République Algérienne Démocratique et Populaire Ministère de

L'Enseignement Supérieur et de la Recherche Scientifique

Université de BLIDA 1

Faculté des Sciences

Département d'Informatique

Master Thesis

An Experimental Study Of Sound Scene Classification Techniques For Smart Systems

By:

BOUKABOUS Ali Nazim	**Ingénierie des Logiciels**
BOUZIDI Sohaib Mouhcine	**Systèmes Informatique et Réseaux**

In front of a jury composed of:

Ms Hichem KAMECHE	President
Mr Mohamed HAMMOUDA	Examiner
Ms. YKHLEF Hadjer	Supervisor
Mr. YKHLEF Farid	Supervisor

2018/2019

Abstract

This thesis is dedicated to the study of Acoustic Scene Classification systems. The primary goal is to provide researchers and practitioners with guidelines that describe key steps for developing efficient scene classification systems. To this end, we have carried out two experimental case studies using a large set of sound scenes DCASE 2016 dataset. We have supported our analysis using numerous statistical tests. In the first one, we have conducted a comparative study among various systems, which were trained using 3 learning paradigms (Feed-Forward Neural Network (FNN), Support Vector Machine (SVM) and K-nearest neighbors (KNN)) on 3 sets of features (Mel Frequency Cepstral Coefficients (MFCC), MFCC+ΔMFCC, and Spectrogram). The obtained results indicate that ΔMFCCs do not have significant impact on the predictive performance. Moreover, FNN exhibits very robust and high scores compared with the other learning paradigms. In the second case study, we have tested the use of feature selection in order to reduce the computational cost of training. Our analysis shows the positive role of feature selection in this case. Specifically, we can conclude that systems that were built using 40% ΔMFCC and 60% MFCC can increase the generalization ability of FNN.

Keywords: Acoustic Scene Classification, Machine Learning, Feature Extraction, Feature Selection, Statistical Tests.

Résumé

Cette thèse est dédiée pour une étude des systèmes de classification de scènes sonore. Le premier but est de pouvoir apporter aux chercheurs et aux appliquant des guides qui décrivent les étapes essentielles pour le développement d'un système performant pour la classification des scènes. Pour cela, nous avons réalisé deux cas d'études différents en utilisant une très large base de données proposée par DCASE 2016. Pour des fins d'analyse, nous avons utilisé plusieurs méthodes de calcul de statistiques. Pour le premier cas, nous avons réalisé une étude comparative de multiples systèmes, lesquels ont été entrainés en utilisant trois paradigmes d'apprentissage (Réseau de neurones à propagation avant (FNN), Machine à vecteurs de support (SVM) et l'algorithme de k plus proches voisins (KNN)) sur trois ensembles de caractéristiques (Coefficients de fréquence de Mel (MFCC), MFCC+ ΔMFCC et Spectrogramme). Les résultats obtenus indiquent que ΔMFCC n'ont pas un grand impact pour l'amélioration des performances de prédiction. De plus, FNN est très performant et donne de très bons résultats par rapport aux autres paradigmes d'apprentissage. Pour le deuxième cas, nous avons testé l'utilité de la sélection des caractéristiques dans le but de réduire le coût d'exécution de la phase d'entrainement. Dans ce cas, nos analyses montrent l'effet positif de la sélection des caractéristiques. Nous pouvons conclure que les systèmes qui ont été fondés en utilisant 40% des ΔMFCC et 60% des MFCC peut améliorer la généralisation des FNN.

Mots clés : Classification de Scenes Sonores, Apprentissage Automatique, Extraction des Caracteristiques, Selection des Caracteristiques, Testes Statistique.

ملخص

هذه الأطروحة مخصصة لدراسة أنظمة تصنيف المشاهد الصوتية. الهدف الأول هو تزويد الباحثين والمنفذين بأدلة تصف الخطوات الأساسية لتطوير نظام قوي لتصنيف المشاهد. لهذا السبب، أدركنا دراسة حالتين مختلفتين باستخدام قاعدة بيانات كبيرة جدًا اقترحتها DCASE 2016. لأغراض تحليلية، استخدمنا عدة طرق لحساب الإحصائيات. بالنسبة للحالة الأولى، أجرينا دراسة مقارنة لأنظمة متعددة، تم تدريبها باستخدام ثلاثة نماذج تعليمية ((FNN)، (SVM)، (KNN)) على ثلاث مجموعات من الخصائص ((MFCC)، (MFCC+ΔMFCC) و(Spectrogram)). تشير النتائج التي تم الحصول عليها إلى أن ΔMFCC ليس لها تأثير كبير على تحسين أداء التنبؤ. علاوة على ذلك، FNN فعال للغاية ويعطي نتائج جيدة للغاية مقارنة بنماذج التعلم الأخرى. للحالة الثانية، اختبرنا فائدة (Feature Selection) من أجل تقليل تكلفة تنفيذ المرحلة التدريبية. في هذه الحالة، تظهر تحليلاتنا التأثير الإيجابي لهذه الأخيرة. يمكننا أن نستنتج أن الأنظمة التي تم إنشاؤها باستخدام 40 ٪ من ΔMFCC و60 ٪ MFCCيمكن أن تحسن أداء FNN.

كلمات المفاتيح: أنظمة تصنيف المشاهد الصوتية، أنظمة التصنيف، الإحصائيات.

Acknowledgements

We would first like to thank our thesis supervisors Ms YKHLEF Hadjer and Mr YKHLEF Farid of the Electronic Departement at the University of Blida 1. Their offices were always open whenever we ran into a trouble spot or had a question about our research or writing. they consistently allowed this project to be our own work, but steered us in the right the direction whenever they thought I needed it.

Thanks to the reviewers Mr Mohamed HAMMOUDA and Mr Hicham KAMECHE for the time they spent on carefully reading the thesis, for their constructive comments and valuable suggestions which have helped us to improve the quality of this manuscript.

Contents

List of Figures

List of Tables

Introduction

1. Context and problem statement

Enabling devices to make sense of their environment through the analysis of sounds is the main objective of research in **Acoustic Scene Classification (ASC)**. ASC systems perform analogous processing tasks to the human auditory system, and are part of a wider research theme linking fields such as Machine Learning, Robotics and Artificial Intelligence. ASC refers to the task of associating a semantic label to an audio stream that identifies the environment in which it has been produced. It uses computational algorithms that attempt to automatically recognize scenes using **Signal Processing** and **Machine Learning Methods**.

ASC has a major impact in a wide range of applications. We can cite: audio surveillance and noise pollution monitoring [1]. Unlike video monitoring, acoustic or audio surveillance can be advantageous in many scenarios, since sounds travel through obstacles, is not affected by lighting conditions, and capturing sound typically consumes less power. Moreover, many potentially dangerous events can only be detected at an early stage through the analysis of an audio stream. For instance, the detection of specific sound sources such as gunshots, screams, and sirens. Noise pollution is one of the topmost qualities of life issues for urban residents worldwide. Exposure to harmful levels of noise has proven effects on health such as sleep disruption, stress, hypertension, and hearing loss.

The process of classifying an acoustic scene is divided into two major steps (Figure 1.1): Feature Extraction and Machine Learning. First, **sound pre-processing step**, a sound file is decomposed into small **frames** of a certain length; then, a **feature extraction** method is applied on each frame to extract a vector of data that specifies the pattern of that exact frame, each vector of data is associated to the scene label that corresponds. Existing sound representations often include: **Mel-frequency Cepstral Coefficients** and other low-level **Spectral Descriptors** or more specialized features such as histograms of sound events or histogram of gradients learned from time-frequency representations. Second, **machine learning model** maps the extracted vector of features (sound frame) to its textual label. Finally, predictions are merged together to form the final decision based on majority vote. Many classification algorithms have been introduced in the literature, such as **Neural Networks (FNNs), Support Vector Machines (SVMs)** and more recently deep learning-based approaches. These latter are characterized by a high computational complexity and often have a large number of parameters. Diagram below show the overall mechanism of acoustic scene classification system.

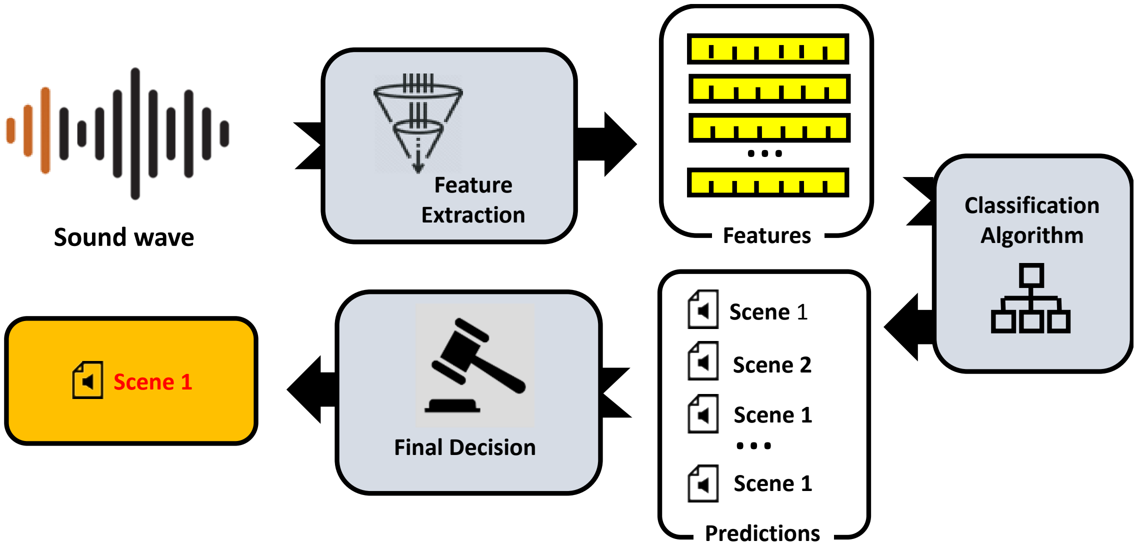


Figure 1.1. Overall mechanism of acoustic scene classification system

A possible way of improving predictions, while decreasing the computational burden, consists of including a Feature Selection step before invoking a classification algorithm. Feature selection aims at finding a compact and effective subset of features. The thrust consists of reducing the number of features while maintaining or even improving the generalization power of the learning model. Given a set of n features, one straightforward strategy consists of searching for a subset that best optimizes a criterion indicative of the generalization accuracy. This task involves evaluating $2^{n} - 2$ subsets (excluding the empty set and the entire set), which becomes intractable for moderate and large number of features [2]. To cope with this shortcoming, numerous filter-based approaches have been developed in the literature such as Mutual Information Feature Selection (MIFS) [3], Conditional MIFS (CMIFS) [4], min-Redundancy Max-Relevance (mRMR) [5], and the well-known Joint Mutual Information (JMI) [6].

2. Related work

Khunarsal, Lursinsap, and Raicharoen have conducted their works based on Environmental Sound Classification using K-Nearest Neighbor classifier and Feed Forward Neural Network by dividing their dataset into 5-Folds cross-validation [7]. They have used combination of Spectrogram, Mel Frequency Cepstral Coefficients (MFCCs), Linear Prediction Coefficients (LPCs) and matching pursuit (MP) features as feature extraction approaches with varying the window size in each time. For that, they concluded that Both k-NN and feed-forward neural network can effectively classify unstructured environmental sound. In particular, the feed-forward neural network gave the best result in their experiments. Using a neural network and K-NN, the average accuracy of the **spectrogram + LPC + MP** features combination is the best of all. They obtained an overall accuracy of 86% and 94.98% for FNN and K-NN respectively.

Shuiping, Wang & Zhenming, Tang & Shiqiang, Li. Have used Short-Time Average Zero-Crossing Rate, Short-Time Energy, Centroid of audio frequency spectrum, Sub-Band Energy and MFCC features as the characteristic parameters and designed an audio classification system based on SVM. They achieved an overall accuracy 90.43% for MFCC [8].

3. Contributions

Sound scene classification has been a subject of mounting interest in the last decade. Incredible efforts have been deployed into building such systems. The challenge consists of extracting the features that best represent the problem at the hand, and fitting the most effective classification model. To put it simply, the aim is to find the best combination of a feature extraction approach and a classification algorithm. This problem has been addressed by the research community using various methodologies. The aim of this thesis is to **derive guidelines for researchers and beginners in this field to assist them in building effective scene classification systems**. To this end, we have carried out experimental comparisons among various scene classification systems. The contribution of our work is two folds:

- **First**, we have conducted extensive experimental comparison among sound analysis methods using a large set of sound scenes DCASE 2016 dataset [9]. We have thoroughly examined 3 acoustic features (Mel Frequency Cepstral Coefficient, a combination of Mel Frequency Cepstral Coefficient with the delta coefficients, and Spectrograms) and 3 classification paradigms (Feed Forward Neural Network, K Nearest Neighbors, Support Vector Machine). We have backed our analysis and conclusions based on well-known statistical tests.

- **Second**, we have investigated the effect of the number of features on the generalization performance. Specifically, we have invoked a feature selection approach, namely JMI [6] in order to automatically determine the optimal set of extracted features. **To the extent of our knowledge, only very few attempts have considered incorporating feature selection step into the design of their systems**.

4. Structure of the thesis

This thesis is divided into 2 parts:

- In the first part, we survey the background knowledge surrounding sound representations and machine learning that are relevant to the enquiries pursued in this thesis. We present, in Chapter 1, basic notions on acoustic scene classification, starting from pre-processing to feature extraction. Then, in Chapter 2, we provide a throughout description of some relevant classification concepts.
- In the second part, we present the setup and the experimental results of our enquiries. We provide, in Chapter 3, a description the methodology that we have followed for comparing sound scene classification approaches. Finally, in Chapter 4, we present the results of the experimental comparisons, while backing our analysis and discussions based on robust statistical tests.

Fundamentals of Acoustic Scene Classification

Acoustic scene classification aims at characterizing the environment of an audio stream by selecting a semantic label for it. This process involves two primary steps: Feature Extraction, Machine Learning. Feature extraction consists of transforming the signal into a representation which maximizes the sound recognition performance of the analysis system. The acoustic features provide a numerical representation of the signal content relevant for machine learning, characterizing the signal with values which have connection to its physical properties, for example, signal energy, its distribution in frequency, and change over time. On the other hand, machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from data, the result of the learning process is known as machine learning model, this latter takes as an input a set of features extracted from a sound scene and assigns a label to it. The first part of this thesis covers state-of-the-art surrounding feature extraction techniques and machine learning models that are necessary for comprehending the ideas discussed in this thesis.

Chapter 1: Sound Representation

1.1. Introduction

Sound Analysis has become a large educated domain in our days, it gets an important attention from physicians and also other scientists trying to understand its nature and behavior in the environment, in the purpose of making it understandable in machines. It has been known that the audio sounds have multiple representative parameters (Temporal, Cepstral, Spectral, Spectrogram) which can be used for recognition.

In this Chapter, we describe some theory behind sound representation and signal processing principles that are required for perceiving this work. We begin by introducing the process of sound acquisition in Section 1.2. Then, we briefly discuss time/frequency representations, Fourier transform, and some preprocessing mechanisms in Sections 1.3 and 1.4, respectively. Finally, we provide some acoustic features that have been widely employed in the literature in Section 1.5.

1.2. Signal/Sound Acquisition

Sound is the result of a vibration that propagates as waves through a medium such as air or water. Sounds can be recorded under the form of an electric signal x.t/ where t represents the time by means of an electroacoustic transducer such as a microphone. This analog signal x.t/ can then be converted to a digital signal x(n) n being the digital signal and stored on a computer before further analysis. The necessary steps to perform this analog-digital conversion include:

- **A filtering stage**: the analog signal x.t/ is low-pass filtered in order to limit its frequency bandwidth in the interval (0; B) where B is the cut-off frequency of the low-pass filter.

- **A sampling stage**: the low-passed analog signal is then digitally sampled at a sampling rate fs D 2B to avoid the well-known frequency aliasing phenomenon.

- **A quantification stage**: the obtained digital signal is then quantized (e.g. the amplitude of the signal can only take a limited number of predefined values to preserve storage capacity).

1.3. Time/Frequency representation

Time-frequency representations such as Short Time Fourier Transform were designed mainly according to mathematical rules leading, for example, to linear-frequency scales. Human perception studies have shown that we do not perceive sound similarly in each region of the spectrum and that the resolution of the human ear also varies along the frequency axis. Therefore, non-linear-frequency scales have been introduced in an attempt to mimic human perception and provide a better way to extract information from sound signals. The Figure below represents the frequency in function of time.

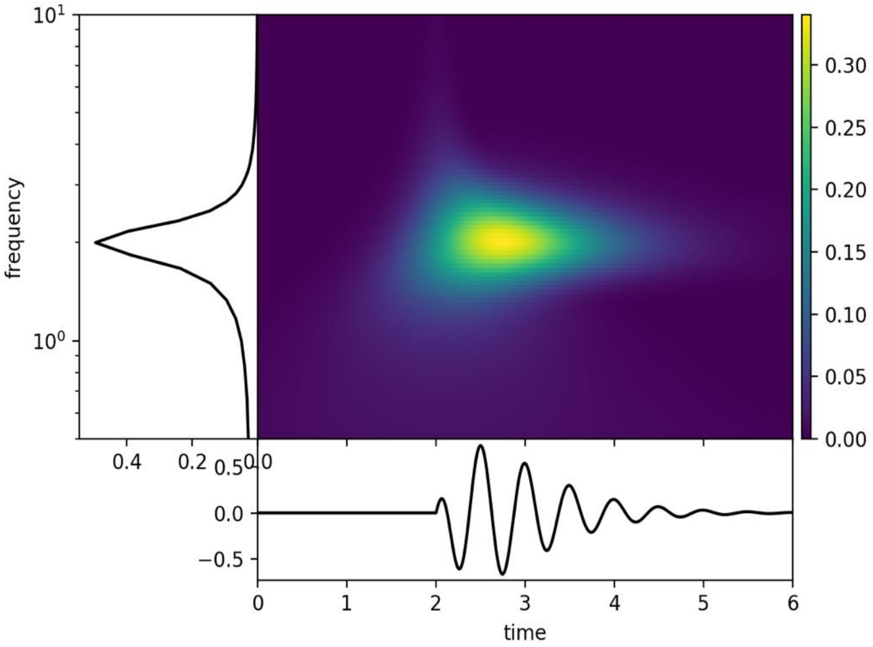


Figure 1.1. Representation of frequency in function of time.

1.4. Fourier transform

All the signals observed in the nature can be decomposed into a sum of pure sinusoids with different frequencies. Fourier Transform is a mathematical technique for obtaining the spectral composition of the signal by decomposing it into pure frequencies that make up the original signal [10]. The resulting sinusoids of Fourier Transform on a signal represented as a function of time is a complex value, whose imaginary part represents the phase off-set of the pure sinusoid and its absolute value represents the corresponding frequency component. Applying inverse Fourier

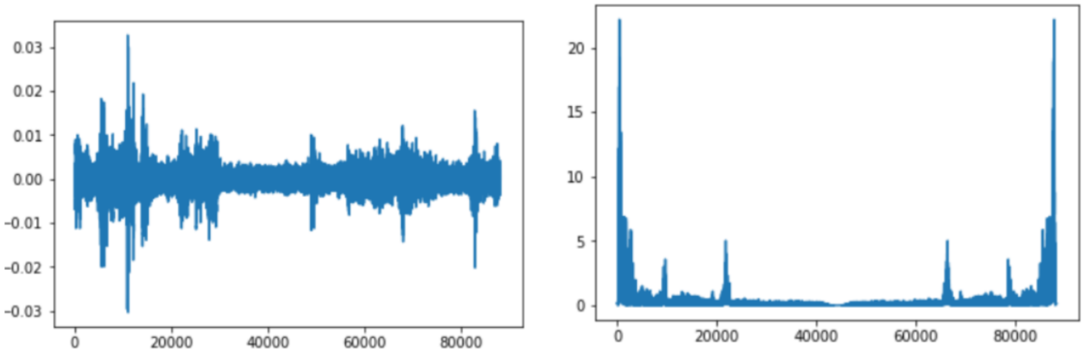
Transform on the resulting signal reconstructs the original signal when the condition provided for sampling theorem is satisfied.

Fast Fourier Transform (FFT) converts a signal from the time domain to the frequency domain as shown in Figure 1.2 below. Each frame having N_m samples are converted into frequency domain [10]. Fast Fourier transform is a fast algorithm to apply Discrete Fourier Transform (DFT) on the given set of N_m samples as shown below:

$$D_K = \sum_{m=0}^{N_m-1} D_m e^{\frac{-j2\pi km}{Nm}} \tag{1.1}$$

where, k = $0 \ldots . N_m - 1$ and D_m represents the DFT of a frame given m. where, k = $0 \ldots . N_m - 1$ and D_m represents the DFT of a frame given m. It is worth mentioning that the DFT algorithm has a complexity of $\mathrm{O}(N^2)$, whereas, the Fast Fourier transform implementation has a quasi-logarithmic complexity $\mathrm{O}(NLog_2N)$.

Here are two plots that show the effect of the FFT function applied to a simple raw audio waveform, it finds out the frequency domain representation of a time domain signal (see Figure 1.2).



(a) Raw audio waveform (b) Frequency domain signal after applying DFT

Figure 1.2. Graphical representation of the effect of DFT on raw audio wave form.

1.5. Pre-processing

Audio is prepared and processed for machine learning algorithms in the audio processing phase of the overall system design. Pre-processing is applied to the audio signal before the process of machine learning starts. The main role of this stage is to **enhance certain characteristics of the incoming signal** in order to maximize audio analysis performance in the later phases of the analysis system. For instance, this is achieved by reducing the effects of noise or by *emphasizing the target sounds in the signal* [11].

1.6. Feature engineering

Feature engineering consists of extracting features from raw data and transforming them into formats that are suitable for the machine learning model. It is a crucial step in the machine learning process, because the right features can ease the difficulty of modeling, therefore, it can improve the performance of a scene classification system [11]. Practitioners agree that the vast majority of time in building such systems is spent on feature engineering and data processing.

1.7. Temporal features

These features are computed directly on the temporal waveform. Therefore, usually rather straightforward to compute. A large number of features has been introduced for audio signals the energy of signal, zero crossing rate (ZCR), Time domain Envelope and Temporal waveform moments. But few examples of acoustic features that are frequently used by the research community.

1.8. Spectral features

Spectral features are obtained by converting the time-based signal into the frequency domain using Fourier Transform thus it we can obtain other features as: spectral centroid, spectral flux, spectral envelop, spectral roll-off, etc. These features can be used to identify the notes, pitch, rhythm, and melody [11].

1.9. Cepstral features

Cepstral features allow decomposition of the signal according to the so-called source-filter model widely used to model audio sound production [12]. The signal is decomposed into a carrier (source) and modulation (filter). **Mel Frequency Cepstral Coefficients (MFCC)** are the most common used cepstral coefficients. They are obtained as the inverse discrete cosine transform (DCT) of the log energy in Mel Frequency bands [12].

$$mfcc(t,c) = \sqrt{\frac{2}{M_{mfcc}}\sum_{m=1}^{M_{mfcc}} \log\left(\tilde{X}(t)\right) \cos\left(\frac{c\left(m-\frac{1}{2}\right)\pi}{M_{mfcc}}\right)} \tag{1.2}$$

where M_{MFCC} is the number of Mel Frequency bands, m the frequency band index, $\tilde{X}_m$ is the energy in the m^{th} Mel Frequency band and c is the index of the cepstrum coefficient ($c \in \{1,2,\dots,M_{MFCC}\}$). An example of $\boldsymbol{MFCC}$ features as shown below (see Figure 1.3).

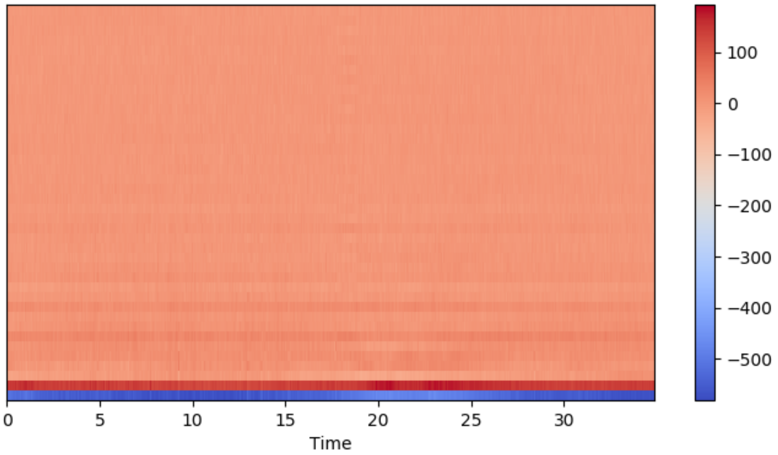


Figure 1.3. MFCCs features of an Office Scene example.

A common implementation uses a triangular filter bank where each filter is spaced according to a Mel Frequency scale. The energy coefficients $\tilde{X}_m(t)$ in the band m are obtained as a weighted sum of the spectral amplitude components $|\tilde{X}(t,f)|$ (where the weights are given according to the amplitude value of the corresponding triangular filter).

The MFCC contains the information of only the power spectral envelope of a signal frame, but it fails to capture the temporal dynamics of the audio signal. Delta features are used to capture these dynamics. They are basically time derivative of the MFCC features. Delta-Mel Frequency cepstral coefficients or ΔMFCCs is also referred as differential coefficients [13]. It has been widely used in the field of sound Analysis, where generally they are used in conjunction with MFCC feature vectors. Delta coefficients are calculated from MFFCs in the following equation below:

$$delta = \frac{\sum_{n=1}^{N} n(c_T - c_t)}{2\sum_{n=1}^{N} n^2} \tag{1.3}$$

Where C_n is the MFCC vector corresponding to n^{th} signal frame. MFCC vectors for frames ranging from (n – l) to (n + l) are utilized to compute delta coefficient vector $\boldsymbol{del_n}$ for $\boldsymbol{n_{th}}$ frame, l being the window size. An example of $\Delta MFCC$ features is given (see Figure 1.4).

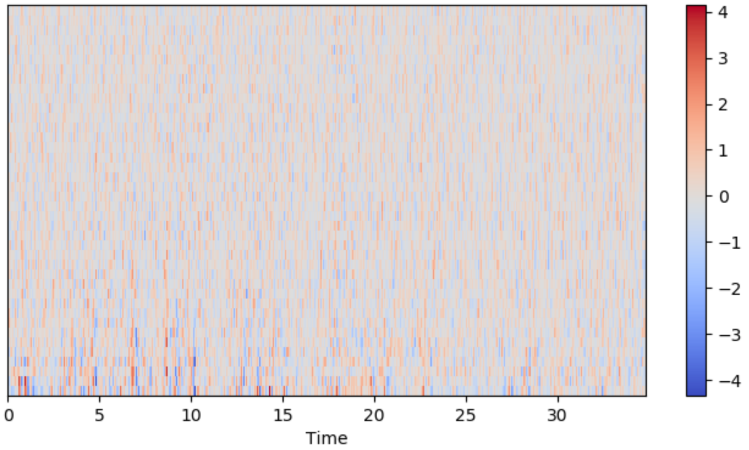


Figure 1.4. ΔMFCC features of an Office Scene example

1.10. Spectrogram features

Features can also be extracted from the time-frequency representation of a sound scene [14]. Spectrogram features rely on techniques inspired by computer vision to characterize the shape, texture and evolution of the time-frequency content in a sound scene. An example of Spectrogram features is given below (see Figure 1.5).

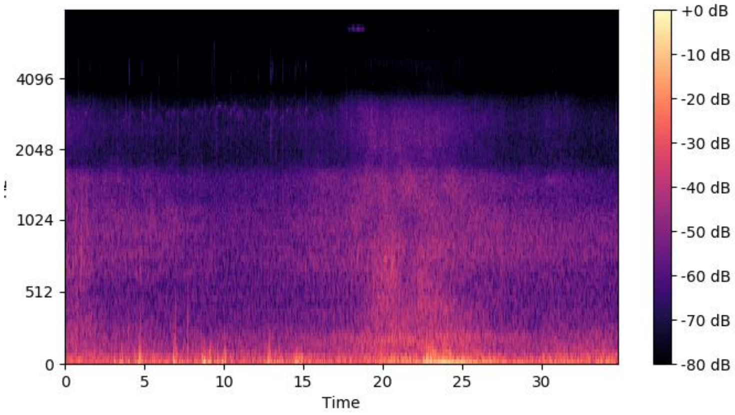


Figure 1.5. Spectrogram features

1.11. Other approaches

Studies on human perception have allowed for a better understanding of the human hearing process. Some results from these studies have been exploited in feature engineering and led to widely used features such as **Mel Frequency Cepstral Coefficients (MFCC)** [11]. However, there is still a large variety of perceptual properties that could be exploited in feature extraction:

- **Loudness** is the amplitude of a sound wave determines its loudness or volume. A larger amplitude means a louder sound, and a smaller amplitude means a softer sound.

- **Sharpness** can be interpreted as a spectral centroid based on psychoacoustic principles. It is commonly estimated as a weighted centroid of specific loudness.

- **Perceptual spread** is a measure of the timbral width of a given sound. It is computed as the relative difference between the largest specific loudness and the total loudness.

1.12. Conclusion

In this chapter, we have reviewed the basic concepts of sound representation that are necessary to understand the ideas treated in this work. We have presented various sound features (temporal, spectral, cepstral features and others) that are widely discussed by the research community. These features are prepared to be used as an input for a learning algorithm. In the next chapter, we will give an overview of basic machine learning notions, including classification steps, classifiers algorithms, some evaluation measurements and statistical tests.

Chapter 2: Machine Learning for Acoustic Scene Classification

2.1. Introduction

Machine learning is about programming computers to optimize a performance criterion using example data or past experience. It is used in cases where we cannot directly write a computer program to solve a given problem, but need example data or experience. Different categories of machine learning approaches have been introduced in the literature. They fall into three primary categories: Supervised Learning, Unsupervised Learning, and Semi-Supervised Learning. Supervised learning takes as an input a set of labeled data and learns a classification model. Unsupervised learning deals with input data that is not labeled. Semi-supervised learning builds a model from data made of a mixture of labeled and unlabeled examples. We undertake a supervised learning approach to address scene classification problem.

2.2. Fundamentals of classification

Machine learning has multiple fundamentals and rules which can be used to process and structure the Acoustic data that goes into system algorithms. The most rules used are described below.

In machine learning each data from the dataset refers to a specific scene which is tagged to its own label. These class labels are accompanied with more data called **features** (so-called **dimensions**) and they act as the input of the system [11]. New features can also be obtained from the oldest using a method known as '***feature engineering***'.

Machine learning algorithms are described as learning a target function that maps input variables (features) to an output variable (class label). The most common objective of machine learning is to learn the mapping (training) $Y = f(X)$ to make predictions (testing) of Y for new X

on f mapping function (learning algorithms) [11]. This is called **predictive modeling** or **predictive analytics** and the goal is to make the most accurate predictions possible.

While training, the learning algorithm maps the training samples with their class labels in order to build a model. The model obtained is tested on unseen samples (testing data). The output consists of a list of predictions that are used to get measurements referring to the model performance.

Generalization is an important fundamental in machine learning [15]. It refers to how well the approximation of the target function generalizes a new data. Generalization works best if the signal or the sample that is used as the training data has a high signal to noise ratio. If that is not the case, generalization would be poor and we will not get good predictions.

The classification step has gotten two major problems like overfitting or underfitting. A model is overfitting if it fits the training data too well and there is a poor generalization of new data. As well a model is underfitting if it fits the training data is not enough to get important results. The main goal in this phase is building a model with favorable fitting and gives good performances.

Cross-validation is an important statistical method that helps to get meaningful results about the machine learning models included in the system [11, 11]. It consists of splitting data into training and testing sets. The most used method is K-Fold Cross-validation.

After building all desired models, it is important to differentiate between them and to be able to do that, statistical tests are used. These tests use models' performances obtained previously to compare each model with the others. "Friedman Test", "Nemenyi Test" and "Wilcoxon signed Rank Test" are applied.

2.2.1 Common classifiers

A. Feed Forward Neural Network

Artificial Neural Networks (ANNs), also known simply as neural networks (NNs), are computing systems that process information by their dynamic state response to external inputs [16]. In its fundamental structure, a NN is a network of interconnected nodes called neurons, weighted connections join the neurons and scale the strength of the transmitted signals, representing the synapses in the brain. An ANN can be described as mapping an input space to an output space [17].

ANNs in general are composed of nodes or units connected by directed links. A link from unit to another unit serves to propagate the activation. Each link also has a numeric weight associated with it, which determines the strength and sign of the connection. Each unit j first computes a weighted sum of its inputs, then it applies an **activation function** on this sum to derive the output [18]. Three activation functions are cited below (Sigmoid, ReLU, Softmax):

- **Sigmoid Function**

The sigmoid function is the most common form of activation function used in the construction of neural networks. It is a strictly increasing function, which holds an excellent balance between linear and non-linear behavior. It is defined by:

$$S(x) = \frac{e^x}{e^x + 1} \tag{2.1}$$

- **Rectified linear unit (ReLU)**

ReLU is an activation function that scales the output linearly. ReLU function has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. It is defined by:

$$ReLU(x) = \max(0, x) \tag{2.2}$$

- **Softmax function**

Softmax function computes the probabilities of each scene over all possible scenes. The output is equivalent to a categorical probability distribution. It measures the probability that any of the classes are true. It is defined by:

$$Softmax(x_j) = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \tag{2.3}$$

where x is a vector of inputs to the output layer, j indexes the output units, so $j = 1, 2, \ldots, K$.

NNs use generally a procedure called **Backpropagation**, which is the practice of learning the weights of a neural network based on the error rate obtained from the previous training epoch. Proper tuning of the weights ensures lower error rates and makes the model reliable.

NNs use another method called dropout. It is a technique where randomly selected neurons are ignored during training. (randomly “dropped-out”). This means that their contribution to the activation of neurons is temporarily removed on the forward pass and any weight updates are not applied to the neuron on the backward pass.

The effect is that the network becomes less sensitive to the specific weights of neurons. This in turn results in a network that is capable of better generalization and is less likely to overfit the training data [19].

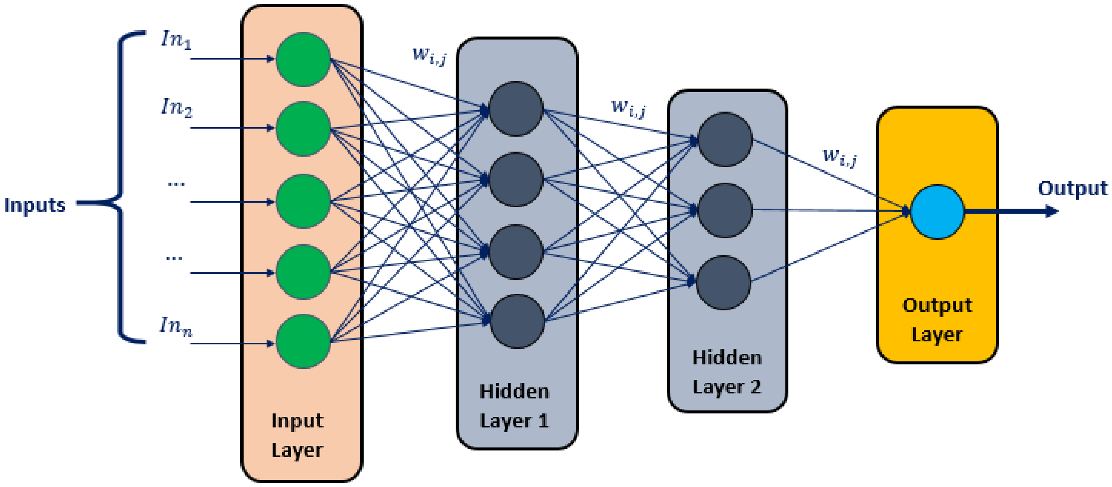


Figure 2.1. Multi-Layers Neural Network Structure

Figure 2.1 shows a simple structure of a Multi-Layer Neural Network, In represents input features which get through the different layers (input layer, hidden layers and finally output layer) passing by nodes in each layer and attributed with weights $w_{i,j}$, i and j are nodes indexes.

Feed Forward Neural Network or **FNN** is a NN that does not contain any feedback connection(loops), i.e., each layer receives inputs only from the previous layer. It is a directed graph, where information is always traveling forward. There are many architectures of FNN, one of the most used in classification is Multi-Layer Perceptron (**MLP**). MLP can partition the input space into exponentially more linear regions than a shallow network with the same number of neurons. For this reason, they can more easily represent highly structured and complex functions. MLP have obtained excellent results in speech and sound classification [20].

B. Support Vector Machine

Support Vector Machine (called SVM) constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space. A good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point (these points are known as Support vectors) of any class (so-called functional margin). Generally, the generalization error of the classifier decreases as the margin gets larger [21].

It is believed that mapping data into higher dimensional spaces could make data easily separated or better structured. This procedure is known as Kernel trick. We describe in what follows two Kernel methods that have been widely used in machine learning experiments.

- **Polynomial Kernel**

Polynomial kernels are well suited for problems where all the training data is normalized. It has adjustable parameters are the slope $\boldsymbol{\alpha}$, the constant term c and the polynomial degree d as below:

$$k(x, y) = (\alpha x y + C)^d \tag{2.4}$$

- **Radial Basis Function Kernel**

Radial basis function kernel (RBF) is a function whose value depends on the distance from the origin or from some point. It is defined in the following formula (2.5):

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \qquad (2.5)$$

The parameter $\boldsymbol{\sigma}$ plays a major role and should be carefully set to the problem at hand.

- **K-Nearest Neighbor**

K-Nearest neighbor algorithm called KNN is one of the most popular learning techniques. KNN classifier is a supervised learning algorithm based on clustering. It divides data inputs into clusters that defines class labels or categories [22].

KNN is a sample-based learning algorithm or lazy learner. It uses all the training data to predict the class labels of testing data [23].

KNN works by finding the distances between test data and all the training data, then selecting the specified number (K) closest to the test. Finally, it votes for the most frequent label. These distances are computed using different functions as like as: Euclidean, Manhattan and Minkowski Distance. A simple example of KNN classifier is given by Figure 2.2. White points are data labeled to the class A, black points are data which belongs to class B and the blue point represents the testing data. Distances are computed between test data (blue) and all other data (white and black) using a distance function. KNN gets all distances smaller than k chosen; then, it votes for the most frequent label present in these distances.

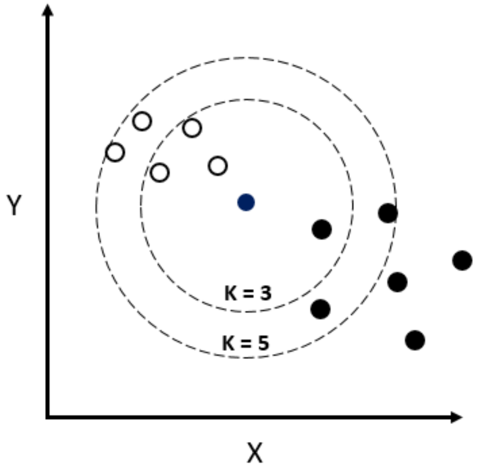


Figure 2.2. KNN classification

This approach is effective, non-parametric and easy to implement. However, the prediction takes too long due to its high calculation complexity. Furthermore, the performance is dependent only on the training set there is no weight difference among samples.

2.3. Evaluation of acoustic scene classification models

2.3.1 Cross-validation

Cross-validation is a statistical method, also known as resampling procedure, used to estimate the skill of a machine learning model on a limited data sample. It is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem. Several resampling techniques have been introduced such as k-fold cross validation, leave-one-out, 5×2 cross validation, and 10×10 cross validation.

K-Fold Cross-validation is an iterative approach: during iteration i, it randomly divides the set of observations into K groups or folds, of approximately equal size. Fold i is treated as a testing set, and the remaining $K - 1$ folds assigned as a training set. This procedure is repeated K times as shown below (see Figure 2.3):

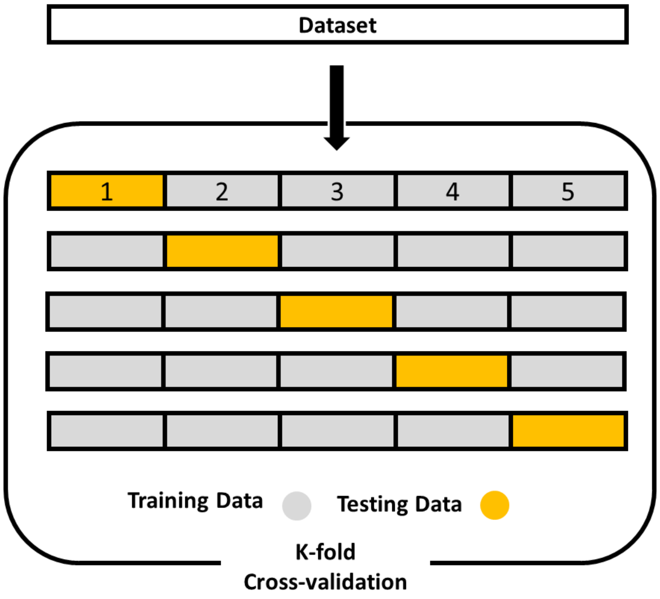


Figure 2.3. 5-Fold Cross-validation.

The results of a k-fold cross-validation runs are often summarized with the mean of the model skill scores. It is also good practice to include a measure of the variance of the skill scores, such as the standard deviation or standard error. The choice of K is usually 5 or 10, but there is no formal rule [24].

Stratification is a technique where we rearrange the data in a way that each fold has a good representation of the whole dataset. It forces each fold to have at least m instances of each class. This approach ensures that one class of data is not overrepresented especially when the target variable is unbalanced, which is depicted in Figure 2.4.

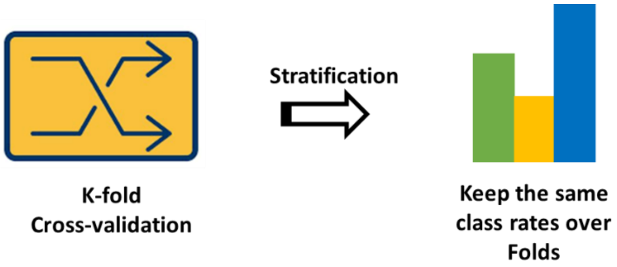


Figure 2.4. Stratified K-Fold Cross-validation.

2.3.2 Performances metrics

The choice of the metrics used to evaluate machine learning model is very important. It influences how the performance of a learning algorithm is measured and compared. Numerous performance metrics have been used in machine learning experiments. In what follows, we describe some of them [25].

Confusion Matrix

The Confusion matrix is one of the most intuitive and easiest metrics used for finding the correctness and accuracy of a model. It is a matrix of $n \times n$, where n represents the number of classes. The row dimension is called Ground truth, whereas the column is known as Predicted. Table 2.2 provides a representation of the confusion matrix with $n = 2$.

Table 2.1. Confusion Matrix

		Predictions	
	Classes	0	1
Ground Truth	0	TN	FP
	1	FN	TP

where:

- **True Positives (TP)** are the cases when the actual class of the data point was 1 (True) and the predicted is also 1 (True).
- **True Negatives (TN)** are the cases when the actual class of the data point was 0 (False) and the predicted is also 0 (False).
- **False Positives (FP)** are the cases when the actual class of the data point was 0 (False) and the predicted is 1 (True).
- **False Negatives (FN)** False negatives are the cases when the actual class of the data point was 1 (True) and the predicted is 0 (False).
- **Accuracy/error rate**

Accuracy is the ratio between the number of correct predictions made by the model and the total number of instances. Accuracy can be computed using the previous metrics (TP, TN, FP, FN) is given by:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.6}$$

- **Precision and recall**

Precision also called positive predictive is the fraction of predicted positives which are actually positive. Recall is the fraction of actual positives which are correctly predicted. We can calculate them from the confusion matrix using the equations (2.8-2.9) below:

$$Precision = \frac{TP}{TP + FP} \tag{2.7}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.8}$$

The precision and recall can be used in multi-class problems to measure the predictive performance of the classifier for a particular scene.

- **F1 Score**

F1 Score is the average between precision and recall. It measures how precise the classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances). It is given by the following formula (2.10):

$$F1 = 2 \times \frac{Precision + recall}{Precision \ \times recall} \tag{2.9}$$

High precision but lower recall, gives an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better is the performance of our model.

2.3.3 Statistical tests

Given multiple learning algorithms, model evaluation aims at identifying which algorithm produces the most accurate classifiers. This concern is one among the fundamental issues in machine learning [4]. In order to address it, García et al. [26] [13], and Japkowicz et al. introduced several statistical tests such Friedman, Nemenyi and Wilcoxon for performance comparison [27].

A. Friedman test

The Friedman test is useful for comparing several algorithms over multiple domains. It first ranks the techniques for each dataset separately according to the generalization measure in descending order. The best performing technique gets the rank 1, the second best gets rank 2... etc. In case of ties, average ranks are assigned. Let r_i^j be the rank attributed to the j^{th} algorithm on the i^{th} dataset; and let R_j denote the average rank of algorithm $j \in \{1, \dots, t\}$ over N datasets. Under the null hypothesis, it is assumed that all techniques are equivalent; hence, their average ranks should be equal.

$$R_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j \tag{2.10}$$

$$x_F^2 = \frac{12N}{t(t+1)}\left[\sum\nolimits_{j=1}^{k} R_i^j - \frac{t(t+1)^2}{4}\right] \tag{2.11}$$

The test statistic is given in equation (2.12) chi-squared distribution with $t - 1$ degrees of freedom for sufficiently large N and t (usually $N > 10$ and $t > 5$). This test provides only an assessment whether the observed differences in the performances are statistically significant.

B. Nemenyi test

This test is invoked when all techniques are compared with each other. The performance of two methods is significantly different if their corresponding average ranks differ by at least the critical difference CD

$$CD = q_\alpha \sqrt{\frac{t(t+1)}{6N}} \tag{2.12}$$

where the critical value q_α is defined based on the Studentized range statistic divided by $\sqrt{2}$.

C. Wilcoxon signed rank test

Wilcoxon signed-ranks test is a non-parametric test. It is considered the best strategy to compare two algorithms over multiple domains. The formulation of this test is the following. We designate by d_i the differences between the performance scores of two techniques on N datasets, $i \in \{1 \dots N\}$. We first rank these differences according to their absolute values; in case of ties average ranks are attributed. Then, we compute the sum of ranks for the positive and the negative differences, which are denoted as R^+ and R^-, respectively. Their formal definitions are given by:

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2} \sum_{d_i=0} rank(d_i) \tag{2.13}$$

$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2} \sum_{d_i=0} rank(d_i). \tag{2.14}$$

Notice that the ranks of $d_i = 0$ are split evenly between R^+ and R^-. Finally, the statistics T_w is computed as $T_w = min(R^+, R^-)$. The statistics z follows the normal distribution with 1 mean and 0 variance. For instance, the hypothesis which states that two approaches perform equally is rejected if $z \leq -1.96$ at a 5% significance level.

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}} \tag{2.15}$$

2.4. Feature Selection

Feature Selection is the process of determining what inputs should be presented to a classification algorithm. Originally feature selection was performed by domain experts as they chose what properties of an object should be measured to try and determine the class label. Modern classification problems attempt to collect all possible features, and then use a statistical feature selection process to determine which features are relevant for the classification problem.

Feature selection algorithms of all kinds rely upon a single assumption about the data, that the feature set contains irrelevant and/or redundant features. Irrelevant features contain no useful information about the classification problem, and redundant features contain information which is already present in more informative features. It may also improve classification performance by reducing the potential for overfitting when shrinking the feature set. The strongly relevant features are not redundant as each one contains useful information that is not present in any other combination of features.

2.4.1 Joint Mutual Information

An information theoretic filter algorithm is one that uses a measure drawn from Information Theory as the evaluation criterion. Evaluation criteria are designed to measure how useful a feature or feature subset is when used to construct a classifier. We will use J to denote an evaluation criterion which measures the performance of a particular feature or set of features in the context of the currently selected set. The most common heuristic evaluation criteria in information theoretic feature selection is simply Joint Mutual Information (**JMI**). The JMI score for feature X_k is:

$$J_{JMI}(X_k) = \sum_{X_j \in S} I(X_k X_j; Y) \tag{2.16}$$

This is the information between the target and a joint random variable $X_k X_j$, Defined by pairing the candidate X_k with each feature previously selected. The idea is if the candidate feature is complementary' with existing features, we should include it [28].

2.5. Conclusion

In this part, we have reviewed the background knowledge of classification and sound representations that are relevant to the enquiries pursued in this thesis. The main goal of our work is to design proper Machine Learning experiments for acoustic scene classification; from cross validation to results discussion. In the next part, we will first introduce the experimental setup and describe the general schema of the main step that we have followed. Then, we will present and discuss the obtained results based on robust statistical tests.

Experiments

The challenge consists of extracting the features that best represent the problem at the hand, and fitting the most effective classification model. To put it simply, the aim is to find the best combination of a feature extraction approach and a classification algorithm. This problem has been addressed by the research community using various methodologies. The aim of this thesis is to **derive guidelines for researchers and beginners in this field to assist them in building effective scene classification systems**. To this end, we have carried out experimental comparisons among various scene classification systems. Our work is two folds:

- **First**, we have conducted extensive experimental comparison among sound analysis methods using a large set of sound scenes DCASE 2016 dataset [9]. We have thoroughly examined 3 acoustic features (Mel Frequency Cepstral Coefficient, a combination of Mel Frequency Cepstral Coefficient with the delta coefficients, and Spectrograms) and 3 classification paradigms (Feed Forward Neural Network, K Nearest Neighbors, Support Vector Machine). We have backed our analysis and conclusions based on well-known statistical tests.
- **Second**, we have investigated the effect of the number of features on the generalization performance. Specifically, we have invoked a feature selection approach, namely JMI [6] in order to automatically determine the optimal set of extracted features. **To the extent of our knowledge, only very few attempts have considered incorporating feature selection step into the design of their systems**.

Chapter 3: Experimental Setup

3.1. Introduction

The main goal of this thesis is to derive guidelines for researchers based on empirical comparisons among various scene classification systems. In this Chapter, we present the setup for conducting these experiments. We have carried out two case studies. We present in Section 2.2 the first case study: we have evaluated numerous **machine learning classification systems (FNN, SVM, and KNN)** combined with multiple **feature extraction techniques (MFCC, ΔMFCC and Spectrogram)**. Then, in Section 2.3, we describe the second case study: we have tested the effect of **feature selection** on the generalization power of a given system.

3.2. Dataset

We have selected TUT Acoustic scenes 2016 (11 GB) dataset from the DCASE 2016 Community [9] containing 1,560 sound files that defines 15 different scenes with 104 instances for each scene from the urban environment. The 15 scenes are given by:

- Beach
- Bus
- Cafe/Restaurant
- Car
- City Center
- Forest Path
- Grocery Store
- Home
- Library
- Metro Station
- Office
- Park
- Residential Area
- Train
- Tram

TUT Acoustic scenes 2016 dataset consists of recordings from various acoustic scenes, all having distinct recording locations. For each recording location, 3-5-minute-long audio recording was captured. The original recordings were then split into 30-second segments. For all acoustic scenes, the recordings were captured each in a different location: different streets, different parks, different homes. Recordings were made using a Soundman OKM II Klassik/studio A3, electret binaural microphone and a Roland Edirol R-09 wave recorder using 44.1 kHz sampling rate and 24 bits resolution. The microphones are specifically made to look like headphones, being worn in the ears. As an effect of this, the recorded audio is very similar to the sound that reaches the human auditory system of the person wearing the equipment. For this experimental study, we have divided each sound file into frames of 25 milliseconds with an overlapping of 50% of the frame length; we gotten then 1501 frames per sound file and around **2 million frames** over the entire dataset.

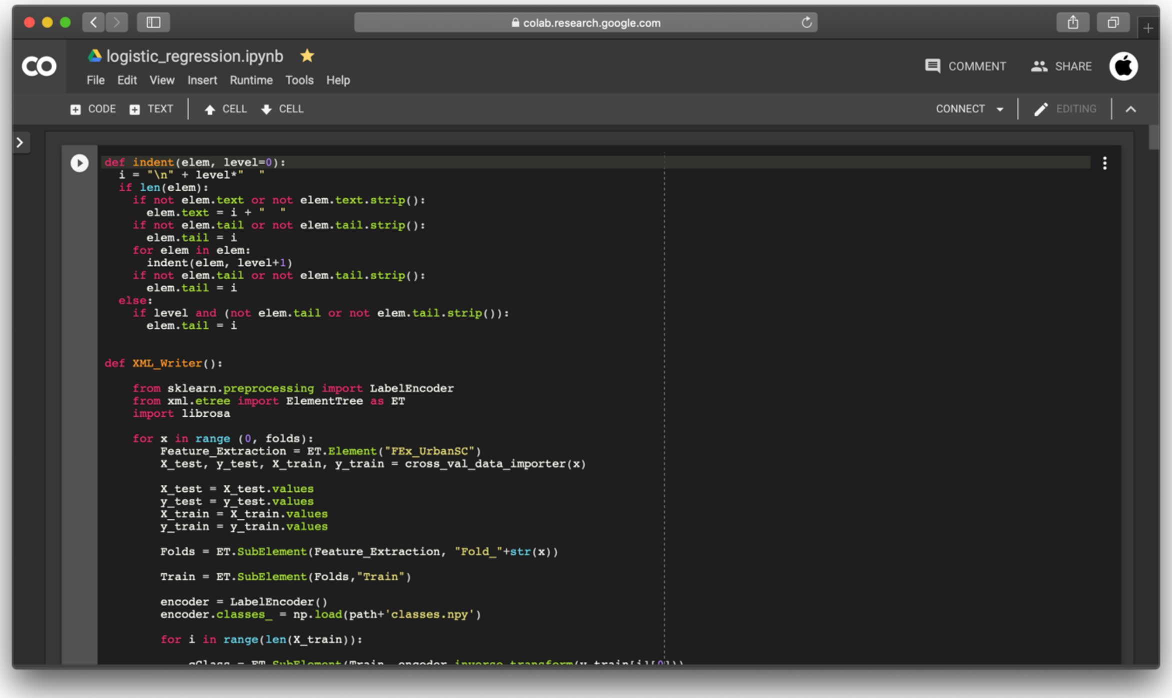
3.3. Cross-validation

We have divided the dataset following 5 cross-validation and have gotten 5 subsets containing in each fold the training and testing data denoted $train_i$, $test_i$, i = 1...5, respectively. Cross Validation consists of, first, shuffling the dataset randomly. Second, it splits the dataset into 5 folds(groups); we take each fold as a test dataset and the remaining groups as a training data set. Third, we fit a model on the training set and evaluate it on the test set. We repeat these three steps 5 times. Finally, we report the mean of scores over these 5 iterations.

3.4. Programming language and execution platform

We have carried out our experiments using Python programming language. Python is a powerful programming language frequently used for conducting machine learning experiments. It offers a myriad of libraries such as Scikit-Learn for classification algorithms and Librosa for signal processing.

DCASE 2016 is a huge dataset made of 11 Gb of acoustic scenes data. This mass entails an increase in the computational cost, specifically, the training time. To cope with this, we have decided to carry out our experiments using an online platform called **Google Colab** [29]. This platform provides a high-performance machine where we can execute a Python code and export results directly to Google Drive. Figure 3.1 shows a screenshot of Colab notebook.



```
def indent(elem, level=0):
  i = "\n" + level*"  "
  if len(elem):
    if not elem.text or not elem.text.strip():
      elem.text = i + "  "
    if not elem.tail or not elem.tail.strip():
      elem.tail = i
    for elem in elem:
      indent(elem, level+1)
    if not elem.tail or not elem.tail.strip():
      elem.tail = i
  else:
    if level and (not elem.tail or not elem.tail.strip()):
      elem.tail = i


def XML_Writer():

    from sklearn.preprocessing import LabelEncoder
    from xml.etree import ElementTree as ET
    import librosa

    for x in range (0, folds):
        Feature_Extraction = ET.Element("FEx_UrbanSC")
        X_test, y_test, X_train, y_train = cross_val_data_importer(x)

        X_test = X_test.values
        y_test = y_test.values
        X_train = X_train.values
        y_train = y_train.values

        Folds = ET.SubElement(Feature_Extraction, "Fold_"+str(x))

        Train = ET.SubElement(Folds,"Train")

        encoder = LabelEncoder()
        encoder.classes_ = np.load(path+'classes.npy')

        for i in range(len(X_train)):

            cClass = ET.SubElement(Train, encoder.inverse_transform(y_train[i][0]))
```

Figure 3.1. Screenshot of Colab notebook

3.5. Description of the conducted experiments

The primary goal of this thesis is to provide researchers and beginners with some guidelines for developing scene classification systems. To this end, we have carried out **2 case studies**.

A. First case study

The purpose of this case study is to compare the performances of several scene classification systems, varying the learning paradigms and the mechanism for extracting features. Specifically, we have employed 3 different classification models, namely Feed Forward Neural Network (**FNN**), Support Vector Machine (**SVM**) and K-Nearest Neighbors (**KNN**); and 3 sets of acoustic features: MFCC, MFCC+ΔMFCC, SPEC. Note that we have implemented multiple variants of each learning model, obtained by varying **hyperparameters** such as number of epochs, nature of the kernel, etc. We have organized this case study in **3 scenarios**. In each scenario, we compare the performances of several systems, obtained by combining multiple classifiers trained using a single learning algorithm with the 3 sets of acoustic features. We analyze and discuss the experimental scores based on statistical tests. A detailed description of these scenarios can be found in Figure 3.2.

B. Second case study

The aim of the **second case study** is to investigate how the number of selected features affects the performance. We have carried out the following experiment: We have first trained a system which combines 40 MFCCs + 40 ΔMFCCs features along with FNN; we have used 3 hidden layers and set the number of epochs to 100. We refer to this system as the **control system (FNN-80F).** Second, we have invoked **Joint Mutual Information (JMI)** filter to select the top most effective L features. Third, we have varied L from 15 to 75, while increasing L by 5 each time. We obtained 14 scene classification systems. We refer to them as **FNN-15F, FNN-20F, …, FNN-75F**. We have tested and compared their scores using Wilcoxon signed-ranks test. Figure 3.3 exhibits a general schema that highlight the primary steps for conducting the second case study.

3.5.2 Case Study 1 Setup

A. Acoustic features

We have extracted the Mel Frequency Cepstral coefficients (MFCCs), delta Mel Frequency Cepstral coefficients (ΔMFCCs) and the spectrogram features (SPEC) from the training and testing sound data. We have carried out this step using Librosa library [30]. Table 3.1 summarizes the setup of this step.

Table 3.1. Acoustic Features Setup.

Parameter name	Value
Sample rate	44.1 kHz
Frame length	20ms
Overlap [%]	50%
MFCC coefficients	40
ΔMFCC coefficients	40
Spectrogram coefficients	128

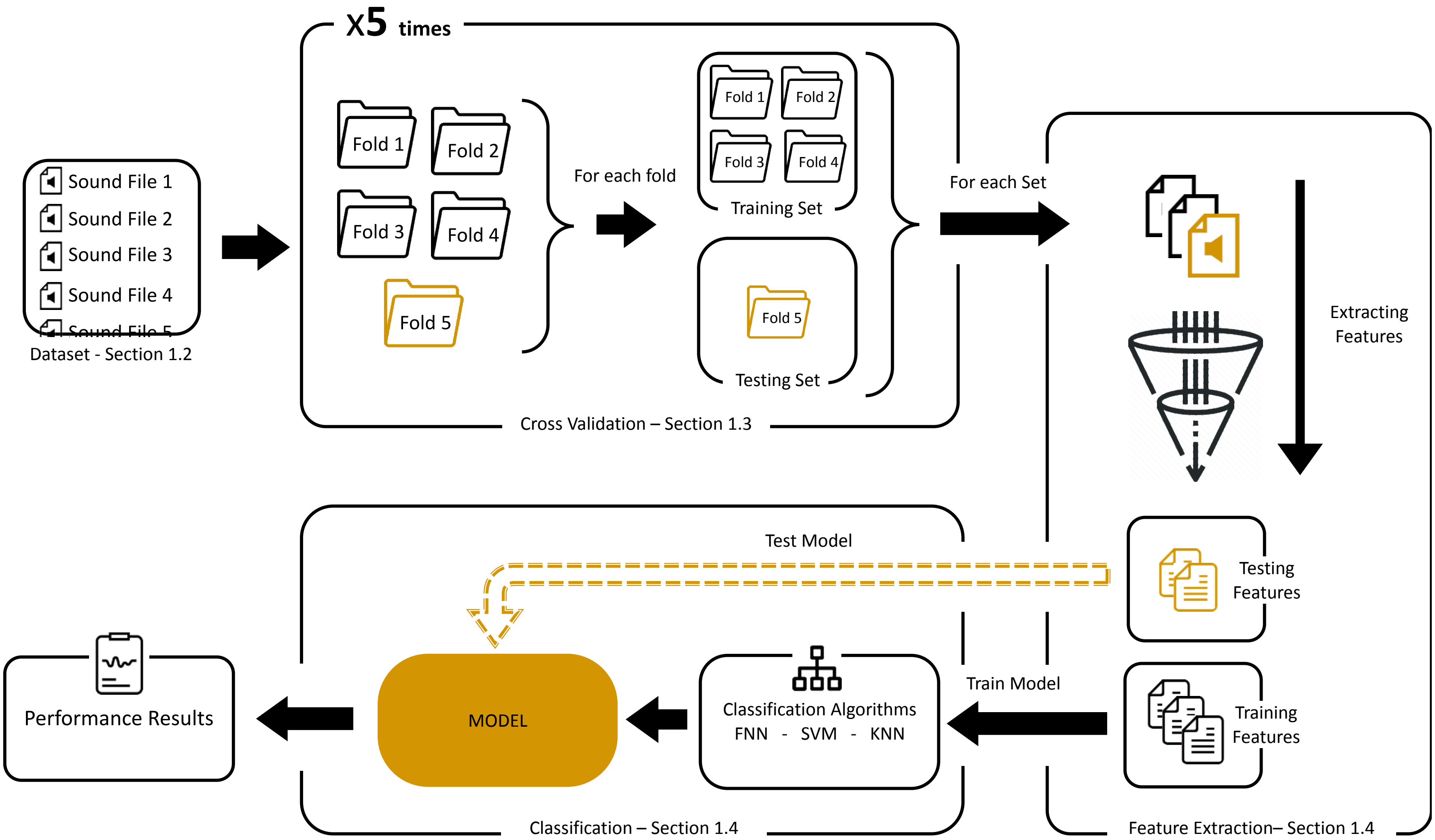


Figure 3.2. General schema that highlight the primary steps for conducting the first case study.

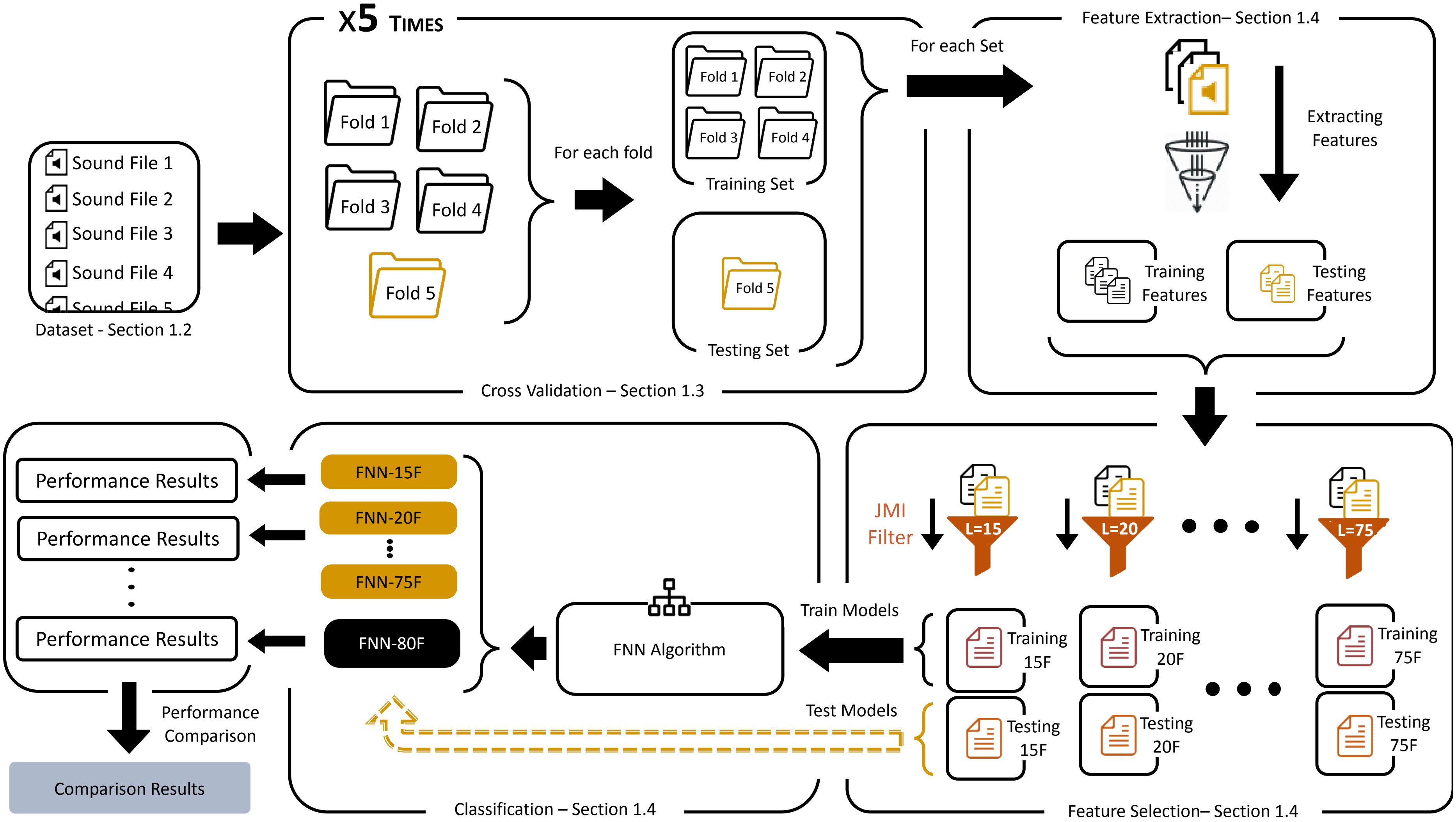


Figure 3.3. Exhibits a general schema that highlight the primary steps for conducting the second case study.

B. Classification models

We have fitted our classifiers using FNN, SVM and KNN learning algorithms, varying some hyperparameters such as number of epochs, nature of the kernel, etc. We have invoked Scikit-Learn library [31] and Keras [32] for implementing these models. Table 3.2, Table 3.3 and Table 3.4 give a description of these variants.

- **FNN description**

Table 3.2. FNN Setup Description.

Parameter	Value
Feature	MFCC
Number of hidden layers	2, 3
Number of hidden neurons in hidden layers	1024
Number of hidden neurons in output layer	15
Activation function for hidden layer neurons	sigmoid
Activation function for output layer neurons	softmax
Dropout value for hidden layer neurons	0.5

Parameter	Value
Feature	MFCC + ΔMFCC
Number of hidden layers	2, 3
Number of hidden neurons in each layer	1024
Number of hidden neurons in output layer	15
Activation function for hidden layer neurons	sigmoid
Activation function for output layer neurons	softmax
Dropout value for hidden layer neurons	0.5

Parameter	Value
Feature	Spectrogram
Number of hidden layers	2, 3
Number of hidden neurons in each layer	1024
Number of hidden neurons in output layer	15
Activation function for hidden layer neurons	sigmoid
Activation function for output layer neurons	softmax
Dropout value for hidden layer neurons	0.5

- **KNN description**

Table 3.3. KNN Setup Description.

Parameter	Value
Feature	MFCC
Number of nearest neighbors	10, 20, 30, 40
Distance function	Euclidean distance

Parameter	Value
Feature	MFCC + ΔMFCC
Number of nearest neighbors	10, 20, 30, 40
Distance function	Euclidean distance

- **SVM description**

Table 3.4. SVM Setup Description.

Parameter	Value
Feature	MFCC
Kernel	Radial basis function
Regularization	1

Parameter	Value
Feature	MFCC + ΔMFCC
Kernel	Radial basis function
Regularization	1

C. Scene classification systems

We have divided this case study into 3 scenarios: FNN scenario, SVM scenario and KNN scenario. Each scenario compares the performances of several systems, obtained by combining multiple classifiers trained using a single learning algorithm with the 3 sets of acoustic features. Table 3.5, Table 3.6 and Table 3.7 describe the scene classification systems that were tested in this case study.

Table 3.5. FNN Scene Classification Systems.

Abbreviation	Classification Model	Feature Set
FNN-100_2 MFCC	FNN with 100 epochs and 2 hidden layers	MFCC
FNN-100_2 MFCC+ΔMFCC	FNN with 100 epochs and 3 hidden layers	MFCC+ΔMFCC
FNN-100_2 SPEC	FNN with 100 epochs and 2 hidden layers	SPECTROGRAM
FNN-100_3 MFCC	FNN with 100 epochs and 3 hidden layers	MFCC
FNN-100_3 MFCC+ΔMFCC	FNN with 100 epochs and 2 hidden layers	MFCC+ΔMFCC
FNN-100_3 SPEC	FNN with 100 epochs and 3 hidden layers	SPECTROGRAM
FNN-500_2 MFCC	FNN with 500 epochs and 2 hidden layers	MFCC
FNN-500_2 MFCC+ΔMFCC	FNN with 500 epochs and 3 hidden layers	MFCC+ΔMFCC
FNN-500_2 SPEC	FNN with 500 epochs and 2 hidden layers	SPECTROGRAM
FNN-500_3 MFCC	FNN with 500 epochs and 3 hidden layers	MFCC
FNN-500_3 MFCC+ΔMFCC	FNN with 500 epochs and 2 hidden layers	MFCC+ΔMFCC
FNN-500_3 SPEC	FNN with 500 epochs and 3 hidden layers	SPECTROGRAM

Table 3.6. KNN Scene Classification Systems.

Abbreviation	Classification Model	Feature Set
KNN10 MFCC	KNN with K=10	MFCC
KNN10 MFCC+ΔMFCC	KNN with K=10	MFCC+ΔMFCC
KNN20 MFCC	KNN with K=20	MFCC

KNN20 MFCC+ΔMFCC	KNN with K=20	MFCC+ΔMFCC
KNN30 MFCC	KNN with K=30	MFCC
KNN30 MFCC+ΔMFCC	KNN with K=30	MFCC+ΔMFCC
KNN40 MFCC	KNN with K=40	MFCC
KNN40 MFCC+ΔMFCC	KNN with K=40	MFCC+ΔMFCC

Table 3.7. SVM Scene Classification Systems.

Abbreviation	Classification Model	Feature Set
SVM MFCC	SVM with radial basis function kernel	MFCC
SVM MFCC+ΔMFCC	SVM with radial basis function kernel	MFCC+ΔMFCC

3.6. Conclusion

In this chapter, we have described the setup used to conduct our experimental enquiries, starting from cross validation to classification step. We have presented two schemes that highlight the key steps for carrying out a proper Machine Learning experiment. In the following chapter, we will present the results of these experiments and analyze them in order to derive guidelines based on numerous statistical comparisons.

Chapter 4: Experimental results and discussion

4.1. Introduction

In this chapter, we have carried out two different case studies. The first case, given in Section 4.2, consists of combining various machine learning approaches with multiple feature extraction methods. The aim is to evaluate the performance of these models and to derive guidelines based on statistical tests**.** In the second case, provided in Subsection 2.3, in order to alleviate the computational burden and reduce the complexity of data, we have included a feature selection step. We have run our experiments using ***Joint Mutual Information*** that automatically selects the most effective set of features.

4.2. Case Study 1

A. FNN Classifier

Table 4.1 gives the average F1-score results of the first scenario. The last row specifies the mean rank of each method over all scenes.

Analysis and discussion

We have statistically compared the performances of these techniques using Friedman test. Under the null hypothesis, we assumed that all systems are equivalent and the observed differences are due to chance. Friedman test rejects this hypothesis with $\mathrm{FF} = 45.49 > \mathrm{F}(11{,}154) = 12.51$ for $\alpha = 1 \times 10 - 16$ (FF is distributed according to the F distribution with $12 - 1 = 11$ and $(12 - 1) \times (15 - 1) = 154$ degrees of freedom), and therefore **confirms the existence of at least one pair of scene classification systems with significantly different performances.**

Table 4.1. F1 Scores using FNN.

	Mel Frequency Cepstral Coefficient				**Delta Mel Frequency Cepstral Coefficient**				**Spectrogram**			
Epochs	100	100	500	500	100	100	500	500	100	100	500	500
H-L **Scenes**	2	3	2	3	2	3	2	3	2	3	2	3
Beach	90.59 ± 3.57	**91.38 ± 4.12**	90.59 ± 3.57	90.59 ± 3.57	**91.38 ± 4.12**	90.59 ± 3.57	90.59 ± 3.57	90.59 ± 3.57	53 ± 7.10	45 ± 6.96	73 ± 15.83	73 ± 15.36
Bus	99.13 ± 1.74	99.13 ± 1.74	99.13 ± 1.74	98.18 ± 2.24	99.13 ± 1.74	**100.0 ± 0.00**	99.13 ± 1.74	99.13 ± 1.74	90 ± 4.58	89 ± 7.91	93 ± 6.39	95 ± 4.13
Cafe/Restaurant	96.36 ± 1.83	95.64 ± 2.64	95.64 ± 2.64	94.69 ± 1.52	95.64 ± 2.64	**96.44 ± 1.79**	95.49 ± 0.20	95.64 ± 2.64	76 ± 6.33	80 ± 4.82	86 ± 5.58	84 ± 4.74
Car	**98.26 ± 2.13**	**98.26 ± 2.13**	**98.26 ± 2.13**	97.31 ± 2.2	**98.26 ± 2.13**	**98.26 ± 2.13**	**98.26 ± 2.13**	**98.26 ± 2.13**	85 ± 8.43	84 ± 7.72	89 ± 5.26	91 ± 3.92
City Center	**98.33 ± 3.33**	**98.33 ± 3.33**	**98.33 ± 3.33**	**98.33 ± 3.33**	**98.33 ± 3.33**	97.46 ± 3.35	**98.33 ± 3.33**	**98.33 ± 3.33**	88 ± 2.38	87 ± 2.13	91 ± 4.60	91 ± 4.60
Forest Path	**98.95 ± 2.11**	**98.95 ± 2.11**	96.94 ± 2.50	98.00 ± 2.46	97.90 ± 2.58	**98.95 ± 2.11**	**98.95 ± 2.11**	97.89 ± 2.58	0.0 ± 0.00	0.0 ± 0.00	39 ± 19.30	41 ± 20.66
Grocery Store	**98.95 ± 2.11**	97.78 ± 4.44	96.83 ± 4.38	93.87 ± 4.03	97.78 ± 4.44	97.78 ± 4.44	95.87 ± 4.09	95.77 ± 4.11	80 ± 11.31	78 ± 14.58	85 ± 13.87	85 ± 12.28
Home	93.23 ± 3.9	95.22 ± 2.89	94.18 ± 3.56	**96.18 ± 3.46**	94.28 ± 3.58	93.31 ± 2.36	95.22 ± 2.89	94.18 ± 3.56	0.0 ± 0.00	0.0 ± 0.00	0.0 ± 0.00	32 ± 7.92
Library	96.95 ± 4.03	96.95 ± 4.03	95.78 ± 5.18	95.77 ± 4.11	96.95 ± 4.03	**98.00 ± 4.00**	97.99 ± 2.46	95.79 ± 6.14	66 ± 13.63	67 ± 14.66	69 ± 11.95	66 ± 13.10
Metro Station	**99.05 ± 1.90**	**99.05 ± 1.90**	98.00 ± 2.46	97.13 ± 2.36	**99.05 ± 1.90**	**99.05 ± 1.90**	97.99 ± 2.46	97.13 ± 2.36	47 ± 17.36	0.0 ± 0.00	69 ± 9.90	69 ± 9.80
Office	96.94 ± 2.50	**97.90 ± 2.58**	97.03 ± 2.45	96.08 ± 3.73	96.94 ± 2.50	96.94 ± 2.50	96.94 ± 2.50	96.95 ± 4.03	52 ± 4.20	52 ± 4.29	55 ± 4.48	55 ± 3.39
Park	93.96 ± 4.06	95.13 ± 3.18	93.67 ± 6.24	93.60 ± 5.93	93.38 ± 3.82	**96.08 ± 3.73**	93.67 ± 6.24	93.60 ± 5.93	61 ± 7.90	60 ± 9.91	70 ± 5.17	73 ± 7.94
Residential Area	91.86 ± 3.86	92.82 ± 5.00	89.21 ± 3.93	89.11 ± 3.77	92.82 ± 5.00	**93.67 ± 3.97**	91.97 ± 5.05	91.10 ± 3.59	45 ± 9.81	39 ± 14.25	59 ± 6.71	61 ± 6.74
Train	94.72 ± 4.73	93.45 ± 4.19	93.42 ± 6.67	92.24 ± 6.07	**95.67 ± 4.13**	94.50 ± 4.97	92.51 ± 5.44	89.62 ± 9.58	72 ± 9.40	71 ± 6.08	81 ± 7.38	81 ± 7.38
Tram	94.42 ± 1.39	93.65 ± 3.36	94.43 ± 4.20	90.67 ± 5.95	94.43 ± 4.20	**96.51 ± 3.17**	92.69 ± 2.31	91.31 ± 6.57	81 ± 9.57	79 ± 8.84	89 ± 4.44	88 ± 4.73
Mean Ranks	**3.80**	**3.30**	**5.13**	**6.53**	**3.93**	**3.20**	**4.47**	**5.63**	**11.23**	**11.63**	**9.67**	**9.47**

We have followed up the previous findings with a Nemenyi test at a 5% significance level with the critical value and the critical difference. The results of this test are depicted in Figure 4.1.

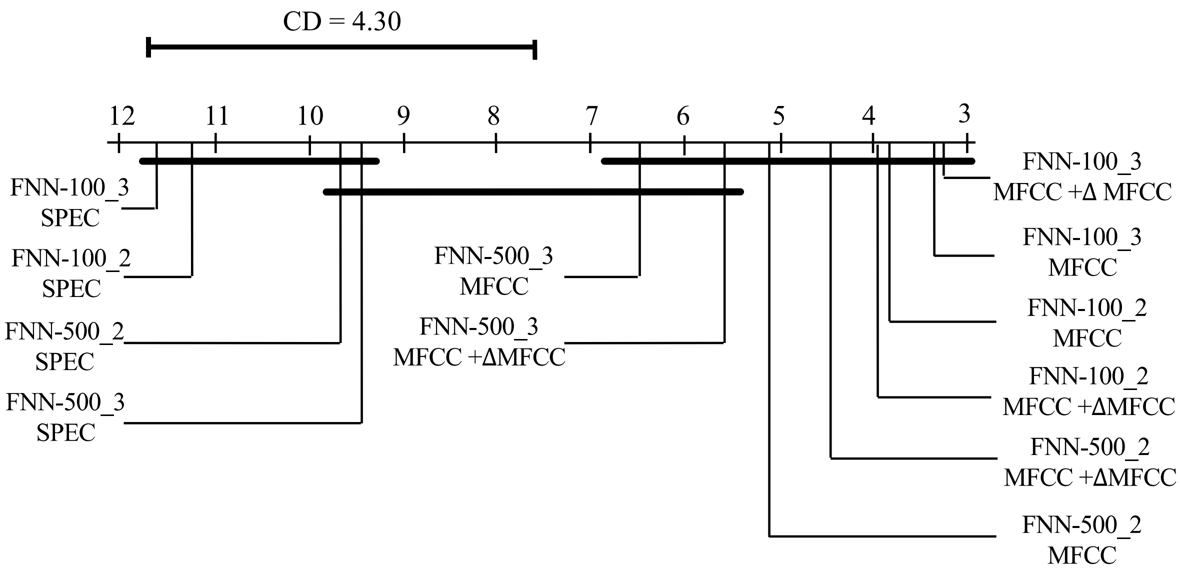


Figure 4.1. Comparison of all systems against each other with the Nemenyi test. Groups of techniques that are not significantly different (at $\alpha = 0.05$) are connected.

The analysis of the previous results can be summarized as follows. Nemenyi test indicates that FNN with MFCC or MFCC+ ΔMFCC are significantly better than SPEC ones. Specifically, FNN with MFCC or MFCC+ ΔMFCC require less epochs, whereas, FNN with SPEC entail more iterations. Additionally, **we observe that ΔMFCC features do not have an important impact on the generalization ability of FNN. We can also conclude that FNN with SPEC may not have converged yet**. Possibly, we can improve these results by increasing the number of epochs, hidden layers, by using more complex neural network architectures, or even introducing a feature selection step.

B. SVM Classifier

The results of this experiment are given in Table 4.2. Columns 2 and 3 represent the F1-score rates scored by MFCC and MFCC+ ΔMFCC, respectively. Columns 4 specifies the difference in performance between these two models, whereas, column 5 indicates the rank of differences for each row entry.

Table 4.2. F1 Scores using SVM.

Scenes	MFCC	MFCC+ ΔMFCC	Difference	Ranks
Beach	85.50±09.35	**86.65±10.25**	1,15	7
Bus	**79.60±08.10**	74.92±11.18	-4,68	-13
Cafe/Restaurant	**89.88±03.72**	89.12±03.19	-0,76	-5
Car	**68.73±14.10**	67.87±09.64	-0,86	-6
City Center	91.54±04.76	**92.13±03.82**	0,59	3
Forest Path	82.30±07.99	**83.67±14.56**	1,37	9
Grocery Store	**94.47±06.46**	88.70±08.98	-5,77	-14
Home	**85.23±11.11**	84.85±07.50	-0,38	-1
Library	76.97±19.43	**79.25±07.78**	2,28	10
Metro Station	**96.44±03.38**	89.91±06.21	-6,53	-15
Office	**93.95±05.06**	92.67±03.74	-1,28	-8
Park	**86.77±07.34**	86.35±08.43	-0,42	-2
Residential Area	71.81±09.31	**75.16±04.93**	3,35	11
Train	**57.28±15.46**	53.79±08.19	-3,49	-12
Tram	59.17±12.50	**59.90±07.29**	0,73	4

Analysis and discussion

The next step consists of testing the influence of **Δ**MFCC features on the generalization performance of SVM using Wilcoxon signed-ranks test. We have found: the sum for ranks for positive and negative differences $R^{+} = 44$, $R^{-} = 76$, and the statistics $z = -0.91 > -1.97$ for a significance level $\alpha = 0.05$. The value of z provides a strong evidence that the observed differences are not significant; hence, these two models perform similarly. **We can conclude that adding ΔMFCC features do not improve the predictive performance in this case.**

C. KNN Classifier

Table 4.3 shows the average F1-score results of KNN Classifier. The last row designates the mean rank of each method over all scenes.

Table 4.3. F1 Scores using KNN.

	MFCC				**MFCC+ ΔMFCC**			
K **Scenes**	10	20	30	40	10	20	30	40
Beach	57.10±2.46	**58.32±2.21**	56.20±2.45	56.20±2.45	56.26±3.11	56.79±2.31	55.89±2.09	56.16±1.99
Bus	**92.46±6.19**	91.59±6.04	91.59±6.04	91.59±6.04	**92.46±6.19**	91.59±6.04	91.59±6.04	91.59±6.04
Cafe/Restaurant	**86.70±4.76**	85.54±4.51	85.54±4.51	84.10±7.14	85.54±4.51	85.54±4.51	85.54±4.51	84.10±7.14
Car	**98.10±2.33**	97.14±2.33	97.14±2.33	97.14±2.33	**98.10±2.33**	97.14±2.33	97.14±2.33	97.14±2.33
City Center	**98.33±3.33**	**98.33±3.33**	**98.33±3.33**	97.38±3.40	**98.33±3.33**	**98.33±3.33**	**98.33±3.33**	97.38±3.40
Forest Path	**97.99±2.46**	97.04±2.42	**97.99±2.46**	**97.99±2.46**	**97.99±2.46**	**97.99±2.46**	**97.99±2.46**	**97.99±2.46**
Grocery Store	**86.47±4.58**	**86.47±4.58**	**86.47±4.58**	85.69±4.96	**86.47±4.58**	**86.47±4.58**	**86.47±4.58**	85.69±4.96
Home	**87.50±8.65**	86.26±10.44	84.14±8.35	83.84±10.18	**87.50±8.65**	85.31±9.29	84.14±8.35	83.84±10.18
Library	**81.17±5.84**	79.57±5.88	77.42±6.72	76.75±7.53	79.89±5.73	78.33±7.01	77.42±6.72	77.03±8.01
Metro Station	**85.90±10.71**	**85.90±10.71**	84.85±9.98	84.85±9.98	**85.90±10.71**	84.85±9.98	84.85±9.98	84.85±9.98
Office	**95.99±3.75**	**95.99±3.75**	**95.99±3.75**	**95.99±3.75**	**95.99±3.75**	**95.99±3.75**	**95.99±3.75**	**95.99±3.75**
Park	**94.84±5.77**	93.97±5.23	93.97±5.23	93.97±5.23	**94.84±5.77**	93.97±5.23	93.97±5.23	93.97±5.23
Residential Area	**79.80±5.28**	78.37±6.48	76.90±6.24	76.90±6.24	78.33±5.39	78.37±6.48	76.90±6.24	76.90±6.24
Train	**74.48±5.46**	73.42±7.52	71.98±7.23	70.91±8.78	**74.48±5.46**	74.25±6.41	72.81±6.26	70.91±8.78
Tram	**80.69±7.55**	**80.69±7.55**	79.45±8.61	79.45±8.61	**80.69±7.55**	**80.69±7.55**	79.45±8.61	79.45±8.61
Mean Ranks	**2.17**	**3.97**	**5.17**	**6.33**	**2.77**	**4.07**	**5.27**	**6.37**

Analysis and discussion

First, we statistically compared the performances of these systems using the average ranks over 15 scenes. Friedman test rejects the null hypothesis that all systems perform similarly with $FF = 9.44 > F(7,98) = 9.23$ for $\alpha = 1.0 \times 10^{-8}$. Then, we have tested the pairwise significance differences using a Nemenyi test at a 5% significance level with the critical value $q_{0.05} = 3.04$ and the critical difference $CD = 2.72$.

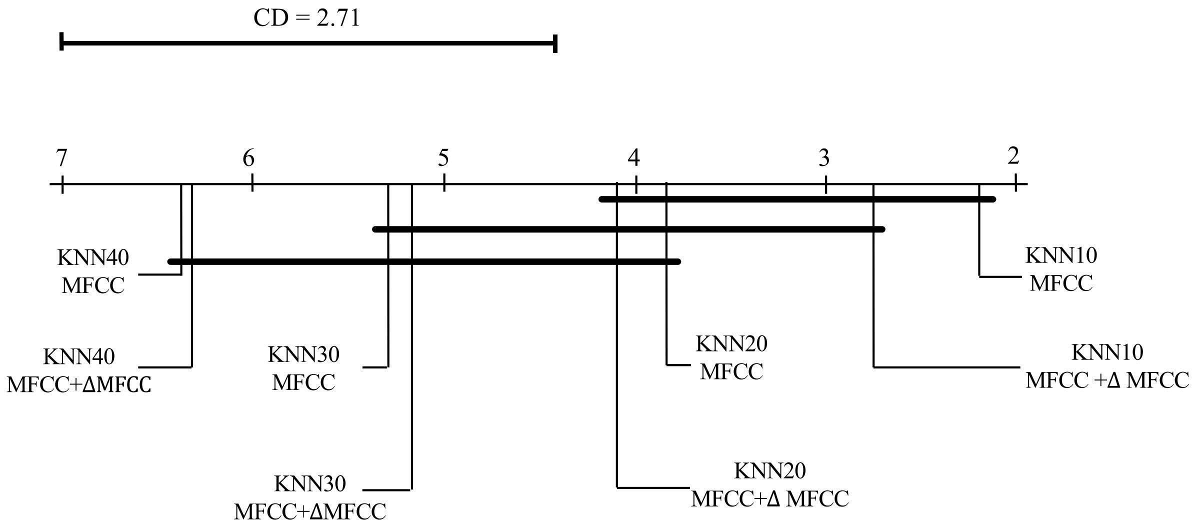


Figure 4.2. Comparison of all systems against each other with the Nemenyi test. Groups of techniques that are not significantly different (at α = 0.05) are connected.

The analysis of the test results, illustrated by Figure 4.2, confirms our previous findings regarding ΔMFCC coefficients. It provides **a strong evidence that ΔMFCC coefficients do not significantly improve the predictive performance of the system in this case**. Moreover, as the number of nearest neighbors increases, the generalization ability deteriorates significantly; for instance, KNN10+MFCC and KNN30+MFCC are not connected, which proves the superiority of the first system over the second.

D. FNN vs SVM vs KNN

As a final step of this case study, we have compared the performances of the top two models discussed in each scenario, namely FNN-100_3 MFCC and FNN-100_3 MFCC+ΔMFCC, KNN10 MFCC and KNN10 MFCC+ΔMFCC, with the SVM-based systems. We have conducted a Friedman test, assuming that all systems perform similarly. This test rejects the null hypothesis with $FF = 32.85 > F(5{,}70) = 26.78$ for $\alpha = 5.0 \times 10^{-15}$. Then, we have tested the pairwise significance differences using a Nemenyi test at a 5% significance level with the critical value $q_{0.05} = 2.85$ and the critical difference $CD = 1.95$.

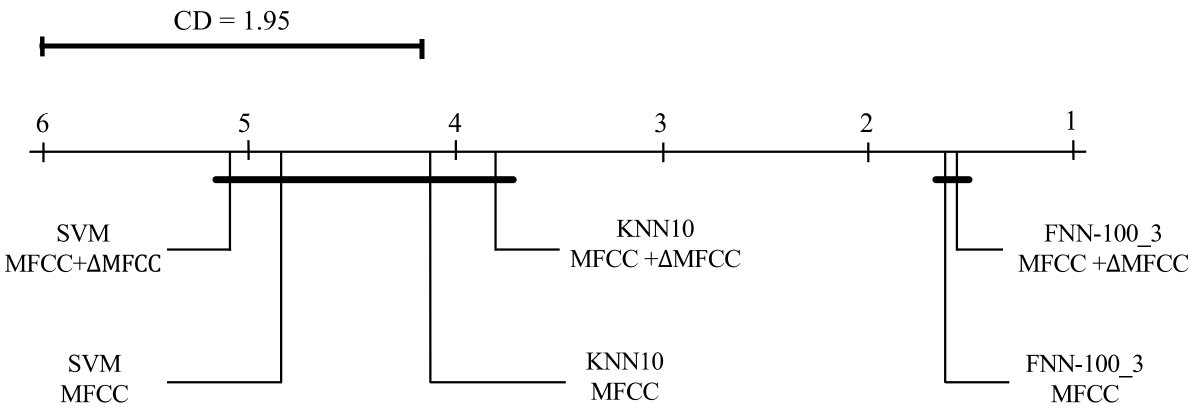


Figure 4.3. Comparison of all systems against each other with the Nemenyi test. Groups of techniques that are not significantly different (at α = 0.05) are connected.

Analysis and discussion

The pairwise comparisons given by Nemenyi test (Figure 4.3) reveal the existence of three categories of systems: FNN, KNN, and SVM variants from the best performing approach to the worst one. Specifically, these results demonstrate the superiority of FNN over KNN and SVM, which is expected since FNN usually exhibits remarkable scores when enough data are provided. However, data are not sufficient to determine the superiority of KNN over SVM. Furthermore, ΔMFCC+MFCC-based systems perform similarly to MFCC ones.

Based on this and on our previous findings, we can conclude that **higher number of features is not always beneficial and can negatively affect the performance of the classification system**. In the next section, we will investigate the influence of number of features on the generalization ability of the system. We will include before classification a feature selection step which automatically determines the most effective subset of MFCC and ΔMFCC coefficients.

4.3. Case study 2

Table 4.4 gives the average F1-score results of the second case study.

Table 4.4. FNN F1-Scores using multiple combinations of selected features with JMI.

SF Scenes	FNN-15F	FNN-20 F	FNN-25 F	FNN-30 F	FNN-35 F	FNN-40 F	FNN-45 F	FNN-50 F	FNN-55 F	FNN-60 F	FNN-65 F	FNN-70 F	FNN-75 F	FNN-80 F
Beach	87.57±6.62	89.74±5.13	91.38±4.12	89.75±5.82	90.54±5.65	90.59±3.57	90.59±3.57	90.59±3.57	**91.38±4.12**	**91.38±4.12**	90.59±3.57	90.59±3.57	90.54±5.65	90.59±3.57
Bus	91.60±3.52	93.59±2.19	95.32±0.17	95.32±2.88	95.57±4.77	99.13±1.74	99.13±1.74	99.13±1.74	99.13±1.74	99.13±1.74	99.13±1.74	99.13±1.74	99.13±1.74	**100.0±0.00**
Cafe/Restaurant	83.01±6.45	91.59±3.76	92.23±4.57	94.59±1.50	94.59±1.50	94.44±1.80	94.69±3.25	**96.44±1.79**	**96.44±1.79**	95.64±2.64	**96.44±1.79**	95.49±2.88	95.64±2.64	**96.44±1.79**
Car	96.51±3.17	**98.26±2.13**	**98.26±2.13**	97.31±3.62	97.31±3.62	96.26±3.69	97.39±2.13	**98.26±2.13**	**98.26±2.13**	**98.26±2.13**	**98.26±2.13**	**98.26±2.13**	**98.26±2.13**	**98.26±2.13**
City Center	95.56±3.91	95.72±5.44	97.38±3.40	96.52±4.27	96.51±3.17	96.52±4.27	95.72±5.44	**99.13±1.74**	96.51±3.17	98.33±3.33	97.46±3.35	98.33±3.33	96.51±3.17	97.46±3.35
Forest Path	96.94±2.50	96.95±4.03	96.94±2.50	**98.95±2.11**	97.89±2.58	97.89±2.58	**98.95±2.11**	**98.95±2.11**	97.89±2.58	**98.95±2.11**	97.89±2.58	**98.95±2.11**	**98.95±2.11**	**98.95±2.11**
Grocery Store	82.53±7.26	92.05±5.12	94.51±3.22	92.88±6.16	95.16±4.44	97.28±3.47	95.96±4.08	96.91±4.35	96.91±4.35	96.91±4.35	**98.95±2.11**	**98.00±4.00**	96.83±4.38	97.78±4.44
Home	86.73±8.52	89.49±6.19	90.98±4.25	93.23±3.90	91.20±4.16	92.18±3.92	93.13±3.85	92.18±3.92	93.23±3.90	94.18±4.76	93.31±2.36	92.18±3.92	**94.28±3.58**	93.31±2.36
Library	88.28±4.33	91.52±6.11	93.47±4.71	94.26±2.15	94.33±3.43	95.13±3.18	96.08±3.73	96.08±3.73	96.08±3.73	97.13±3.94	96.95±4.03	96.08±3.73	96.95±4.03	**98.00±4.00**
Metro Station	91.47±7.11	92.80±5.57	95.99±3.75	93.67±6.24	96.94±2.50	97.05±3.98	95.99±3.75	97.99±2.46	97.99±2.46	97.99±2.46	**99.05±1.90**	**99.05±1.90**	97.99±2.46	**99.05±1.90**
Office	96.94±2.50	96.07±1.99	**97.89±2.58**	96.94±2.50	96.94±2.50	96.94±2.50	96.95±4.03	96.94±2.50	95.99±3.75	96.94±2.50	**97.89±2.58**	95.99±3.75	96.94±2.50	96.94±2.50
Park	91.07±6.19	92.33±6.72	93.96±4.06	96.94±2.50	94.11±4.89	94.18±3.80	96.94±2.50	95.05±4.47	**97.03±2.45**	95.05±4.47	93.97±6.11	95.77±4.11	96.08±3.73	96.08±3.73
Residential Area	86.19±5.76	89.35±3.35	90.84±4.00	92.40±2.86	90.51±3.96	93.67±3.97	91.65±5.09	**95.04±5.22**	92.82±5.00	93.87±5.78	93.77±4.00	92.92±5.04	91.76±3.78	93.67±3.97
Train	81.69±8.77	86.11±3.23	88.04±4.39	93.45±4.19	92.62±2.62	93.45±4.19	94.50±4.97	94.62±3.52	**95.67±4.13**	**95.67±4.13**	93.31±5.85	**95.67±4.13**	93.45±4.19	94.50±4.97
Tram	79.24±6.83	84.31±4.07	86.35±3.23	90.62±2.67	92.19±4.94	90.15±4.66	92.62±3.26	94.43±3.45	96.33±3.24	94.43±4.20	92.84±4.76	95.39±4.78	94.52±4.22	**96.51±3.17**

Analysis and discussion

Because we are only interested in testing whether feature selection significantly improves the predictive performance, we have carried out pairwise comparisons between FNN-80F (system with 80 features) with each of the above systems. Due to its robustness, we have considered using the Wilcoxon signed-ranks tests. A summary of this test statistics is shown in Table 4.4. The first row specifies the number of win/tie/loss of the system in the row over the system in the column. The second shows the p-values; it is worth noting that $p - values \leq 0.05$ indicates that the system in the column is significantly worse than the system in the row at 5% significance level i.e. feature selection step has a negative effect on the predictive performance.

Table 4.5. Summary of Wilcoxon signed-ranks test statistics. Differences at 5% significance level are marked with $*$, and at 1% with +.

		Number of selected features					
		FNN-20F	FNN-25F	FNN-30F	FNN-35F	FNN-40F	FNN-45F
FNN-80F	*W/T/L*	$0/1/\mathbf{14}$	$2/1/\mathbf{12}$	$1/2/\mathbf{12}$	$0/1/\mathbf{14}$	$0/3/\mathbf{12}$	$2/3/\mathbf{10}$
	p-value	$7.2 \times 10^{-4^{+}}$	$2.9 \times 10^{-3^{+}}$	$2.4 \times 10^{-3^{+}}$	$7.2 \times 10^{-4^{+}}$	$1.2 \times 10^{-3^{+}}$	$7.6 \times 10^{-3^{+}}$
		FNN-50F	FNN-55F	FNN-60F	FNN-65F	FNN-70F	FNN-75F
FNN-80F	*W/T/L*	$3/5/\mathbf{7}$	$3/2/\mathbf{10}$	$5/3/\mathbf{7}$	$3/\mathbf{6}/\mathbf{6}$	$3/4/\mathbf{8}$	$1/4/\mathbf{10}$
	p-value	2.2×10^{-1}	8.8×10^{-2}	3.06×10^{-1}	2.2×10^{-1}	1.2×10^{-1}	$1.06 \times 10^{-2^{*}}$

Based on the test results, we can reach the following conclusions. Selecting less than 60% (around 47 features) of the number of features considerably deteriorates the F1-score results, whereas, keeping more than 60% results in systems with statistically similar performances.

In order to get an insight into the nature of the most relevant features, we represent in Figure 4.4 the rate of MFCC et ΔMFCC chosen for each system.

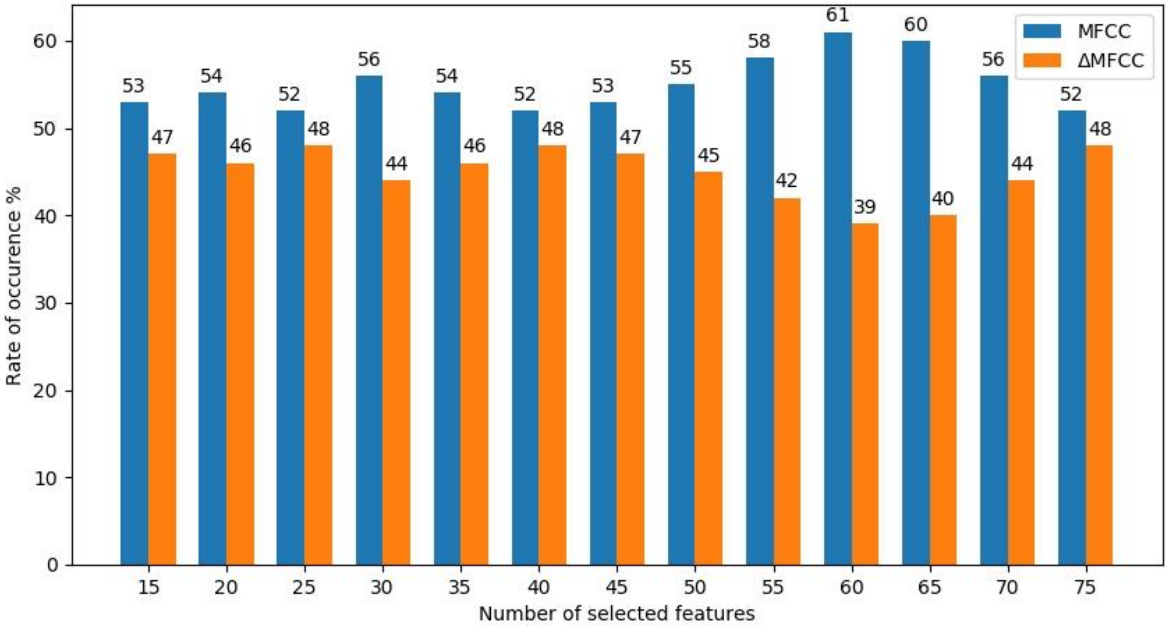


Figure 4.4. Rate of selected MFCC and ΔMFCC for each system

Figure 4.4 indicates that the selection criterion JMI promotes MFCCs over ΔMFCC. Most importantly, the rate of selected MFCC ranges from 50% to 60%, while the ΔMFCC falls between 40% and 50%. Recall that the systems trained with number of features between 47 to 70 are statistically similar to FNN-80F. These approaches were built using 40% ΔMFCC and 60% MFCC. Moreover, as the number of selected ΔMFCC increases, the generalization ability of these systems deteriorates considerably.

4.4. Execution time

As the high complexity of the data used for these experiments, it took a very long time to finish the execution, especially during the training phase. For each model, it took around **6-7 hours to train it on a single fold**. It also took around **8-9 hours to upload all the data to the Google Drive**.

4.5. Conclusion

In this chapter, we have presented the results of our experimental enquiries. Several lessons can be derived from our analysis:

- FNN-based systems have demonstrated superiority over SVM and KNN ones. However, data are not sufficient to determine which one is significantly better KNN or SVM.

- ΔMFCC features do not significantly improve the predictive performance of system built using FNN, SVM, or KNN.
- As the number of nearest neighbors increases, the generalization ability of KNN deteriorates significantly. We recommend using less neighbors $K \leq 10$.
- Higher number of features is not always beneficial and can negatively affect the performance of the classification system.
- Selecting less than 60% (around 47 features) of the number of features considerably deteriorates the F1-score results, whereas, keeping more than 60% results in systems with statistically similar performances.
- Systems that were built using 40% ΔMFCC and 60% MFCC can increase the generalization ability of FNN.

Conclusion

This thesis is devoted to study Acoustic Scene Classification systems. The initial goal was to provide beginners in this field with practical guidelines for building such systems. To address this matter, we have designed several experimental case studies using a huge dataset made of thousands of sound files. Our contributions are two folds:

First, we have conducted extensive experimental comparison among sound analysis methods. We have thoroughly examined 3 acoustic features (MFCC, MFCC+ΔMFCC, and Spectrograms) and 3 classification paradigms (Feed Forward Neural Network, K Nearest Neighbors, Support Vector Machine), while varying their parameters. Four lessons can be learned from the analysis of the obtained results:

- FNN-based systems have demonstrated superiority over SVM and KNN ones. However, data are not sufficient to determine which one is significantly better KNN or SVM.
- ΔMFCC features do not significantly improve the predictive performance of system built using FNN, SVM, or KNN.
- As the number of nearest neighbors increases, the generalization ability of KNN deteriorates significantly. We recommend using less neighbors K≤10.
- Higher number of features is not always beneficial and can negatively affect the performance of the classification system.

Second, we have investigated the effect of the number of features on the generalization performance. Specifically, we have invoked a feature selection approach, namely JMI, in order to automatically determine the optimal set of extracted features. To the extent of our knowledge, only very few attempts have considered incorporating feature selection step into the design of their systems. Based on our analysis, we can conclude that selecting less than 60% (around 47 features) of the number of features considerably deteriorates the F1-score results, whereas, keeping more than 60% results in systems with statistically similar performances. Most importantly, systems that were built using 40% ΔMFCC and 60% MFCC can increase the generalization ability of FNN.

Limits and Future work

This thesis has revealed several interesting areas for improvement. The first area is based upon the insights gained from the first case study. We can improve the results of FNN with SPEC by increasing the number of epochs, hidden layers, by using more complex neural network architectures, or even introducing a feature selection step. Another extension of this work would be testing other learning paradigms such as Gaussian Mixture Model (GMM) and ensemble-based learners like Adaboost, Bagging, Arcing; or even training using other feature sets.

Another appealing work direction would be to study in depth the impact of feature selection on the performance of scene classification systems, using other approaches like Mutual Information Feature Selection (MIFS) [3], Conditional MIFS (CMIFS) [4], min-Redundancy Max-Relevance (mRMR) [5].

During this project, we have encountered many struggles. The training of the learning models took a very long time due to the lack of dedicated computational platforms. In addition, when performing model selection, storing the trained classifiers caused a considerable increase in the usage of memory space.

Bibliography

[1] R. Serizel, V. Bisot, S. Essid and G. Richard, Acoutic Features for Environmental Sound Analysis, Tuomas Virtanen; Mark D. Plumbley, 2017.

[2] G. Brown, A. Pocock, M.-J. Zhao and M. Luján, "Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection," p. 27−66, 2012.

[3] R. Battiti., "Using mutual information for selecting features in supervised neural net learning.," *IEEE*, vol. 5, no. 4, p. 537–550, 1994.

[4] H. Cheng, Z. Qin, C. Feng, Y. Wang and F. Li., "Conditional mutual information-based feature selection analyzing for synergy and redundancy," *Electronics and Telecommunications Research Institute (ETRI) Journal*, vol. 33, no. 2, 2011.

[5] H. Peng, F. Long and a. C. Ding, "Feature selection based on mutual information: Criteria of max dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, p. 1226–1238, 2005.

[6] J. Moody and H. Yang, "Data visualization and feature selection: New algorithms for non-gaussian data," *Advances in Neural Information Processing Systems*, p. 12, 1999.

[7] Khunarsal, Lursinsap, Peerapol, Raicharoen, Chidchanok and Thanapant., Very short time environmental sound classification based on spectrogram pattern matching, Information Sciences, 2013.

[8] Shuiping, W. Zhenming and T. Shiqiang, "Design and Implementation of an Audio Classification System Based on SVM," in *Procedia Engineering*, 2011, pp. 4031-4035.

[9] A. Mesaros, T. Heittola and T. Virtanen., Acoustic Scene Classification: an Overview of DCASE 2017 Challenge Entries, International Workshop on Acoustic Signal Enhancement, 2018.

[10] a. Shikha G., "Feature Extraction using MFCC," *Signal & Image Processing: An International Journal (SIPIJ)* , vol. 4, no. 4, 2013.

[11] R. Serizel, V. Bisot, S. Essid and G. Richard, Acoutic Features for Environmental Sound Analysis, Tuomas Virtanen; Mark D. Plumbley, 2017.

[12] X. Valero and F. Al´ıas, "Gammatone cepstral coefficients: biologically inspired features for nonspeechaudio classification," . *IEEE Trans Multimed*, vol. 14, no. 6, p. 1684–1689, 2012.

[13] Gaurav and Naithani, "Acoustic Analysis of Infant Cry Signals," *Tuomas Virtanen,* 2015.

[14] T. Virtanen, M. D. Plumbley and D. Ellis., "Acoustic Features for Environmental Sound Analysis," in *Computational Analysis of Sound Scenes and Events*, Springer, 2017.

[15] Herrera, S. Garc´ıa and Francisco, "An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons," in *Journal of Machine Learning Research*, vol. 9, 2009, p. 2677–2694.

[16] G. Parascandolo, "Recurrent Neural Networks For Polyphonic Sound Event Detection," Tempere university of technology, Tempere, USA, 2015.

[17] Kevin, L. Priddy and P. E. Keller, "Artificial Neural Network: An Introduction," *The International Society for Optical Engineering,* 2005.

[18] S. Russell and P. Norvig, Artificial Intelligence: a modern approach, Pearson, 2010.

[19] Srivastava, N. Hinton, G. Krizhevsky, A. Sutskever, I. Salakhutdinov and Ruslan., "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research,* vol. 15, pp. 1929-1958, 2014.

[20] G. E. D. G. H. Abdel-Rahman Mohamed, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 20, no. 1, p. 14–22, 2012.

[21] T. Hastie, R. Tibshirani and J. Friedman, "The elements of Statistical Learning," *Springer,* p. 134, 2009.

[22] P. Kalaivani and K. L. Shunmuganathan, "An improved K-nearest-neighbor algorithm using genetic algorithm for sentiment classification," in *IEEE*, 2014.

[23] S. Jiang, G. Pang, M. Wu and L. Kuang, "An improved K-nearest-neighbor algorithm for text categorization," *Expert Systems with Applications,* vol. 39, no. 1, pp. 1503-1509, 2012.

[24] M. Kuhn and K. Johnson., Applied Predictive Modeling, Springer, 2013, p. 70.

[25] G. C. and G. E., A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation, vol. 3408, ECIR, Ed., Berlin, Heidelberg: Springer, 2005.

[26] S. Garcıa, A. Fernandez, J. Luengo and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," in *Information Sciences*, vol. 180, Elsevier, 2010, p. 2044–2064.

[27] Shah, N. Japkowicz and Mohak, Evaluating learning algorithms, Cambridge: Cambridge University Press, 2011.

[28] A. C and Pocock, "Feature Selection Via Joint Likelihood," University of Manchester, 2012.

[29] T. Carneiro, Nóbrega, R. V. M. Da, Nepomuceno, Thiago, Bian, Gui-Bin, V. H. C. D. Albuquerque, Filho and P. P. Rebouças, Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications, IEEE, 2018.

[30] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg and O. Nieto, "librosa: Audio and Music Signal Analysis in Python," in *14Th Python In Science Conf,* 2015.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher and P, Scikit-learn: Machine Learning in Python, JMLR, 2011.

[32] F. hollet, "Keras.," 2015.

[33] S. García and F. Herrera, "An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons.," *Journal of Machine Learning Research,* vol. 2677–2694, p. 9, 2009.