recall* 100%. Con respecto a la identificación del estado “sin abeja reina”, el primer modelo tiene mejores resultados.

4.2.3 Clasificación usando MFCCs extraídos desde la serie de tiempo de la amplitud

Ambos modelos, también fueron utilizados para clasificar el estado de salud de la colmena utilizando los MFCCs como caracterización del audio, debido a que el primer modelo tiene una capa adicional de *Conv2D* + *MaxPooling2D* + *Dropout* solo se pudieron utilizar 20 y 40 MFCCs, para el segundo modelo se utilizaron 13, 20 y 40 MFCCs. En general, estos modelos fueron entrenados en un rango de 18 - 41 [s], esto varía dependiendo del dataset utilizado, la cantidad de MFCCs utilizados y la arquitectura de la CNN. A continuación se mostrarán los resultados obtenidos.

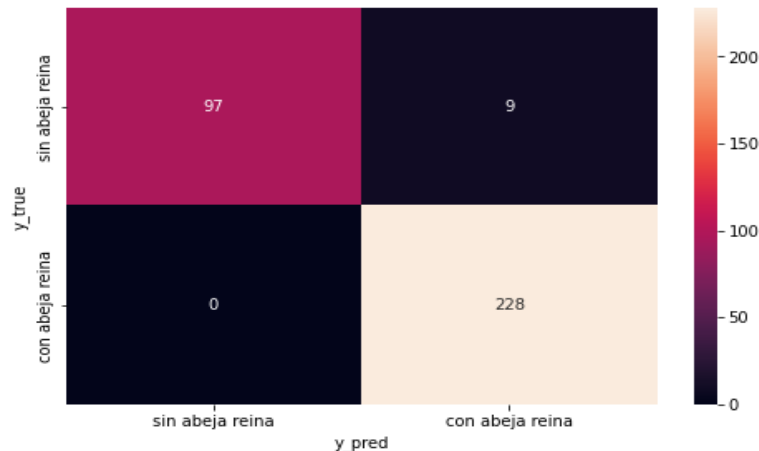


Figura 48: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 13 MFCCs, *dataset* 1 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).

Fuente: Elaboración propia.

Para el caso de los 13 MFCCs, se puede apreciar que los resultados de la clasificación al utilizar los datos de prueba son bastante buenos. Las métricas obtenidas son *precision* de 100% y *recall* de 91.51% para el caso de las colmenas que presentan el estado “sin abeja reina”, en el segundo estado los resultados fueron *precision* 96.20% y *recall* 100%.



Figura 49: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 1* [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.



Figura 50: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 1* [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20).
Fuente: Elaboración propia.

Para el caso de las CNNs que fueron entrenadas utilizando 20 MFCCs, se puede apreciar que los resultados fueron bastante similares, además fueron mejores que los obtenidos al usar 13 MFCCs como caracterización del zumbido de la colmena. Las métricas obtenidas por el primer modelo fueron *precision* 99.06% y *recall* 99.06% para el estado “sin abeja reina”, en cambio en el segundo estado se obtuvo *precision* 99.56% y *recall* 99.56%. Con respecto al segundo modelo, las métricas obtenidas fueron *precision* 100% y *recall* 99.06% para el estado “sin abeja reina”, los resultados del segundo estado fueron *precision* 99.56% y *recall* 100%. En este experimento, los mejores resultados fueron obtenidos usando el segundo modelo.

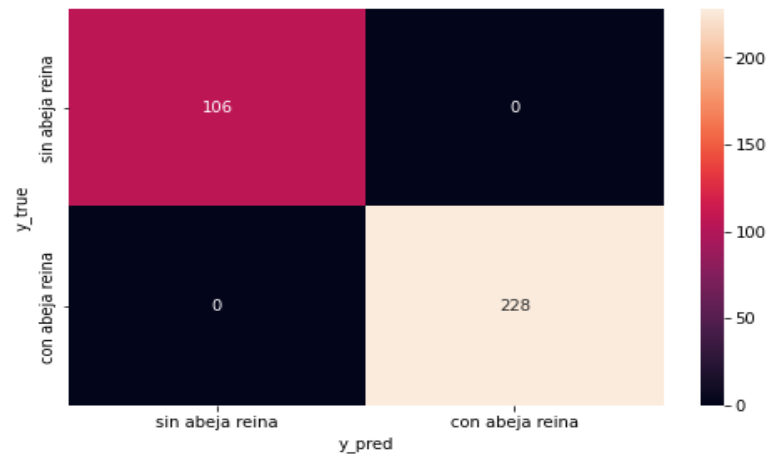


Figura 51: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset 1* [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.



Figura 52: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset 1* [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20).
Fuente: Elaboración propia.

Para el caso de las CNNs entrenadas usando 40 MFCCs, se puede ver que los resultados de la clasificación utilizando los datos de prueba son muy buenos. Ambos modelos obtuvieron una *precision* y *recall* de 100% para ambos estados de salud de la colmena.

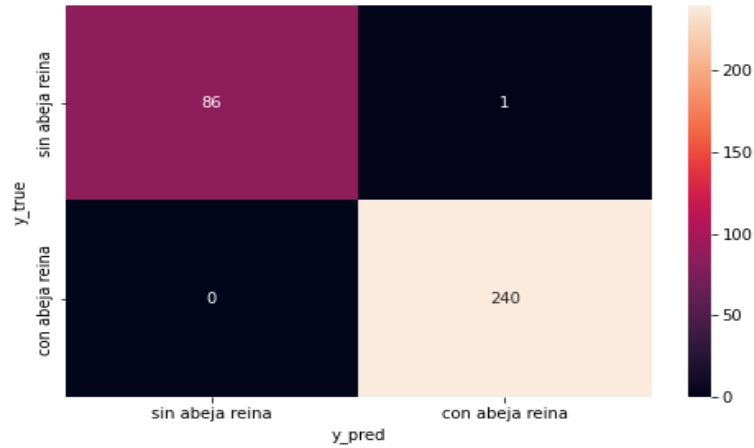


Figura 53: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 13 MFCCs, *dataset 2* [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.

Como se puede apreciar en este caso, al utilizar el primer modelo descrito en 3.2.7, usando 13 MFCCs extraídos desde zumbidos de colmenas de abejas con una mayor duración (2 [s] en lugar de 1 [s]), los resultados de la clasificación mejoraron. Las métricas obtenidas por el modelo empleado fueron *precision* 100% y *recall* 98.85% para el estado “sin abeja reina”, en cambio para el segundo estado “con abeja reina” se obtuvo *precision* 99.59% y *recall* 100%.

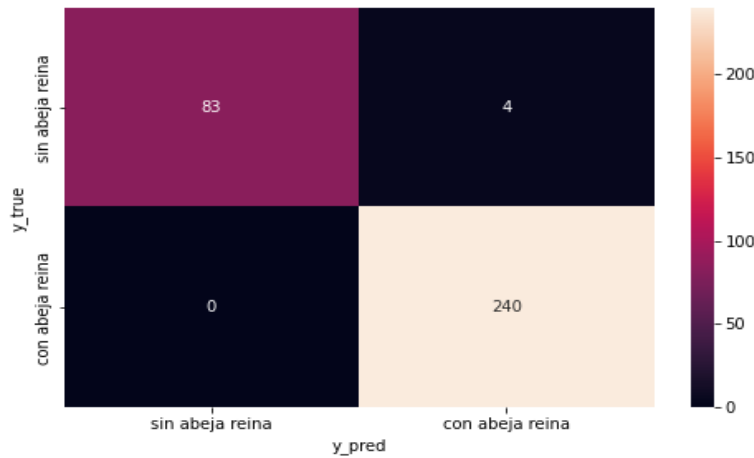


Figura 54: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 2* [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.

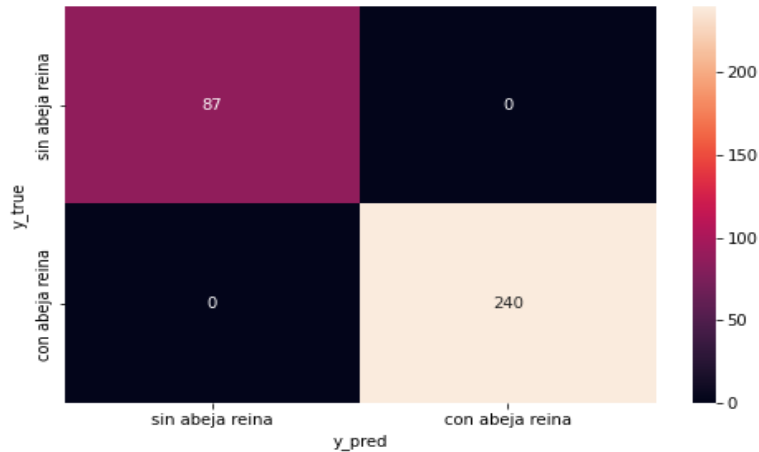


Figura 55: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 2* [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20).
Fuente: Elaboración propia.

Para el caso del entrenamiento de las CNNs usando 20 MFCCs extraídos a partir del *dataset* de 2 [s], se puede observar que ambos modelos pueden clasificar bastante bien el estado de las colmenas de abejas. Las métricas obtenidas por el primer modelo fueron *precision* 100% y *recall* 95.40% para el estado “sin abeja reina”, en el caso del segundo estado se obtuvo *precision* 98.36% y *recall* 100%. Los resultados del segundo modelo fueron *precision* y *recall* de 100% para ambos estados de salud. Para este experimento, los mejores resultados fueron obtenidos por el segundo modelo.

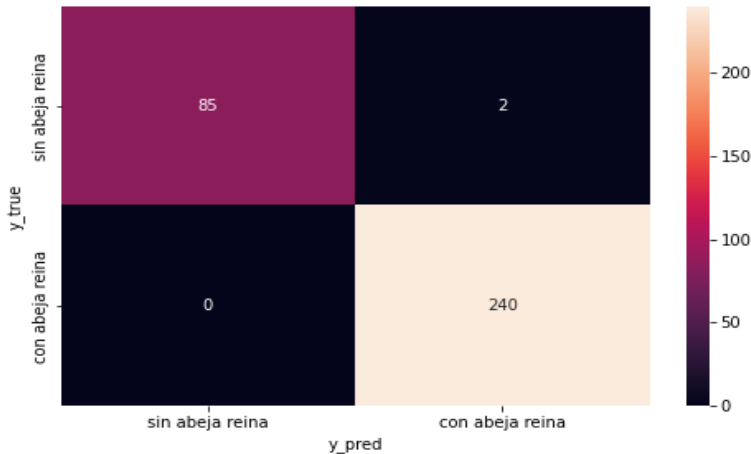


Figura 56: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset 2* [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.

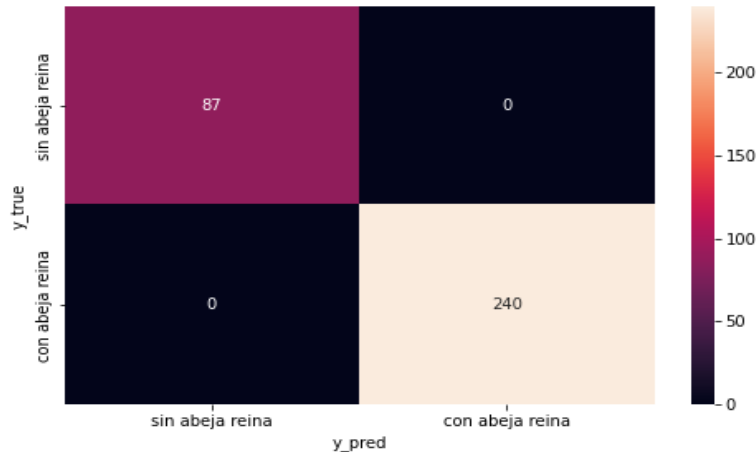


Figura 57: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset 2* [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20).
Fuente: Elaboración propia.

Al utilizar el *dataset* de 2 [s] para generar los 40 MFCCs, se puede ver que los resultados de la clasificación del estado de la colmena de abejas son bastante similares. El primer modelo obtuvo *precision* de 100% y *recall* de 97.70% con respecto al estado “sin abeja reina”, en cambio para el segundo estado se obtuvo *precision* 99.17% y *recall* 100%. El segundo modelo logró una *precision* y *recall* de 100% para ambos estados de la colmena. En este experimento los mejores resultados fueron obtenidos por el segundo modelo.

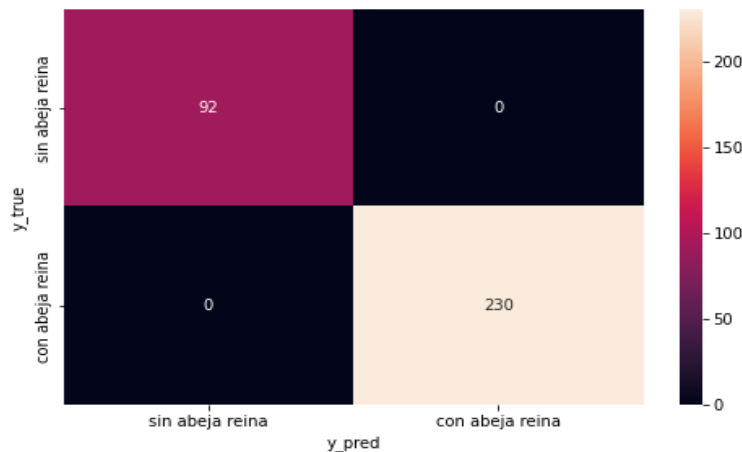


Figura 58: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 13 MFCCs, *dataset 3* [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.

Para este caso, al utilizar los 13 MFCCs extraídos a partir de los zumbidos de las colmenas de abejas del *dataset* de 3 [s], el modelo pudo clasificar bastante bien los estados de las colmenas de abejas a partir de las muestras de prueba. El modelo obtuvo

una *precision* de 100% y un *recall* de 100% para ambos estados de salud de la colmena.

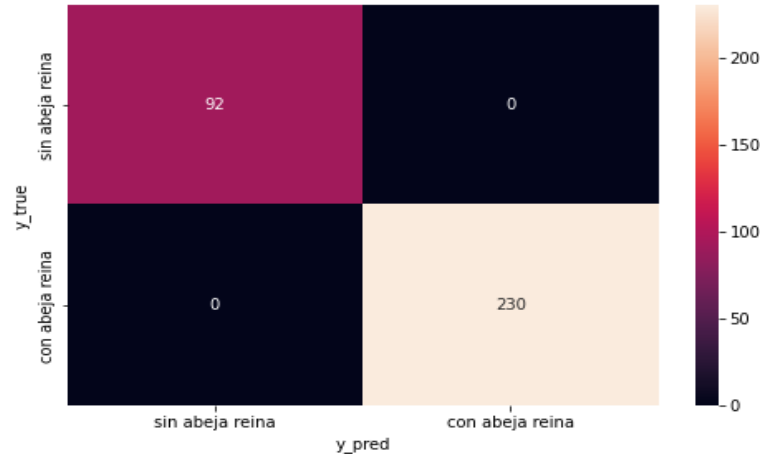


Figura 59: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset* 3 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.

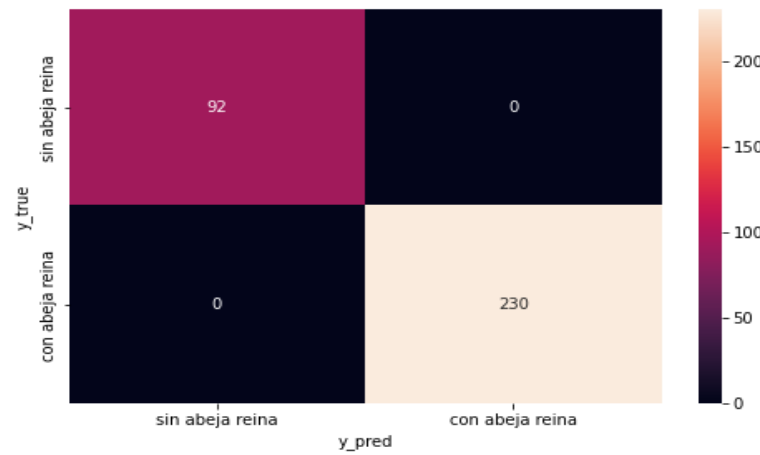


Figura 60: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset* 3 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20).
Fuente: Elaboración propia.

Para este caso, ambos modelos obtuvieron los mismos resultados con respecto a la clasificación del estado de salud de la colmena. Ambos modelos consiguieron una *precision* de 100% y *recall* de 100% para los dos estados de las colmenas. Es por ello que en este experimento, ambos modelos son igual de buenos clasificando el estado de la colmena.



Figura 61: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset* 3 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.

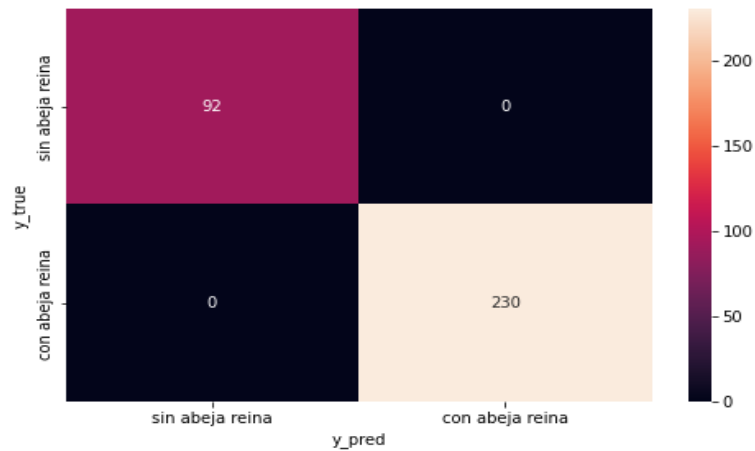


Figura 62: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset* 3 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20).
Fuente: Elaboración propia.

Para este caso también se puede apreciar que ambos modelos obtuvieron los mismos resultados al clasificar el estado de salud de las colmenas. Ambos modelos obtuvieron una *precision* y *recall* de 100% para los dos estados de la colmena. En este experimento, los dos modelos en cuestión son igual de buenos.

4.2.4 Clasificación usando MFCCs extraídos desde el espectrograma de Mel

Para este caso, se ha realizado el mismo procedimiento que el descrito en 4.2.3, obteniendo valores similares para el tiempo de entrenamiento de los modelos. A continuación se presentan los resultados obtenidos.



Figura 63: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 13 MFCCs, *dataset* 1 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).

Fuente: Elaboración propia.

Al extraer los 13 MFCCs desde los espectrogramas de Mel generados a partir de las muestras de zumbidos de las colmenas del *dataset* de 1 [s] para entrenar la CNN, se puede ver que los resultados de la clasificación son bastante buenos. Las métricas obtenidas por el modelo son *precision* 100% y *recall* 98.11% para el estado “sin abeja reina”, en el caso del estado “con abeja reina” se obtuvo *precision* 99.13% *recall* 100%.



Figura 64: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset* 1 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).

Fuente: Elaboración propia.



Figura 65: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 1* [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20).
Fuente: Elaboración propia.

Para este caso, ambos modelos presentaron buen desempeño al momento de clasificar el estado de salud de las colmenas. Las métricas obtenidas por el primero son *precision* 97.22% y *recall* 99.06% con respecto al estado “sin abeja reina”, en cambio para el segundo estado se obtuvo *precision* 99.56% y *recall* 98.68%. El segundo modelo obtuvo una *precision* de 99.05% y *recall* de 98.11% para el estado “sin abeja reina” y para el segundo estado los resultados fueron *precision* 99.13% y *recall* 99.56%. En este experimento la primera CNN es menos precisa que la segunda, pero puede detectar mejor las colmenas sin abeja reina en comparación con la segunda.



Figura 66: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset 1* [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.



Figura 67: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset 1* [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20).
Fuente: Elaboración propia.

En este experimento, el primer modelo detecta mejor las colmenas que no tienen abeja reina y el segundo detecta mejor aquellas colmenas que si las tiene. Las métricas del primero son *precision* 96.36% y *recall* 100% con respecto al estado “sin abeja reina”, en cambio para el segundo estado se obtuvo *precision* 100% y *recall* 98.25%. El segundo modelo presenta una *precision* de 100% y *recall* de 98.11% para el estado “sin abeja reina”, el segundo estado presenta *precision* 99.13% y *recall* 100%.

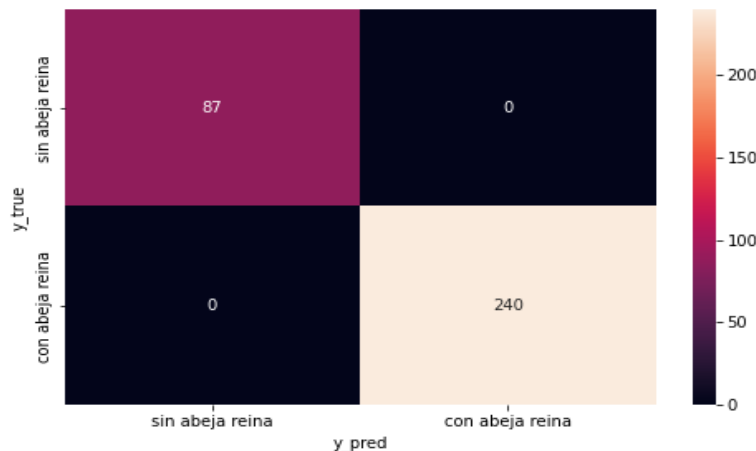


Figura 68: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 13 MFCCs, *dataset 2* [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.

Al utilizar 13 MFCCs extraídos desde el espectrograma de Mel generado a partir de zumbidos de colmenas con duración de 2 [s] como caracterización del audio, se puede ver que el modelo ha obtenido una *precision* de 100% y un *accuracy* de 100% para ambas clases al momento de clasificar los datos de prueba.

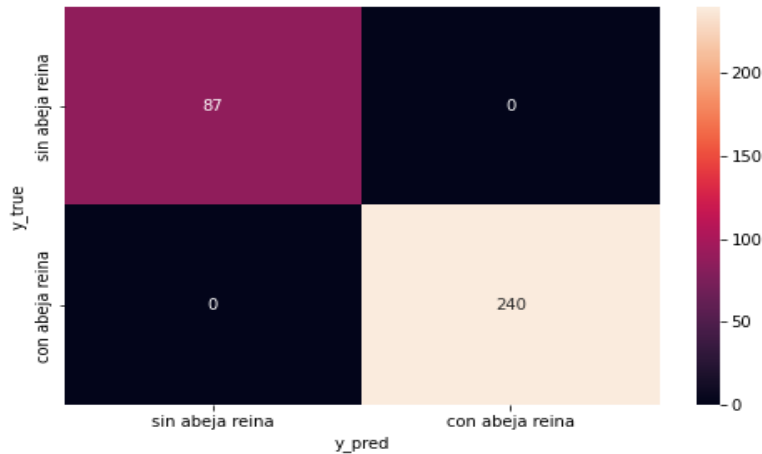


Figura 69: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 2* [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.

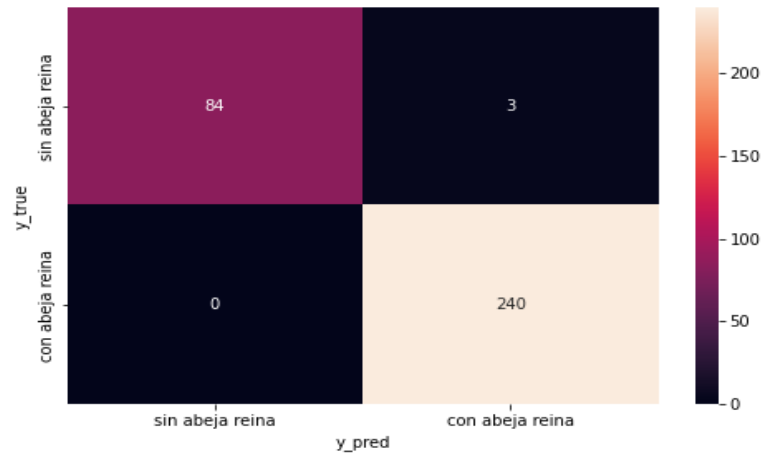


Figura 70: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 2* [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20).
Fuente: Elaboración propia.

Para este caso, se puede ver que el primer modelo obtuvo mejores resultados en comparación con el segundo. Las métricas del primero son *precision* y *recall* de 100% para ambas estados de salud de la colmena, en cambio el segundo modelo obtuvo *precision* 100% y *recall* 96.55% para el estado “sin abeja reina” y *precision* 98.77% y *recall* de 100% para el segundo estado.

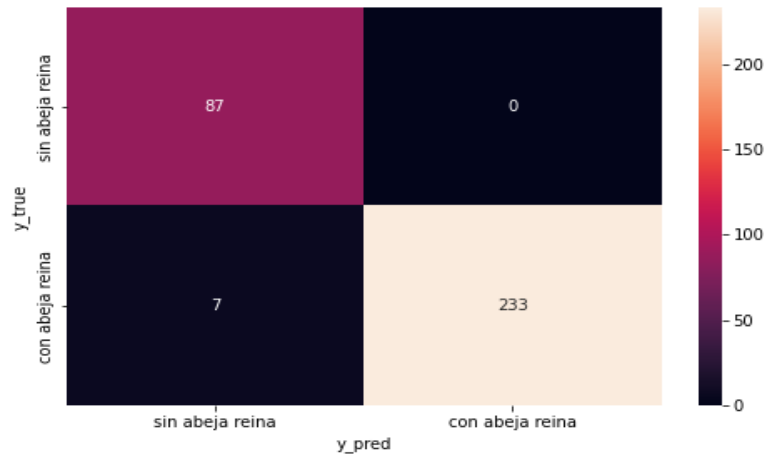


Figura 71: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset 2* [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.

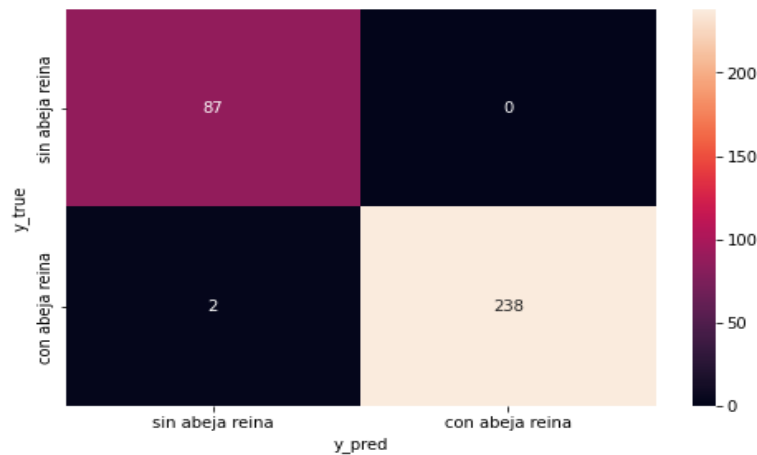


Figura 72: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset 2* [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20).
Fuente: Elaboración propia.

Como se puede apreciar en este experimento, el segundo modelo ha obtenido mejores resultados en comparación con el primero. Con respecto a las métricas, para el primer modelo se obtuvo *precision* 99.55% y *recall* 100% asociados al estado “sin abeja reina”, para el segundo estado se ha obtenido *precision* 100% y *recall* 97.08%. Las métricas obtenidas por el segundo fueron *precision* 97.75% y *recall* 100% para el estado “sin abeja reina”, en cambio para el segundo estado la *precision* fue de 100% y el *recall* 99.17%.

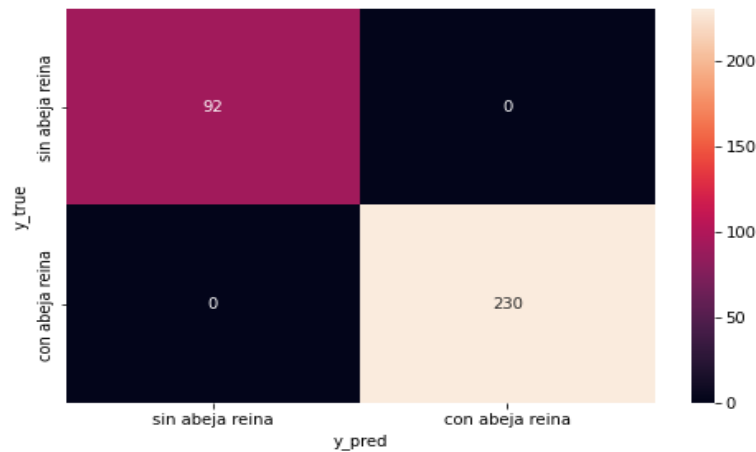


Figura 73: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 13 MFCCs, *dataset 3* [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.

Como se puede observar en este experimento, la CNN logró identificar correctamente los estados de salud de las colmenas mediante su zumbido, usando las muestras de prueba. El modelo obtuvo una *precision* y *recall* de 100% para ambos estados.

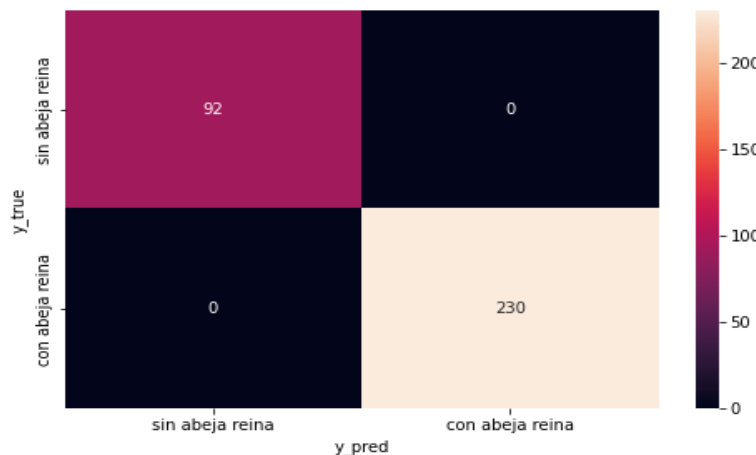


Figura 74: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset 3* [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.

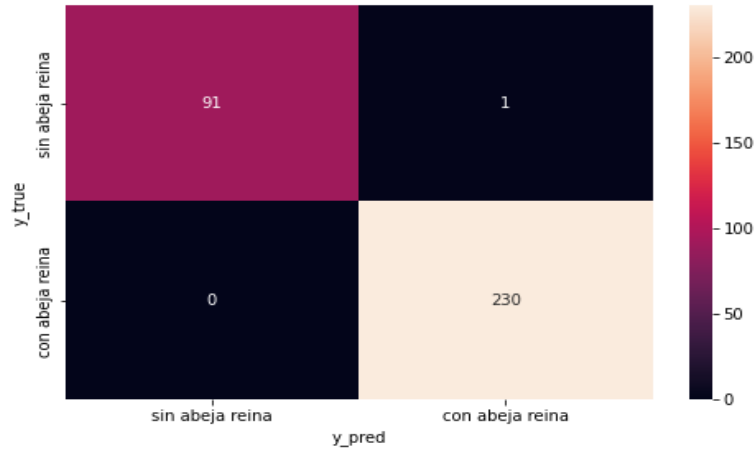


Figura 75: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 20 MFCCs, *dataset* 3 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20).
Fuente: Elaboración propia.

En este caso, ambos modelos lograron tener un buen desempeño con respecto a la clasificación del estado de salud de la colmena, siendo el primero mejor que el segundo. Las métricas del primer modelo fueron *precision* y *accuracy* 100% para ambos estados de salud, en cambio el segundo obtuvo una *precision* de 100% y *recall* de 98.91% para el estado “sin abeja reina”, con respecto al segundo estado se obtuvo *precision* 99.57% y *recall* 100%.

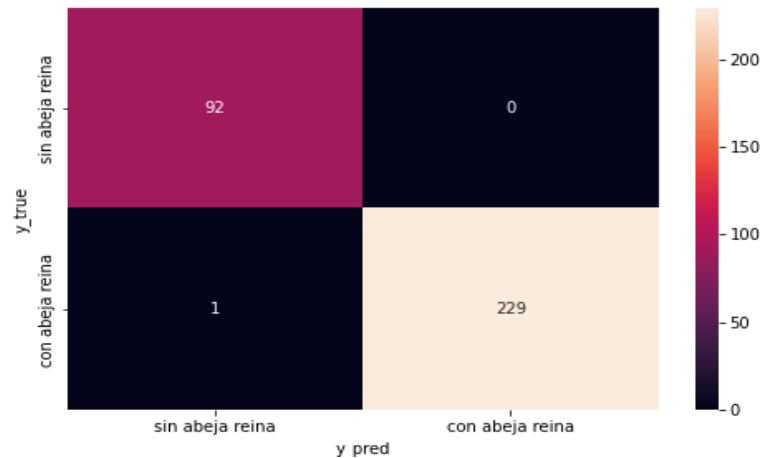


Figura 76: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset* 3 [s], 3 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 13).
Fuente: Elaboración propia.

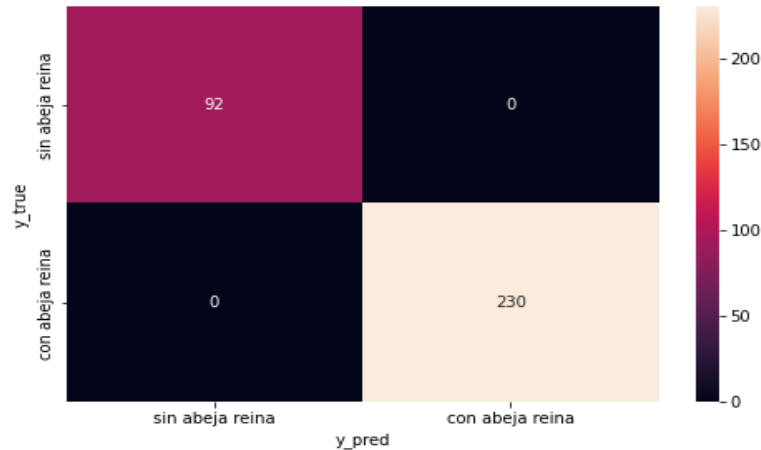


Figura 77: Matriz de confusión generada por el primer modelo descrito en 3.2.7. Métrica usada *accuracy*, número de épocas 72, 40 MFCCs, *dataset* 3 [s], 4 capas *Conv2D* + *MaxPooling2D* + *Dropout*. (mfcc 20).
Fuente: Elaboración propia.

Para este caso, ambas CNNs presentaron un buen desempeño al momento de clasificar los estados de salud de la colmena, siendo el segundo modelo mejor que el primero en la realización de dicha labor. Con respecto a las métricas obtenidas, la primera CNN logró una *precision* de 98.92% y *recall* de 100% para el estado “sin abeja reina”, en cambio en el segundo estado se obtuvo *precision* 100% y *accuracy* 99.57%. Para el segundo modelo se obtuvo *precision* y *recall* 100% para ambos estados.

4.2.5 ¿Cómo afecta la duración de la grabación en los resultados de la clasificación?

Al hacer la comparación entre los resultados obtenidos de las matrices de confusión mostradas anteriormente, se puede observar que al aumentar la duración del zumbido ciertas características se vuelven más descriptivas, posiblemente esto se deba a que el aumento en la duración de la grabación del zumbido, permita captar mejor las señales relevantes emitidas por la colmena. Lo anterior provoca que los modelos entrenados con ellas aumenten sus métricas al momento de clasificar los estados de salud de las colmenas, por ejemplo, en el caso de los MFCCs extraídos desde el espectrograma de Mel, se pudo ver que la *precision* y *recall* aumentaron desde *precision* 100% y *recall* 98.11% para el estado “sin abeja reina”, y para el estado “con abeja reina” *precision* 99.13% y *recall* 100% a *precision* y *recall* 100% para ambos estados usando los datasets de 2 y 3 segundos, aunque también se debe tener en cuenta que esto conlleva un aumento en el uso de la memoria RAM y el tiempo de entrenamiento. Así que, una alternativa al momento de entrenar modelos que traten de detectar estados de salud de la colmena, puede ser variar la duración de los zumbidos, para verificar si las métricas del modelo mejoran.

4.2.6 Uso de 13, 20 o 40 MFCCs

Al analizar los resultados obtenidos en las matrices de confusión previamente mostradas, se puede ver que en general los modelos entrenados con 40 MFCCs casi siempre obtienen métricas con *precision* y *recall* 100% para ambas clases, aunque al aumentar los tiempos de la duración de los zumbidos de 1 [s] a 2 o 3 [s], se puede ver que los 13 y 20 MFCCs pueden lograr los mismos resultados. Al parecer, 13 MFCCs son suficientes para caracterizar el zumbido de una colmena de abejas para detectar si posee o no abeja reina, sólo basta variar la duración del zumbido de la colmena para que el modelo basado en CNN obtenga mejores resultados en sus métricas. Aunque los 20 y 40 MFCCs también permiten obtener buenas métricas con respecto a la clasificación producida por las CNNs, puede ser que al emplearlos en el análisis de otro tipo de estado de salud de colmenas de abejas u otro fenómeno asociado a clasificación de estados basados en audios, sean mucho más relevantes que el uso de 13 MFCCs.

4.2.7 Comparación entre los diferentes enfoques

Si bien todas las características utilizadas para entrenar los modelos de clasificación del estado de salud de la colmena basados en CNNs mostraron buenos resultados, existen varias diferencias entre ellas que podrían ayudar a seleccionar una por sobre otra.

En el caso de los espectrogramas logarítmicos, hay que tener en cuenta que solo se pudo realizar el entrenamiento de la red con el *dataset* de audios de 1 [s], esto se debe a que la cantidad de memoria RAM utilizada por ellos, fue mayor a la disponible. Otra consideración a tener en cuenta es que el uso de estos espectrogramas provoca que el modelo tome bastante tiempo en ser entrenado, para este caso tomó aproximadamente 3:47 y 4:23 [min] entrenar cada uno, por lo que al aumentar la duración de los zumbidos de la colmena, el tiempo de entrenamiento debería ser mayor. Si bien los modelos basados en CNNs y espectrogramas logarítmicos pueden dar buenos resultados, los pesos iniciales de la red influenciarán bastante en ellos, es por eso que en ciertas ocasiones será necesario entrenarlo varias veces, hasta obtener los resultados esperados.

En cuanto a los espectrogramas de Mel, la cantidad de memoria RAM ocupada por ellos es menor en comparación a la de los espectrogramas logarítmicos, pudiéndose realizar el entrenamiento de los modelos con los *datasets* de duración 1, 2 y 3 segundos. También se debe tener en cuenta que el tiempo de entrenamiento es mucho menor tomando alrededor de 41 [s] (*dataset* 1 [s]), 1:22 [min] (*dataset* de 2 [s]) y 1:26 [min] (*dataset* 3 [s]), cabe tener en cuenta que estos tiempos varían según la arquitectura de la CNN. En general, el uso del espectrograma de Mel mostró mejores resultados en comparación al espectrograma logarítmico, aunque comparten el mismo problema de los pesos iniciales.

Los MFCCs obtenidos tanto de la serie de tiempo de la amplitud como del

espectrograma de Mel, son los que dan mejores resultados. Los pesos iniciales de las CNNs en general afectaron en poca medida el desempeño de estos modelos (bastó con entrenar el modelo una sola vez para obtener buenos resultados con respecto a la clasificación). Además, debido a que la cantidad de MFCCs usados son mucho menores en comparación con las frecuencias extraídas de los espectrogramas, el entrenamiento de estos modelos fue mucho más rápido, el rango va desde 18 - 41 [s] (para los tres *datasets*), cabe tener en consideración que esto varía según la arquitectura de la CNN y la cantidad de MFCCs usados.

Teniendo en consideración los puntos señalados anteriormente, al momento de seleccionar características para caracterizar los zumbidos de una colmena y construir un modelo basado en CNNs que clasifique su estado de salud, sería conveniente usar en primera instancia los MFCCs extraídos de la serie de tiempo de la amplitud (presentan tan buenos resultados como los extraídos desde el espectrograma de Mel y no se necesita generar dicho espectrograma previamente), en segundo lugar se podrían utilizar los MFCCs extraídos desde el espectrograma de Mel (se han colocado en segundo lugar, ya que requieren generar previamente el espectrograma de Mel). Si por algún motivo no se pueden usar los MFCCs, la siguiente opción es el espectrograma de Mel, debido a que ocupa menos memoria y se puede entrenar a la CNN más rápido, además su desempeño es un poco mejor que el del espectrograma logarítmico. Finalmente el último enfoque sería el del espectrograma logarítmico, el cual consume una mayor cantidad de memoria RAM, además de aumentar considerablemente el tiempo que demora la CNN en ser entrenada.

4.3 Pruebas en terreno

Una vez concluido el proceso del análisis de los resultados de los audios mencionados en la sección 3.2.1 y el desarrollo de la aplicación web, se procedió a probar lo llevado a cabo en la sección 4.2 en colmenas ubicadas en las instalaciones de la UTFSM (colmenas del USM Bee Lab), esto se llevó a cabo en colaboración con el Dr. José Lino Contreras Véliz, la Dra. Karen del Pilar Yañez Arellano, Enrique y Fernando. A continuación se dará el contexto en el cual se realizó el experimento, la forma en la que se llevó a cabo la obtención de los audios y los resultados obtenidos.

4.3.1 Contexto

Los audios fueron extraídos de dos colmenas llamadas **colmena 6** y **colmena extra** que se encuentran ubicadas en el apiario de la casa central de la UTFSM (Universidad Técnica Federico Santa María - Av España 1680, Valparaíso, Chile) frente a la Avenida España (hay aproximadamente unos 100 o 200 metros de distancia entre la Avenida España y el apiario). Debido al alto tráfico de locomoción vehicular que pasa por dicho lugar, se contó con algo de presencia de ruido en las grabaciones, cabe destacar que estas

fueron realizadas entre las 15:00 hrs y las 20:00 hrs aproximadamente en los días 25/11/2022 (día nublado), 02/12/2022 (día soleado) y 09/12/2022 (día soleado). Dentro de las colmenas existían las siguientes características:

- **Colmena 6:** Tenía presencia de abeja reina entre las fechas 25/11/2022 y 09/12/2022.
- **Colmena extra:** No tenía presencia de abeja reina. Su pérdida ocurrió en algún momento antes de la experimentación, pero dentro de ella habían larvas de abeja reina. Entre el 25/11/2022 y 02/12/2022 no había presencia de abeja reina, en cambio el 09/12/2022 ya había abeja reina.

4.3.2 Obtención de los audios

La obtención de audios se logró mediante la instalación de un micrófono dentro de cada colmena mencionada en el punto anterior. Los micrófonos usados fueron **Micrófonos de escritorio de diseño fino marca Trust**, micrófono condensador omnidireccional con una relación de señal ruido de 58 dB, respuesta en frecuencia de 100 Hz - 12000 Hz, impedancia de 2200 Ohm, sensibilidad de -45 dB, nivel de presión sonora de 115 dB, conexión por cable, tipo de conector 3.5 mm, largo del cable 1.8 m, peso total de 47 g, sin reducción de ruido, con número de referencia 21674. Los micrófonos fueron ubicados en medio de los cuadros de las colmenas. La tasa de muestreo usada fue 44100 muestras por segundo, para grabar las colmenas se utilizaron un teléfono *Galaxy A31* y un *laptop LENOVO ideapad 330S* con sistema operativo *Microsoft Windows 10 Home Single Language* procesador *Intel(R) Core(TM) i7-8550U*.



Figura 78: Instalación de micrófono dentro de la colmena extra, realizada por el Dr. José Lino Contreras Véliz y la Dra. Karen del Pilar Yañez Arellano. Fuente: Elaboración propia.

4.3.3 Resultados preliminares

Se probó el sistema con una de las CNNs que obtuvo uno de los mejores resultados descritos en la sección 4.2, esta CNN fue la que procesa 13 MFCCs extraídos desde la serie de tiempo de la amplitud de audios con duración de 3 segundos. Al momento de probarlo este no pudo reconocer correctamente los audios que procedían de colmenas que no tenían presencia de abeja reina, catalogando todos esos audios como poseedores de abeja reina.

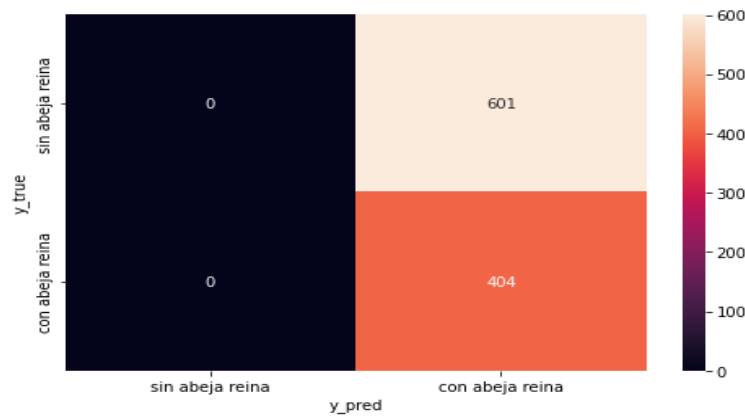


Figura 79: CNN probada con audios del día 25/11/2022. Fuente: Elaboración propia.

Como se puede observar en la anterior matriz de confusión, la *precision* asociada a la clase sin abeja reina es de 0% al igual que su *recall*. En cambio la *precision* asociada a la clase con abeja reina es de 40.20% y su *recall* de 100%. Lo anterior indica que la CNN puede detectar correctamente todas las muestras de zumbidos provenientes de colmenas con abeja reina pero existirá un 40.20% de probabilidad de que efectivamente exista abeja reina en la colmena, en cambio para el caso de colmena sin abeja reina, la CNN no puede detectar colmenas sin abeja reina. Este problema pudo haberse producido debido a la gran presencia de ruido que hubo ese día, sobreajuste del modelo, el *dataset* inicial con el que se entrenó la CNN era pequeño o sus muestras eran poco variadas lo que provocó que el modelo no pudiera aprender a generalizar correctamente la clase sin abeja reina, u otros motivos.

4.3.4 Reentrenamiento de la red

Debido al inconveniente mencionado anteriormente, se decidió reentrenar la CNN con un nuevo *dataset* más robusto, para ello se combinaron los audios obtenidos desde *hiveeyes* con los audios usados por el proyecto *LongHive* (Nolasco et al., 2018). Dicho proyecto fue el ganador del concurso *#IoTforGood* organizado por *Hackster.io* y consiste en una solución integral para el monitoreo de colmenas formado por varios sensores relevantes (temperatura, humedad, CO2, otros) los cuales son usados para indicar al apicultor el estado del ambiente dentro de la colmena, además posee una CNN que

procesa los zumbidos de la colmena mediante espectrogramas de Mel provenientes de audios con duración de 4.5 segundos para detectar presencia o ausencia de abeja reina con una precisión de 89%. Al combinar los audios de *hiveeyes* con los audios usados en el proyecto *LongHive* se obtuvieron 7246 muestras de audio con duración de 3 segundos (aproximadamente 6 horas de grabaciones de zumbidos de abejas) donde el 80% de ellos fue usado para el entrenamiento de la red y el 20% para probarla, además se cambió la métrica de *accuracy* usada inicialmente en el entrenamiento de la CNN por la métrica *F1 score*. A continuación se presentan los resultados obtenidos al reentrenar la red.

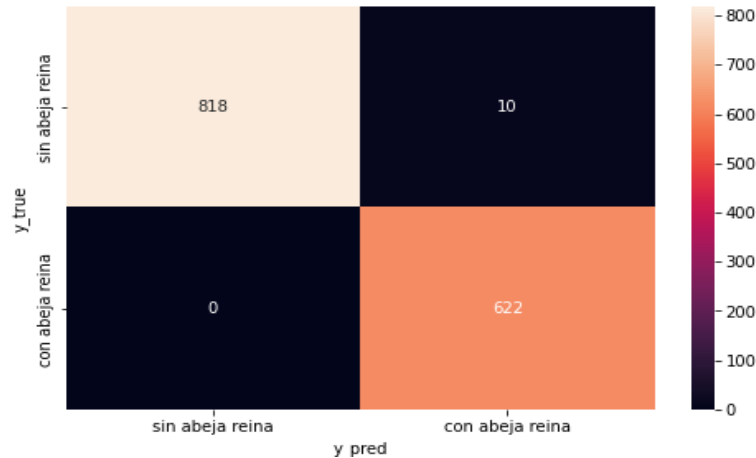


Figura 80: Matriz de confusión obtenida por la “CNN modelo final”. Probada con los audios del conjunto de prueba. Fuente: Elaboración propia.

Como se puede observar en la matriz de confusión, la CNN reentrenada obtuvo una precisión promedio de 99.21% identificando de buena manera las colmenas con y sin abeja reina, esto se logró reentrenando la red una sola vez, demorando 15 minutos con 17 segundos usando la CPU.

4.3.5 Resultados finales

Luego de reentrenar la CNN con el *dataset* formado por los audios obtenidos de *hiveeyes* y *LongHive*, se procedió a probarla con los audios obtenidos de las colmenas del USM Bee Lab casa central, debido a que las grabaciones obtenidas el día 25/11/2022 presentaron demasiado ruido producido principalmente por actividades realizadas dentro de la universidad en esa fecha, estas fueron descartadas y en su lugar se probó la CNN con las muestras obtenidas el día 02/12/2022 y 09/12/2022. A continuación se presentan las matrices de confusión generadas al probar la red con las grabaciones mencionadas anteriormente.



Figura 81: Matriz de confusión obtenida por la “CNN modelo final”. Probada con los audios del día 02/12/2022. Fuente: Elaboración propia.

El día 02/12/2022 se recopilieron 3853 muestras de zumbidos con duración de 3 segundos, 2646 provenientes de la colmena extra (sin abeja reina en esa fecha) y 1207 provenientes de la colmena 6 (con abeja reina en esa fecha) recopiladas entre las 16:00 y 18:20 hrs aproximadamente. La CNN obtuvo una precisión promedio de 95.44% al identificar los estados de las colmenas a través de los audios entregados, la red pudo detectar el 95.50% (*recall* de la clase sin abeja reina) de los audios que provenían de la colmena sin abeja reina.

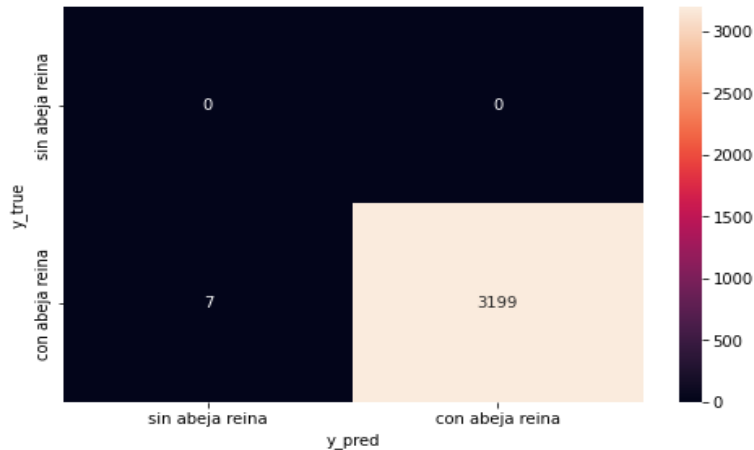


Figura 82: Matriz de confusión obtenida por la “CNN modelo final”. Probada con audios del día 09/12/2022. Fuente: Elaboración propia.

El día 09/12/2022 tanto la colmena extra como la colmena 6 contaban con presencia de abeja reina, se recopilaron en total 3206 muestras de audio con duración de 3 segundos entre las dos colmenas recopiladas entre las 14:00 y 15:30 hrs aproximadamente. La CNN pudo identificar correctamente el 99.78% de las muestras que correspondían a una colmena con abeja reina.

CAPÍTULO 5: CONCLUSIONES

En la presente memoria, se han mostrado los resultados obtenidos al utilizar diferentes enfoques basados en caracterización del audio por medio de espectrogramas logarítmicos, espectrogramas de Mel y MFCCs en combinación con CNNs, para la identificación de estados de salud de colmenas de abejas, en este caso para determinar si la colmena posee o no abeja reina. Esto se ha realizado generando 3 *datasets* con muestras de audios con duración de 1, 2 y 3 segundos (todos los *datasets* están desbalanceados, existiendo mayor cantidad de muestra de colmenas con abeja reina), con 1667, 1635 y 1606 muestras respectivamente, con la intención de examinar en qué medida afecta la duración del zumbido emitido por la colmena en la identificación del fenómeno producida dentro de ella. Además, con respecto a la caracterización del zumbido emitido por las colmenas, se han utilizado 13, 20 y 40 MFCCs, para examinar cuantos coeficientes son suficientes para caracterizar de buena manera el zumbido de una colmena. También se ocuparon dos modelos de CNNs, uno de cuatro capas y otro con tres capas, esto debido a que el primer modelo se puede entrenar con 20 y 40 MFCCs pero producto de la cuarta capa no se puede entrenar con los 13 MFCCs, en cambio el segundo modelo no presenta esta dificultad.

5.1 Sobre los resultados de los diferentes enfoques

Al analizar los resultados obtenidos a partir de los 38 experimentos realizados, se pudo apreciar que cada uno de estos enfoques tiene sus ventajas y desventajas, como la cantidad de memoria RAM utilizada, el tiempo que demora en entrenar la CNN, la dificultad asociada a generar la característica a usar, entre otras. A continuación se darán más detalles sobre cada uno de ellos.

5.1.1 Enfoque basado en MFCCs

Al hacer la comparación entre cada uno de los enfoques, se pudo ver que las CNNs entrenadas con MFCCs (obtenidos desde la serie de tiempo de la amplitud y del espectrograma de Mel) fueron las que obtuvieron mejores resultados y en general bastaba con entrenarlas una vez para obtener un buen clasificador (los pesos aleatorios iniciales de la red, afectaron en poca medida los resultados del clasificador), además el rango de tiempo asociado al entrenamiento de la CNN estaba entre 18 - 41 [s]. Debido a que tanto el uso de MFCCs extraídos de la serie de tiempo de la amplitud y del espectrograma de Mel ofrecían resultados similares, sería conveniente partir aplicando el primer enfoque en lugar del segundo, ya que este último necesita el paso adicional de generar previamente el espectrograma de Mel.

5.1.2 Enfoque basado en espectrograma de Mel

Para el caso de los espectrogramas, la caracterización del audio mediante el espectrograma de Mel presenta resultados un poco mejores en comparación con el enfoque del espectrograma logarítmico. Además, este enfoque consume menos memoria RAM al momento de entrenar la CNN (permitiendo usar los tres *datasets*) y es mucho más rápido su entrenamiento encontrándose en un rango de 38 [s] - 1:30 [min]. Por lo mencionado anteriormente, si por algún motivo no se puede ocupar un enfoque basado en CNNs y MFCCs para generar un clasificador basado en audio, este sería el siguiente enfoque recomendado.

5.1.3 Enfoque basado en espectrograma logarítmico

El enfoque basado en espectrograma logarítmico, quedó en último lugar, en comparación con el desempeño de los anteriores. La cantidad de memoria RAM ocupada al momento de entrenar la CNN (superior a los 12.68 GB disponibles), solo permitió que el clasificador pudiera ser entrenado con el *dataset* con audios de 1 [s], además el tiempo empleado en entrenar la CNN es mucho mayor en comparación con los otros enfoques, tomando entre 3:47 - 4:23 [min]. Algo que tiene en común este enfoque y el del espectrograma de Mel, es que los pesos iniciales aleatorios de la red, afectan en gran medida su desempeño al momento de tener que clasificar el conjunto de *test*. Es por ello que a veces es necesario entrenar el modelo más de una vez, para mejorar los resultados de la clasificación.

5.2 Acerca del análisis visual de las caracterizaciones

A continuación, se presentarán algunas de las conclusiones que se obtuvieron al comparar y analizar visualmente los resultados de las caracterizaciones de los zumbidos de las colmenas de abejas, que se utilizaron a lo largo de la presente investigación.

5.2.1 Sobre el espectrograma de Mel

Se pudo apreciar que el espectrograma de Mel era visualmente más diferenciador en comparación al espectrograma logarítmico. Se pudo ver que en los audios obtenidos desde *hiveeyes* en las colmenas sin abeja reina, aparecían unas franjas alrededor de las frecuencias 4096 y 8192 [hz] muy marcadas, que en el caso de las colmenas con abeja reina no aparecían o simplemente aparecían pero mucho menos marcadas, la cuál podría ser ocupada como un posible patrón para identificar si una colmena se encontraba en el estado “con abeja reina” o “sin abeja reina”.

5.2.2 Sobre los MFCCs extraídos del espectrograma de Mel

Para este caso, se pudo ver a partir de los audios recopilados desde *hiveeyes* que en general cuando una colmena se encontraba en el estado “con abeja reina” se obtenía una imagen anaranjada o rojiza en la cual no se podía observar con claridad ningún coeficiente marcado, en cambio cuando la colmena estaba bajo el estado “sin abeja reina” se observó que el 4° coeficiente comenzaba a destacarse y se vislumbraba un poco el 7° y 9° coeficiente, lo cual podría sugerir que la aparición del 4° MFCC indicaría la presencia de una colmena sin abeja reina. Además en comparación al espectrograma de Mel, es más fácil detectar visualmente un patrón que permita diferenciar entre una colmena que tiene o no la presencia de abeja reina, debido a que en el caso de MFCCs es necesario examinar una pequeña cantidad de coeficientes (13, 20 o 40 MFCCs que fueron utilizados en la presente investigación), en comparación con el rango de frecuencias que presenta el espectrograma de Mel que llegaba hasta 16384 [hz] aproximadamente.

5.2.3 Sobre los MFCCs extraídos desde el espectrograma de Mel y desde la serie de tiempo de la amplitud

Para el caso de la comparación visual entre los MFCCs extraídos desde la serie de tiempo de la amplitud y del espectrograma de Mel, se podía apreciar que en el primer caso aparecían más coeficientes cepstrales en comparación con el segundo (esto podría deberse a la forma en que el algoritmo utilizado fue implementado), pero en ambos, cuando la colmena no tenía la presencia de la abeja reina, el 4° MFCC era el que más resaltaba. Si bien inicialmente se pensó que los MFCCs extraídos desde el espectrograma de Mel tendrían un mejor desempeño en comparación con los provenientes de la serie de tiempo de la amplitud (debido a que los primeros, son más fácil de identificarlos para una persona), resultó que las CNNs no tuvieron mayores problemas al identificar el estado de salud de la colmena usando ambas caracterizaciones.

5.3 Otras conclusiones

Un punto a tener en cuenta, es que una posible forma de mejorar el desempeño de la CNN al momento de clasificar el estado de salud de la colmena, es aumentar la duración de las muestras de audios para entrenar la red. En general, se pudo observar que en algunos casos al aumentar el tamaño de la duración del zumbido, también podría mejorar el desempeño del clasificador. Por ejemplo, en el caso del clasificador que utilizó 13 MFCCs extraídos desde audios del dataset de 1 [s], obtuvo tan buenos resultados como los clasificadores que utilizaron 40 MFCCs extraídos desde los audios contenidos en los datasets de 1 [s] y 3 [s].

Para el caso de los MFCCs, se pudo ver que en general para identificar si una colmena posee o no la presencia de abeja reina, el uso de 13 MFCCs era suficiente para

generar un modelo de clasificación que obtuviera un buen desempeño al momento de identificar el estado de la colmena. Posiblemente para identificar otro tipo de fenómeno dentro de la colmena, podría ser necesario el uso de una mayor cantidad de coeficientes o puede ser que el enfoque de los 13 MFCCs con variación en la duración del zumbido de la colmena sea suficiente.

Según los resultados obtenidos a lo largo del presente trabajo, se pudo observar que es una buena idea dotar a un modelo de reconocimiento de patrones de audio con características similares a las del oído humano. Donde los MFCCs y espectrogramas de Mel son los que más se asemejan a como el oído humano percibe los sonidos, en comparación a los espectrogramas logarítmicos que no están transformados a una escala similar a como el oído humano percibe el sonido.

5.4 Trabajo futuro y posibles mejoras

Con respecto a mejoras y trabajo futuro que se puede realizar a partir de la investigación y resultados obtenidos en la presente memoria, se podría aplicar el mismo procedimiento para identificar otras anomalías dentro de las colmenas de abejas, por ejemplo: ataque de abejorros, ataque de hormigas, presencia de pesticidas, Varroa u otros. También cabe tener en consideración que las CNNs utilizadas en los experimentos realizados en la presente memoria, solo podían clasificar si la colmena se encontraba con abeja reina o sin abeja reina (sólo podía detectar si la colmena presenta o no un determinado estado de salud), pero en realidad dentro de una colmena pueden estar ocurriendo varios fenómenos, ya que estos no son excluyentes, por lo que otra mejora que se podría realizar, es modificar la cantidad de neuronas de la capa de salida junto a su función de activación para que pueda generar una probabilidad asociada a cada fenómeno que pueda estar ocurriendo dentro de la colmena y que hayan sido utilizados para entrenar el modelo de clasificación. Cabe tener en cuenta que para generar los espectrogramas logarítmicos, espectrogramas de Mel y MFCCs, se han usado mayoritariamente parámetros por defecto, por lo que otra posible mejora podría ser variar los parámetros utilizados en la generación de la caracterización de los zumbidos de las colmenas para examinar qué tanto afectan ellos al momento de clasificar el estado de salud, ya que puede ser que exista una mejor combinación de parámetros que puedan caracterizar de mejor manera el zumbido.

Debido a que el problema tratado en esta memoria es la identificación de anomalías mediante el análisis del sonido emitido por la entidad cuyo estado se quiere identificar (para este caso específico, fue la detección de anomalías en colmena de abejas, mediante el análisis del sonido que ella emite), los enfoques propuestos y experimentos realizados pueden ser utilizados para resolver otros problemas similares, por ejemplo se podría usar el enfoque del espectrograma de Mel y CNNs para procesar el sonido emitido por un motor e identificar las piezas que podrían estar fallando. Es por ello que personas

que estén buscando solucionar problemas asociadas a otras áreas, que requieran del procesamiento del sonido emitido por alguna entidad de interés para determinar algún estado o fenómeno en ella, se pueden basar en los experimentos y enfoques utilizados en este trabajo, para generar una solución a dichos problemas.

Con respecto a la *API*, para procesar archivos de audio con *librosa* con un formato diferente al *.wav*, es necesario tomar el archivo que está cargado en memoria y escribirlo en el sistema de archivos para que pueda ser procesado. Una mejor solución sería poder implementar alguna solución que permitiera tomar directamente el archivo que ya se encuentra cargado en memoria y extraer la información que el sistema requiere desde el.

Para el caso de trabajar con una CNN que detectara más de una anomalía dentro de la colmena, sería necesario colocar una nueva CNN dentro de la *API*, que pueda lograr dicho cometido y modificar la función que formatea la respuesta para que esté acorde a dicho escenario. También habría que modificar el frontend para ajustar las vistas al nuevo formato de salida.

Para esta memoria se trabajó en un entorno de desarrollo y entorno de producción basado en planes gratuitos. Un posible trabajo a futuro sería llevar todo el sistema a un entorno de producción más robusto para que pueda ser usado por usuarios interesados, podría ser usado inicialmente dentro de las instalaciones de la universidad y después ser llevado a un público más amplio.

También se podría utilizar un *dataset* que tenga una mayor cantidad de muestras de diferentes colmenas tomadas en diferentes condiciones climáticas y variar los parámetros usados por la CNN como tasa de aprendizaje, tamaño del lote u otros, con el fin de mejorar el desempeño presentado por las CNNs mostradas a lo largo del presente trabajo.

REFERENCIAS BIBLIOGRÁFICAS

- Acha Ruiz, G. A. (n.d.). *Biometría de voz con Librosa y CNN – Guillermo André Acha Ruiz*. Guillermo André Acha Ruiz. Recuperado mayo 13, 2022, de <https://guillermoacharuiz.com/portfolio/biometriadevoz/>
- Achury, F., & Smith, B. (2020). Resumen. En *Varroa un problema de gran impacto a nivel sanitario y productivo en la apicultura, métodos de diagnóstico, tratamientos y prevención (tesis pregrado)*. Universidad de Ciencias Aplicadas y Ambientales UDCA (Colombia). <https://repositorioslatinoamericanos.uchile.cl/handle/2250/3133923>
- Alwaysdata. (n.d.). alwaysdata: Hosting made for everyone. Recuperado enero 25, 2023, de <https://www.alwaysdata.com/en/>
- Amado Sarmiento, O. A. (2013). *Sistema de monitoreo del sonido emitido por una colmena de abejas como herramienta para estudios comportamentales*. Universidad Nacional de Colombia Facultad de Ingeniería, Departamento de Ingeniería Eléctrica y Electrónica. (tesis magister) <https://repositorio.unal.edu.co/bitstream/handle/unal/51525/1010180936.2013.pdf?sequence=1&isAllowed=y>
- Araos, R. (2021, marzo 22). *Cómo controlar a Varroa destructor, principal amenaza sanitaria en abejas melíferas*. El Mercurio Campo. <https://www.fraunhofer.cl/content/dam/chile/es/media-2021/csb/docs/El%20Mercurio%20Campo%20Varros%2022%20marzo%202021.pdf>
- Coordinación Nacional de Ganadería. (n.d.). 10.2 Como apoyar una colonia de obreras ponedoras. En *Manual Básico de Apícola* (p. 23). https://www.mieldemalaga.com/data/manual_basico_apicultura.mex.pdf
- Departamento de ingeniería informática. Universidad de Santiago de Chile. (2020, septiembre 28). *Aplicación desarrollada por egresados DIINF es utilizada en diagnósticos de apicultura*. FACULTAD DE INGENIERÍA UNIVERSIDAD DE SANTIAGO DE CHILE. Recuperado julio 28, 2021, de <https://www.fing.usach.cl/noticias/aplicacion-desarrollada-por-egresados-diinf-es-utilizada-en-diagnosticos-de-apicultura>
- Django Software Foundation. (n.d.). *Django overview*. Django. Recuperado noviembre 17, 2022, de <https://www.djangoproject.com/start/overview/>
- Docker Inc. (n.d.). *Get Docker*. Docker Documentation. Recuperado noviembre 17, 2022, de <https://docs.docker.com/get-docker/>
- Docker Inc. (n.d.). *Overview*. Docker Documentation. Recuperado noviembre 17, 2022, de

<https://docs.docker.com/compose/>

Eberhardt, C. (2006). 2.7 Daño producido por Varroa destructor. En *Caída natural de Varroa destructor Anderson & Trueman (Mesostigmata: Varroidae), como indicador del nivel de infestación y efecto del tratamiento otoñal con ácido fórmico sobre la colmena (tesis pregrado)* (pp. 11-12). Universidad Austral de Chile. <http://cybertesis.uach.cl/tesis/uach/2006/fae.16c/doc/fae.16c.pdf>

Espin, D. (n.d.). ▷ *Colmena Zanganera: Qué hacer y como salvarla [Guía]*. La Tienda del Apicultor. Recuperado mayo 20, 2022, de <https://www.latiendadelapicultor.com/blog/que-hacer-con-una-colmena-zanganera/>

Fernandez, D. M. (2016, agosto). *Despliegue de Django en Pythonanywhere*. Molanco. Recuperado enero 25, 2023, de <https://www.molanco.com/es/blog/despliegue-de-django-en-pythonanywhere>

Ferrari, S., Silva, M., Guarino, M., & Berckmans, D. (2008). Monitoring of swarming sounds in bee hives for early detection of the swarming period. En *Computers and Electronics in Agriculture* (1st ed., Vol. 64, pp. 72-77). ELSEVIER. https://www.researchgate.net/publication/229224190_Monitoring_of_swarming_sounds_in_bee_hives_for_early_detection_of_the_swarming_period

García, E. (2019). 2.1 Teoría de las redes neuronales convolucionales, 2.1.1 Capas de convolución y 2.1.2 Pooling. En *Introducción a las redes neuronales de convolución. Aplicación a la visión por ordenador (tesis de pregrado)* (pp. 17-20). Universidad de Zaragoza. <https://core.ac.uk/download/pdf/290002463.pdf>

Git - Submodules. (n.d.). Git SCM. Recuperado noviembre 19, 2022, de <https://git-scm.com/book/en/v2/Git-Tools-Submodules>

Gómez, M. (2017, octubre 11). *Procesamiento digital de señales Semana 5. DFT*. Recuperado julio 29, 2021, de <https://ccc.inaoep.mx/~pgomez/cursos/pds/slides/S5-DFT.pdf>

Google Colab. (n.d.). Google Colab. Recuperado mayo 20, 2022, de <https://research.google.com/colaboratory/intl/es/faq.html>

Harding, J. (2021, enero 30). *The HoneyBee, Varroa, Vibration, and CCD*. Modern Survival Blog. Recuperado junio 22, 2021, de <https://modernsurvivalblog.com/natural-threats/the-honeybee-varroa-vibration-and-ccd/>

hostabee. (2020). *Solutions pour Apiculteurs - Ruche connectée I Monitoring et Antivols*. Hostabee.com. Recuperado mayo 23, 2022, de

<https://hostabee.com/es/apicultores/>

Innovatione AgroFood Design. (2019, junio 17). *Polinización: tipos y polinizadores en el campo- Innovatione*. Innovatione AgroFood Design. Recuperado noviembre 24, 2022, de <https://innovatione.eu/2019/06/17/polinizacion/>

Lagerman, B. (n.d.). *DETECCIÓN GRATIS DE VARROA DESTRUCTOR*. Recuperado julio 14, 2021, de <https://beescanning.com/es/deteccion-gratis-de-varroa-destructor/>

La tienda del apicultor. (n.d.). *Cómo revisar una colmena: Inspección interna y externa*. La Tienda del Apicultor, blog de apicultura. Recuperado mayo 20, 2022, de <https://www.latiendadelapicultor.com/blog/como-revisar-una-colmena/>

La tienda del apicultor. (n.d.). *Quién es quien en la colmena: Una sociedad muy estructurada*. La tienda del apicultor, blog de apicultura. Recuperado mayo 20, 2022, de <https://www.latiendadelapicultor.com/blog/quien-es-quien-en-la-colmena-una-sociedad-muy-estructurada/>

La tienda del apicultor. (n.d.). *Varroa destructor: Qué es y como tratarla*. La tienda del apicultor, blog de apicultura. Recuperado junio 20, 2021, de <https://www.latiendadelapicultor.com/blog/varroa-destructor-que-es-y-como-tratarla/>

Lee, J. (n.d.). *Computer Pioneers*. IEEE computer society. Recuperado julio 28, 2021, de <https://history.computer.org/pioneers/cooley.html>

López Monfort, J. J. (2015, diciembre 11). *Escala Mel*. Universitat Politècnica de València. Recuperado mayo 13, 2022, de <https://www.youtube.com/watch?v=6LHSzer-gs4>

Manzano, J. (n.d.). *El declive de las abejas Apis mellifera*. Ecocolmena innovación social en apicultura. Recuperado Junio 28, 2021, de <https://www.ecocolmena.org/el-declive-de-las-abejas-melliferas-y-polinizadores-silvestres/#h-ccd-colapso-de-las-colmenas>

McLuckie, C. (2012, junio 28). *Google Compute Engine: computación sin límites*. Esto es Google. Recuperado mayo 20, 2022, de <https://latam.googleblog.com/2012/06/google-compute-engine-computacion-sin.html>

National Institute on Deafness and Other Communication Disorders. (2022, junio 14). *¿Cómo oímos? Estructura del oído y del nervio auditivo | NIDCD*. NIDCD. Recuperado diciembre 8, 2022, de <https://www.nidcd.nih.gov/es/espanol/como-oimos>

- Neumann, S. (2020, agosto 6). *App móvil usa Inteligencia Artificial para “escuchar” a las abejas y detectar si están enfermas*. Diario Financiero. Recuperado mayo 20, 2022, de <https://www.df.cl/df-lab/innovacion-y-startups/app-movil-usa-inteligencia-artificial-para-escuchar-a-las-abejas-y>
- Nolasco, I., Terenzi, A., Orcioni, S., Bear, H., & Benetos, E. (2018, noviembre). *AUDIO-BASED IDENTIFICATION OF BEEHIVE STATES*. https://www.researchgate.net/publication/328997639_Audio-based_identification_of_beehive_states
- Nolasco, I. & Benetos, E., “To bee or not to bee: Investigating machine learning approaches to beehive sound recognition,” en Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE), 2018, submitted.
- Orellana, F. (2021, noviembre 26). *¿Qué es el espectrograma? Usos en el análisis musical | UNIR*. UNIR Ecuador. Recuperado mayo 16, 2022, de <https://ecuador.unir.net/actualidad-unir/espectrograma/>
- Paz, M., Friedrich, G., & Galasso, C. (2020). Procesamiento de señal visualizado sobre un espectrograma. En *Revista elektron* (Vol. 4, pp. 35-39). <http://elektron.fi.uba.ar/index.php/elektron/article/view/90/194>
- Pérez, N. (2006). 2.7 Diagnóstico de la varroosis. En *Control alternativo de Varroa destructor Anderson & Trueman utilizando panales zanganeros sintéticos* (p. 8). Universidad Austral de Chile. <http://cybertesis.uach.cl/tesis/uach/2006/fap438c/doc/fap438c.pdf>
- Preguntas frecuentes de Google Compute Engine | Documentación de Compute Engine. (2022, Marzo 11). Google Cloud. Recuperado mayo 20, 2022, de <https://cloud.google.com/compute/docs/faq?hl=es-419>
- Qandour, A., Ahmad, I., Habibi, D., & Leppard, M. (2014). REMOTE BEEHIVE MONITORING USING ACOUSTIC SIGNALS. En (Vol. 42, pp. 204-209). Acoustics Australia. <https://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1505&context=ecuworkspost2013>
- Qandour, A., Ahmad, I., Habibi, D., & Leppard, M. (2014). Remote Beehive Monitoring Using Acoustic Signals. En (Vol. 42, pp. 204-209). Acoustics Australia. https://www.researchgate.net/publication/275517604_Remote_Beehive_Monitoring_using_Acoustic_Signals
- Quezada, C. (2020). 2.1.2.3. Capa fully connected. En *EVALUACIÓN DE LA INCORPORACIÓN DE CARACTERÍSTICAS EXTRAÍDAS POR MÉTODOS CONVENCIONALES EN REDES*

- NEURONALES CONVOLUCIONALES PARA EL PROBLEMA DE CLASIFICACIÓN DE CIFAR10 (tesis pregrado) (p. 11). UNIVERSIDAD DE CHILE. <http://repositorio.uchile.cl/bitstream/handle/2250/178226/Evaluacion-de-la-incorporacion-de-caracteristicas-extraidas-por-metodos-convencionales-en-redes-neuronales-convolucionales.pdf?sequence=1&isAllowed=y>
- Rico, S., Escribano, B., & Escribano, A. (2013, enero). *CONSECUENCIAS PRODUCTIVAS Y ECONÓMICAS DE LA PRESENCIA DEL ÁCARO VARROA SP. EN LAS EXPLOTACIONES APÍCOLAS*. Recuperado junio 20, 2021, de <https://es.slideshare.net/ajescc/consecuencias-productivas-y-economicas-de-la-presencia-del-acaro-varroa-en-las-explotaciones-apicolas>
- Rincón Llorente, C. (2007). 1.1.1.16 MFCC. En *Diseño, implementación y evaluación de técnicas de identificación de emociones a través de la voz (Proyecto fin de carrera)*. (p. 45-47). UNIVERSIDAD POLITÉCNICA DE MADRID. Recuperado mayo 09, 2022, de <http://lorien.die.upm.es/barra/pfcs/2007-carmenr/docs/proyecto.pdf>
- Robles, A., Saucedo, T., González, E., & Galván, C. (2017). Frequency Analysis of Honey Bee Buzz for Automatic Recognition of Health Status: A Preliminary Study. En *Frequency Analysis of Honey Bee Buzz for Automatic Recognition of Health Status: A Preliminary Study. En Research in Computing Science* (Vol. 142, pp. 89-98). https://www.rcs.cic.ipn.mx/2017_142/Frequency%20Analysis%20of%20Honey%20Bee%20Buzz%20for%20Automatic%20Recognition%20of%20Health%20Status.pdf
- Ruiz Vargas, V. H. (2018). 2.3.3.5. Coeficientes Cepstrales de Frecuencia Mel (MFCC). En *EVALUACIÓN DE MÉTODOS PARA EL RECONOCIMIENTO DE COMANDOS DE VOZ PARA PERSONAS CON DISLEXIA (Tesis de pregrado)*. (pp. 32-35). Universidad Señor de Sipán. <https://repositorio.uss.edu.pe/bitstream/handle/20.500.12802/5250/Ruiz%20Vargas.pdf?sequence=1&isAllowed=y>
- Stancich, E. (2020, abril 24). *EL DECLIVE DE LAS POBLACIONES DE ABEJAS*. PUNTO VERDE BLOG. Recuperado junio 28, 2021, de <https://puntoverdeblog.net/2020/04/24/el-declive-de-las-poblaciones-de-abejas/>
- Team Bootstrap. (n.d.). Bootstrap · The most popular HTML, CSS, and JS library in the world. Recuperado noviembre 19, 2022, de <https://getbootstrap.com/>
- Terenzi, A., Cecchi, S., & Spinsante, S. (2020, octubre 31). On the Importance of the Sound Emitted by Honey Bee Hives. *veterinary sciences*, 7(4), 168. Recuperado enero 25, 2022, de <https://www.mdpi.com/2306-7381/7/4/168>
- You, E. (n.d.). *What is Vue?* Vue.js. Recuperado noviembre 17, 2022, de <https://vuejs.org/guide/introduction.html#what-is-vue>