DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

SOUND ANALYSIS FOR INDOOR COMPATIBLE EVENT DETECTION WITH DEEP LEARNING

by
Halil Özgen DİNDAR

December, 2022
İZMİR

SOUND ANALYSIS FOR INDOOR COMPATIBLE EVENT DETECTION WITH DEEP LEARNING

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Master of
Science in Computer Engineering

by
Halil Özgen DİNDAR

December, 2022
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"SOUND ANALYSIS FOR INDOOR COMPATIBLE EVENT DETECTION WITH DEEP LEARNING"** completed by **HALİL ÖZGEN DİNDAR** under supervision of **ASSOC. PROF. DR. GÖKHAN DALKILIÇ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Gökhan DALKILIÇ

Supervisor

Assoc. Prof. Dr. Ömer AYDIN

(Jury Member)

Asst. Prof. Dr. Yunus DOĞAN

(Jury Member)

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

I would like to thank to my supervisor Assoc. Prof. Dr. Gökhan Dalkılıç for his guidance and help throughout this work.

Halil Özgen DİNDAR

SOUND ANALYSIS FOR INDOOR COMPATIBLE EVENT DETECTION WITH DEEP LEARNING

ABSTRACT

Smart systems analyze acquired data from the environment and generate results that make the user's life easier. Smart systems improve the quality of life for users by analyzing data collected from the environment. The scope of application and demand for smart systems expands daily. The success of smart systems depends on the quality and quantity of the data used for analysis. For this reason, sound data is becoming increasingly popular as a result of its accessibility and a vast number of samples. In our work, we aim to build a smart system that is affordable, mobile, and easily accessible. We transformed sound data into spectrograms and used them to train a convolutional neural network. Three distinct approaches (image combination, multi-input, and transfer learning) were used during training. The highest success rate is achieved by the transfer learning approach with 87.26 percent. From the ESC 50 dataset, eight sounds are selected that are suitable for indoor environments. In order to assess the trained models' performance in the real world, we also use recordings of real-life examples. Even though the overall results are not high as the ESC dataset test, the system achieved almost 100 percent for some sound types. For this purpose, a new smartphone prototype application is developed to record and analyze sound.

Keywords: Smart home, internet of things, sound analysis, sensor, transfer learning

KAPALI ORTAMA UYGUN AKTİVİTE TESPİTİ İÇİN DERİN ÖĞRENME İLE SES ANALİZİ

ÖZ

Akıllı sistemler çevreden topladıkları verileri analiz ederek kullanıcının hayatını kolaylaştıran sonuçlar üretir. Akıllı sistemlere olan ihtiyaç ve kullanım alanı gün geçtikçe artmaktadır. Akıllı sistemlerin analiz etmesi için gerekli olan verinin niceliği ve niteliği ise analizin başarısını etkilemektedir. Bu sebeple ulaşılabilirliği ve örnek sayısının fazla olması ses verisinin popülerliğini arttırmıştır. Biz bu çalışmamızda uygun fiyatlı, kolay erişilebilir ve taşınabilir bir akıllı ortam yaratmayı amaçladık. Analiz için ses verisini spektrogramlara çevirerek elde edilen görüntüleri evrişimli sinir ağını eğitmek için kullandık. Eğitim için resim birleştirme, çoklu girdi ve öğrenme aktarımı olmak üzere üç farklı yöntem kullanıldı. En başarılı sonuçlar yüzde 87,26 başarı ile transfer öğrenme yöntemiyle elde edildi. Veri seti olarak ise ESC 50 veri setinden alınmış sekiz farklı kapalı ortama uygun ses seçilmiştir. Elde edilen sonuçların teoride kalmaması için eğitilen modeller ev ortamında kaydedilmiş ses verileri ile test edildi. Test sonuçlarının ortalaması, ESC veri seti ile elde edilen test sonucundan düşük olsa da bazı ses tiplerinde yüzde yüze varan başarı elde edilmiştir. Bu amaçla kayıt ve analiz işlemlerini yapabilmesi için akıllı telefon uygulaması prototipi geliştirildi.

Anahtar kelimeler: Akıllı ev, nesnelerin interneti, ses analizi, sensör, transfer öğrenme

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER 1
INTRODUCTION

The incorporation of technology into our lives has resulted in the creation of a vast number of new application areas that make our day-to-day lives easier. As a solution for these specialized application areas, smart systems using environmental data to process, evaluate, and provide user-appropriate outcomes are developed.

Smart home applications are a subcategory of smart systems. The primary objective of smart home applications is to create a harmonious and user-friendly environment to make users' lives simpler. The user may monitor and configure the system according to their unique needs, while devices connect with one another and receive system messages. The infrastructure for smart home applications may be incorporated or built subsequently. Due to the need for various IoT and embedded home devices, the lack of mobility of these systems is also a significant drawback. The objective of this project is to create an inexpensive and transportable smart home environment. Future enhancements will also include the creation of a customized smart home environment in which the user may specify bespoke actions for event detection.

In order to detect activity, smart home applications evaluate data collected from several IoT devices. For visual data, a camera can be employed, a microphone for audio data, and for other measurements, devices like a barometer, a thermometer, and motion sensors can be utilized.

The event detection success rates are high when camera feeds are analyzed due to the abundance of meaningful data. However, placing cameras in the home may compromise the privacy of people's private lives. Moreover, regulations prohibit the use of inside cameras due to privacy concerns (Boyle et al. 2000; Boyle & Greenberg, 2005; Hudson, & Smith, 1996; Liranzo & Hayanjneh, 2017).

The effectiveness of sound analysis may be inferior to that of video feed analysis, but the privacy concern is not high as the camera. The increasing demand for smart

home voice assistants supports this claim. Even though the camera feed provides richer, more informative data, the camera cannot recognize certain events, such as a doorbell ringing or an alarm going off. In addition, microphones are inexpensive and readily available due to their incorporation into smartphones and laptops. All the aforementioned factors improve the desirability of microphones.

Gathering data from other measurement devices alone is insufficient for event detection. Consequently, numerous measurement instruments are required to achieve better results. Due to the intricacy of mathematics, creating relevant data for event detection requires sensitive measurement data from IoT sensors. Sensitive measurement devices are more costly, and this procedure requires numerous of them, reducing their affordability. In addition, large regions may require many units of the same equipment for accurate measurement in all areas, and installation locations can affect measurement quality, necessitating home-specific installations, hence increasing costs. Although there are no privacy concerns with this strategy because the data collected from sensors are meaningless, the price may become prohibitive.

After weighing the advantages and disadvantages of the aforementioned methods, the microphone is the best option due to its affordability and discretion. Therefore, a microphone was chosen as the data collection device for event detection in this study. The next phase is data analysis. We chose a technique with a high success rate for image recognition because the popularity and interest in studying this topic is growing. Because of its high success rates when applied to image-related analysis, we decided to use the convolutional neural network (CNN) algorithm in our thesis. In order to train the system, we used eight indoor-compatible sounds from the ESC50 dataset, which is one of the public popular datasets. These sounds include;

- Keyboard Typing
- Glass Breaking
- Water Pouring
- Clock Alarm
- Door Knocking
- Can Opening
- Washing Machine
- Vacuum Cleaner

1.1 Motivation

Smart home applications are expanding and providing ever-more intelligent and effective solutions. Analysis and transmission of sensor data are essential components of smart environments. However, customization and cost are two major drawbacks. Using the microphone of a smartphone, our project aims to create an affordable smart environment. The primary objective is to detect activity using sound data. If we are able to achieve this, we will be able to analyze and communicate with other IoT devices in order to create intelligent environments.

Our motivation for this project is to create an affordable, mobile, and easily accessible smart environment. This requires a system that works well in theory but also stands up to practical testing with real-world data.

1.2 Contribution

The study's main contributions can be explained in three parts;

On the basis of the success rates of the analyzed studies, we chose image conversion as a technique for sound analysis. For the visual representation of sound, spectrograms were utilized. From the ESC-50 dataset, eight distinct indoor-compatible sounds are chosen. We trained models using a convolutional neural network (CNN). For sound classification, three distinct techniques are used: image combination, multi-input, and transfer learning. For image combination, different spectrograms are combined into one image. In multi-input, full-sized spectrograms are used as input for the CNN model. Lastly, for transfer learning, state-of-the-art models' frozen weights are used to train the model.

We also published "Indoor Event Detection with Sound Data" titled paper in Innovations Intelligent Systems and Applications Conference 2021 (Dindar & Dalkılıç, 2021).

In addition, a prototype application is created to assess the system's performance in the real world. Six samples of eight sound classes are recorded in the home

environment for this purpose. Real-world records are used to evaluate trained models.

1.3 Outline of Thesis

The second chapter contains a literature review of past related works about waveforms and spectrograms, which are image representations of sound that can be analyzed and categorized.

In the third chapter sound-to-image conversion techniques, spectrograms, and image classification techniques are described.

In chapter four the results of the experiments on classification methods are explained. In addition, the system design and test results of a real-world prototype application are clarified.

The fifth chapter contains the study's conclusion and additional improvements.

CHAPTER 2
RELATED WORKS

Various sound analysis approaches have been proposed over time, but the recent popularity and high success rate of image recognition with CNN have led to the conversion of sound to images. As waveforms or spectrograms, sound data can be converted to images. A spectrogram has more information than a waveform, which influences CNN's output and workload.

2.1 Literature Review

Kim, Lee & Nam (2018) and Abdoli, Cardinal, & Lameiras Koerich (2019) studies have introduced the waveform as input for their neural network. To maximize the use of the information, six waveforms with varying frame widths are constructed in the study. Abdoli et al. (2019) further note that waveform-based systems have fewer data to examine, and thus fewer trainable parameters. A system with fewer trainable parameters is easier to train but has a lower success rate against a spectrogram.

Su, Zhang, Wang, & Madani (2019) combined chroma, mel frequency cepstral coefficients (MFCC), mel spectrogram, spectral contrast, and Tonnetz spectrograms into one image to create richer informative input for CNN. Their proposed CNN model output is composed of Dempster-Shafer (DS) evidence theory. To evaluate the success of the system UrbanSound8K dataset is used. The disadvantage of this dataset is having distinctive sounds which are easy to classify.

To use the benefits of the 2 approaches, Li et al. (2018), combined the waveform and mel spectrogram. For training waveform, they used rawnet and for mel spectrogram, they used melnet. Both models consist of five convolutional layers and the prediction results are combined with the use of DS evidence theory.

Similar to Li et al. (2018), Dong, Yin, Cong, Du, & Huang (2020) combined the 1D and 2D approaches in their work. They conducted separate experiments with the

constant q-transform (CQT), MFCC, and mel spectrogram. The result revealed that the mel spectrogram has the highest percentage of correct predictions. In addition, they added raw waveforms for the missing time domain data in the mel spectrogram. Their proposed two-stream CNN model, comprised of raw audio CNN (RACNN) and logmel CNN (LMCNN), combines their fully connected layers via addition. Following the softmax method, the prediction result is calculated. Testing on the UrbanSound8k dataset yields a success rate of 95.7%.

Boddapati, Petef, Rasmusson, & Lundberg (2017), propose to create a red, green, and blue (RGB) image from the three different representations of the same sound which are red for the mel spectrogram, green for MFCC, and blue for cross recurrence plot (CRP). For investigating the impact of sample rate, 8 kHz, 16 kHz, and 32 kHz are used in the classification. Recurrent neural networks (RNN) and CNN are combined to create convolution recurrent neural networks (CRNN) for training purposes. The results are 73%, 91%, and 93% for ESC50, ESC10, and UrbanSound8K datasets respectively.

Another study (Agrawal, Sailor, Soni, & Patil, 2017) worked on a series of processes for feature extraction over the raw sound data. They filtered the audio signal with the gammatone filter bank to apply teager energy operator (TEO) because of TEO cannot be applied directly. In their experiment, they used two different classifiers gaussian mixture model (GMM) and CNN. Spectral features preserve more data than cepstral features, therefore more suitable for convolution and pooling operations. Thus, spectral features are used for CNN whereas cepstral features are used for GMM.

Two different attention mechanisms are proposed by Zhang, Xu, Zhang, Qiao, & Cao (2019). Temporal attention focuses on characteristic parts of sound rather than blank ones. On the other hand, channel attention takes advantage of the channel-wise relationship with CNN. In their experiment, they used three distinct public datasets: DCASE-2016, ESC-10, and ESC-50. The outcomes show that the attention mechanism increases the success rate of prediction.

Apart from other approaches Sharma, Granmo, & Goodwin (2020), used four different spectrograms that are MFCC, gammatone frequency cepstral coefficients (GFCC), the CQT, and chroma for their attention based Deep Convolutional Neural Network (DCNN). The proposed DCNN model has two blocks main and attention. They also showed that using data augmentation affects the system in a positive way.

In Mushtaq, & Su (2020)'s work authors presented their new feature extraction technique based on the logarithmic scale of the mel spectrogram. For the training model, they used the power of transfer learning via DenseNet-161 and they tested their system on ESC10, ESC50, and Urban8k.

Guzhov, Raue, Hees, & Dengel (2021) focus on the image domain rather than the audio domain for feature extraction. Their proposed model is based on an attention mechanism that highlights the relevant parts.

The dataset developed by Dubey, Emmanouilidou, & Tashev (2019)'s work comprises of five-second-long sounds that are suited for those with hearing loss. They implemented transfer learning using models trained on the AuidoSet and ESC datasets. Specifically, they utilized extreme learning machines and support vector machines as classifiers and compared their results.

In their thesis (Vafeiadis et al., 2020), they intend to determine how distance from the microphone and events influence recognition accuracy. Also, compare the event detection capabilities of the 1D CNN and 2D CNN. Their conclusion is that event detection up to 6 meters is acceptable and that 2D CNN performs better than 1D CNN. Lastly, they wanted to determine if the overlapping events can be distinguished, and they discovered that the louder event is detected.

CHAPTER 3
ANALYZATION AND CLASSIFICATION OF SOUND DATA

The objective of our research was to analyze sound data for event detection. Figure 3.1 shows the stages of our work. First, we transformed sound data into spectrograms, which are two-dimensional representations of sound. The subsequent step is to classify these images; we used CNN for this. We used three distinct techniques for classification: image combination, multi-input, and transfer learning. We wanted to test our models in the real world as well. Therefore, we evaluated our trained models using real-world recordings. In the remainder of the chapter, the steps are described in detail.

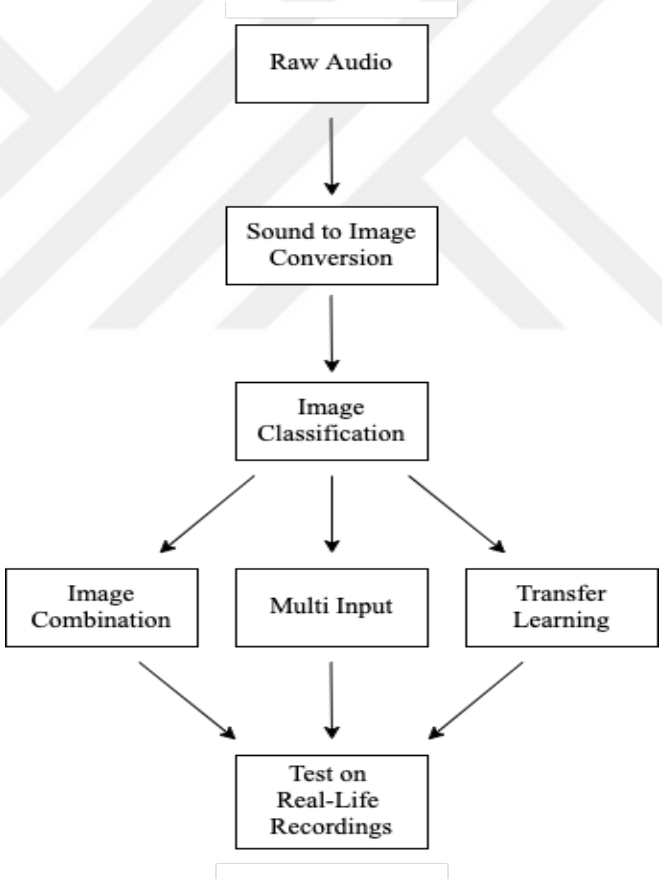

Figure 3.1 Block diagram of the system

3.1 Sound to Image Conversion

Waveform and spectrogram are two distinct visual representations of sound data. Waveform depicts the amplitude variations over time. As shown in Figure 3.2 when

the sound is louder, waves are drawn higher, the same thing goes if the area is silent there will be a flat line. The frequency is not shown in the waveform. The spectrogram depicts the frequency variations over time as well as the amplitude variations for the frequency with color. Figure 3.2 shows a waveform and spectrogram representation. Unlike the waveform, the spectrogram offers a large amount of valuable data for analysis. As described by Su et al. (2019), training the model with fewer data does not result in a large process load, but this has an effect on the success rate. Waveform-analyzing studies have a lower success rate than spectrogram-analyzing studies (Abdoli et al., 2019). In our research, we sought a high success rate, hence we employed spectrograms.

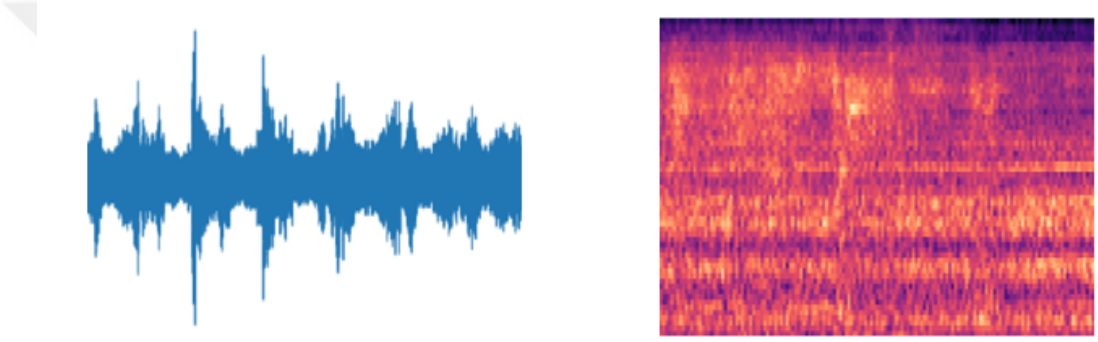

Figure 3.2 Sound image representation of waveform (left) and spectrogram (right)

3.2 Spectrograms

Mel spectrogram, Tonnetz, MFCC, chroma, and spectral contrast are the most common spectrograms for sound classification, according to our literature review (Sharma et al., 2020; Su et al., 2019). Therefore, we utilized them in our research. Each of these spectrograms focuses on and emphasizes the distinct characteristics of the sound. The specifics are detailed below.

3.2.1 Mel Spectrogram

Mel scale is a logarithmic transformation of frequency that demonstrates meaningful changes over time as opposed to linear changes. The formula for calculating the mel scale is shown in Eq. (3.1).

$$L = 10 \log\left(\frac{I}{I_o}\right) \tag{3.1}$$

Because humans respond more effectively to small amplitudes than to larger ones, the decibel scale is also a logarithmic transformation of amplitude. The mel spectrogram uses frequencies converted to the mel scale, while the decibel scale is used for coloring. As shown in Figure 3.3 x-axis shows the time, the y-axis shows the frequency, and the color shows the amplitude changes for the specific frequency in time.

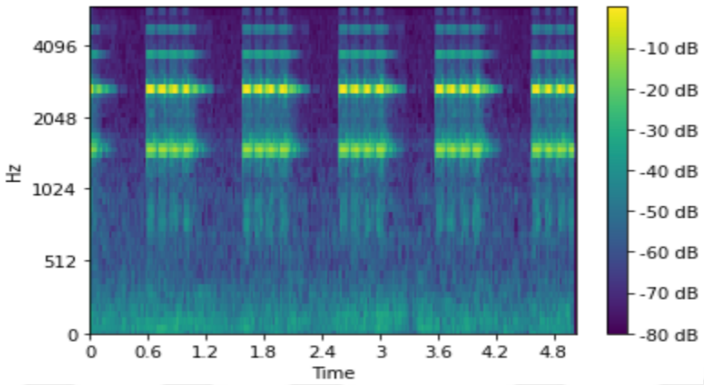

Figure 3.3 Example of mel spectrogram

3.2.2 Mel Frequency Cepstral Coefficients

The mel scale is also used by MFCC, as described in the section on the mel spectrogram. After converting hertz to the mel scale, the logarithm is applied. In the final step, discrete cosine transformation (DCT) is used. On the mel spectrogram, DCT is used to create the MFCC. Figure 3.4 shows an example of MFCC where the color shows the decibel changes over time and frequency.

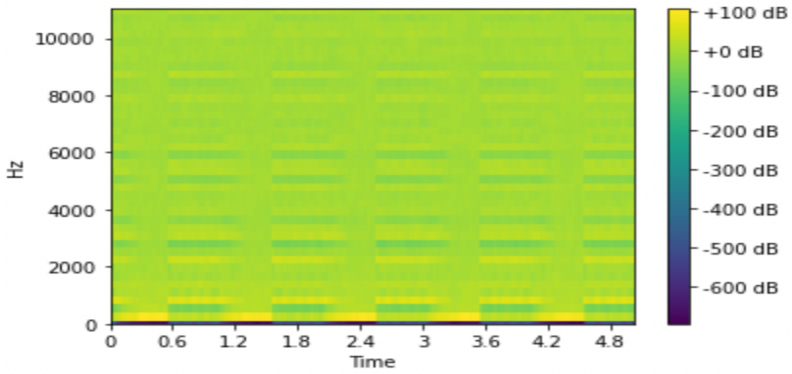

Figure 3.4 Example of MFCC spectrogram

3.2.3 Tonnetz

Tonnetz, according to Harte, Sandler, & Gasser (2006), demonstrates the harmonic changes of audio signals. This spectrogram is a two-dimensional representation of pitch relationships that categorizes pitches into six distinct classes. These classes are shown in the y-axis in Figure 3.5.

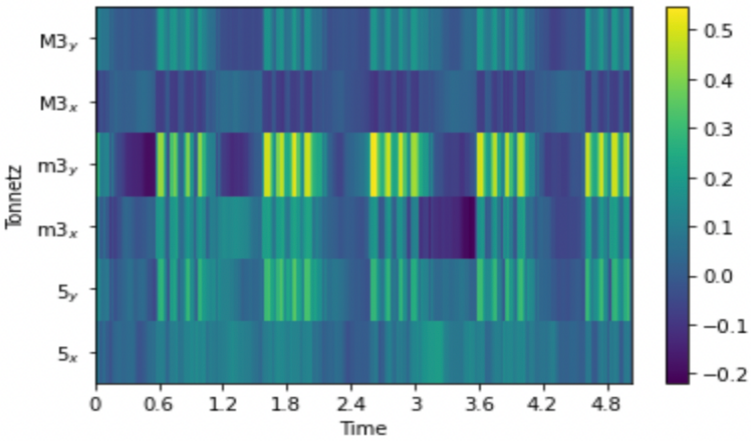

Figure 3.5 Example of Tonnetz spectrogram

3.2.4 Chroma

A chromatic scale consists of all twelve notes in ascending or descending pitch order. Similar to the Mel scale, chroma representation depicts frequency changes over time using a chroma scale. We used three distinct types of chroma spectra: short-time Fourier transform (STFT), chroma energy normalized statistics **(**CENS), and CQT.

Chroma STFT uses short-time Fourier transformation for computing chroma features. It represents pitch and signal structure with color for the high values and black for the low values (Das et al. 2020). The example of STFT shown in Figure 3.6 y-axis shows the pitch classes, and the color shows the signal structure for the frequency over the chroma scale.

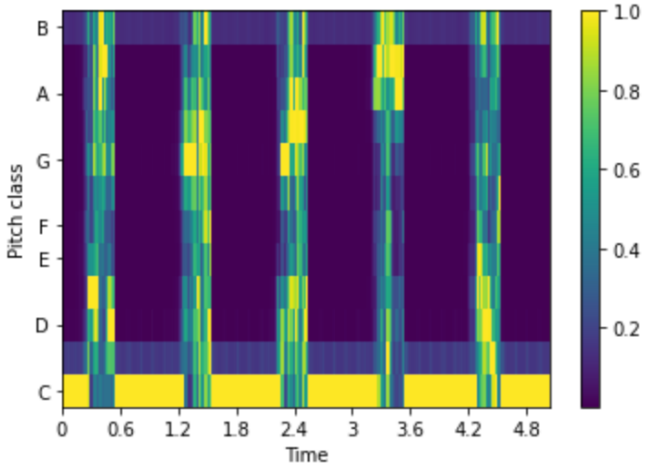

Figure 3.6 Example of chroma STFT

The idea of CENS is to take statistics over large windows and smooths the deviation of audio features. Chroma CENS is generally used for audio matching and similarity tasks. Figure 3.8 shows the CENS example, y-axis shows the pitch classes over time, and because of the deviation of audio features, the image looks blurry.

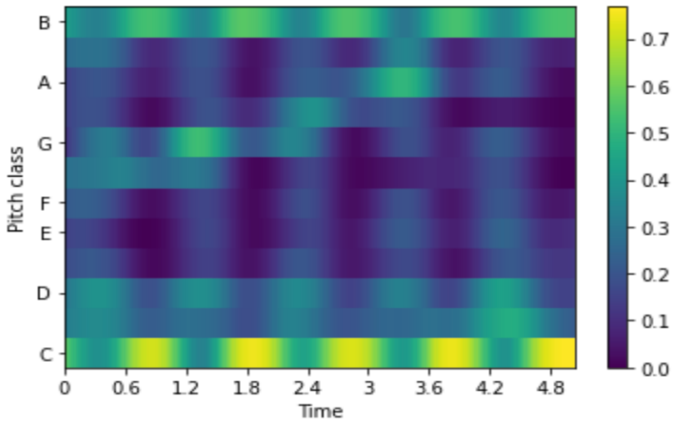

Figure 3.7 Example of chroma CENS

According to Muller (2015), chroma CQT is computing chroma features using the constant q-transform algorithm. The frequency scale is logarithmic, as opposed to linear as in STFT, and the window length varies according to frequency, with long windows for low frequencies and short windows for high frequencies (O'Hanlon, & Sandler, 2019; Ziang, Kavitha, Miyao, & Kurita, 2019). An example of the chroma CQT is shown in Figure 3.8. The y-axis shows the pitch class over time, whereas the color indicates the frequency according to the chroma scale.

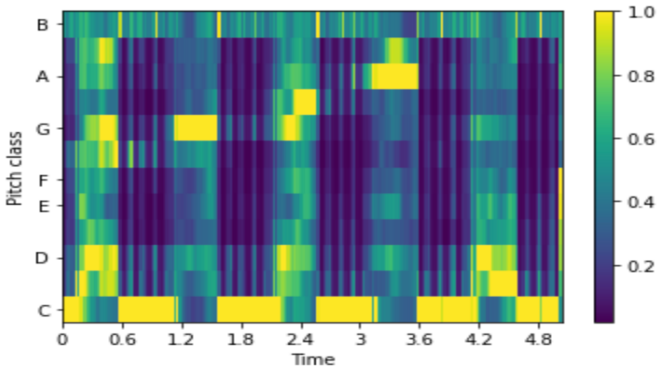

Figure 3.8 Example of chroma CQT

3.2.5 Spectral Contrast

Spectral contrast indicates the decibel difference between the spectrum's peaks and valleys. As explained in Jiang, Lu, & Zhang (2003)'s work spectral contrast is concerned with the magnitude of spectral peaks and valleys and their difference in each subband, and it represents the relative spectral characteristics. An example is shown in Figure 3.9 with the colors corresponding to the contrast value.

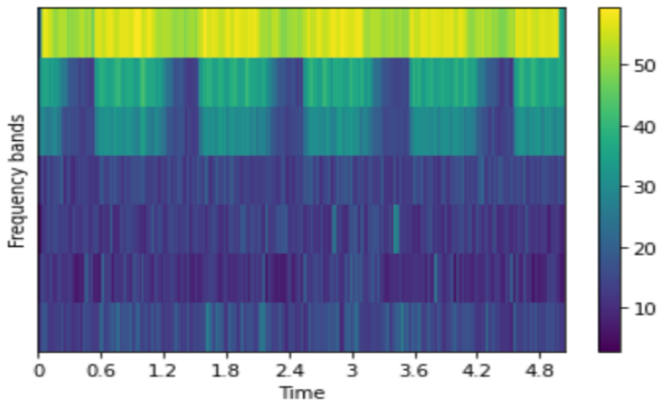

Figure 3.9 Example of spectral contrast

3.3 Image Classification

In our research, we implemented three distinct approaches: image combining, multi-input, and transfer learning. Our objective is to categorize images for event detection.

Following the creation of our input data, we analyze the converted image. We picked the convolutional neural network (CNN) technique, which is one of the deep learning (DL) algorithms, due to its ability to generate high classification success rates with hidden layers. DL is a subsection of machine learning (ML), which is a subsection of artificial intelligence (AI). Figure 3.10 shows the relationship between AI, ML, and DL.

The primary goal of artificial intelligence (AI) is to equip machines with human-like cognitive abilities that enable them to learn and comprehend. By learning and adapting, AI can generate the optimal solution for a given task rather than automating repetitive manual tasks.

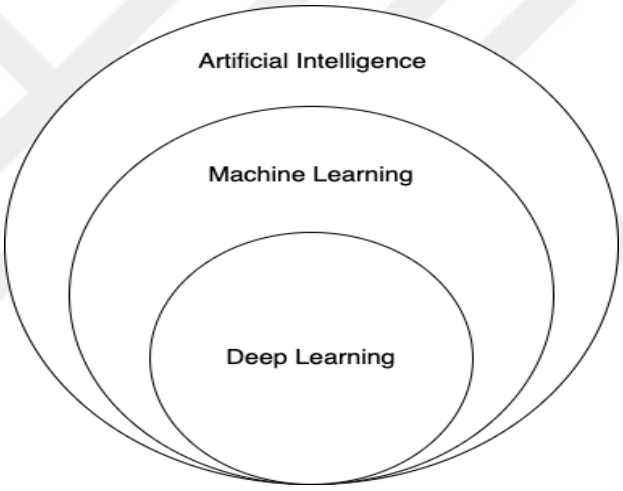

Figure 3.10 AI, ML, and DL relationship

3.4 Machine Learning

Machine learning is a subset of artificial intelligence in which the implementation of the learning process is determined. Implementing solutions to complex problems can be time-consuming and prone to error for humans. On the contrary, ML can learn from the input and determine the optimal solution for the given task. Figure 3.11 shows the three distinct learning techniques:

- Supervised Learning
- Unsupervised Learning
- Reinforced Learning

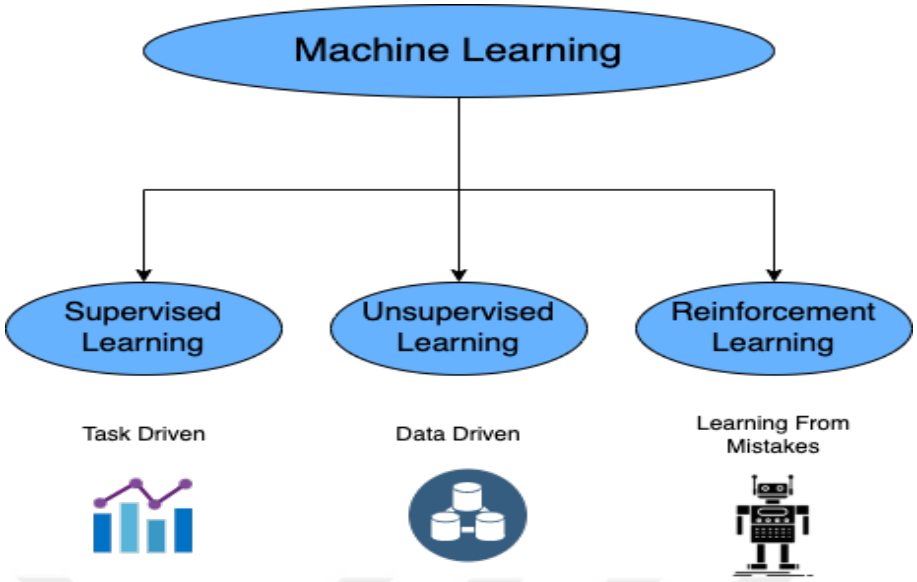

Figure 3.11 Types of learning

The most vital aspect of machine learning is the data to be processed. The success rate of the system is directly influenced by the variety and amount of data. The data is divided into two groups:

- Training
- Testing

As suggested by the name, training data is used to train the model. From the training data, the model derives its own experiences, which it then adjusts and improves.

Testing data are used to validate the performance of the model. Testing and training data should be kept separate and not used interchangeably. because the objective is to create an impartial model. The ML model evaluates the success rates of systems based on test data using the knowledge gained from training data.

3.4.1 Supervised Learning

As the name suggests, systems learn, with us being the supervisors of learning. To do that, data is labeled before being fed to the model. The advantage of supervised

learning is that you can learn better from previous experiences since the data is labeled. On the other hand, it's hard to find huge, labeled datasets. Even more challenging if one were to create such a dataset.

How well an AI performs when presented with new cases is determined by the diversity of the data. If there are insufficient examples in the training data set, the model will fail to provide accurate results. Duplicate data will distort the AI's comprehension, so training data must be balanced and purged. Data scientists must be cautious with the data used to train the model.

3.4.2 Unsupervised Learning

Unsupervised learning is the process of training a model without labeling or classifying the data. The model must discover similarities, patterns, and differences without any prior knowledge.

Due to the absence of external factors during unsupervised learning, the outcome can be unpredictable. For instance, if the system is programmed to classify dogs and cats, it may produce unexpected results for their undesirable characteristics, such as breeds. It's possible that this will result in disorder rather than order.

In unsupervised learning, there is no feedback mechanism. Because the data is unlabeled, the system cannot determine if the result is accurate. This method does not require supervision during the learning phase, but human intervention is required to validate the output variables.

To learn, unsupervised learning requires a massive amount of data, and even if unlabeled data is easier to locate than labeled data, this causes computational complexity. It also implies longer training periods.

3.4.3 Reinforcement Learning

Reinforcement learning is the third type of learning. There is a supervisor present in this learning environment, but it is not as effective as supervised learning. Because in supervised learning, the supervisor has the correct answer, whereas in reinforced learning, the supervisor, here referred to as the agent, does not. Instead, the model makes assumptions, and based on those assumptions, it provides feedback such as;

- Rewards if the assumption is correct
- Penalty if the assumption is incorrect.

The objective of reinforcement learning is to train a system to maximize the agent's cumulative reward by selecting the set of actions most likely to lead to that goal.

3.5 Artificial Neural Network

The human brain contains numerous neurons connected by synapses. To mimic the human brain's capabilities, artificial neural networks use artificial neurons, also known as nodes, that are connected to one another. These nodes transmit calculated values to other nodes in the network. The three main layers of artificial neural networks are as follows:

- Input Layer
- Hidden Layers
- Output Layer

The input layer displays the network's initial input data. Each input layer node must be connected to the nodes of the subsequent hidden layer.

The second is hidden layers, which, as the name suggests, can consist of multiple layers. Here, all relevant features, or pattern extraction, take place. According to the significance of these features or patterns, weights are assigned.

The final layer is the output layer, which displays the network's output. All output layer nodes are connected to the final node of the hidden layer. As stated previously, factors with a higher weight have a greater impact on the final outcome than those with a lower weight.

3.6 Deep Learning

Deep learning is a technique for machine learning that teaches itself from data without human intervention. Deep learning algorithms continue to process data to produce human-like results. They employ artificial neural networks for this purpose. Deep learning does not require in-process corrections because artificial neural networks repeat themselves to find the optimal solution. Deep learning identifies the low- to high-level characteristics of input by applying various filters to the hidden layers. These characteristics are represented separately by each neuron or node in the network, and each neuron has a weight that indicates the strength of its connection to the output. During network learning, the weight is calculated and adjusted.

There are two algorithms for optimizing the network's weights and producing the best possible results. Forward propagation begins at the input layer and concludes at the output layer. Each node's weights are multiplied by the current value, and the resulting product is added to the bias value. The current value varies by network element type. If it is input, the value is the feature value; if it is a node, it uses the most recent calculated value. This calculated value is finally passed to the activation function, which generates the result. This concludes the function of the node and advances the neural network to the subsequent node.

The other is backpropagation, which modifies the weights of the neural network based on the calculated loss value at the epoch. Adjusting the weight reduces the loss value and generalizes the network, thereby enhancing its reliability. Deep learning is gaining popularity, and its benefits can be summarized as follows:

- Deep learning can reveal human-unfindable properties, relations, and patterns when working with large datasets.

- Since feature extraction is performed by the network, the human factor and human errors should be minimized. In numerous instances, DL feature extraction outperforms human feature extraction.
- Using parallel computation, thousands of computations can be performed in a short amount of time. Furthermore, the weight system can be used repeatedly by saving and reusing the weights.

3.7 Convolutional Neural Network

The high success rate of image classification makes CNN one of the most popular deep neural networks. CNN divides images into smaller parts with certain features for the system to facilitate processing and then analyzes these aspects to generate predictions without losing data. CNN employs a unique technique known as convolution, which generates distinct images by applying filters. The dot-product operation is used to multiply a filter by a portion of an input that has the same size as the filter.

Pooling layers summarize the region's characteristics to reduce the network's computational load. The activation function computes the output value from the input's weighted sum. The choice of activation function depends on the task at hand. Rectified Linear Activation (ReLu), Logistic (Sigmoid), and Hyperbolic Tangent (Tanh) are often employed activation functions.

3.7.1 Convolution Layer

This is the most fundamental component of CNN. This layer's purpose, similar to that of the human brain, is to detect the image's distinguishing features. Using filters, the layer shows the lower and higher properties of the image. To detect these properties, the entire image undergoes a filtering step. The main goal of this application is to locate a given feature class anywhere in the image.

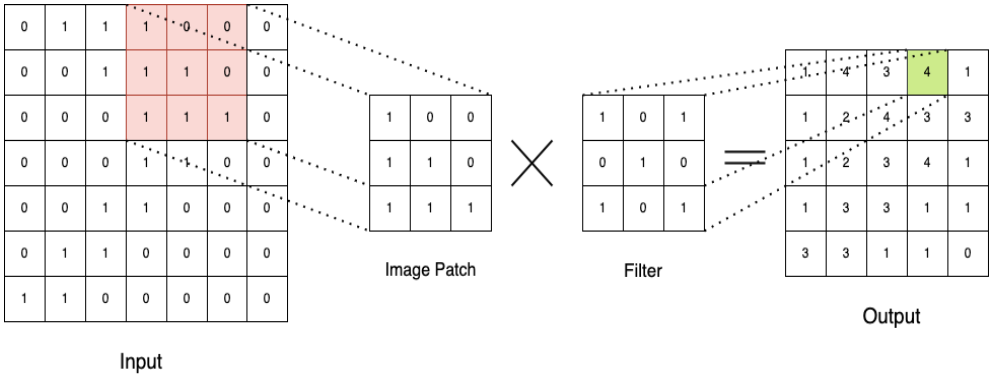

Figure 3.12 Application of filter in convolutional neural network

The application of the filter is shown in Figure 3.12. During calculations, the supplied filter size, which is smaller than the actual image, moves over the image. Figure 3.12 demonstrates that a dot product is applied between the filter and the image. When the filter is moved across the image, the procedure is complete. This process is done n times in this layer, where n is the number of filters.

3.7.2 Activation Function

An activation function specifies how the weighted sum of the input is converted into output by a node or nodes in a network layer.

3.7.2.1. Rectified Linear Activation (ReLu)

It is a straightforward activation function. The function essentially returns itself if the value is positive and zero otherwise.

3.7.2.2. Sigmoid (Logistic) Function

The sigmoid activation function is also known as the logistic function. The function accepts any real number as input and returns values between 0 and 1. Output values are closer to 1.0 for larger (positive) input values and closer to 0.0 for smaller (negative) input values.

3.7.2.3. Hyperbolic Tangent (Tanh)

The hyperbolic tangent activation function, also known as Tanh, is extremely similar to the sigmoid activation function; both have an s-shaped curve. The function accepts input and returns values between -1 and 1. In contrast to a sigmoid, the output is closer to -1 rather than 0 as the value decreases.

3.7.3 Pooling Layer

Pooling is an additional layer that typically follows convolution layers. The purpose of this layer is to reduce the calculation load by removing image values. This layer contains no learning. For the elimination rule, maximum, minimum, and average poolings are possible. According to what their names imply, max selects the value that is the highest, min selects the value that is the lowest, and average selects the value that is the average of all the values in the pool. Figure 3.13 shows the maximum pooling calculation. As described previously, this layer selects the largest value in the pooling size, which in this case is 2x2.

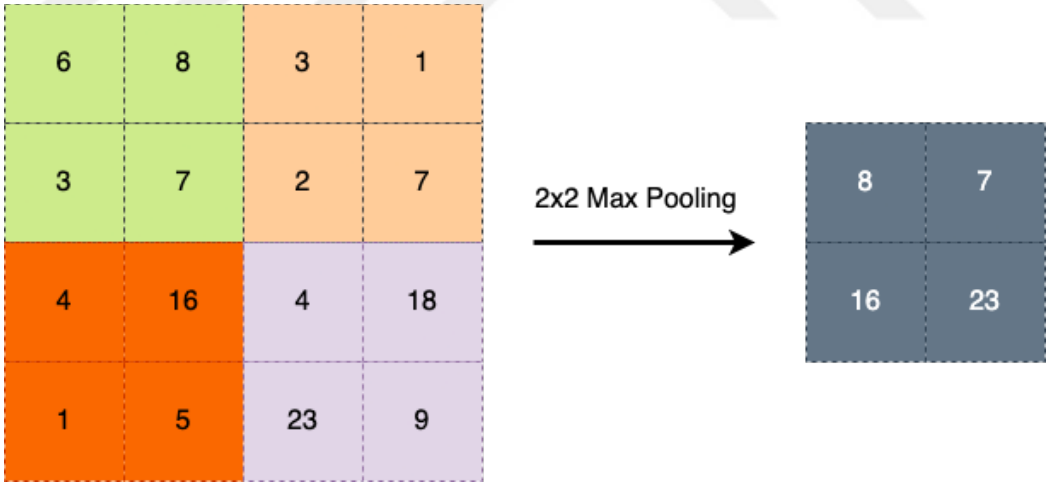

Figure 3.13 Max pooling implementation

3.7.4 Fully Connected Layer

Fully connected layers, also known as linear layers, connect each neuron input to each neuron output. In order to generate input for this layer, which is typically the

convolution or pooling layer, the previous layer was flattened. The value of the weight defines its impact on connected output.

The output layer is the final layer. This layer's node count must match the classification count. For instance, if a network is intended to classify five distinct types, the output layer node count must equal five. The classifier function determines the final value.

The softmax function transforms any n-value vector into an n-value vector whose sum is one. The result values range from 0 to 1. Because the function ensures that the resulting sum is one, it is ideally suited for multiclass classification networks.

As was previously mentioned, the sigmoid function is used for the binary classifications because it returns either 0 or 1.

3.7.5 Dropout Layer

Increasing the success rate of neural networks is not always the desired action. Overfitting occurs when a neural network memorizes too many specific properties and, as a result, cannot generalize effectively to new data. Error rates that are low during training but high during testing are often indicative of overfitting. As previously described, the network has memorized the training data excessively well and cannot perform well on testing data. To eliminate this memorization, certain nodes are eliminated. The elimination process is random. Figure 3.14 shows the neural network after dropout has been implemented. Some of the nodes were eliminated; consequently, fewer connections remain.

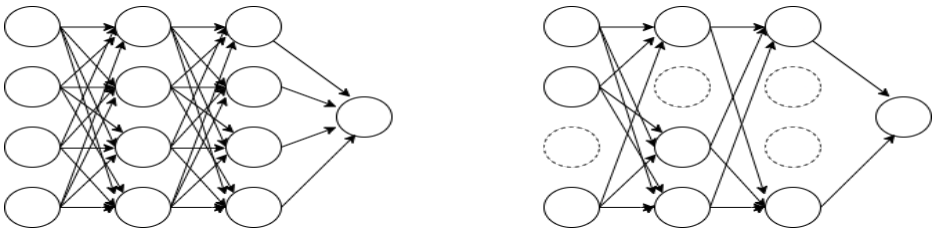

Figure 3.14 Neural network before (left) and after (right) dropout operation

3.8 Image Combination

In this method, the sound is transformed into various spectrograms. The spectrograms are then combined into a single image of varying sizes. The output image serves as the input for CNN. According to Su et al. (2019), four distinct spectrograms —the mel spectrogram, Tonnetz, chroma, and spectral contrast— are used to create a 41x85 size image called the LMC feature. Figure 3.15 shows an example of an input.

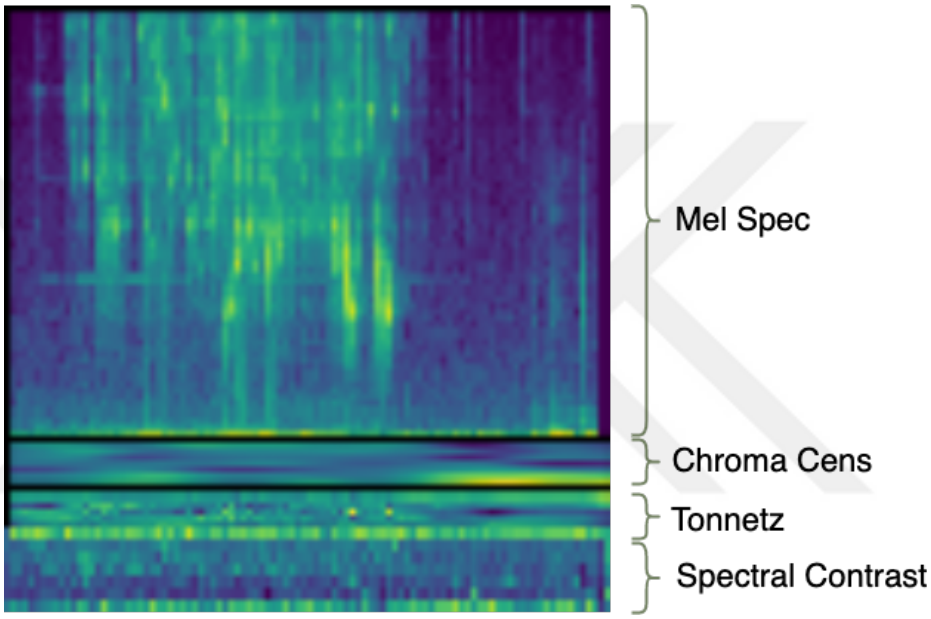

Figure 3.15 Image combination input example

As demonstrated in Su et al. (2019), the success rate of sound classification decreases as the convolutional layer of the CNN model increases. To that end, as shown in Figure 3.16, we eliminated the final convolutional layer in their model. Our model explanation is presented in four steps:

1) The initial layer uses 32 filters with a 3x3 kernel size, 2x2 strides, and ReLu as its activation function.
2) In the second layer, we repeat the first layer but with 2x2 max pooling and 2x2 strides.
3) At the third level, 64 filters are used, each with a 3x3 kernel size, 2x2 strides, and the ReLu activation function.

4) The last layer is the fully connected layer with 1024 units. ReLu is the activation function. 0.5 rate dropout is added in order to reduce overfitting.

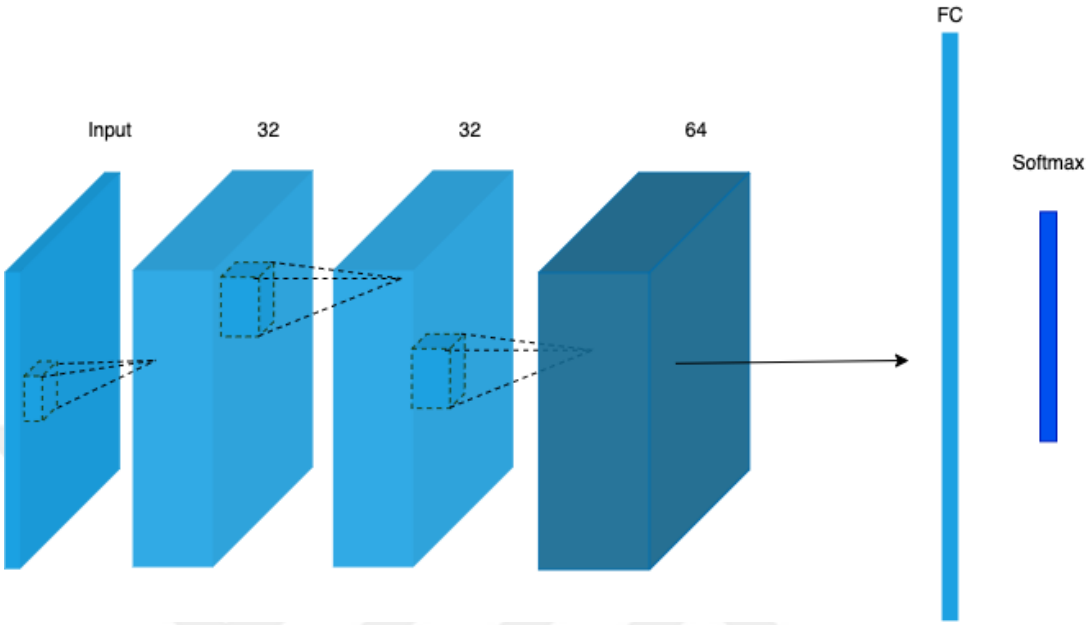

Figure 3.16 Proposed model design

3.9 Multi-Input

Multiple spectrograms are used as input to the CNN model in this approach as well, but instead of scaling the images down, the full-size images are used. The objective is to maximize the effectiveness of the spectrogram without cropping it. Controversially, using uncropped images as input raises the number of trainable parameters and consequently the computing load. Due to the extensive training time, we were compelled to limit ourselves to three spectrograms. The mel spectrogram Figure 3.3, MFCC Figure 3.4, and Tonnetz Figure 3.5 are all inputs that are fed into the system simultaneously.

3.10 Transfer Learning

Transfer learning is sharing the experience gathered while training the model. When a neural network is trained, weights are calculated between layers, and as the system learns, the weights are updated. Transfer learning in a neural network involves storing the calculated weights and using them for similar problems. In this

method, state-of-the-art algorithms for image classification, such as ResNet, inception, DenseNet, and VGGNet, were used for transfer learning.

3.10.1 Residual Network

The Residual Network (ResNet) was the first neural network we used for transfer learning. According to He, Zhang (2016), the error rate of a network increases as the number of layers increases. ResNet has residual blocks that skip some of the layers to address this issue. Various ResNet implementations, such as ResNet50, ResNet101, and ResNet152, are designated by their layer counts.

3.10.2 Inception

CNN models typically employed a constant filter size in their convolutional layers. In contrast, inception's architecture utilizes filters of varying sizes to make the most accurate predictions. This network accomplishes this through the use of parallel processing to extract feature maps simultaneously without requiring additional computational resources.

3.10.3 Visual Geometry Group Network

The Visual Geometry Group Network (VGGNet) is another architecture for transferring learning. This network requires a static input size of 224 x 224 and uses 3 x 3 kernel-sized filters in succession.

3.10.4 DenseNet

The last neural network we used for transfer learning in this study was DenseNet. The main difference from other neural networks is that in DenseNet, for each layer, the feature maps of all the preceding layers are used as inputs, and its own feature maps are used as inputs for each subsequent layer (Huang, Liu, Van Der Maaten & Weinberger, 2017). With the usage of preceding feature maps, DenseNet lowers the computational and memory load. DenseNet has different versions, which are named after the calculations of the layer count.

CHAPTER 4
EXPERIMENT AND ANALYSIS

Urban8k (Salamon, Jacoby, & Bello, 2014) and ESC (Piczak, 2015) datasets are the most popular public ones for environmental sound classification. Our study is about detecting indoor events, so we eliminated the Urban8k data set, which consists mostly of outdoor sounds. The ESC50 dataset involves fifty different sounds. We picked eight sounds from ESC50: keyboard typing, vacuum cleaner, glass breaking, washing machine, can opening, alarm clock, water pouring, and door knocking.

For training CNN models, the TensorFlow framework (Abadi et al., 2016) is used in the google collaboratory environment (Bisong, 2019). For converting audio data to spectrograms, librosa is used, and for drawing, Matplotlib (Hunter, 2007) libraries are used.

As suggested in this paper (Piczak, 2015), five-fold cross-validation is used for random and fair testing results. The accuracy results, which we shared in this thesis, are the result of average 5-fold cross-validation.

4.1 N-fold Cross Validation

The dataset is divided into two groups for classification tasks: testing and training. After learning from the training dataset, the model will be evaluated on the testing dataset. Even if data is separated at random, there is still a possibility of creating a bias. To overcome this problem, n-fold cross-validation is used. The two-fold cross-validation process consists of:

1) Divide the data set into n different, equal pieces
2) Choose one piece for testing and the rest for training.
3) Repeat the second step until n distinct train/test pairs are used.

Five-fold cross-validation is explained in Figure 4.1. The dataset was first divided into five equal parts. Then, for each iteration, a different test item is chosen. The

remainder was utilized for training. The average of the results is determined after five iterations. The average result of cross-validation was used in our work.

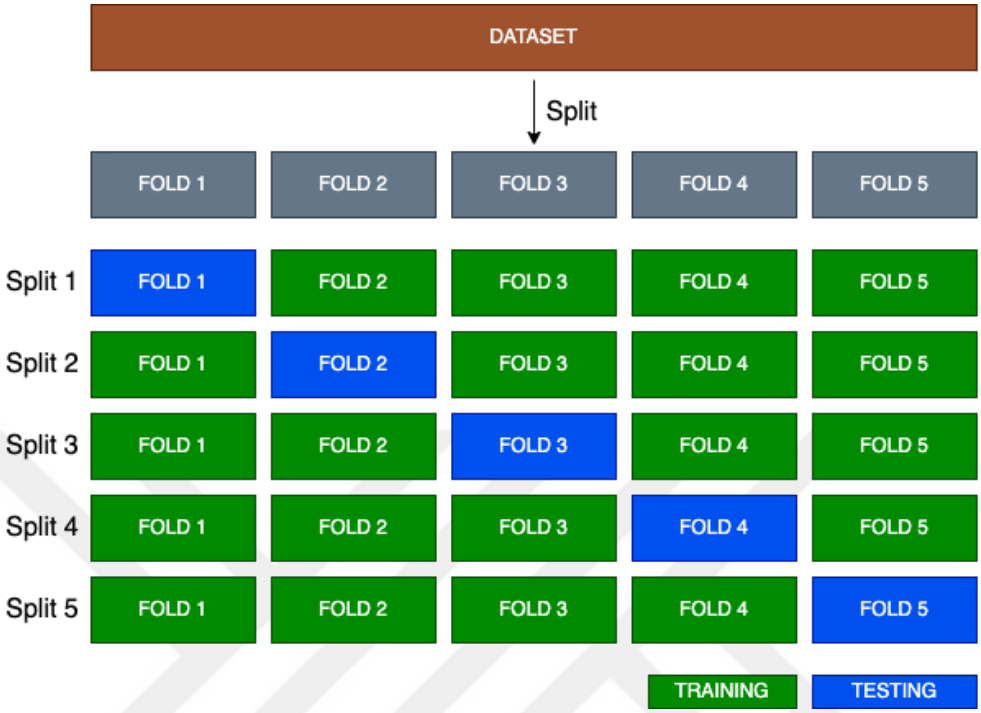

Figure 4.1 Five-fold cross-validation illustration

4.2 Image Combination Testing

To determine which spectrogram has the greatest impact on event detection, we analyzed each of the five spectrograms separately prior to analyzing the spectrograms collectively. Appendix 1 displays the results of a single spectrogram.

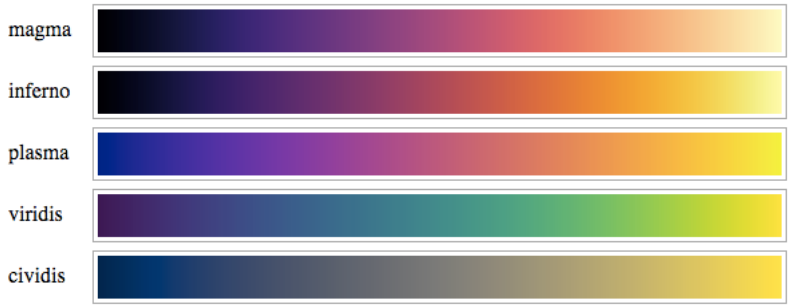

Figure 4.2 Color types for spectrogram drawing

We also tested various color palettes to determine if a change in color affected image classification. Figure 4.2 shows the various color palettes. At the conclusion

of our testing, we determined that color has no or negligible impact on event detection. We used viridis for the remainder of the study.

We desired to combine spectrograms because the results of individual spectrograms are insufficient. First, we combined them in pairs using a 1:1 ratio. The results are shown in Appendix 3. This has shown that the highest combination of two spectrograms does not produce the best results. Because the spectrograms focus on similar aspects of sound, their combination lacks rich, unique values and therefore performs poorly in event detection. We, therefore, concluded that, for event detection, analyzing different aspects is essential.

As explained in the image combination chapter, four different spectrograms are combined into one 41*85 input image. In version 0.8 of librosa (McFee et al., 2015) the library has three different chroma spectrograms. There is no information about which chroma spectrogram is used in the LMC feature set (Su et al. 2019), so we created three distinct LMC features for each chroma spectrogram. Chroma CENS for LMC1, chroma CQT for LMC2, and chroma STFT for LMC3. We tested these inputs with their LMCNet model Su et al. (2019), and also with our model, which is explained in Section 4.6. The results in Table 4.1 show us that with fewer convolutional layers, our model performed better.

In addition, we tested the other possible input combinations of spectrograms. According to our literature review, the mel spectrogram is the most common and contains the most significant information for image classification, so we used it for all the combinations. We utilized three different combinations of MFCC, chroma, spectral contrast, and Tonnetz for the remainder. As shown in Table 4.1, we utilized STFT, the most successful chroma, for further analysis. First, we substituted spectral contrast with MFCC and named it IC_MCTM. Second, we changed chroma to MFCC and renamed the variable IC_MMTS. We substituted STFT for Tonnetz for the last one and named it IC_MCMS. The outcomes are shown in Table 4.1 Despite similar outcomes, IC_MCTM has the highest success rate. We realized that spectral

peak and valley differences have the least influence on the environment. The other combinations we tried for the IC are added in Appendix 4.

Table 4.1 Model accuracy rate single input

Input	Model	Accuracy Rate	Parameter Count
LMC-1	LMCNet	74.32%	300K
LMC-2	LMCNet	72.84%	300K
LMC-3	LMCNet	77.68%	300K
IC-1	IC	82.04%	300K
IC-2	IC	80.26%	300K
IC-3	IC	82.18%	300K
IC_MCMS	IC	82.81%	300K
IC_MMTS	IC	83.43%	300K
IC_MCTM	IC	84%	300K

4.3 Multi-Input Testing

We also evaluated the multi-input technique, which uses full-size spectrograms as input instead of cropping them. We tried with two inputs first as Appendix 5 shows the results. Since the results are not high enough, we added another input and tried with three inputs. We had to limit inputs to three due to processing time and system load.

As described in the preceding chapter, the mel spectrogram is used for all possible combinations. We analyzed MFCC, chroma STFT, and Tonnetz for the remaining two slots. According to the results of the IC testing, we eliminated the spectral contrast for this test. The names of the models are derived from the spectrograms used as inputs: MI_MTC for mel spectrogram, Tonnetz, and chroma CENS; MI_MTM for mel spectrogram, Tonnetz, and MFCC; and MI_MCM for mel spectrogram, chroma CENS, and MFCC. Due to the use of three full-size inputs, the number of trainable parameters has been increased to 7M. Table 4.2 shows the results. The results of our indoor testing indicate that Tonnetz is superior to chroma STFT.

Table 4.2 MI model accuracy rates

Input	Model	Accuracy Rate	Parameter Count
MI_MTC	MI	84.06%	7M
MI_MCM	MI	82%	7M
MI_MTM	MI	84.38%	7M

4.4 Transfer Learning Testing

Transfer learning is the final testing method for model training. Four distinct neural networks are utilized for transfer learning: ResNet, VGGNet, inception, and DenseNet. Due to the system's workload, we chose a single image as input rather than multiple parallel ones. Based on past test results, IC_MCTM has the highest success rate; consequently, we chose it as an input for transfer learning.

For testing all the models, we used frozen weights minus the top layer. As explained below, we removed the top layer and added a different-sized, fully connected layer.

First, we tried the VGGNet model. At the end, 1024 fully connected layers were added. The VGGNet model only accepts inputs of size 224x224, so we altered our input accordingly.

The ResNet-152 model was utilized for the second transfer learning test. We were compelled to restrict the size of a fully connected layer to 512 due to the excessive number of trainable parameters.

The third model is the inception model. We added a 1024 fully connected layer at the end. Since the inception model does not accept inputs smaller than 75x75 pixels, we adjusted the input size to 100x100.

DenseNet is the final transfer learning model we tested. 1024 size fully connected layer added at the end. The input size remains 41x85.

Trainable parameter counts and the results are shown in Table 4.3. The transfer learning technique has the highest success rate in our study. Even though the parameter count is high (3.9M) DenseNet outperforms the other variations.

Table 4.3 Accuracy of transfer learning models based on real-world examples

Input	Model	Accuracy Rate	Parameter Count
IC_MCTM	VGGNet	84.64%	1.5M
IC_MCTM	Inception	85.99%	2.1M
IC_MCTM	ResNet	85.06%	6.2M
IC_MCTM	DenseNet	87.26%	3.9M

4.5 Real World Test

In this section, we evaluated the real-world applicability and success of the system by testing its three different approaches with real-world data. Six samples for each of eight categories, totaling forty-eight distinct sounds, are recorded from the real world and the internet for this purpose.

We used four distinct micro-electromechanical systems (MEMS) microphones: a four-facing top microphone, two front-facing bottom microphones, and a rear-facing top microphone, all of which are embedded in our smart mobile phone with the 14.2 operating system version. The backend is written with the python programming language, and Flask library version 1.0.2 handles server requests. This simple server prototype application is compatible with all platforms that support python script. We utilized a computer with a speed of 2.2 GHz.

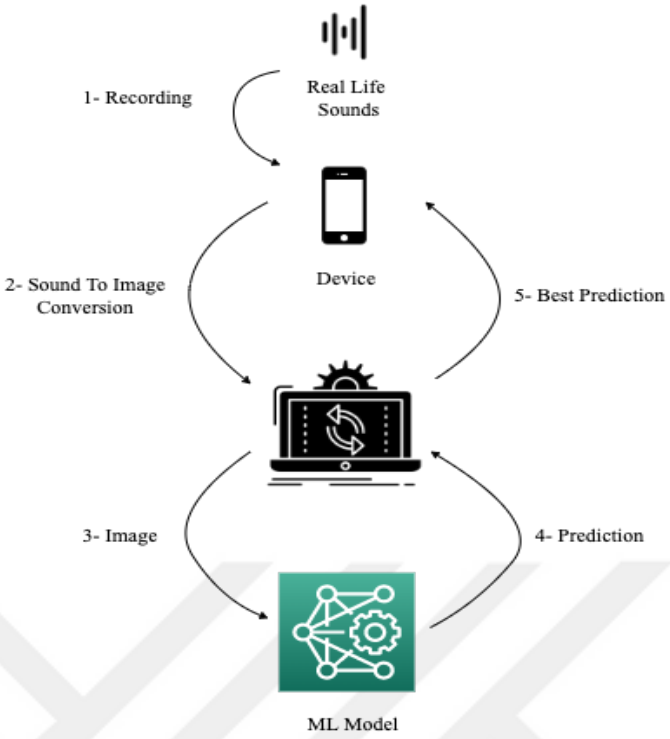

Figure 4.3 System design of the real-time testing prototype

Figure 4.3 shows the system design. In detail;

1) Sounds are recorded with a mono channel at 44.1 kHz and saved as the m4a extension. Each record is five seconds in length.
2) Five-second-long records are transmitted via the Restful API to the backend.
3) On the server side, the conversion mechanisms described in Chapter 3 are used to transform sound data into an image. The image is used as input to a CNN model.
4) Each array element can have a value between 0 and 1, with the sum of all array elements being 1.
5) The event with the highest accuracy is returned to the front end. The result is displayed to the user.

Table 4.4 Accuracy rate of real-life example predictions

Sound Type	IC	MI	DenseNet
Alarm clock	83.33%	83.33%	100%
Keyboard typing	0%	0%	33.33%
Door knocking	100%	66.66%	66.66%
Washing machine	66.66%	50%	33.33%
Vacuum cleaner	83.33%	83.33%	33.33%
Glass breaking	33.33%	66.66%	83.33%
Pouring water	0%	33.33%	0%
Can opening	83.33%	33.33%	83.33%

For the real-life examples, we used the model with the highest accuracy rate within five models, which we used for five-fold cross-validation. As seen in Table 4.4, even though the total success rate is lower than the other results that we tested on ESC dataset examples, some of the sound predictions are quite successful. For instance, models performed well with door knocking, vacuum cleaner, and alarm clock sounds, but struggled with water pouring and keyboard sounds.

We compared recorded sounds with the ESC dataset sounds and realized sound volume and clarity were not high as the ESC dataset. Our prediction is that the problem might be the recording devices' quality, which can be the reason for the low accuracy rate.

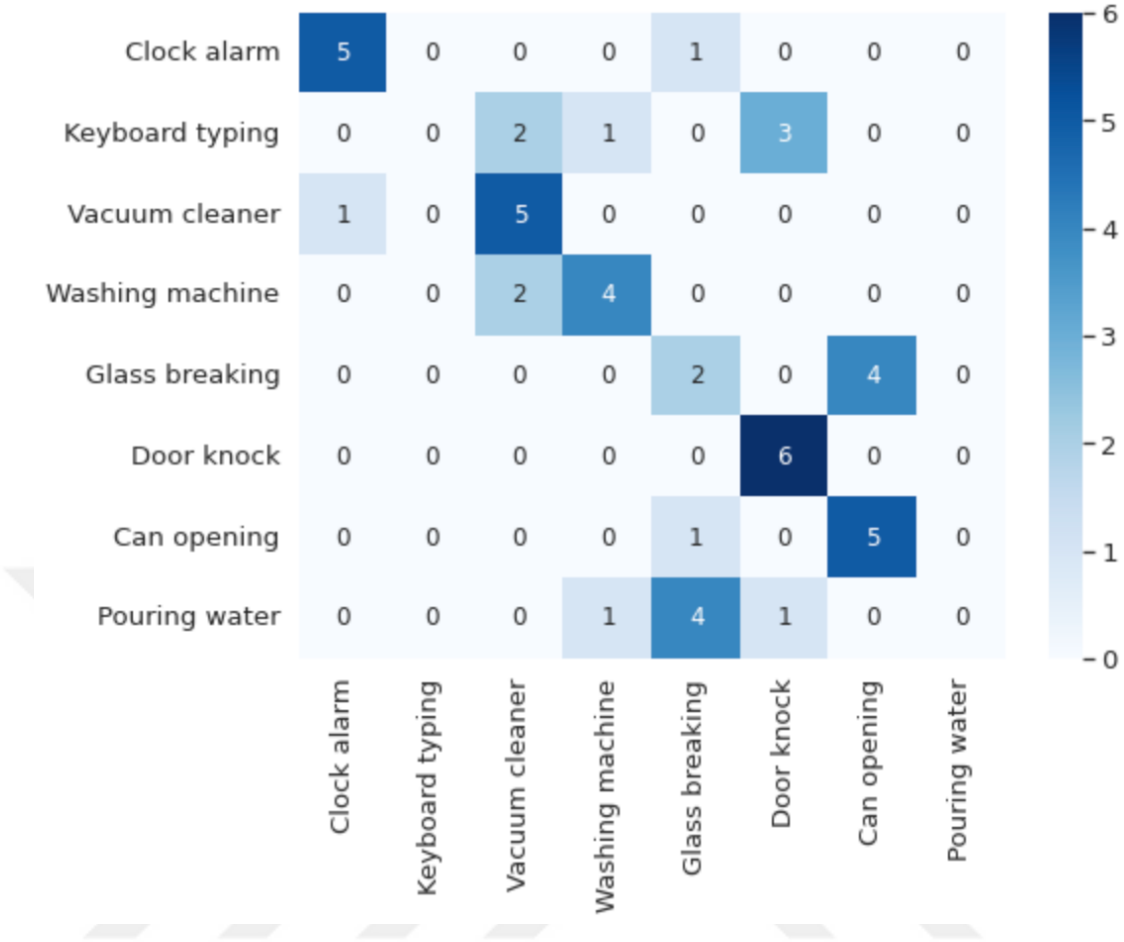

Figure 4.4 Confusion matrix of IC model

The confusion matrix indicates which sounds can be confused with others. As shown in Figure 4.4, the IC model can accurately predict alarm clock, door knocking, can opening, and vacuum cleaner events, but has difficulty predicting keyboard typing and pouring water events. We offer two different options for keyboard typing. The first keyboard is standard, while the second is a mechanical keyboard. Because the keys on mechanical keyboards sound louder, they are frequently mistaken for a doorbell. First, we were unable to determine why keyboarding is confused with vacuuming. Then, we tested a silent environment to determine its effects. The model resulted in a vacuum cleaner for the silent example because vacuum cleaners produce a constant static sound. We realized that the key sounds on a standard keyboard are too low and are therefore mistaken for vacuum cleaner sounds.

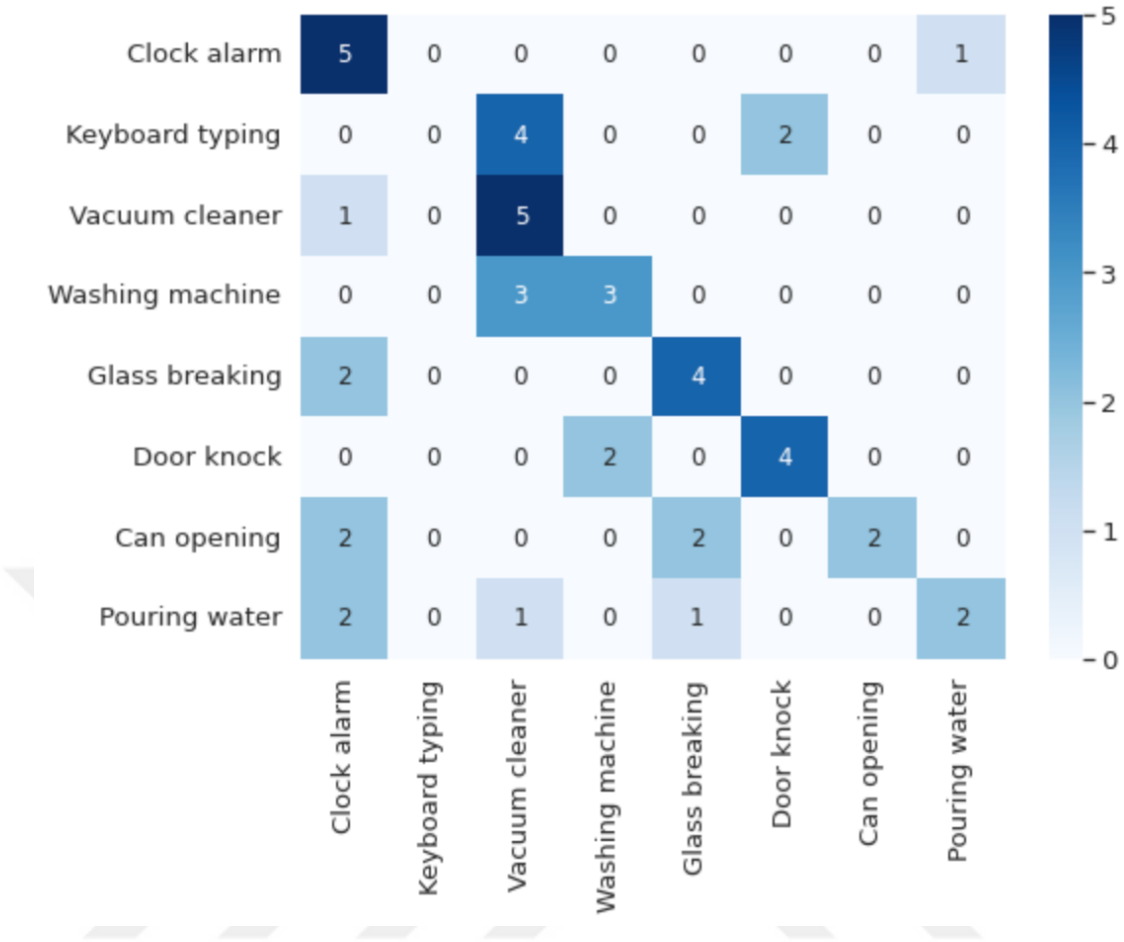

Figure 4.5 Confusion matrix of MI model

Figure 4.5 shows the performance of the MI model on real-life sounds. The MI model has the same difficulties with keyboard typing events as the IC model. The MI model has a moderate success rate for most of the sounds.

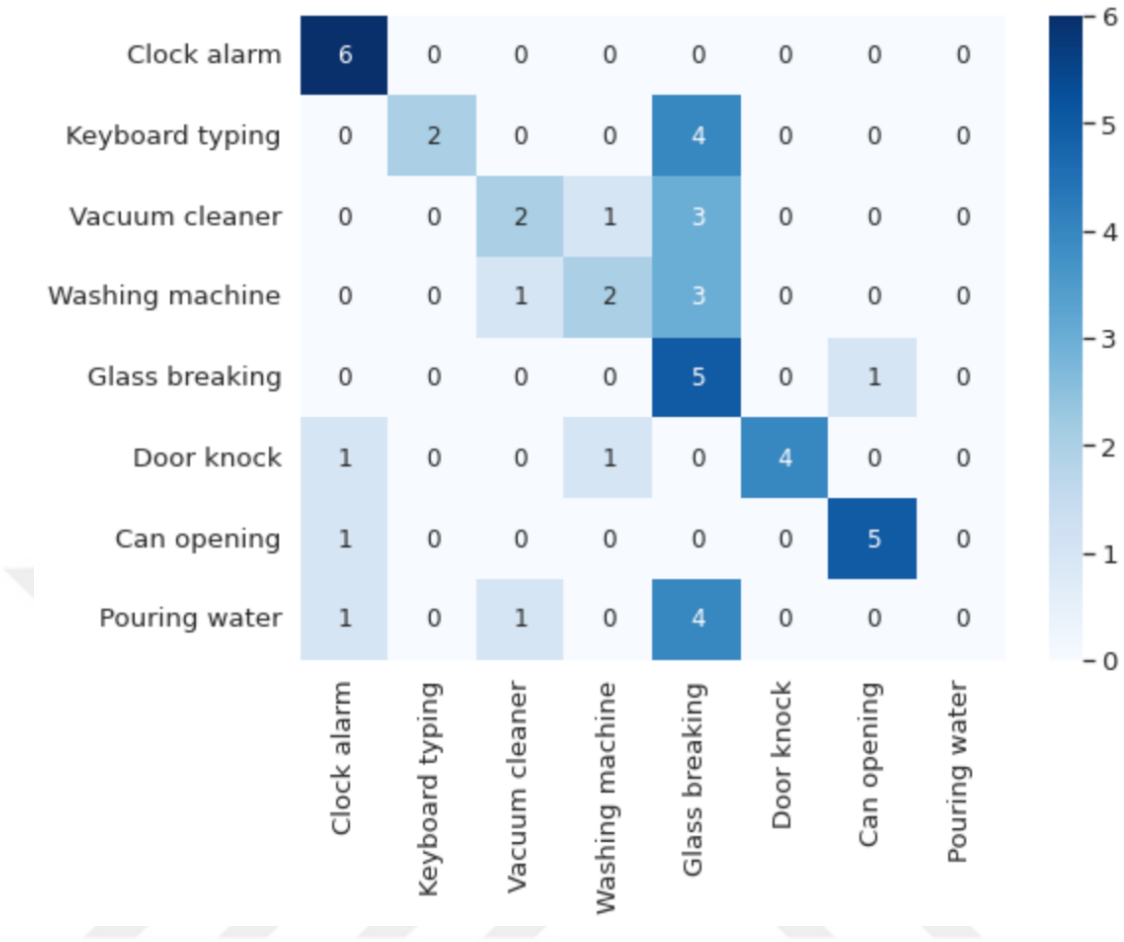

Figure 4.6 Confusion matrix of the DenseNet model

DenseNet has the highest accuracy, just as it did in the ESC dataset evaluation. Different from the other two models, this model mostly confuses other sounds with glass breaking. DenseNet is also capable of detecting some of the keystroke events which is undetectable for the other two models. Details are shown in Figure 4.6.

CHAPTER 5
CONCLUSION

In this study, a system is developed that analyzes sound data to detect indoor events. For events, we selected eight distinct sounds, including keyboard typing, door knocking, glass breaking, can opening, washing machine, vacuum cleaner, water pouring, and an alarm clock.

For sound analysis, we converted sound data into a spectrogram rather than a waveform for richer informative data.

We used various techniques to classify images. First, we independently evaluated spectrograms to determine which had the greatest impact. Then, we experimented with combining two images into one. We realized that combining the two highest spectrograms did not yield the best result, so we need a combination of spectrograms that focuses on different sound areas.

We also tested color changes in spectrograms. After our testing, we found out that color changes do not affect image classification in our case.

We evaluated the image combination (IC) technique described in Chapter 4.6. To accomplish this, we combined four distinct spectrograms into a single image input. The highest rate of precision is 84%. In addition, we tested various portion combinations, with the results presented in Appendix 4.

We also feed our convolutional neural network (CNN) multiple inputs for training, as described in section 4.7. In the multi-input (MI) technique, full-size images are fed to the model instead of cropped versions. We experimented with two and three inputs in parallel. Due to training time and workload, we had to limit the number to three. The mel spectrogram, Tonnetz, and mel frequency cepstral coefficients (MFCC) spectrograms have the highest success rate, at 84.38%. The remainder results are displayed in Appendix 5.

For our last approach, we used the transfer learning technique. We used state-of-the-art image classification models, namely VGGNet, ResNet, DenseNet, and inception, for this purpose. We achieved an 87.26% accuracy rate with 3.9 million trainable parameters on the DenseNet model.

We recorded six samples for each of the eight different sound classes in real life. These were evaluated using the most accurate model of each technique (IC, MI, and transfer learning). Some of the sound prediction values are quite high, even though the overall values are not as high as the environmental sound classification (ESC) dataset. We believe that models can perform better with an improved recording device.

Improving the precision with which events are detected is a priority for the future of work toward a fully functional smart environment. In addition to this, new sound classes should be added to improve event coverage.

Moreover, in our work, real-world examples are five seconds long, but to create intelligent environments, we must manage real-time recordings. This system must perform real-time event analysis and prediction without time restrictions.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., … Zheng, X. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. doi: 10.48550/ARXIV.1603.04467

Abdoli, S., Cardinal, P., & Lameiras Koerich, A. (2019). End-to-end environmental sound classification using a 1D convolutional neural network. *Expert Systems with Applications*, *136*, 252–263. doi: 10.1016/j.eswa.2019.06.040

Agrawal, D. M., Sailor, H. B., Soni, M. H., & Patil, H. A. (2017, August). Novel TEO-based Gammatone features for environmental sound classification. In *2017 25th European Signal Processing Conference (EUSIPCO)* (pp. 1809-1813). IEEE.

Bisong, E. (2019). Building machine learning and deep learning models on Google cloud platform: A comprehensive guide for beginners. Berkeley, CA: Apress.

Boddapati, V., Petef, A., Rasmusson, J., & Lundberg, L. (2017). Classifying environmental sounds using image recognition networks. *Procedia Computer Science*, *112*, 2048–2056. doi: 10.1016/j.procs.2017.08.250

Boyle, M., Edwards, C., & Greenberg, S. (2000). The effects of filtered video on awareness and privacy. *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work - CSCW '00*. New York, New York, USA: ACM Press.

Boyle, M., & Greenberg, S. (2005). The language of privacy: Learning from video media space analysis and design. *ACM Transactions on Computer-Human Interaction: A Publication of the Association for Computing Machinery*, *12*(2), 328–370. doi: 10.1145/1067860.1067868

Das, J. K., Ghosh, A., Pal, A. K., Dutta, S., & Chakrabarty, A. (2020). Urban sound classification using convolutional neural network and long short term memory based on multiple features. *2020 Fourth International Conference On Intelligent Computing in Data Sciences (ICDS)*. IEEE.

Dindar, H. O., & Dalkilic, G. (2021). Indoor event detection with sound data. *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE.

Dong, X., Yin, B., Cong, Y., Du, Z., & Huang, X. (2020). Environment sound event classification with a two-stream convolutional neural network. *IEEE Access:*

Practical Innovations, Open Solutions, *8*, 125714–125721. doi: 10.1109/access.2020.3007906

Dubey, H., Emmanouilidou, D., & Tashev, I. J. (2019). Cure dataset: Ladder networks for audio event classification. *2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*. IEEE.

Guzhov, A., Raue, F., Hees, J., & Dengel, A. (2021). ESResNet: Environmental sound classification based on visual domain models. *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE.

Harte, C., Sandler, M., & Gasser, M. (2006). Detecting harmonic change in musical audio. *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia - AMCMM '06*. New York, New York, USA: ACM Press.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. IEEE.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Hudson, S. E., & Smith, I. (1996). Techniques for addressing fundamental privacy and disruption tradeoffs in awareness support systems. *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work - CSCW '96*. New York, New York, USA: ACM Press.

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. doi: 10.1109/mcse.2007.55

Jiang, D.-N., Lu, L., Zhang, H.-J., Tao, J.-H., & Cai, L.-H. (2003). Music type classification by spectral contrast feature. *Proceedings. IEEE International Conference on Multimedia and Expo*, *1*, 113–116 vol.1. IEEE.

Kim, T., Lee, J., & Nam, J. (2018). Sample-level CNN architectures for music auto-tagging using raw waveforms. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Li, S., Yao, Y., Hu, J., Liu, G., Yao, X., & Hu, J. (2018). An ensemble stacked convolutional neural network model for environmental event sound recognition. *Applied Sciences (Basel, Switzerland)*, *8*(7), 1152. doi:

10.3390/app8071152

Liranzo, J., & Hayajneh, T. (2017). Security and privacy issues affecting cloud-based IP camera. *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*. IEEE.

McFee, B., Raffel, C., Liang, D., Ellis, D., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and Music Signal Analysis in Python. *Proceedings of the 14th Python in Science Conference*. SciPy.

Muller, M. (2015). *Fundamentals of music processing: Audio, analysis, algorithms, applications* (1st ed.). Cham, Switzerland: Springer International Publishing.

Mushtaq, Z., & Su, S.-F. (2020). Efficient classification of environmental sounds through multiple features aggregation and data enhancement techniques for spectrogram images. *Symmetry*, *12*(11), 1822. doi: 10.3390/sym12111822

O'Hanlon, K., & Sandler, M. B. (2019). Comparing CQT and reassignment based chroma features for template-based automatic chord recognition. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Piczak, K. J. (2015). ESC: Dataset for Environmental Sound Classification. *Proceedings of the 23rd ACM International Conference on Multimedia - MM '15*. New York, New York, USA: ACM Press.

Salamon, J., Jacoby, C., & Bello, J. P. (2014). A dataset and taxonomy for urban sound research. *Proceedings of the ACM International Conference on Multimedia - MM '14*. New York, New York, USA: ACM Press.

Sharma, J., Granmo, O.-C., & Goodwin, M. (2020). Environment sound classification using multiple feature channels and attention based deep convolutional neural network. *Interspeech 2020*. ISCA: ISCA.

Su, Y., Zhang, K., Wang, J., & Madani, K. (2019). Environment sound classification using a two-stream CNN based on decision-level fusion. *Sensors (Basel, Switzerland)*, *19*(7), 1733. doi: 10.3390/s19071733

Vafeiadis, A., Votis, K., Giakoumis, D., Tzovaras, D., Chen, L., & Hamzaoui, R. (2020). Audio content analysis for unobtrusive event detection in smart homes. *Engineering Applications of Artificial Intelligence*, *89*(103226), 103226. doi: 10.1016/j.engappai.2019.08.020

Ziang, Y. &., Kavitha, M., Subash, &., & Kurita, T. (2019). Music score estimation algorithm using octave hierarchy defined on logarithmic frequency.

Zhang, Z., Xu, S., Zhang, S., Qiao, T., & Cao, S. (2019). Learning attentive representations for environmental sound classification. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 130327–130339. doi: 10.1109/access.2019.2939495

APPENDICES

APPENDIX 1: Single spectrogram test results table

Spectrograms	Accuracy
Mel spectrogram	72.42%
MFCC	30.93%
Spectral Contrast	38.43%
Chroma CENS	29.06%
Chroma CQT	35.62%
Chroma STFT	40.31%
Tonnetz	24.68%

APPENDIX 2: Spectrogram color test result table

Color	Accuracy
Viridis	68.42%
Inferno	68.36%
Plasma	67.95%
Cividis	68.06%
Magma	68.16%

APPENDIX 3: Spectrogram combination 1:1 ratio test results

Spectrograms	Accuracy
Mel Spectrogram, Tonnetz	62.40%
Mel Spectrogram, MFCC	75.00%
Mel Spectrogram, Spectral Contrast	59.68%
Mel Spectrogram, CENS	57.81%
Mel Spectrogram, CQT	64.37%
Mel Spectrogram, STFT	65.30%
Tonnetz, MFCC	41.25%

APPENDIX 4: Spectrogram combination test results

Spectrograms	Accuracy	Total Size	Size
Mel Spectrogram	77.50%	[100:1]	[0:60, :]
Tonnetz			[60:70, :]
MFCC			[70:80, :]
CENS			[80:90, :]
Spectral Contrast			[90:, :]
Mel Spectrogram	81.56%	[100:1]	[0:60, :]
Tonnetz			[60:70 , :]
MFCC			[70:80, :]
CQT			[80:90, :]
Spectral Contrast			[90:, :]
Mel Spectrogram	79.68%	[100:2]	[0:60, :]
MFCC			[60:72, 0:1]
Tonnetz			[60:72, 1:]
STFT			[72:84, :]
Spectral Contrast			[84:, :]
Mel Spectrogram	75.31%	[100:2]	[0:60, :]
MFCC			[60:72, 0:1]
Tonnetz			[60:72, 1:]
CQT			[72:84, :]
Spectral Contrast			[84:, :]
Mel Spectrogram	71.87%	[100:2]	[0:60 , :]
MFCC			[60:72, 0:1]
Tonnetz			[60:72, 1:]
CENS			[72:84, :]
Spectral Contrast			[84:, :]
Mel Spectrogram	79.62%	[100:1]	[0:60 , :]
MFCC			[60:80]
CQT			[80:, :]

APPENDIX 5: Multi-input convolutional neural network test results

Spectrograms	Accuracy
Mel Spectrogram Spectral Contrast Chroma CQT	70.31%
Mel Spectrogram Spectral Contrast Chroma STFT	74.68%
Mel Spectrogram Spectral Contrast Chroma CENS	58.43%
Mel Spectrogram MFCC Chroma CENS	78.12%
Mel Spectrogram Tonnetz Spectral Contrast	76.72%
Mel Spectrogram Spectral Contrast	74.68%
Mel Spectrogram MFCC	80.21%
Mel Spectrogram Tonnetz	73.12%
Mel Spectrogram CENS	75.93%
Mel Spectrogram STFT	79.84%
Mel Spectrogram CQT	76.56%