NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES

DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

PROGRAM OF POSTGRADUATE STUDIES

“BIG DATA AND ARTIFICIAL INTELLIGENCE”

MSc THESIS

Deep Learning Methods for Auditory Scene Analysis

Georgios K. Charitos

ATHENS

JUNE 2022

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

«ΜΕΓΑΛΑ ΔΕΔΟΜΕΝΑ ΚΑΙ ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ»

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Μέθοδοι Ανάλυσης Ακουστικών Σκηνών με Νευρωνικά Δίκτυα»

Γεώργιος Κ. Χαρίτος

ΑΘΗΝΑ

ΙΟΥΝΙΟΣ 2022

MSc THESIS

Deep Learning Methods for Auditory Scene Analysis

Georgios K. Charitos
DS1.19.0019

SUPERVISOR: Stavros Perantonis, Professor UoA

THREE-MEMBER ADVISORY COMMITTEE:

Stavros Perantonis, Professor UoA

Haris Papageorgiou, Professor UoA

Theodoros Giannakopoulos, NCSR "Demokritos" Researcher (B)

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ

Μέθοδοι Ανάλυσης Ακουστικών Σκηνών με Νευρωνικά Δίκτυα

Γεώργιος Κ. Χαρίτος
DS1.19.0019

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Σταύρος Περαντώνης, Καθηγητής ΕΚΠΑ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:

Σταύρος Περαντώνης, Καθηγητής ΕΚΠΑ

Χάρης Παπαγεωργίος, Καθηγητής ΕΚΠΑ

Θεόδωρος Γιαννακόπουλος, Ερευνητής (Β), ΕΚΕΦΕ «ΔΗΜΟΚΡΙΤΟΣ»

ABSTRACT

Real world environments consist of different acoustic scenes. From a hectic city street to a calming suburban village, sounds do not come from a unique outcome, but it is a combination of a wide range of sounds that derive from different origins.

Environmental audio scene and sound event recognition is the main topic of the present Thesis. It will utilize and analytically present modern Deep Learning and Transfer Learning techniques in order to analyze and study the quality of different environments. For this purpose, it will use real-world data that have been collected and annotated by human annotators.

In the present Thesis many experiments under a wide range of setups have been made, including classic Deep Learning as far as modern and top-tier Transfer Learning techniques to solve the audio classification problem.

The goal of this work is to provide DL developers with an end-to end solution on audio analytics and recognition of soundscape quality in both urban and rural areas, which may lead to strong tools to reduce noise pollution for a better and more sustainable urban living.

SUBJECT AREA: Auditory Scene Analysis

KEYWORDS: soundscape analysis, audio classification, deep learning, transfer learning, audio analysis, sound recognition

ΠΕΡΙΛΗΨΗ

Ο πραγματικός κόσμος αποτελείται από διάφορες και ποικίλες ακουστικές σκηνές. Από δρόμους γεμάτους κίνηση των σύγχρονων μεγαλουπόλεως μέχρι ήρεμα εξοχικά χωριά, οι διάφοροι ακουστικοί ήχοι δεν προέρχονται από μια και μοναδική πηγή αλλά είναι ένας συνδυασμός διαφόρων ακουστικών προελεύσεων.

Η ανάλυση των περιβαλλοντικών ηχητικών σκηνών καθώς και η αναγνώριση γεγονότων αποτελεί το κύριο θέμα της παρούσας Διπλωματικής Εργασίας. Χρησιμοποιούνται και παρουσιάζονται αναλυτικά οι πλέον σύγχρονες τεχνικές Νευρωνικών Δικτύων και Βαθιάς Μάθησης με στόχο την ανάλυση και μελέτη της ποιότητας των διαφόρων ακουστικών πηγών σε μια πληθώρα διαφορετικών περιβαλλόντων. Προς το σκοπό αυτό, όλα τα δεδομένα για την ανάλυση έχουν συλλεχθεί και αναγνωρισθεί με τη βοήθεια ανθρώπινου παράγοντα.

Στη παρούσα Διπλωματική Εργασία έχουν διατυπωθεί και αναλυθεί πολλά και διάφορα πειράματα, με διαφορετικές παραμέτρους κάθε φορά, όπως κλασσικές αρχιτεκτονικές Βαθιάς Μάθησης και Νευρωνικών Δικτύων καθώς και κορυφαίες τεχνικές μεταφοράς γνώσεις μεταξύ διαφόρων Νευρωνικών Δικτύων για προβλήματα κατηγοριοποίησης.

Απώτερος σκοπός της παρούσας Εργασίας είναι να προσφέρει και να καθοδηγήσει νέους προγραμματιστές Νευρωνικών Δικτύων μια ολοκληρωμένη λύση στην ανάλυση ηχητικών σκηνών τόσο σε ήρεμα όσο και σε πολύ θορυβώδη περιβάλλοντα, που ίσως τους οδηγήσει μελλοντικά στη δραστική μείωση του θορύβου εντός των μεγαλουπόλεων για μια πιο βιώσιμη κατοίκιση εντός αυτών.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Ανάλυση Ακουστικών Σκηνών

ΛΕΞΕΙΣ - ΚΛΕΙΔΙΑ: ανάλυση ήχου, κατηγοριοποίηση ήχου, βαθιά μάθηση, μεταφορά γνώσης, ανάλυση ήχου, αναγνώριση ήχου

AKNOWLEDGEMENTS

Special thanks must go to my supervisor, Theodoros Giannakopoulos, for his enthusiastic support of this work. He has been the perfect mentor, offering the best possible advice and encouragement. I am proud of his patience and grateful for working with such a motivational and knowledgeable Researcher

My professor, Stavros Perantonis, taught the course of Machine Learning in the first semester of my Master Studies. I knew from the very beginning that is the field of science that I would like to follow and get through. I really want to thank him for being such an inspiring personality and helping me on my first academic steps.

Harris Papageorgiou completed the puzzle in the second semester with the course of Deep Learning. Among his lectures, he gave me the chance to get into Deep Learning in depth an understand even better the utter need of it and its existence.

Also, I would like to thank the National and Kapodistrian University of Athens for the curriculum and the structure of the Master Program.

Finally I would like to thank all my colleagues of this Master Program for their devotion, collaboration, guidance and laughs.

Thanks to my parents and my beloved brother for successful career advice.

Immense gratitude, as always, to my wife Kelly for her true love, inexhaustible patience and perpetual support.

CONTENTS

FIGURES LIST

PICTURES LIST

TABLES LIST

PREFACE

In 2017 one of my colleagues and I started diving into Data Science and Analytics. It was the first time I encountered so much information behind raw numbers and simple words. “What is this?” I asked myself. I could never imagine that you can uncover so much information that will definitely lead to data driven decisions. I was so much excited about this field of science that I decided, 2 years later, to start attending a dedicated Master Program in National and Kapodistrian University of Athens (UoA).

The program was very challenging and required a lot of hard work and effort to get through it successfully.

My first lectures were about Machine and Deep Learning. I was so eager to uncover more about this era since it seemed to my eyes very promising. At the end of the program Mr Perantonis, my professor at the Machine Learning course, introduced me Theodoros Giannakopoulos, which finally became my supervisor of the present senior Thesis. Together and after his meticulous guidance I managed to get into the audio world. He has been the perfect mentor, offering the best possible advice and encouragement. I am proud of his patience and grateful for working with such a motivational and knowledgeable Researcher.

This work aims to help developers to engage in end-to-end Deep Learning problems that revolve around to audio classification. It will provide them with lots of information in the way of collecting and preprocessing the data, building the models and finally make the appropriate inference

Finally, to my other colleagues at UoA, I would like to thank you for your amazing collaboration as well. It was always helpful to bat ideas about my research around with you. I also benefitted from debating issues with my friends and family. If I ever lost interest, you kept me motivated. My wife Kelly deserves a particular note of thanks.

I hope you enjoy your reading.

Georgios Charitos

Athens, June 2022

1. INTRODUCTION

1.1. Motivation

Real life includes many diverge acoustic environments. From a very calm village area to a hectic downtown street, different sounds do not derive from a unique source, but it is a complex combination of a diversity of sounds that derive from different origins.

The present Master Thesis was inspired by the problem of soundscape quality and sound recognition and it is addressed in the context of which sound / audio gathered from different environments and corresponds to either a bad or good quality, in terms of sound pollution.

For this analysis, there is a vast array of different areas that systems dealing with the problem can be applied; amidst them, event detection, noise pollution among cities and during siesta hours, IoT etc.

To this end, it would be very beneficial for the researcher to be able to analyze sound clips from different environments and take measures accordingly for enhancing and improving humans' life.

1.2. Approach

Environmental audio scene and sound event recognition is the main topic of the present Thesis. It will utilize and analytically present modern Deep Learning and Transfer Learning techniques in order to analyze and study the quality of different environments. For this purpose, it will use real-world data that have been collected and annotated by human annotators.

In order to train and validate the models, Python programming environment is used and the main library is Deep Audio Features Package (DAFP). DAFP takes the audio clip as input, transforms the audio part into MEL Spectrograms and then applies CNN and DL techniques in order to classify the sound clips to different classes according to their quality.

1.3. Structure

In Chapter 2 of the present Thesis, basic concepts and background are presented; Terms like Machine Learning, Deep Learning, Reinforcement Learning, Linear Regression and classification are stated.

In Chapter 3, there is an introduction of Auditory Scene Analysis, Event Detection and Machine Listening. Also, there are proposed some Deep Learning methods for analyzing and predicting auditory scenes. Finally, the main dataset used for the purposes of the present Thesis is proposed and described.

In Chapter 4, all the experiments done are presented. The models evaluated are described meticulously and there is a small discussion at the end.

Finally, in Chapter 5 the overall conclusion is stated as far as future work and further analysis.

1.4. Related Work

The present Master Thesis is inspired by a lot of related work that has been made so far in the field of Soundscape Analysis and Event Detection.

Bregman proposed in [23], Auditory Scene Analysis focuses on the problem of hearing complex auditory environments, using a series of creative analogies to describe the

process required of the human auditory system as it analyzes mixtures of sounds to recover descriptions of individual sounds. In a unified and comprehensive way, Bregman establishes a theoretical framework that integrates his findings with an unusually wide range of previous research in psychoacoustics, speech perception, music theory and composition, and computer modeling.

Related work [26] proposed many methods for extracting different audio features according to any occasion. The solid audio feature selection plays a crucial role in audio surveillance of the environment. In fact, audio features are intended to grab the discriminative information useful for classification purposes while decreasing background noises and other redundancies. The complex recognition task with more data as discussed in related work [26], can be effectively managed by DL methods where classic ML methods cannot guarantee a very good performance.

CNN is one of the most popular NN architectures used in DL. The DL approach for ASA has been proposed in Petetin et al. (2015) using MFCC, spectral centroid, and spectral flatness features. DL model-based techniques outperformed the classical ML classifiers[1]. The results have been significantly good for DL with cepstral and frequency features compared with well-known features such as HOG classified by the SVM approach. In Han and Lee (2016), multi-width frequency-delta data augmentation was applied on input features for training using the CNN models. The frequency-delta features and Melspectrograms are used as input features for data augmentation to represent examples with same labels.

Another related work in Mafra et al. (2016) reviewed different time aspects when combining the features using different classic ML classifiers. This specific representation with temporal averaged Mel-log spectrograms using SVM achieved better recognition accuracy.

Also, the authors in Phan et al. (2017) suggested an approach called Convolutional Neural Network–Label Tree Embeddings (CNN-LTE) strategy. Using the CNN-LTE approach, the features were represented in the form of label tree embedding images. Then these features were learned using the simple 1D pooling layers of CNNs.

As far as the dataset is concerned and according to related work [27], ATHens Urban Soundscape (ATHUS) is a dataset of audio clips of audio clips from urban environments, which has been humanly annotated by proposing a specific soundscape quality for each clip.

The dataset was made publicly available (in http://users.iit.demokritos.gr/~tyianak/soundscape) as an audio feature representation form. In addition, in [27] is presented a basic method that shows how the specific dataset can be used to train supervised models in order for a developer to predict soundscape quality levels in different environments. In other words, the main purpose of this attempt was to provide to different developers and ML engineers, an introduction to audio recognition and soundscape analysis in different and diverge urban spaces, which could lead to powerful assessment tools in the hands of policy makers with regards to noise pollution and sustainable urban living.

[1] SVM classifiers

2. BACKGROUND

2.1. Artificial Intelligence

2.1.1. Introduction

In the modern times, most people are neither confident nor familiar with the term of Artificial Intelligence (AI). When 1,500 senior business leaders in the United States in 2017 were asked about AI, only 17 percent said they were familiar with it [12]. In spite of this lack of familiarity, AI is a tool that is transforming every sector of peoples' daily life and perspective. It is a super technology that makes humans to rethink how to analyze and process large amount of information, deal with a vast array of data, and use the resulting insights to improve decision making.

A very brief and naive answer to the question "*what is AI?"* depends on who is asked. One with a shallow knowledge of technology would link it to robots and complex machines. A researcher or programmer would claim that it is a set of rules or orders that can produce results without having to be explicitly instructed to do so.

AI is one of the newest scientific sectors in modern engineering and sciences. This attempt started after World War II, and the name itself was founded in 1956. AI is commonly referred as the "field I would most like to be in" by scientists of other disciplines. A student in physics might, without doubt, feel that all the good ideas have already been taken by Newton, Galileo and Einstein. AI currently is related to a huge variety of subfields, such as playing chess, proving mathematical theorems, writing poetry, driving a car on a crowded street, and diagnosing diseases.

AI, at the beginning, was treated with fear, followed by disappointment and the loss of enough funding. After that temporary crisis, AI research has tried and discarded many different approaches since its founding, including simulating the brain, modeling human problem solving, large databases of knowledge and imitating animal behavior. In the first decades of the 21st century, highly mathematical statistical machine learning has dominated the field, and this technique has proved highly successful, helping to solve many challenging problems throughout industry and academia.

AI is an exciting subfield of modern technology in the modern times, since it promises to change the world in the near future. AI had been a science fiction concept for many decades. People could never imagine that a simple machine can perceive humans' behavior, interact with them in an intermediate or even in a very advanced level.

Past decade has been proved a stepping stone for the development of AI. It has achieved a great invasion in humans' daily lives, for a vast array of tasks. It is not undoubted that in the near future, AI could probably give the trigger for a machine building that would be able to think, act and feel as an integrated human. Indisputably, driving force for this majestic concept is Machine Learning.

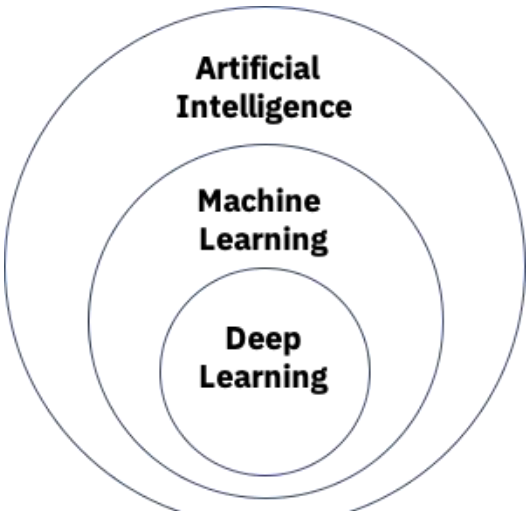


Figure 2.1: AI, ML and DL domains (Source: Google)

The difference between ML and AI is frequently misunderstood (Figure 2.1). ML learns and predicts based on past observations, whereas AI implies an agent interacting with the environment to learn and take actions that maximize its chance of successfully achieving its goals.

2.1.2. Types of Artificial Intelligence

AI can be categorized into two (2) different types; **Weak AI** and **Strong AI**.

As Week AI can be named this type of AI that is mainly trained and focused to perform specific everyday jobs. In other words Weak AI drives most of the AI that surrounds humans on everyday activities. Some of them are Apple's Siri, Amazon's Alexa, IBM Watson, and self-driving vehicles.

Strong AI, on the other hand is a type of AI in a rather theoretical form, where a machine could hold intelligence equal to humans. Furthermore, it could have an inherit consciousness and ability to learn how to solve problems and make specific plans for the near future. Strong AI is also known as super intelligence and surpasses both the intelligence and ability of the common human brain. While strong AI is still a theoretical notion with no practical examples yet, it does not mean that AI developers are not in favor of exploring and developing it further.

The fields that AI has a general application are the following:

- **Automation** process consists of a type of software that automates tasks traditionally done by humans. When combined with machine learning and AI tools, can automate bigger portions of enterprise jobs, enabling RPA's tactical bots to pass along intelligence from AI and respond to process changes.

- **Machine Learning** constructs algorithms that learn from data in order to make appropriate predictions. Such algorithms take into consideration experimental data in order to provide with exclusive data driven predictions.

- **Deep** Learning is a ML technique that tries to teach a machine to manipulate and understand input data in order to forecast, make inferences, classify or predict the desired result.

- **Computer Vision** gives to a machine the ability to see. This architecture analyzes visual information using a camera and digital signal processing and it is often compared to human eyesight.

- **Cognitive Computing,** algorithms try to imitate a human's brain and intelligence by processing and analyzing objects[2] in a way that a human performs and tries to give the desired result.

- **Natural Language Processing (NLP),** is the processing of human language by a computer program. One of the most known examples of NLP is spam detection, which looks at the subject line and text of an email and decides if it's junk. Current approaches to NLP are based on machine learning. NLP tasks include text translation, sentiment analysis and speech recognition.

- **Robotics** is the field of engineering focuses on the design and manufacturing of robots.

- **Self-driving cars** are vehicles that use a combination of computer vision, image recognition and deep learning to build automated skill at piloting a vehicle

2.1.3. Advantages and Disadvantages of Artificial Intelligence

AI technologies are quickly evolving, primarily because AI processes large amounts of data much faster and makes predictions more accurately than humanly possible.

While the huge volume of data being created on a daily basis would bury a human researcher, AI applications that use machine learning can take that data and quickly turn it into actionable information. As of this writing, the primary disadvantage of using AI is that it is expensive to process the large amounts of data that AI programming requires.

Advantages

- Keen on detail-oriented jobs usually performed by humans
- Reduced time for data-heavy tasks performed by humans
- Delivers consistent results with the best possible guidance.

Disadvantages

- Implementation can be sometimes expensive
- Requires deep knowledge of maths and statistics, as well as very good technical expertise
- If the models have not built in a meticulous base, it lacks the ability to generalize from one task to another.

[2] Objects can be text, speech, images, videos, sound etc.

2.2. Machine Learning

2.2.1. Introduction

Machine Learning (ML) is a Computer Science (CS) field that has been arisen after meticulous studying of Computer Theory, Pattern Recognition and Maths (most statistics and probability theory). In fact, ML constructs algorithms that learn from observations in order to perform appropriate predictions. Such algorithms take into consideration some experimental, past data and previous experience in order to provide with exclusive data driven predictions.

Generally, ML is the technological bridge where a machine learns to operate prediction or estimation task based on past experience that it is represented by historical data.

In terms of Mathematics, ML is the employment of mathematical functions and equations in order to learn imitate and represent real-world scenarios. The reason why ML models are called function approximation is because it will be extremely tough to extract exact functions which can be utilized to solve or estimate real world problems.

ML is applied to a vast array of computational tasks such as spam filtering, Optical Character Recognition (OCR), search engines, computer vision tasks, forecasting, hypotheses testing, audio recognition, event detection etc. It is sometime confused with data mining techniques which are mostly focused on data analysis and exploration rather than prediction and making decisions.

Tom M. Mitchell suggested a more official term for ML: "*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E".*

In the era of data analytics, ML is a tool or method that is used for implementing and deploying algorithms that lead to forecasting and prediction. The method that allows predicting and forecasting is the uncovering of hidden patterns that are strongly correlated between the data (pattern uncovering).

If one tries to think how the human brain works, will easily conclude that there is no way of injecting knowledge into it. In fact, humans learn with the observation method and come to specific findings. In other words, they learn by interacting with their close environment. For instance, when a child sees an object for first time, would not be able to interpret what exactly observers.

ML performs with a similar logic; Instead of dealing with every possible parameter, programmers develop algorithms which process a large amount of different information. In other words, the algorithm based on the information given, tries to construct its own logic and adapts its functionality.

In order this concept to be crystal clear, a simple example is stated below:

Table 2.1: Simple Logic Example

1	1
2	8
3	27
4	?

What is the missing value of the above Table 2.1? A human brain can easily understand that the missing value is not anything else than 64, because one can understand that the values on the right part of Table 2.1 are the values on the left part to the power of three (3). So, people can experiment with the data given on any conditions and try to find a relation between them.

A simple program, responsible for solving the above toy problem, would probably starting by setting random numbers (1, 2, 3, 4,… etc.) in the question mark position. When it has reached the real value (64) it would require a simple confirmation that the correct answer is 64, or else the algorithm would continue to put numbers up until there would be no enough memory.

Developers should be able then to partially guide their algorithms to the mostly acceptable solution by feeding them with data in order for the program to have a specific baseline.

Via ML, programmers try to give to machines partial logic. They seek to make computers learn based on observations and gain an experience like human brains.

2.2.2. How Machine Learning works

ML consists of three (3) main parts:

- **Decision Process**, where ML algorithms are used to make a general prediction which is based on input data (either labeled or unlabeled)

- **Loss Function,** which is employed in order to evaluate the prediction and the accuracy of the model[3].

- **Model Optimization**, which is the technique of updating and evaluating the model again and again up until a specific threshold of either accuracy or other metrics, is met.

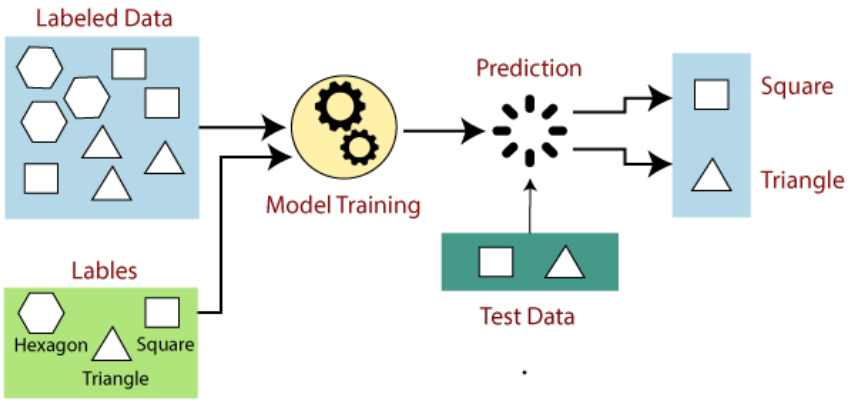
2.2.3. Types of Machine Learning

2.2.3.1. Supervised Learning

It is an attempt to deploying ML and AI tasks, where an algorithm is trained in a specific set of input data (Training Set) that has been labeled for a particular output[4]. The model is trained until it is capable of uncovering hidden relationships and patterns between the input and the output data. During the training phase the model is provided with a vast array of labeled data with an input / output relation. This means that for a specific training pattern of the input set, corresponds a specific label value. Then the trained model is presented with test data in order to fine tune the accuracy and other metrics of the initial model. In other words, the main purpose of the test set is to measure how accurately the algorithm performs on data without labels.

[3] More about Error and Loss functions will be discussed in the next Chapters

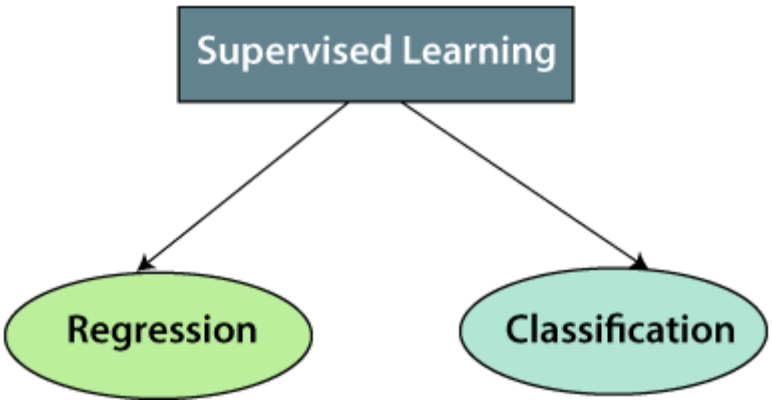
[4] Labeled data are those samples that have been already annotated with a specific characteristic. Labeled data are used more often in Supervised ML algorithms



Picture 2.1: Supervised Learning Architecture (Source: Google)

Supervised learning uses a training set to “teach” a model to yield a desired output. The training dataset includes correct input / output pairs and allows the model to learn iteratively over the time.

This ML type is preferred in most cases for classification and regression problems.



Picture 2.2: Types of Supervised Learning

Classification Algorithms, try to fit all the training parameters in a given, a priori known, number of categories (classes), based on the labeled data that it was trained on. In such problems the machine learns to predict discrete values. These algorithms can be used for email spam filtering, object recognition, handwritten digit classification or any other classification problem solved by Supervised Learning. Some of the most famous algorithms that perform classification tasks are:

- Logistic Regression (Binary Classification)
- Random Forest (Bagging Algorithms of ensembling techniques)
- Decision Trees
- Support Vector Machines (SVM)
- Neural Networks

Regression Algorithms, approach the problem in a diverse way; they expect the model to produce numerical relations between the output and the input data. The output is not a class result as in the classification problems, but a continuous value between given ranges. In such problems the machine is forced to predict the value of a continuous response variable. The most popular regression algorithms are:

- Linear Regression
- Polynomial Regression
- Non-Linear Regression (SVM with non linear kernels)
- Neural Networks

2.2.3.2. Unsupervised Learning

In this approach of ML, the algorithm learns and uncovers patterns and relationships from unlabelled data. They discover hidden patterns without the need of human intervention (training and testing). Also unsupervised learning allows programmers to perform more complex tasks compared to supervised learning. This type of learning can be more unpredictable in accordance to other natural learning methods.

The most important reasons for using Unsupervised Learning are the following:

- It uncovers unknown/hidden patterns between the data
- It is easier to process unlabeled data rather than labeled which need manual intervention.
- It may find out extra features that can be useful for further categorization

The most common unsupervised algorithm is **clustering**.

Picture 2.3: Clustering data into different groups based on similarities (Source: Google)

Clustering, as it can be inferred by the above Picture 2.3, is a data mining technique that groups unlabeled data either on their similarities or differences. This algorithm is good at dealing with raw and unclassified data in the way of categorizing them into different clusters depending on their similarities. Some well-known clustering algorithms are: k-means (distance based clustering algorithms), DBSCAN (density based clustering algorithms), hierarchical clustering, k-NN (k Nearest Neighbors), PCA (Principal Components Analysis), SVD (Singular Value Decomposition) etc.

2.2.3.3. Reinforcement Learning

Reinforcement learning is the training of machine learning models to make a sequence of decisions. To this end, an agent is employed. The agent learns to achieve a goal in an uncertain, potentially complex environment.

In reinforcement learning, an AI faces a game-like situation. The computer employs trial and error to come up with a solution to the problem. To get the machine to do what the programmer desires, AI gets either rewards or penalties for the actions it performs. Its goal is to maximize the total reward.

Although the designer sets the reward policy (that is the rules of the game) they give the model no hints or suggestions for how to solve the game. It's up to the model to figure out how to perform the task to maximize the reward, starting from totally random trials and finishing with sophisticated tactics and superhuman skills.

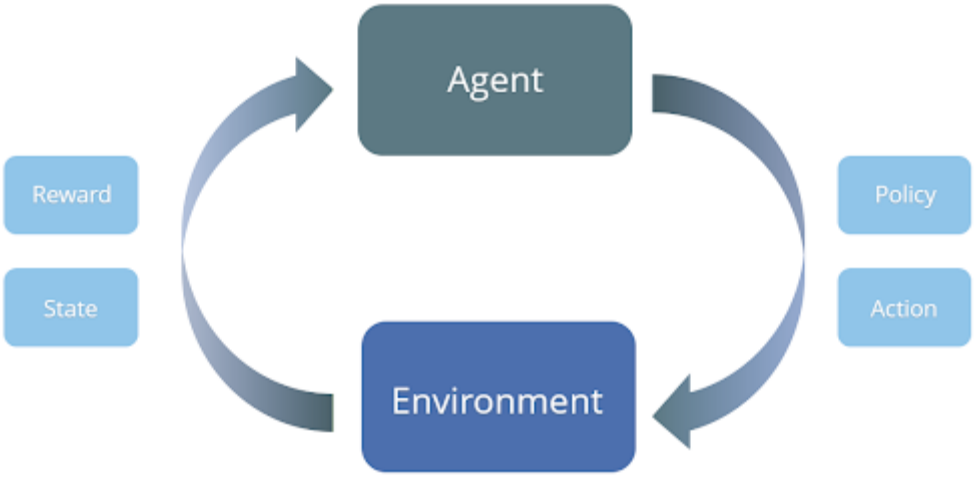


Figure 2.2: Reinforcement Learning Architecture (Source: Google)

By leveraging the power of search and many trials, reinforcement learning is currently the most effective way to hint machine's creativity. In contrast to human beings, AI can gather experience from thousands of parallel game plays if a reinforcement learning algorithm is run on a sufficiently powerful computer infrastructure.

Examples of reinforcement learning

Applications of reinforcement learning were in the past limited by weak computer infrastructure. However, as Gerard Tesauro's 37 backgammon AI superplayer developed in 1990's shows, progress did happen. That early progress is now rapidly changing with powerful new computational technologies opening the way to completely new inspiring applications.

Training machines that control self-driving cars is an excellent example of the possibility of application of reinforcement learning. In an ideal situation, the computer should get no guidance on driving a car. The developer would avoid anything connected with the task and allow the machine to learn from its own errors.

For example, in some circumstances an autonomous vehicle is required to put safety first, minimize ride time, reduce pollution, offer passengers comfort and obey the rules of law. With an autonomous race car, in the opposite, an emphasis should put on speed much more than the driver's comfort. The developer cannot predict everything that could happen on the street. Instead of building lengthy "if-then" instructions, the programmer prepares the reinforcement learning agent to be capable of learning from the system of rewards and penalties. The agent gets rewards for reaching specific goals.

2.2.3.4. Transfer Learning

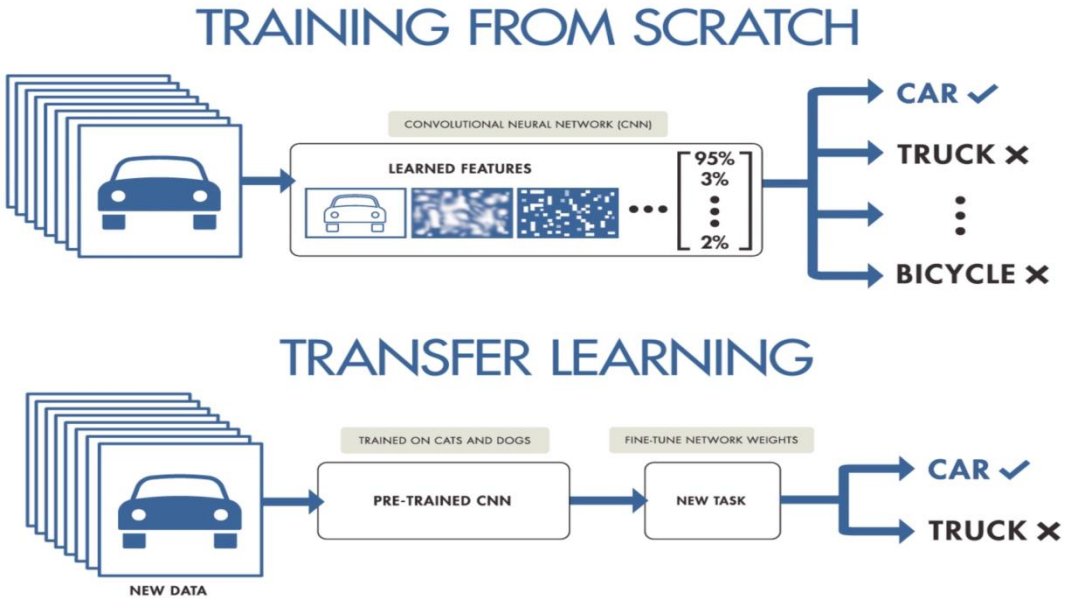
Transfer learning (TL) is an applied problem in ML that focuses on transferring knowledge previously acquired while solving one problem and applying it to a different but correlated problem.

TL is a modern method of ML where the application of knowledge obtained from a model used in one previous task and can be used again as a foundation point for another process.

ML algorithms use previous experience as inputs to make inference and produce new output values. They are typically designed to conduct isolated tasks. A source task is a task from which knowledge is transferred to another target task. A target task is where improved learning occurs because of the transfer of knowledge from a source task.

During TL phase, the knowledge used from a source task is leveraged to improve the learning and development to a new target process. The application of knowledge is using the source task's attributes and characteristics, which will be applied and mapped onto the new target task (Picture 2.4).

When the TL comes to increase the performance of the new process, then it is called a **positive transfer** and when there is a decrease in the performance, it is called a **negative transfer**. One of the major priorities when dealing with TL methods is being capable of providing and ensuring the positive transfer between related tasks while avoiding the negative transfer.



Picture 2.4: TL architecture of an image classification problem (Source: Google)

TL consists of three (3) different types, which are stated below:

- **Inductive Transfer Learning**

In this type of TL, both the source and target processes are the identical. The model uses inductive biases from the source task to help improve the performance of the target task. The source task can contain labeled data, further leading to the model using multitask learning.

- **Unsupervised Transfer Learning**

Unsupervised TL is when an algorithm is subjected to being able to identify patterns in datasets that have not either been labeled or classified. In this scenario, both the source and target are similar. The task here is different, if the data is unlabeled in both source and target. Techniques such as dimensionality reduction and clustering are well known in unsupervised TL learning.

- **Transductive Transfer Learning**

In this type of TL, the source and target processes share similarities but the domains are not similar. The source domain consists of labeled data, whereas there is a lack of it in the target domain, in order for the model to use specific adaptation.

The benefits of utilizing TL techniques are summarized below:

- **Better initial model**. In non TL models, the researcher needs to construct a model without prior knowledge. TL offers a better zero point and can handle processes at some level of experience without training.
- **Higher learning rate**. TL offers a higher learning rate during the training phase because the problem has already trained for another similar job.
- **Higher accuracy**. With a better zero point and higher learning rate, TL provides a ML model to converge at a higher performance level, providing with more robust results.
- **Faster convergence**. The process of learning can acquire the desired result faster than traditional ML methods because it utilizes a pre-trained model.

2.2.4. Mathematics and Concepts of Machine Learning

ML is all about mathematics, which in turn helps in creating algorithms that can learn from data and make accurate predictions / inferences. The prediction could be as simple as classifying amidst dogs and cats from a given set of pictures or what kind of products to recommend to a customer based on past purchases. Nevertheless, it is very important to completely understand the math concepts behind any basic ML algorithm. This fact, may help one picks all the right algorithms for their project in Data Science (DS) and ML.

ML is primarily built on mathematical prerequisites so as long as it is understood why the Maths is used, one will find it more satisfying. With this, it could be crystal clear why to pick one ML algorithm over the other and how it affects the performance of the model.

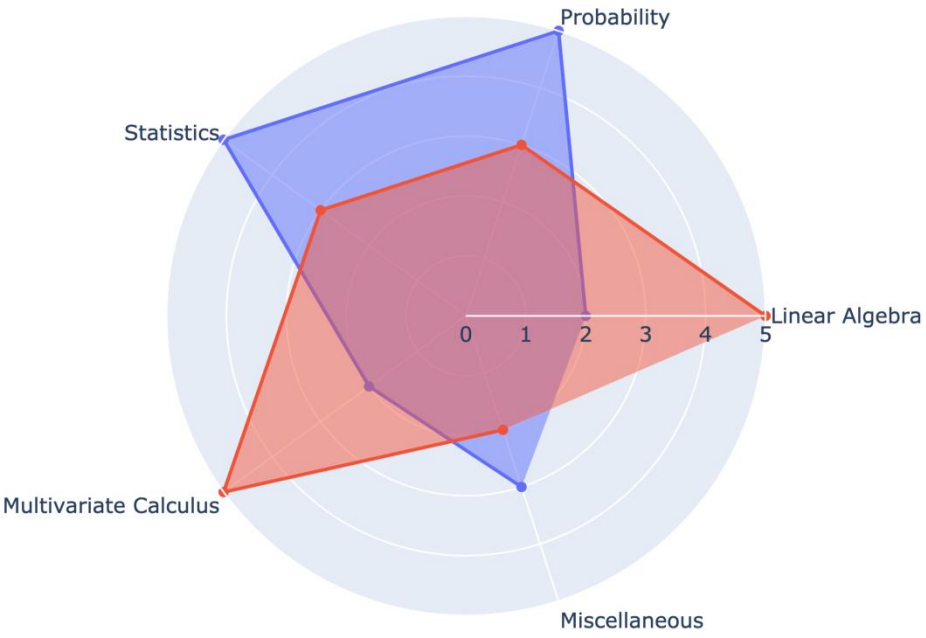


Figure 2.3: Mathematical Concepts for ML (RED) and DS (BLUE) (Source: Google)

The Mathematical concepts that are totally important for ML and further DS implementation belong in a wide range of categories, as following:

- **Linear Algebra**

The importance of linear equations is a fundamental component in developing basic ML concepts. Linear algebra is applied in ML algorithms in loss functions, regularization, covariance matrices, SVD, Matrix Operations, and SVM. It is also applied in ML algorithms like linear regression. These are the concepts that are needed for understanding the optimization methods used for machine learning.

In order to perform a Principal Component Analysis (PCA) that is used to reduce the dimensionality of data, we use linear algebra. Linear algebra is also primarily used in neural networks for the processing and representation of networks.

- **Calculus**

Many students / learners who did not find learning calculus fascinating at school will be in a shock as it is an integral part of ML. Undoubtedly, there is no need for one to master calculus but it is only important to learn and understand the basic components / principles of it. Furthermore, a thorough understanding of the practical applications of ML through calculus during model construction is required.

For instance, the derivative of the function returns its rate of change in calculus, and one should be able to understand the concept of gradient descent. Gradient descent, finds the local minima for a function and so on. Also, some of the necessary topics to ace the calculus part in DS are Differential and Integral Calculus, Vector-Values Functions, Partial Derivatives, Directional Gradients etc.

- **Descriptive Statistics**

Descriptive statistics is a foundamental concept that every aspiring data scientist needs to learn to understand ML when working with classifications tasks like logistic regression, distributions, discrimination analysis, and hypothesis testing.

Statistics is very essential in order for one to become a successful data scientist. Statistics is the main part of mathematics for ML. Some of the fundamental statistics needed for ML are Axioms, Bayes' Theorem, Expectation / Maximization, Variance and Expectation, Random Variables, Conditional, and Joint Distributions.

- **Discrete Maths**

Discrete mathematics is concerned with non-continuous numbers, most often integers. Many applications necessitate the use of discrete numbers. For instance, when scheduling a taxi fleet, cannot be sent 0.88 taxis to pick up a client.

Many of the structures in AI are discrete. A NN, has an integer number of nodes and interconnections between the nodes. Thus, the mathematics used to construct a neural network must include a discrete element, the integer representing the number of nodes and interconnections.

- **Probability Theory**

To properly work through a ML predictive modeling project, it would be reasonable to conclude that probability is essential. ML is the process of creating prediction models from ambiguous or partially unknown data. Working with faulty or incomplete information is what uncertainty matters.

Uncertainty is crucial to ML, yet it is one of the components that create the most difficulties for newcomers, particularly those coming from a programming background.

In ML, there are three major sources of uncertainty:

- Noisy data,
- Imperfect models,
- Limited coverage of the problem area.

However, with the help of the right probability tools, solving a problem can be estimated. Finally, probability is essential for hypothesis testing and distributions like the Gaussian distribution and the probability density function (PDF).

2.2.5. Machine Learning Metrics

Choosing the correct metric while deploying and evaluating ML models is crucial. In some applications looking at a single metric may not give you the whole picture of the problem being solved, and one may want to use a subset different metrics to have a solid evaluation of a model.

Without doubt, different metrics are used for different ML tasks, so a brief discussion will be provided for each type of ML problem.

2.2.5.1. Classification Related Metrics

Classification is one of the most widely used problems in ML with various industrial applications, from face recognition, video categorization, content moderation, medical diagnosis, video summarization to text classification and hate speech detection on social media.

Models such as SVM, logistic regression, decision trees, random forest, Xgboost, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) are some of the most popular classification models.

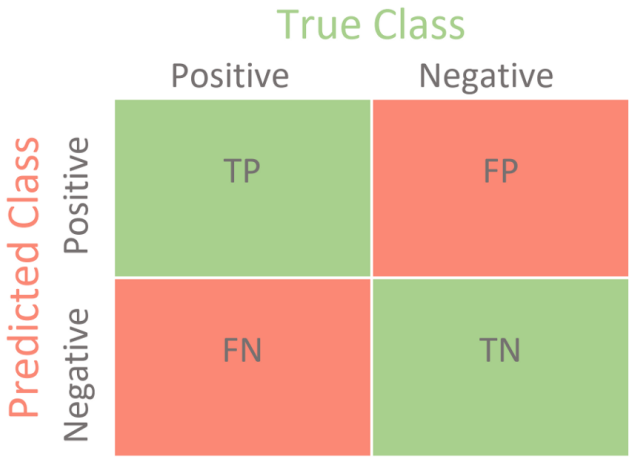


Figure 2.4: Confusion Matrix of a Binary Classification Problem

There are various ways to evaluate a classification model. Some of them are provided below.

- **Classification Accuracy**

Classification accuracy is perhaps the simplest and the most popular amidst the metrics one can imagine and is defined as the number of correct predictions divided by the total

number of predictions, multiplied by 100. So taking Figure 2.4 into consideration, accuracy is given by the following equation:

$$Accuracy = \frac{Correct\ Predictions}{All\ Predictions} = \frac{TP + TN}{TP + FP + TN + FN}$$

- **Precision**

There are many cases where classification accuracy is not a good metric of a model performance. One of these scenarios is when a class distribution is imbalanced[5]. In this case, even if one predicts all samples as the most frequent class they would get a high accuracy rate, which does not make sense at all, because the model is not learning anything, and is just predicting everything as the top class.

So, one needs to look at class specific performance metrics too. Precision is one of such metrics, which is defined as:

$$Precision = \frac{Correct\ Positive\ Predictions}{All\ Positive\ Predictions} = \frac{TP}{TP + FP}$$

It actually shows the percentage of the Correct Positives over the total number of positives predicted by the classifier.

Precision is also known as Specifity of a model

- **Recall**

Recall is another important metric, which is defined as the fraction of samples from a class which are correctly predicted by the model. In mathematical formation:

$$Recall = \frac{Correct\ Positive\ Predictions}{All\ Positives} = \frac{TP}{TP + FN}$$

It actually shows the percentage of the Correct Positives over the total number of positives.

Recall is also known as Sensitivity of a model

- **F1 – Score**

Depending on application, one may want to give higher priority to recall or precision. But there are many applications in which both recall and precision are significant. To this end, it is natural to consider of a way to combine these two metrics into a single one. One popular metric which combines precision and recall is called F1-score, which is the harmonic mean of precision and recall defined as:

$$F1 - score = 2\ \frac{recall \times precision}{recall + precision}$$

[5] **Imbalanced data** refers to those types of datasets where the target class has an uneven distribution of observations, i.e one class label has a very high number of observations and the other has a very low number of observations.

All the above metrics are summarized as following:

$$\text{precision} = \frac{\text{tp}}{\text{tp} + \text{fp}}$$

$$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$$

$$\text{accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}}$$

$$\text{F}_1 \text{ score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Picture 2.5: Most popular Classification Metrics

2.2.5.2. Regression Related Metrics

Regression models are another family of machine ML and statistical models, which are used to predict a continuous target values. They have a wide range of applications, from house price prediction, e-commerce pricing systems, weather forecasting, stock market prediction, to image super resolution, feature learning via auto-encoders, and image compression.

Metrics used to evaluate these models should be able to work on a set of continuous values[6], and are therefore slightly different from classification metrics.

Some Regression Metrics are the following:

- **Mean Squared Error**

Mean Squared Error (MSE) is perhaps the most well known metric used for regression problems. It essentially computes the average squared error between the predicted and actual values.

Assuming a regression model which predicts the house prices in Athens area (show them with $\hat{y}_i$) and for each house there is an actual price the house was sold for (denoted with y_i). Then the MSE can be calculated as:

$$\text{MSE} = \frac{1}{\text{N}} \sum_{\text{i}=1}^{\text{N}} (\hat{y}_\text{i} - y_\text{i})^2$$

Where $\hat{y}_\text{i}$ is the predicted price of the house and y_i is the actual price.

- **Mean Absolute Error**

Mean Absolute Error (MAE) is another metric which finds the average absolute distance between the predicted and target values. MAE is defined as following:

$$\text{MAE} = \frac{1}{\text{N}} \sum_{\text{i}=1}^{\text{N}} |\hat{y}_\text{i} - y_\text{i}|$$

[6] Prerequisite is the variables to be continuous and have infinite cardinality

Where $\hat{y}_i$ is the predicted price of the house and y_i is the actual price.

- **Root Mean Squared Error**

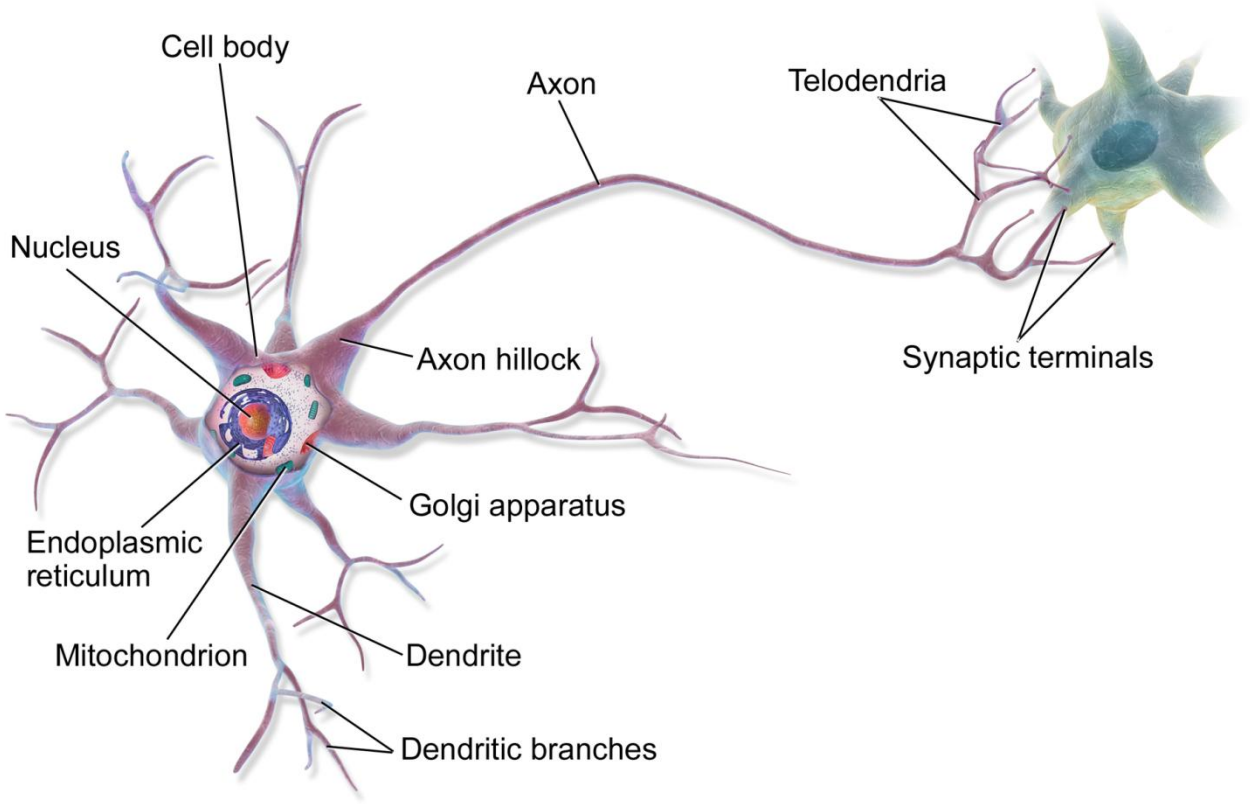
Root Mean Squared Error (RMSE) is nothing but the standard deviation of the prediction error (residuals). In mathematical form, is the squared root of the MSE and it is given by the following mathematical type:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

2.3. Artificial Neural Networks

2.3.1. Introduction

The human brain, its functionality and the way it works and in general, inspired the creation and the development of Artificial Neural Networks (ANN). AI and ML play an essential part in this task. It starts working when a developer enters data and builds a ML algorithm, mostly using simple *"if … else …" clauses* of building a program.



Picture 2.6: Human brain's neurons architecture (Source: Google)

The deep neural network does not only work according to the algorithm but also can predict a solution for a task and make conclusions using its previous experience[7]. In this case, there is no need of using either programming or coding to get an answer.

2.3.2. Deep Neural Networks

The field of AI is essentially on the condition when machines can do tasks that typically require human intelligence. It utilizes ML, where machines can learn by experience and acquire skills without human intervention. DL is a subset of ML where ANN algorithms inspired by the human brain that learn from large amounts of data. Similarly to how a human brain learns from experience, the DL algorithm would perform a task repeatedly, each time changing it a little to improve the desired result. It is mostly referred as DL because the NN have various layers that enable and enhance learning. Just about any problem that requires "thought" to figure out is a problem that DL can learn to solve.

The amount of data generated every day is astonishing[8] and it's the resource that makes DL feasible. Furthermore, DL algorithms benefit from the stronger computing

[7] Past data in the form of training sets

power that is available in the modern times as well as the proliferation of AI as a Service. AI has given smaller organizations access to AI technology and specifically the AI algorithms required for deep learning without a large initial investment.

DL allows machines to solve complex problems even when using a data set that is very diverse, unstructured and interconnected. The more DL algorithms acquire knowledge, the better they perform.

Nodes are little parts of the system, and they are like neurons of the human brain. When a stimulus hits them, a process takes place in these nodes. Some of them are connected and marked, and some are not, but in general, nodes are grouped into layers as the Figure 2.5 depicts.

The system must process layers of data between the input and output to solve a specific problem. The more layers it has to process to get the result, the deeper the network is considered. There is a concept of Credit Assignment Path (CAP)[9] which means the number of such layers needed for the system to complete the task. The neural network is deemed as deep if the CAP index is greater than two.

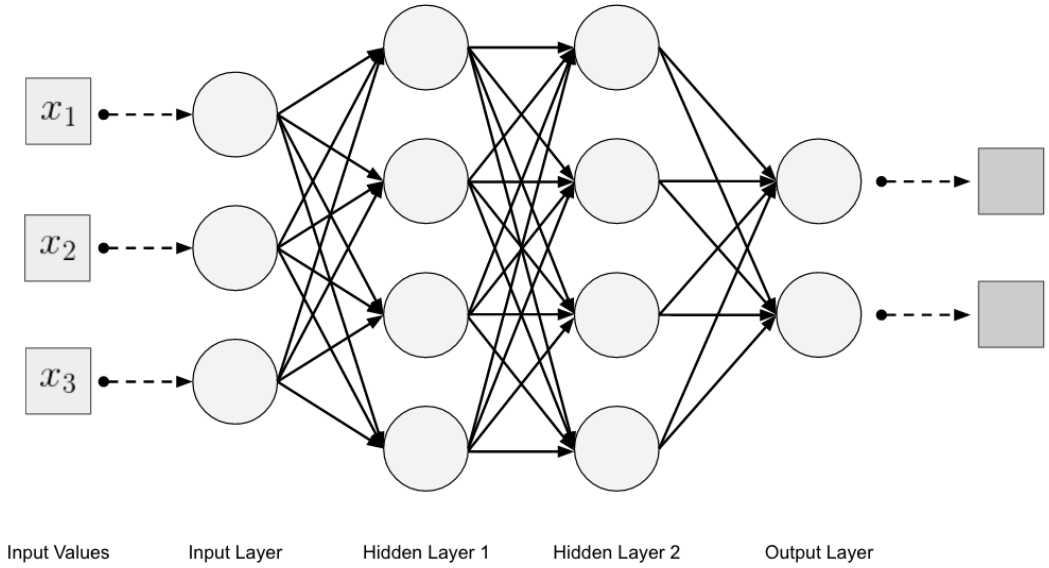


Figure 2.5: DNN Architecture (Source: Google)

A deep neural network is beneficial when you need to replace human labor with autonomous work without compensating for its efficiency. The DNN usage can acquire various applications in real life that will be discussed in the following Paragraphs.

The DNN, in general, consist of the following layers (Figure 2.5):

Input Layer

The input layer takes input values (raw input) from the domain. No computation or other manipulation is performed at this layer. Nodes here just pass on the features (information) to the next hidden layer.

Hidden Layer

[8] Currently estimated at 2.6 quintillion bytes daily

[9] CAD index implies the number of hidden layers a NN Architecture has. Architectures with CAD index greater than 2 are considered to be deep architectures. If CAD index is less than 2, architectures are classified as shallow.

At this phase, the nodes of this layer are not exposed. They provide an abstraction to the neural network.

The hidden layer performs all kinds of computation on the features entered through the input layer and transfers the result to the next layer (output layer).

Output Layer

It is the final layer of the network that brings the information learned through the hidden layer and delivers the final value as a result.

2.3.2.1. Training Artificial Neural Networks

Training ANN is a quite complex process. Some things that one should bear in mind are the following:

Back Propagation

Back-propagation is the need of NN training. It is a way of fine-tuning the weights of a NN based on a specific error index (i.e. loss) obtained in previous training iterations[10]. Proper tuning of the weights ensures lower error rates, making the model reliable by increasing its generalization and stability.

Gradient Descent

Gradient descent is an iterative first-order optimization algorithm used to find a local minimum (in the field of optimizing ANN algorithms) of a given function. This method is widely utilized in ML and DL algorithms, in order to minimize the cost function. Gradient descent algorithm does not work for all functions. In DL algorithms functions are convex and they usually converge in a minimum.

2.3.2.2. DNN in Real Life

Some real examples that DNN help people into their daily lives and routines are the following:

Virtual assistants

Whether it is Alexa or Siri, the virtual assistants of online service providers use DL techniques to help recognize one's speech and the language humans use when they interact with them and provide the best possible services in accordance to their problem.

Machine Translations

In a similar way, DL algorithms can automatically translate between different languages from almost all over the world. This can be powerful for travelers and people in the industry.

Self-Driving Cars

The way an autonomous vehicle understands the realities of the road and how to respond to them whether it is a stop sign, a ball in the street or another vehicle is through DL architectures. The more data the algorithms receive, the better they are able to act human-like in their information processing, even knowing that a STOP sign covered with snow is still a STOP sign depending on its shape and size.

Chatbots

[10] Iterations in NN are well known as Epochs

Chatbots that provide customer service for lots of companies are capable of responding in a rather intelligent and helpful way to an increasing amount of auditory and text questions.

Image Colorization

Transforming black & white images into colorful was formerly a task done meticulously by human hands. In the modern times, DL architectures are capable of using the context and objects in the images to color them. The results are very impressive and so accurate.

Event Detection

In the era of event detection, DL and ANN architectures play a crucial role. Using such architectures and techniques many events can be recognized (fires, explosions, high sound pollution in urban areas) automatically without the need of a human observer 24/7.

Face recognition

DL is being used for face recognition not only for security purposes but also for tagging people on social media posts automatically. The challenges for DL algorithms for facial recognition is to know exactly that it is the same person even when they have changed hairstyles, grown or shaved off a beard or if the image taken is poor due to bad lighting or an obstruction.

Medicine and Pharmaceuticals

From disease and growth diagnoses to personalized medicines created specifically for an individual's genome, DL in the medical field has the attention of many of the largest pharmaceutical and medical companies in the globe.

Entertainment

Ever wonder how Netflix comes up with suggestions for what you should watch next? Or where Amazon comes up with ideas for what you should buy next and those suggestions are exactly what you need but just never knew it before? Totally, it is DL algorithms that do this complex job.

2.3.3. Loss and Loss Functions in DL

2.3.3.1. What is Loss and Loss Functions

Loss and loss functions in general, are nothing else but a method of evaluating how well a DL algorithm fits / models data. If the predictions are fully different from the ground truths, the loss function will result to a higher value than expected. On the other hand If they are pretty good and inside the desired tolerance, it will output a lower value. By changing elements of the program in order to improve your model, the loss function will change accordingly (either gets higher or smaller).

The model loss is related to the model accuracy and in mathematical form can be noted as the difference between the ground truth (actual value) and the predicted value. The general mathematical type for the loss is the following:

$$\text{Loss} = \left| y_{\text{pred}} - y_{\text{actual}} \right|$$

2.3.3.2. Types of Loss Functions

Many types of loss functions can be found in DL problems. To which loss can one utilize, depends on the type of the algorithm. For instance, different types of losses are used for either classifications or regression problems. Some of the most common loss functions are stated in the next paragraphs.

- **Mean Squared Error (MSE)**

This type of metric was mentioned in previous Chapters, as a very common metric for evaluating regression problems in the era of ML. But MSE is also a great and popular loss function referred to DL when dealing with regression issues. The mathematical type is the same as:

$$\mathrm{MSE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\hat{\mathrm{y}}_{\mathrm{i}} - \mathrm{y}_{\mathrm{i}})^2$$

The less value of MSE a model outputs, the most accurate model is in terms of returning more realistic predictions.

- **Binary Cross Entropy (Log Loss for binary classification)**

This type of loss is most commonly used for binary classification problems. Binary cross entropy compares each of the predicted probabilities to actual class output which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value. That means how close or far from the actual value. The mathematical type of the Binary Cross entropy is the following:

$$\mathrm{Log\ Loss} = -\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}[\mathrm{y}_{\mathrm{i}}\log(\mathrm{p}_{\mathrm{i}}) + (1-\mathrm{y}_{\mathrm{i}})\log(1-\mathrm{p}_{\mathrm{i}})]$$

Where N is the length of the training data, y is the value of the corresponding data and p is the referring probability.

- **Categorical Cross Entropy (Log Loss for multiclass classification)**

This type of loss is used for multiclass classification problems. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one. In other words, this loss is a very good measure of how distinguishable two *discrete probability distributions* are from each other. The mathematical type of the Categorical Cross Entropy is the following

$$\mathrm{Log\ Loss} = -\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\mathrm{y}_{\mathrm{i}}\log\hat{\mathrm{y}}_{\mathrm{i}}$$

Where, $\hat{\mathrm{y}}_{\mathrm{i}}$ is the model's output value, y_{i} is the ground truth target value and N is the length of the training data.

2.3.4. Activation Functions in DL

Activation Functions decide if a neuron needs to be activated or not. It will decide whether the neuron's input to the network is significant in the process of prediction. To achieve this simple mathematical operations are used.

The role of the Activation Function is both to derive output from a set of input values fed to a node and add the sense of non-linearity into the NN model. In other words, activation functions add an additional step at each layer during the forward propagation.

2.3.4.1. Types of Activation Functions

- **Binary Step Function**

This type of activation functions is based on a specific threshold value that decides whether a neuron should be activated or not.

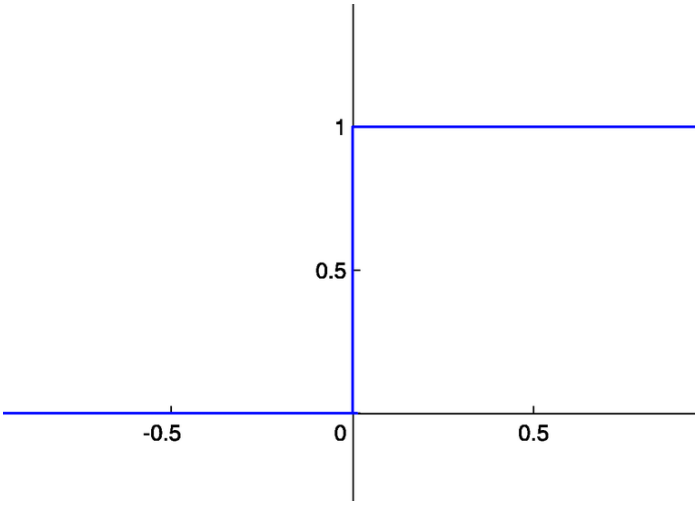


Figure 2.6: Binary Step Function

The input data to the activation function is compared to a specific threshold. If the input is greater than it, then the neuron is activated, else it stays deactivated and the output is not forwarded to the next layer.

The mathematical type of Step Function is the following:

$$f_{step}(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases}$$

- **Linear Activation Function**

The linear activation function is that function where the output is proportional to the input.

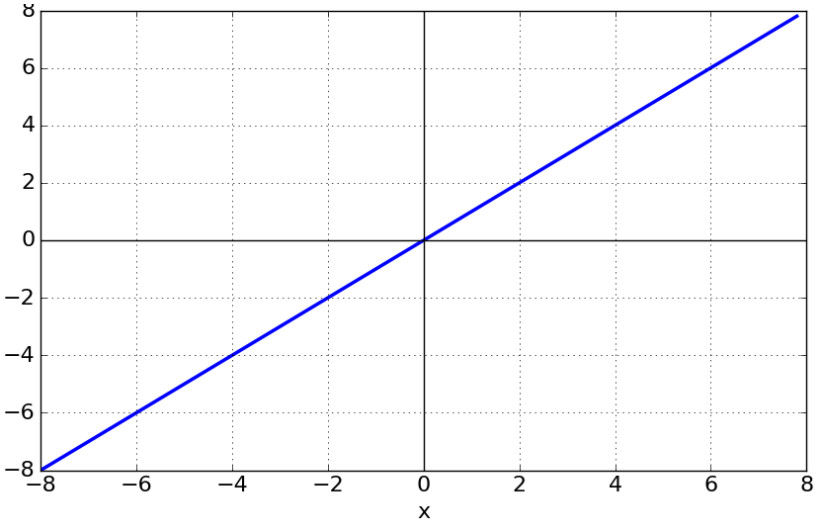


Figure 2.7: Linear Activation Function

The above linear activation function shown above is simply a linear regression model problem.

Due to its limited power, this does not allow the model to create complex mappings between the network's inputs and outputs. In order to solve this limitation, non linear activation functions are used for complex DL models.

The mathematical type of the Linear Function is the following:

$$f_{linear}\ (x) = \ x$$

- **Sigmoid Activation Function**

The Sigmoid function takes real values as input and maps values in the range of [0, 1].

The greater the input, the closer the output value will be to 1, whereas the smaller the input, the closer the output will be to 0, as the above figure depicts.

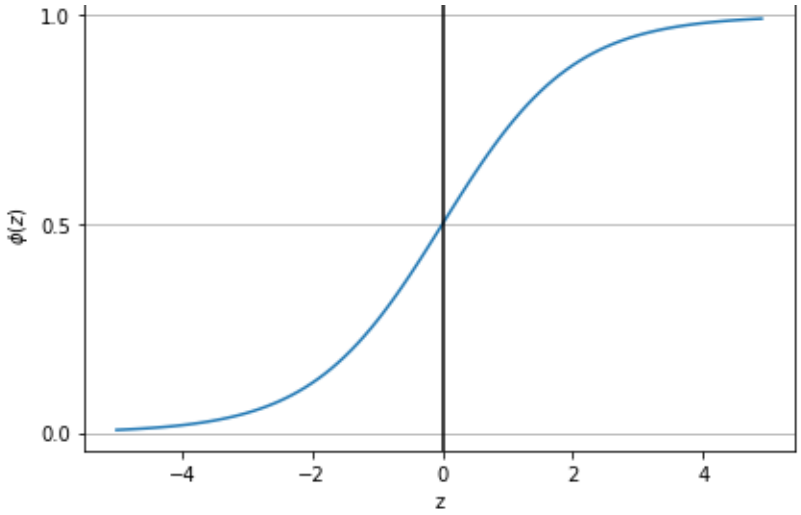


Figure 2.8: Sigmoid Activation Function

The sigmoid function is most commonly used in models where it is needed to predict the probability as an output. Since probability bounds between the range [0, 1], this function is the best possible choice.

The mathematical type of the Sigmoid Function is the following:

$$f_{sigmoid}\,(x) = \frac{1}{1 + e^{-x}}$$

- **Hyperbolic Tangent Activation Function (tanh function)**

Hyperbolic Tangent Activation Function is very close to the sigmoid activation function, with the difference in the output range which is in the range of [-1, 1].

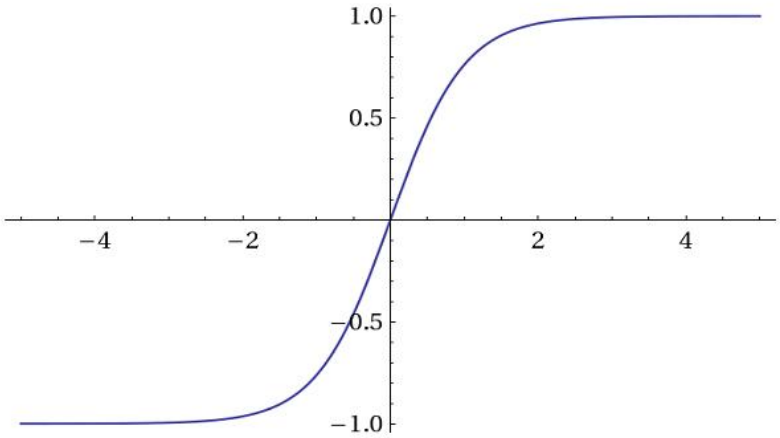


Figure 2.9: Hyperbolic Tangent Activation Function

In Hyperbolic Tangent Activation Function, the greater the input, the closer the output value will be to 1, whereas the smaller the input, the closer the output will be to -1.

The output of the above activation function is zero-centered. Thus, one can easily map the output values as strongly negative, neutral, or strongly positive.

The mathematical type of the Hyperbolic Tangent Activation Function is the following:

$$f_{tan\,h}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

- **Rectified Linear Unit (ReLU) Activation Function**

At first glance it gives the impression of a linear function. ReLU allows backpropagation and simultaneously making the computations more efficient.

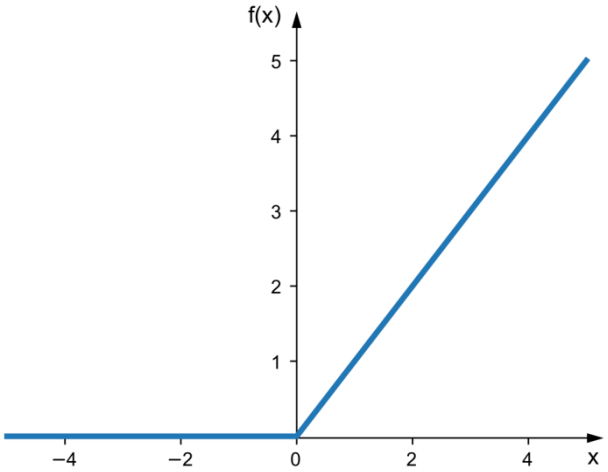


Figure 2.10: ReLU Activation Function

ReLU does not activate all the neurons at the same time. Specifically, the neurons will only be activated if the output of the linear transformation is greater than 0.

Since only a specific number of neurons are activated, ReLU is very computationally efficient when compared to other activation functions (ex. when compared with sigmoid or tanh activation function).

ReLU makes faster convergence of gradient descent to the global minimum of the loss function due to its linear property.

The mathematical type of the ReLU Activation Function is the following:

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

- **Leaky ReLU Activation Function**

Leaky ReLU is another more improved version of ReLU function.

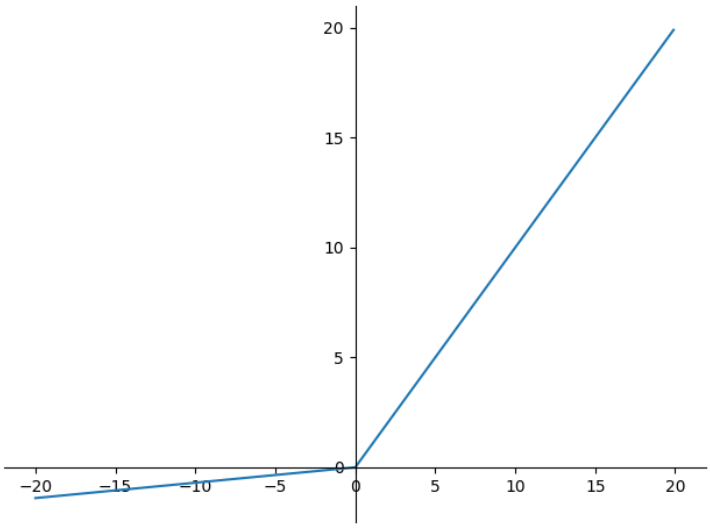


Figure 2.11: Leaky ReLU Activation Function

The pros of Leaky ReLU are more or less the same as that of the normal ReLU, in addition to the fact that it does enable backpropagation, even for negative input values.

By enabling negative values to pass, the gradient of the left side of the graph comes out to be a non-zero value. Therefore, we would no longer encounter dead neurons in that region.

The mathematical type of the Leaky ReLU Activation Function is the following:

$$f_{LReLU}(x) = \begin{cases} x, & x > 0 \\ ax, & otherwise \end{cases}$$

Where a is a user hyperparameter

- **Softmax Activation Function**

Softmax function can be described as a combination of many sigmoid functions.

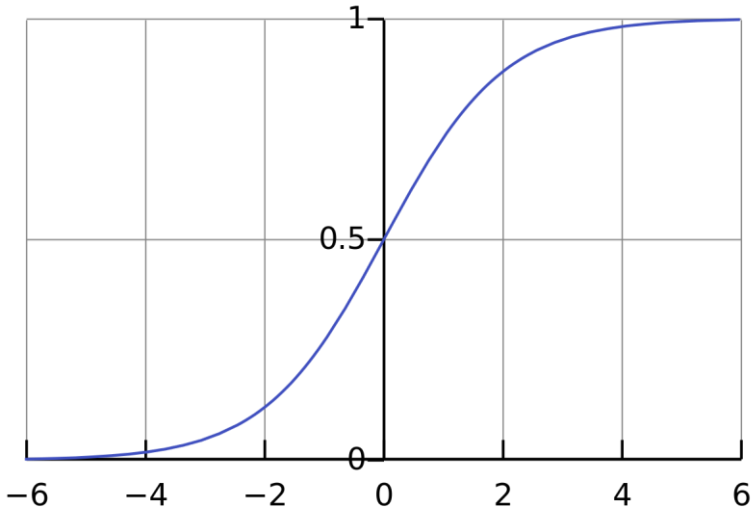


Figure 2.12: Softmax Activation Function

It outputs the relative probabilities. Similar to the sigmoid activation function, the Softmax returns the probability of each class on a multiclass classification problem.

It is most commonly used as an activation function for the last layer of the neural network in the case of multiclass classification.

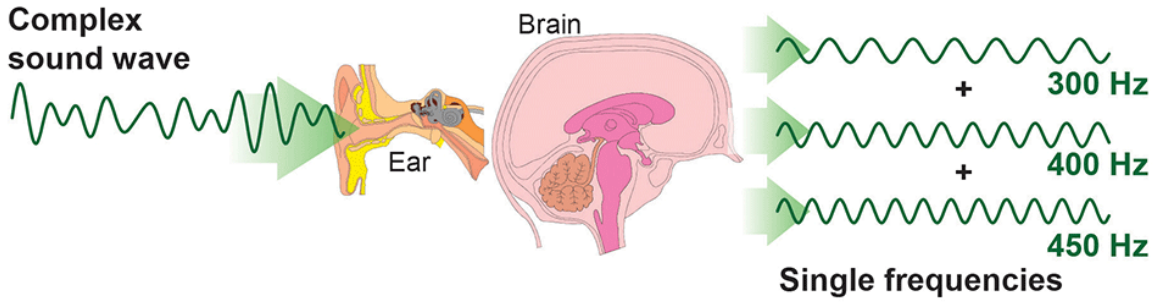
The mathematical type of the Softmax Activation Function is the following:

$$f_{softmax}(z) = \frac{e^{z_i}}{\sum_{i=1}^{K} e^{z_j}}$$

3. AUDITORY SCENE ANALYSIS

3.1. What is Sound?

Sound is produced when an object's vibrations move through a medium until they enter the human eardrum. In physics, sound is produced in the form of a pressure wave. If an object moves rapidly, it causes the surrounding air molecules to move too, initiating a chain reaction of sound wave vibrations throughout the medium. While the physiological definition includes a subject's reception of sound, the physics definition recognizes that sound exists independently of an individual's reception [25].



Picture 3.1: The sound perception of a human brain (Source: Google)

3.1.1. Types of Sounds

There are so many different types of sounds and categories. Some of them are stated below:

- **Infrasounds**

Infrasounds have frequencies under 20000Hz, which makes them inaudible to the human ear. Scientists use infrasound to detect earthquakes and volcanic eruptions, to map rock and petroleum formations underground, and to study activity in the human heart. Despite our inability to hear infrasound, many animals use infrasonic waves to communicate in nature. Whales, hippos, rhinos, giraffes, elephants, and alligators all use infrasound to communicate across impressive distances [25].

- **Ultrasounds**

Ultrasounds have frequencies higher than 20000Hz. Because ultrasounds occur at frequencies outside the human hearing range, it is not audible to the human ear. Ultrasounds are most often used by medical specialists who use sonograms to examine their patients' internal organs. Some lesser-known applications of ultrasound include navigation, imaging, sample mixing, communication, and testing. In nature, bats emit ultrasonic sounds to locate prey and avoid obstacles [25].

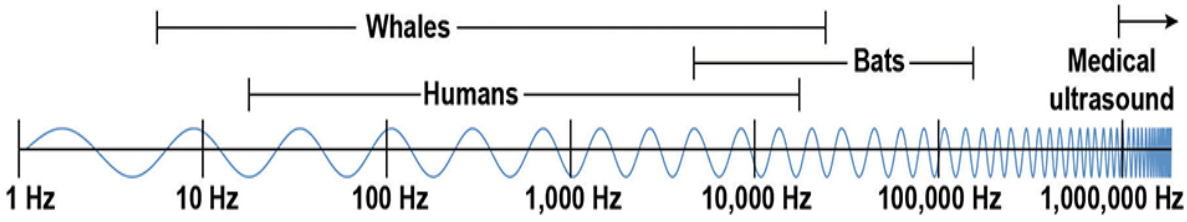


Figure 3.1: Frequencies of sound and average range of hearing (Source: Google)

3.2. Introduction to Auditory Scene Analysis

Auditory Scene Analysis (ASA) refers to the process of deploying complex acoustic input into different auditory objects such as audio clips, music, human interactions, which form the sound waves reaching human ears.

Hearing is one of the five basic human senses. People use this awesome capability naturally in their daily lives, but they often regret its importance. They communicate with other people by talking, feel and perceive this world through acoustic information along with other sensory data.

As Bregman proposed in [23], ASA focuses on the problem of hearing complex auditory environments, using a series of creative analogies to describe the process required of the human auditory system as it analyzes mixtures of sounds to recover descriptions of individual sounds. In a unified and comprehensive way, Bregman establishes a theoretical framework that integrates his findings with an unusually wide range of previous research in psychoacoustics, speech perception, music theory and composition, and computer modeling.

All landscapes include sounds that definitely vary from time to time during the same day or from day to day throughout a year. Natural habitats are dominated by sounds from living organisms / creatures, such as insects, snakes and birds, and non-living particles such as lakes, rivers or even gusts of wind. On the other hand, urban landscapes accommodates human activities that tend to dominate sounds, such as building machines, tires braking on the street, different types of sirens or even people conversations. Sounds derived from a landscape produce its unique soundscape which is a combination of a wide range of distinct components. Soundscape ecology is a relatively new and very active field of research that studies the relationships between soundscape composition, biodiversity patterns, and the definitely the interactions of organisms with their environment live in.

Soundscape ecology originated from the relatively era of bioacoustics, which has been the study of animal communication and behavior for many years. This fact has indisputably led to a new way of grasping biodiversity surveillance, from the perspective of the individual species to assessing the overall biological diversity that generates the soundscapes.

More formally, a soundscape is a dedicated sound event or a combination of sound events that derive from the outer environment. Generally, the study of soundscape environments is a field of the soundscape ecology[11].

The grasp of the soundscape term refers to both the natural surroundings, consisting of sounds like animal vocalizations, biophony[12] and geophony[13]. Without doubt, the term soundscape also consists of the listener's perception of sounds heard. One can claim that the term soundscape can also refer to either different audio recordings or performances of sounds that create the feeling of a specific acoustic environment.

The philosophy of the soundscape concept can be used to evaluate and describe acoustic environments. Basically the focus is not on whether the sounds are loud, but on how humans perceive them in a specific situation. Taking this into consideration, studying meticulously the soundscape analysis can be used for:

- Urban planning,

[11] Soundscape ecology is the study of the acoustic relationships between living creatures or humans and their environments
[12] Biophony is referred to collective habitat expression
[13] Geophony is referred to sounds of the weather and other natural elements

- Noise control and monitor,
- Event detection

3.3. Event Detection

Monitoring of human and social activities is becoming increasingly important in the living habitat from public security to safety applications. The recognition of dangerous or rather suspicious events is significant in all environments (indoor and outdoor), such as smart-homes, residential areas, offices, elevators and undoubtedly in all modern smart cities.

Environmental audio scene and sound event recognition are the basic processes involved in many audio surveillance applications. Despite numerous approaches have been arisen, robust environmental audio surveillance remains a big challenge because of various reasons, like background noises and the lack of universal and multi-modal datasets.

Event detection is the task of manipulating or analyzing different events in order to uncover sets of common patterns within the same event. These patterns define the event type. If some events match a specific pattern, then a specific event occurs. Some examples of these events can be an unexpected explosion, a fire outbreak, a movement in CCTV bank systems, traffic, construction activities during siesta hours, industrial, and social activity, etc. The analysis typically entails filtering and aggregation of events.

In the era of soundscape analysis, Sound Event Detection (SED) is mostly analyzed and studied. The main goal of SED methods is to recognize what is happening in an audio signal and when it is happening. In other words, the target is to recognize at what temporal instances different sounds are active within an audio signal.

In other words, it can be claimed that SED is a process of automatically detecting sound events from an audio source. This benefits many applications such as smart homes, smart speakers, smart house appliances, mobile devices, etc.

Event detection and sound analysis has caught a lot of researchers' eyes in the recent years, because a sound does not often come from a single source but it is commonly a combination of sounds from many different sources.

3.4. Machine Listening

Machine Listening is the audio substitute for Computer Vision. It combines and leverages modern techniques (ex. signal processing, ML, DL) to develop intelligent systems to grab significant and meaningful information from different sounds. Such information can be car engines, vehicle horns, police sirens, construction machines etc. Detecting such sounds is quite challenging taking into consideration the complexity and the variety of the sound sources, the auditory scenes and the background of the modern urban acoustic environments.

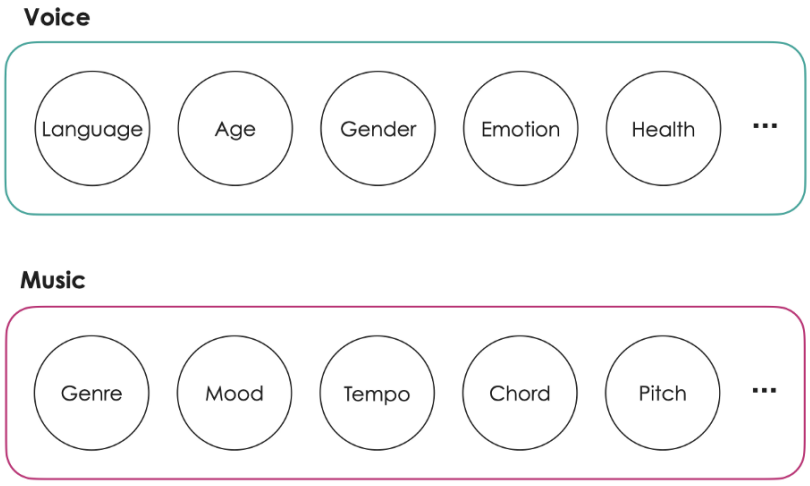
It can be claimed that speech and sound recognition is one of the most widely evolved technology in the modern industry which allows people to interact and communicate with computers in a more natural way. In the past It was very hard for computers to perceive human's speech, but the technological level of it enhanced from 2010 and on, when modern DL techniques came into play.

Computer vision, NLP, and speech recognition in general, are highly significant technologies for AI. However, modern technology miss an important thing; Sound. Speech is nothing else but sound. Despite existing millions or even billions of sounds in the whole environment (ex. urban, rural etc.), machines still do not well understand what is going on around them.

Machine listening, then can be described as a research area to make a system understand non-verbal information from an audio. A formal definition from the machine listening research laboratory from Queen Mary, University of London is the following:

"Machine listening is the use of signal processing and machine learning for making sense of natural / everyday sounds, and recorded music".

People's voices contain linguistic information. Besides this information, humans are also capable of guessing different clues from the voice. Such characteristics could be the age, the gender, a possible emotion etc. Music is another type of audio that contains even more complex information such as genre, mood, tempo etc.



Picture 3.2: Voice and music characteristics (Source: Google)

Undoubtedly, either voices or music are a very small percentage of what one hears during the day. Actually, humans do not know how many sounds can distinguish and there are no clear boundaries between different sounds. In machine listening, all other sounds are often called environmental sounds and they are divided into two main groups of topics which are the **acoustic scenes** and the **acoustic events**.

Picture 3.3: Acoustic scenes and Acoustic events (Source: Google)

The acoustic scenes are location-related information such as buses, parks, libraries, cafes or city centre sounds. It is impossible to recognize the scene with very short audio, so normally researchers assume that at least 10 seconds of audio is required to estimate the scene. On the contrary, acoustic events terminology is normally used for shorter sounds that include specific events such as glass break, knock, car horn, or dog bark. It might be a very short sound like 0.1 seconds, but also can be quite long like continuous water flow.

Despite the machine listening sector has been actively researched since more than a decade ago, it was still quite far away from the point that can be widely applied to real-world problems and applications, even after modern DL techniques and NN algorithms were introduced. Finally, developers have made a breakthrough and DL techniques and approaches outperformed classic ML methods in 2017.

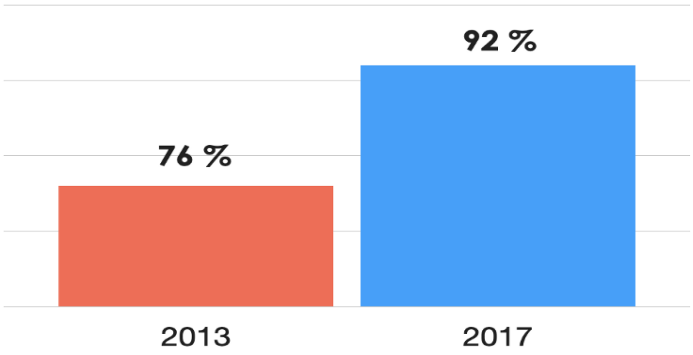


Figure 3.2: ML (red) vs DL (blue) techniques on Machine Listening Problem (Source: Google)

Advanced ML and NN architectures arose millions of opportunities that can give a positive impact on the quality of our daily life. Future machine listening techniques need to aim for general auditory intelligence that can be used in a real-world scenarios. To do so, it requires a range of domain knowledge in vast array of sectors such as signal processing, cognitive sciences, music, psychoacoustics, acoustics, and ML, because the real-world environment and auditory perception of human are rather highly complicated.

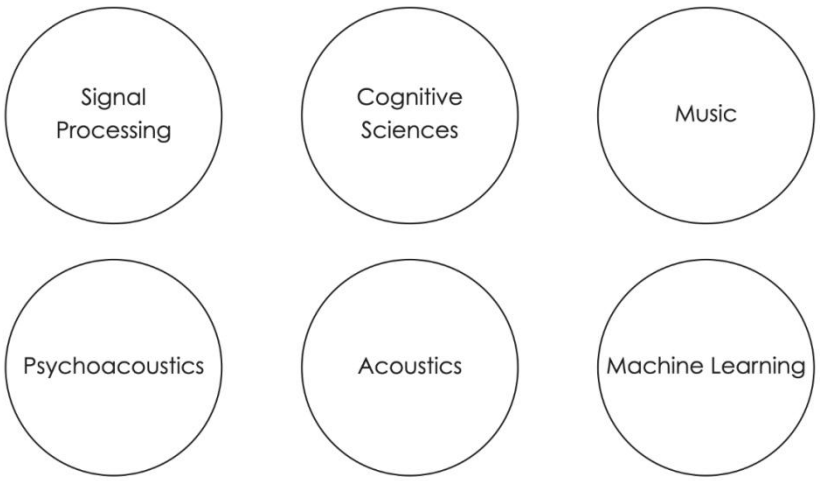


Figure 3.3: Prerequisite domains for Machine Listening (Source: Google)

3.5. Audio Sampling

The term Audio Sampling refers to the conversion of a continuous signal to a discrete signal. A sample is a specific point of the signal at a specific time or space. To this end, the original signal can be reconstructed by the samples. Sampling can be performed in the era of time, space or other dimension. When referring to time, the sampling can be performed by measuring the value of the continuous signal every ***T*** seconds. This value of T is usually called as **Sampling Period.** On the other hand, the number of samples gained in a time period of 1 sec is usually referred as **Sampling Rate.**

In order to rebuild the continuous signal from many samples, special interpolation algorithms are employed. The Whittaker–Shannon interpolation formula is mathematically equivalent to an ideal low-pass filter whose input is a sequence of Dirac delta functions that are modulated (multiplied) by the sample values.

When the time interval between adjacent samples is a constant (T), the sequence of delta functions is called a Dirac comb. Mathematically, the modulated Dirac comb is equivalent to the product of the comb function with s(t). That purely mathematical abstraction is sometimes referred to as impulse sampling [17].

In other and more simple words, *discrete time* refers to the fact that although in nature time runs on a continuum, in the digital world we can only manipulate *samples* of the real-world signal that have been drawn on discrete-time instances. This process is known as *sampling* and it is the first stage in the creation of a digital signal from its real-world counterpart [18].

To provide with a toy example, the first sample may have been taken at time 0 (the time when measurement commenced), the second sample at 0.001 sec, the third one at 0.002sec, and so on. In this scenario, the time instances are equidistant and if we compute the difference between any two consecutive time instances the result is $T_s = 0.001sec$, where T_s is the sampling period. In order to go from the time to frequency domain the mathematical type $F_s = \frac{1}{T_s}$ is employed. So for this specific example, the sampling frequency is F_s = 1000Hz, which means 1000 samples for every 1 sec.

A major issue in the context of sampling is how high the sampling frequency should be (or equivalently, how short the sampling period has to be) as proposed in [18] and [19]. It turns out that in order to successfully sample a continuous-time signal, the sampling frequency has to be set equal to at least twice the signal's maximum frequency [19].

3.5.1. Applications of Audio Sampling

Sampled or Digital Audio is mainly used for reproducing a sound. The phases followed are: Analog to Digital conversion, transmission, Digital to Analog conversion and storage. When one needs to capture the human hearing frequencies[14], audio is typically sampled at 44.100Hz, 48.000Hz, 88.200Hz or 96.000Hz.

The Audio Engineering Society (AES)[15] suggests sampling rate at 48.000Hz for almost all aspects and applications of life.

[14] Human hearing grasps frequencies between 20 – 20.000Hz

[15] The AES is a professional body for engineers, scientists, other individuals with an interest or involvement in the professional audio industry.

Table 3.1: Sampling rates and usage

Sampling Rate	Usage
8.000Hz	Telephones, microphones, voice to voice telecommunications
11.025Hz	MPEG audio, low quality CDs
16.000Hz	VoIP communications
22.050Hz	Low quality CDs, MPEG audio, AM Radio
32.000Hz	miniDV video format, video tapes, high quality microphones
44.056Hz	NTSC videos
44.100Hz	Audio CDs, PAL videos
48.000Hz	Standard Sampling Rate for videos, tape recorders etc. Also used for DVDs and digital TVs
96.000Hz	Blu-Ray Discs, High Quality DVDs,
176.400Hz	Used by High Quality CD cameras and other professional CD applications
192.000Hz	DVDs, Blu-Ray Discs, High Definition CDs and other professional audio applications
352.800Hz	Digital Extreme Definition, used for recording and editing Super Audio CDs,
2.822.400Hz	Super Audio CD (SACD) and Direct Stream Digital
5.644.800Hz	Direct Stream Digital at 2x the rate of the SACD.
11.289.600Hz	Direct Stream Digital at 4x the rate of the SACD
22.579.200Hz	Direct Stream Digital at 8x the rate of the SACD

3.6. STEREO and MONO Audio

As proposed to [20], **MONO** or sound utilizes only one channel in order to convert a signal into a specific sound. Despite being multiple speakers available, the same signal will go the same to all of them. This then gives the effect that the sounds, even if they are coming from separate speakers, are coming from one single and unique source.

On the other hand, **STEREO** sound utilizes more than one channel when converting a signal into a sound, and each signal which is sent out, is unique.

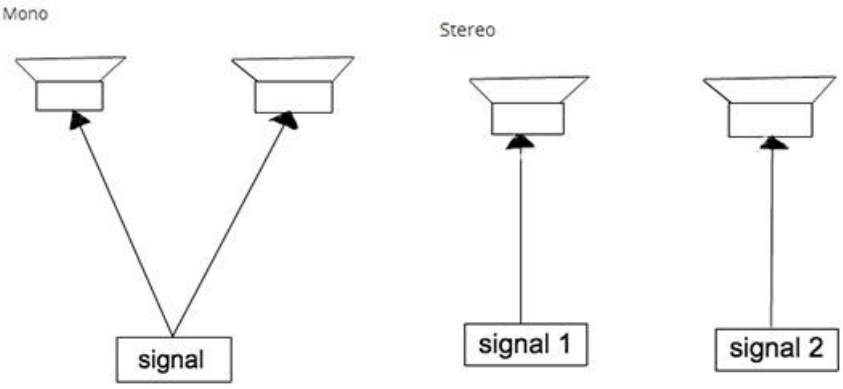


Figure 3.4: STEREO vs. MONO sound architectures (Source: Google)

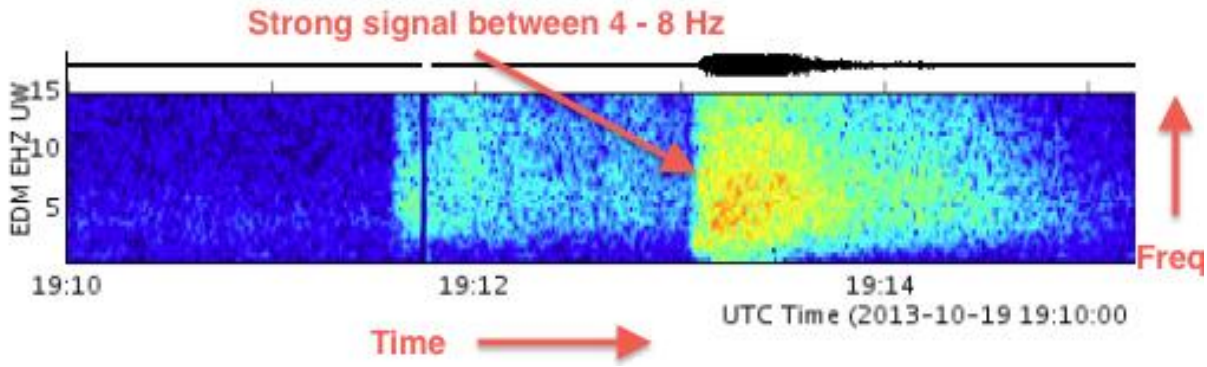
STEREO sound, in other words, gives the effect of sound coming from completely different sources and positions, which is typical and very common in today's technology, especially in speakers that are produced for the surround sound effect.

MONO sound is when only one channel is used whereas STEREO sound is when multiple channels are used to convert multiple signals to sounds.

3.7. Audio Spectrograms

A spectrogram is a visual representation of signal strength over time at different frequencies present in a particular waveform [21]. In other words, a spectrogram is a detailed view of audio, able to represent time, frequency, and amplitude all on the same graph. Not only can one distinguish if more or less energy exists at specific frequencies, but also understand how energy levels change over time, and present the evolution of the signal in the time frequency domain.

In other fields of science spectrograms are mainly used to depict frequencies of sound waves produced by humans, machinery, animals, whales, jets, etc., as recorded by microphones. In the seismic world, spectrograms are increasingly being used to look at frequency content of continuous signals recorded by individuals or groups of seismometers to help distinguish and characterize different types of earthquakes or other vibrations in the earth [21].



`

Figure 3.5: Spectrogram (Source: Google)

Spectrograms are colored two-dimensional graphs. Time runs from left (oldest) to right (most recent), along the x axis. The y axis represents frequency, which can also be thought of as pitch or tone, with the lowest frequencies at the bottom and the highest frequencies at the top. The amplitude (or energy or "loudness") of a particular

frequency at a particular time is represented by color, with dark blues corresponding to low amplitudes and brighter colors up through red corresponding to progressively stronger (or louder) amplitudes.

3.8. Auditory Scene Analysis and Sound Event Recognition in Surveillance

As discussed in [26], capturing of both human and other social activities is becoming increasingly spreading in peoples' living environment for general safety. The recognition of suspicious events is significant in both outdoor and indoor surroundings. These could be smart-homes, residential areas, child-care centers, dwellings, offices, lifts, and smart cities.

ASA and sound event recognition are the fundamental activities involved in many audio surveillance applications. Despite a vast number of approaches have been proposed so far, solid environmental audio surveillance remains a huge challenge because of a wide range of reasons, such as different types of overlapping audio sounds, background noises, and lack of universal and multi-modal datasets dedicated for such purposes.

The main goal then is to review various features of representing audio scenes and sound events and provide appropriate ML techniques for audio surveillance activities.

3.8.1. Features for Audio Surveillance Systems

Related work [26] proposed many methods for extracting different audio features according to any occasion. The solid audio feature selection plays a crucial role in audio surveillance of the environment. In fact, audio features are intended to grab the discriminative information useful for classification purposes while decreasing background noises and other redundancies.

Most feature extraction approaches are based on frame-based processing involves dividing an audio signal into frames. In other words, features are extracted from different frames and this sequence of feature vectors is used to represent an audio signal. Feature extraction can be split into two (2) main categories, according to Cowling and Sitte 2003:

- Stationary
- Non - Stationary

Stationary feature extraction produces meticulous frequency contents from the whole signal. But, it is not able to recognize where these frequencies are available in the signal.

Stationary feature extraction consists of eight (8) main features commonly used in non-speech sounds as following:

1. Frequency,
2. Homomorphic Cepstral Coefficients,
3. Mel Frequency Cepstral Coefficients (MFCC),
4. Linear Prediction Cepstral (LPC) coefficients,
5. Mel Frequency LPC Coefficients,
6. Bark Frequency Cepstral Coefficients,
7. Bark Frequency LPC Coefficients and
8. Perceptual Linear Prediction (PLP) features

On the other hand, non-stationary feature extraction breaks the signals into discrete time units. This helps to analyze and uncovers the occurrence of each frequency component in a specific part of the signal in order to understand the nature of it.

The main features, that use different algorithms to obtain a Time-frequency Representation (TFR) of a signal and are commonly referenced in general literature (Cowling and Sitte 2003) are the following:

1. Short-time Fourier Transform (STFT),
2. Fast Wavelet Transform (FWT),
3. Continuous Wavelet Transform (CWT) and
4. Wigner-Ville Distribution (WVD).

It is stated that in the present Thesis, the MEL Spectrograms method is used and analyzed. All the other methods are simply mentioned.

3.8.2. Deep Learning Approaches for Auditory Scene Analysis

The complex recognition task with more data as discussed in related work [26], can be effectively managed by DL methods where classic ML methods cannot guarantee a very good performance. Table 3.2 depicts many DL approaches used for auditory scene analysis tasks.

CNN is one of the most popular NN architectures used in DL. The DL approach for ASA has been proposed in Petetin et al. (2015) using MFCC, spectral centroid, and spectral flatness features. DL model-based techniques outperformed the classical ML classifiers[16]. The results have been significantly good for DL with cepstral and frequency features compared with well-known features such as HOG classified by the SVM approach. In Han and Lee (2016), multi-width frequency-delta data augmentation was applied on input features for training using the CNN models. The frequency-delta features and Melspectrograms are used as input features for data augmentation to represent examples with same labels.

Another related work in Mafra et al. (2016) reviewed different time aspects when combining the features using different classic ML classifiers. This specific representation with temporal averaged Mel-log spectrograms using SVM achieved better recognition accuracy.

In another related work, the authors in Phan et al. (2017) suggested an approach called Convolutional Neural Network–Label Tree Embeddings (CNN-LTE) strategy. Using the CNN-LTE approach, the features were represented in the form of label tree embedding images. Then these features were learned using the simple 1D pooling layers of CNNs.

Table 3.2: DL Approaches for Auditory Scene Analysis

FEATURES	METHODS	REFERENCES	DATASET
MFCC features, spectral centroid and spectral flatness	DNN	Petetin et al. 2015	LITIS Rouen
MEL Spectrograms	CNN	Han and Lee 2016	TUT-DCASE 2016
MEL LOG Spectrograms	SVM	Mafra et al. 2016	DCASE 2013
Parametrized MFCC features	CNN	Eghbal-zadeh et al. 2017	TUT-DCASE 2016

[16] SVM classifiers

It is mentioned that in the Present Thesis all the TL techniques performed, used the TUT-DCASE 2017 as reference and will be discussed more in the following sections (Appendix A).

3.9. ATHens Urban Soundscape Dataset

3.9.1. Dataset Description

When referring to soundscape one can understand an auditory environment (either urban or rural). As proposed to related work [27], ATHens Urban Soundscape (ATHUS) is a dataset of audio clips of audio clips from urban environments, which has been humanly annotated by proposing a specific soundscape quality for each clip.

To this end, a vast array of different users have perceived and recorded audio sounds by using a simple Android application. Then, each recording was annotated in terms of the level of pleasantness of the soundscape, in a range of 1 (unbearable) to 5 (optimal).

The dataset and according to [27], was made publicly available (in http://users.iit.demokritos.gr/~tyianak/soundscape) as an audio feature representation form. In addition, in [27] is presented a basic method that shows how the specific dataset can be used to train supervised models in order for a developer to predict soundscape quality levels in different environments. In other words, the main purpose of this attempt was to provide to different developers and ML engineers, an introduction to audio recognition and soundscape analysis in different and diverge urban spaces, which could lead to powerful assessment tools in the hands of policy makers with regards to noise pollution and sustainable urban living.

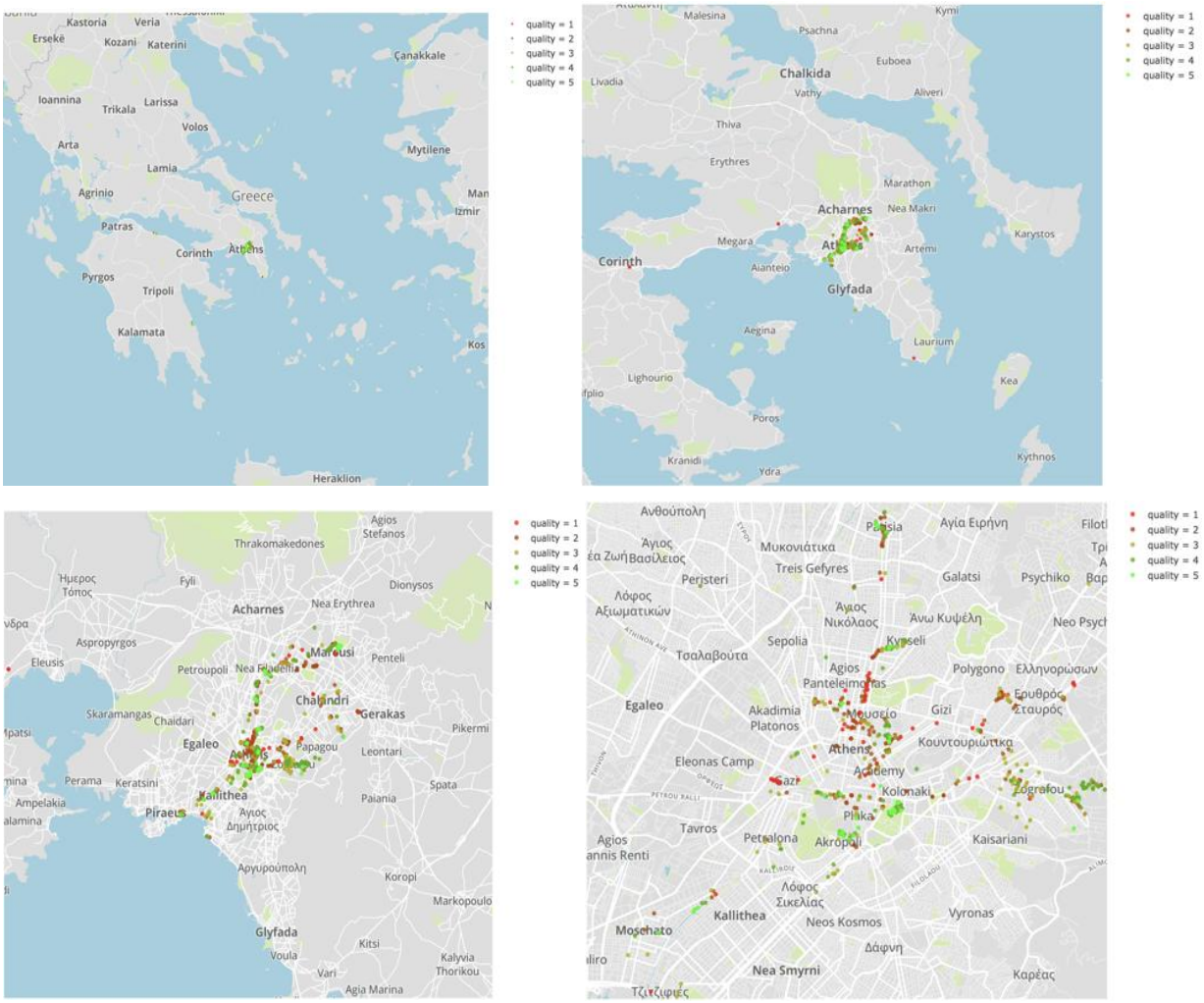
3.9.2. Audio Collection and Annotations

All the audio files were collected in Athens, Greece in many different locations as Picture 3.4. suggests. The files have taken place in a period of almost 4 years, by 10 different humans using 13 different types of smart phone devices. There were 979 recordings and each recording was around 27 seconds of average duration. Some of their statistics are shown in the following table 3.3.

Table 3.3: ATHUS Dataset Statistics

Total Number of Audio Clips:	979
Min Duration:	11.40 sec
Max Duration:	78.83 sec
Avg. Duration:	26.99 sec
Total Duration:	7.33 hrs

Each audio file were annotated by a specific user that performed the file by using a an Android application.



Picture 3.4: Distribution of ATHUS recordings in Athens

The application is available online at:

(http://users.iit.demokritos.gr/~tyianak/soundscape/).

Prior to starting the recording process, the application grabs the geospatial coordinates using the GPS sensor of the mobile phone, and the user provides some demographic information such as their age, gender and educational level.

Then, the recording process starts, and as soon as the user stops it, they finally provides with the perceived soundscape quality, in a range between 1 and 5 (1 corresponds to unbearable soundscape quality and 5 to optimal quality respectively).

Figure 3.6 depicts the distribution of the audio clips per class (1 to 5) and Figure 3.7 shows the distributions of the data after the Train/Test split. It is mentioned that in order to perform the experiments and for testing reasons, the dataset split into

{train: 80%, test: 20%} partitions.

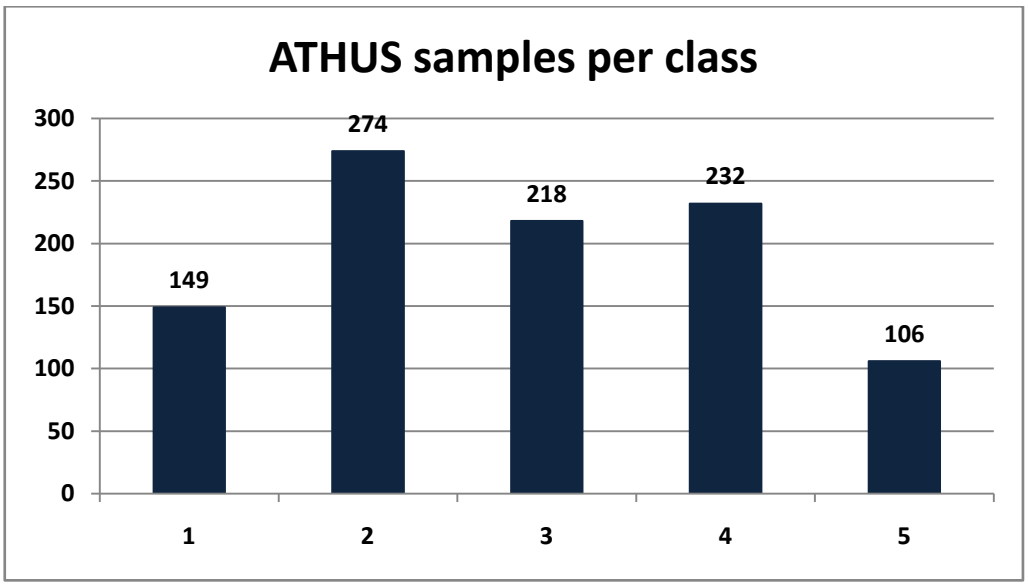


Figure 3.6: ATHUS samples per class

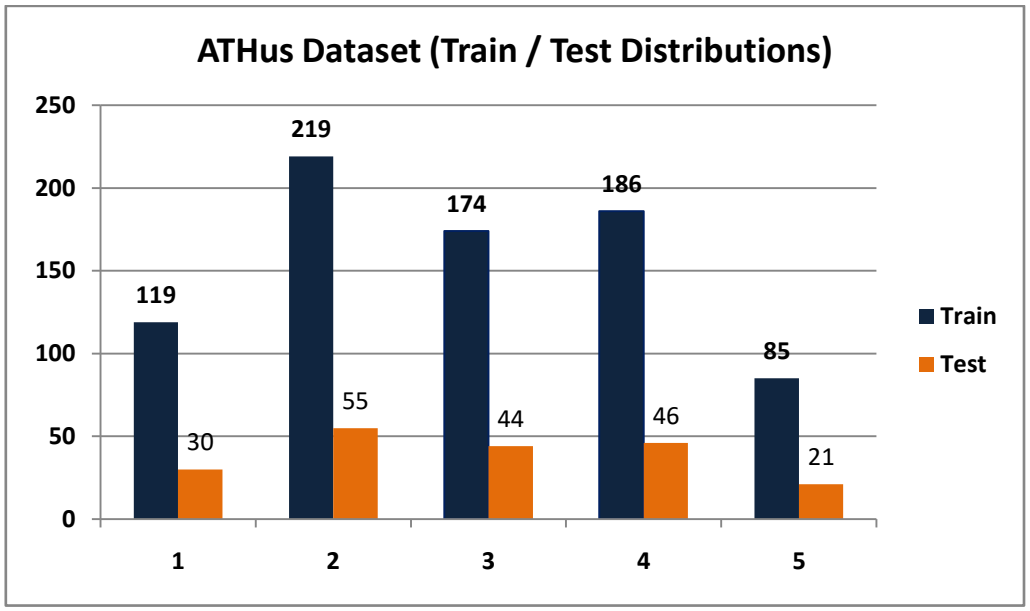


Figure 3.7: ATHus Dataset (Train / Test Distributions)

4. EXPERIMENTS - ATHUS DATASET CLASSIFICATION

4.1. Experimental Setup

In this Chapter will be discussed and analyzed all the experiments performed with the corresponding results. For the present Thesis the ATHUS dataset has been used, which proposed in related work [27] and presented meticulously in previous Chapters. For the TL techniques the TUT 2017 dataset has been used, as related work [28], information of it presented in Appendix A.

The main problem studied and analyzed, is a classification task. The training set is the ATHUS dataset. The main purpose is to classify audio clips in terms of urban quality. As proposed in [27], quality (1) represents unbearable soundscape quality and (5) represents quality. To this end, the dataset has been separated into [train, dev] = [80%, 20%] before moving into the ANN architectures.

The ANN architecture consists of seven (7) layers; four (4) CNN and three (3) linear at the end. More about the DAFP are stated in Appendix B.

The dataset was transformed many times in order to be trained and fitted in many different models. The first approach, presented in this Chapter, was to transform the audio clips into 8KHz and MONO. The same engineering also performed into the TUT dataset, used for the TL evaluation.

The second approach, was to train the dataset as it is and without any other user intervention (44kHz and MONO). The same engineering also performed into the TUT dataset, used for the TL evaluation.

Finally, the third approach was to train the dataset by transforming it into 8kHz and MONO. In addition, 1 sec segmentation was applied to all datasets (1 sec segmentation performed to every single audio clip) used (in both ATHUS and TUT 2017).

The problem has been developed in a Linux Operating System (OS), Python 3 developing environment and has been used all the modern packages and libraries that deal with ML approaches including the DAFP library developed by Theodoros Giannakopoulos (Appendix B). The package utilizes all the modern DL callbacks classes such as the Early Stopping[17], Save Best Model[18] and Reduce Learning Rate on Plateaus[19].

For the needs of the present Chapter more than 60 models have been trained and evaluated and 34 are presented in the present Chapter.

All the experiments of the present Thesis are summarized in the following table 4.1.

[17] Early Stopping stops the training phase at an optimal point where no further improvement of the model is taking place.
[18] Save only the best model produced on the training phase.
[19] Reduce learning rate when a metric has stopped improving.

Table 4.1: Experiments

8kHz and MONO	**MODEL**
	Basic Training Of Soundscape
	Transfer Learning of Soundscape with TUT as Source Model (No Freezing Layers Techniques)
	Transfer Learning of Soundscape with TUT as Source Model (Freezing CNN Layers)
	Transfer Learning of 5 class approach Soundscape with TUT as Source Model (No Freezing Layers Techniques)
	Transfer Learning of 5 class approach Soundscape with TUT as Source Model (Freezing CNN Layers)
	Transfer Learning of 3 class approach Soundscape with TUT as Source Model (No Freezing Layers Techniques)
	Transfer Learning of 3 class approach Soundscape with TUT as Source Model (Freezing CNN Layers)
	Transfer Learning of 3 class approach Soundscape with TUT as Source Model (Freezing 3 out of 4 CNN Layers)
44.1kHz and MONO	**MODEL**
	Basic Training Of Soundscape
	Transfer Learning of Soundscape with TUT as Source Model (No Freezing Layers Techniques)
	Transfer Learning of Soundscape with TUT as Source Model (Freezing CNN Layers)
	Transfer Learning of 5 class approach Soundscape with TUT as Source Model (No Freezing Layers Techniques)
	Transfer Learning of 5 class approach Soundscape with TUT as Source Model (Freezing CNN Layers)
	Transfer Learning of 3 class approach Soundscape with TUT as Source Model (No Freezing Layers Techniques)
	Transfer Learning of 3 class approach Soundscape with TUT as Source Model (Freezing CNN Layers)
	Transfer Learning of 3 class approach Soundscape with TUT as Source Model (Freezing CNN Layers with class weighting balance)
	Transfer Learning of 5 class approach Soundscape with TUT as Source Model (Freezing 3 out of 4 CNN Layers)
8kHz, MONO and 1sec segmentation	**MODEL**
	Basic Training Of Soundscape
	Transfer Learning of Soundscape with TUT as Source Model (No Freezing Layers Techniques)
	Transfer Learning of Soundscape with TUT as Source Model (Freezing CNN Layers)
	Transfer Learning of 5 class approach Soundscape with TUT as Source Model (No Freezing Layers Techniques)

	Transfer Learning of 5 class approach Soundscape with TUT as Source Model (Freezing CNN Layers)
	Transfer Learning of 3 class approach Soundscape with TUT as Source Model (No Freezing Layers Techniques)
	Transfer Learning of 3 class approach Soundscape with TUT as Source Model (Freezing CNN Layers)
	Transfer Learning of 3 class approach Soundscape with TUT as Source Model (Freezing 3 out of 4 CNN Layers)

4.2. ATHUS Basic Training using DAFP

The first approach to the classification problem is a simple / basic training of the ATHUS dataset using the DAFP library. The dataset used has been transformed into MONO and 8kHz.

The model trained for 20 epochs and produced the results below:

- F1-Score = 33% (testing amidst of 20% of the samples during training),
- F1-Score = 31% (testing on the unseen development dataset).

With the following confusion matrix:

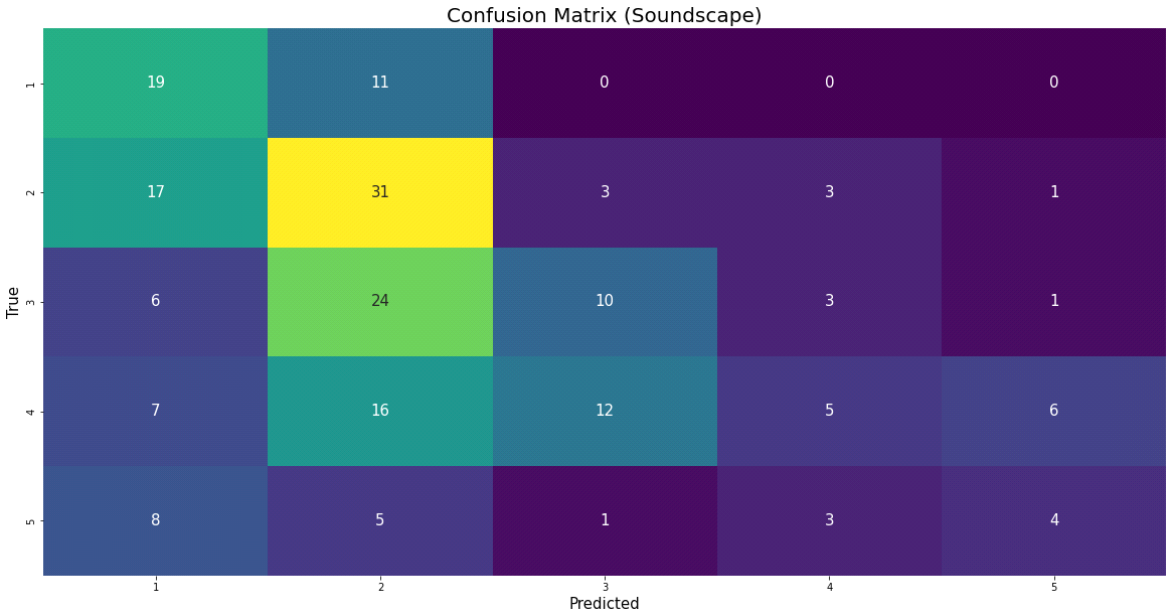


Figure 4.1: Confusion Matrix of ATHUS basic training using DAFP (8kHz and MONO)

The second approach, to the classification problem and similar to the previous, is a simple / basic training of the ATHUS (44.1kHz and MONO this time) dataset using the DAFP library.

The model trained for 22 epochs and produced the results below:

- F1-Score = 35% (testing amidst of 20% of the samples during training),
- F1-Score = 30% (testing on the unseen development dataset).

With the following confusion matrix:

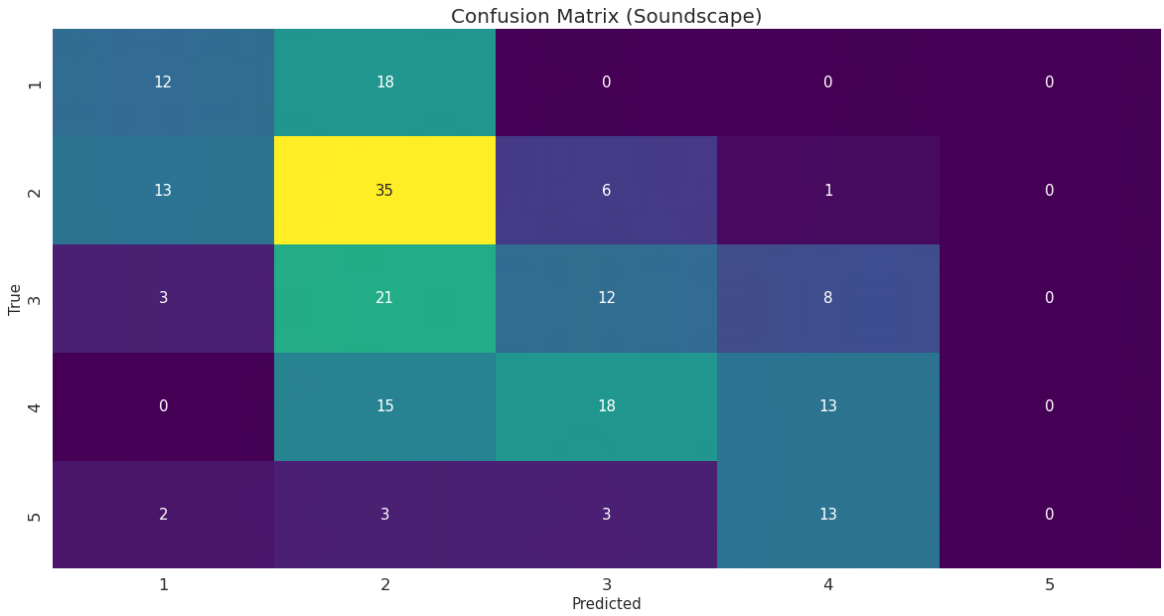


Figure 4.2: Confusion Matrix of ATHUS basic training using DAFP (44.1kHz and MONO)

The third approach, to the classification problem, is a simple / basic training of the ATHUS (8kHz, 1sec segmentation and MONO) dataset using the DAFP library.

The model trained for 53 epochs and produced the results below:

- F1-Score = 55% (testing amidst of 20% of the samples during training),
- F1-Score = 41% (testing on the unseen development dataset).

With the following confusion matrix:

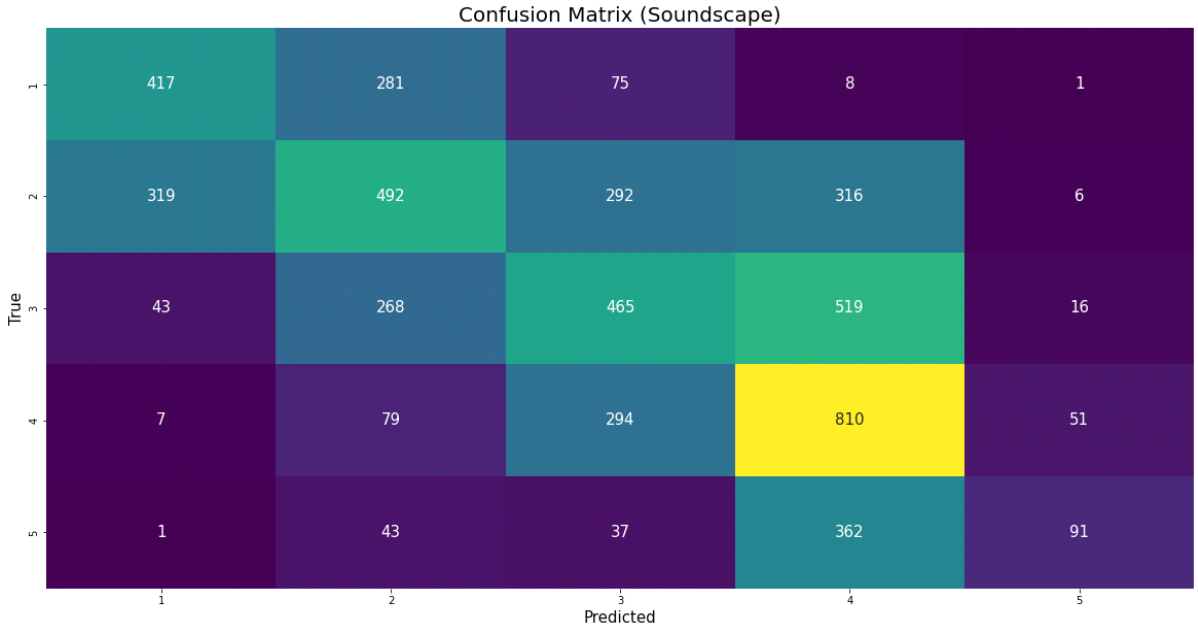


Figure 4.3: Confusion Matrix of ATHUS basic training using DAFP (8kHz, MONO and 1sec segmetation)

4.3. TUT 2017 Basic Training using DAFP

In order to perform TL technique a reference dataset is required. For this purpose the TUT Acoustic Scenes 2017 dataset is employed. To this end, the TUT dataset was trained used DAFP.

Again, here, there are three (3) approaches as the ATHUS dataset follows; The first approach is to train the TUT dataset in 8kHz and MONO. Thus:

The model trained for 50 epochs and produced the results below:

- F1-Score = 79% (testing amidst of 20% of the samples during training)
- F1-Score = 77% (testing on the unseen development dataset)

With the following confusion matrix:

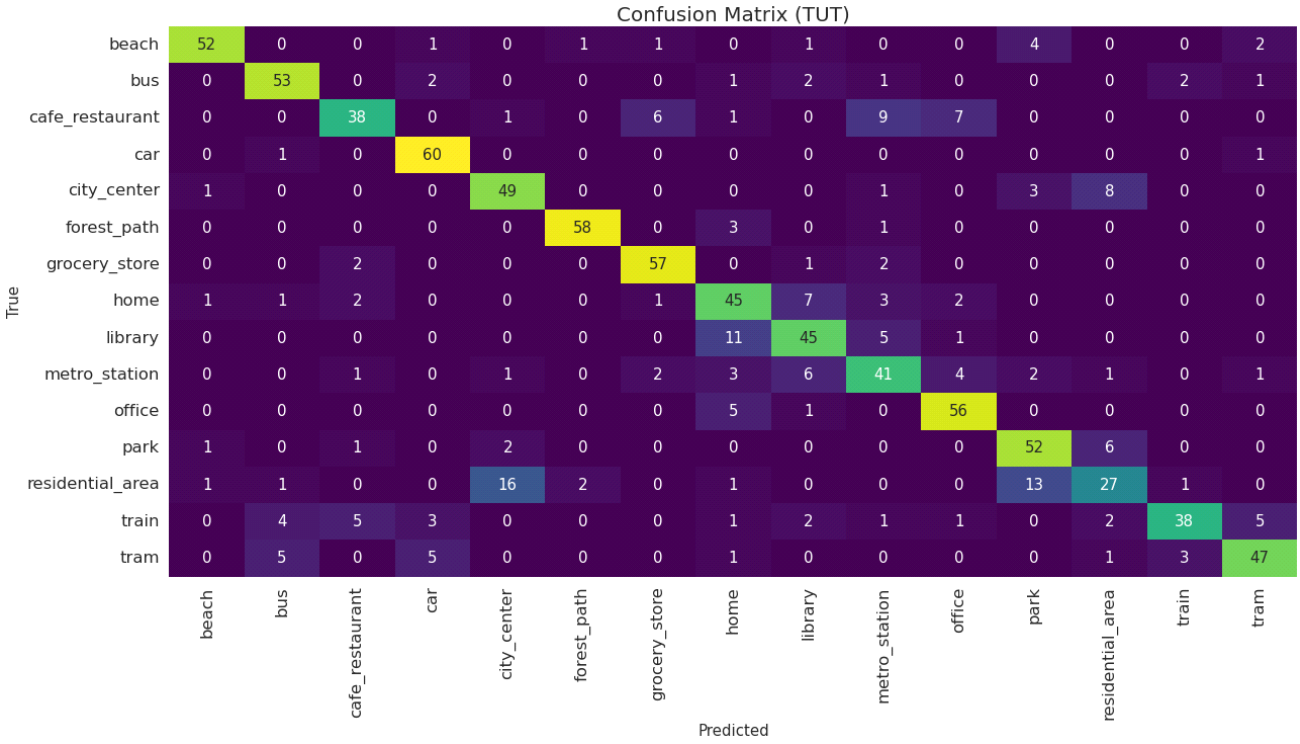


Figure 4.4: Confusion Matrix of TUT 2017 basic training using DAFP (in 8kHz and MONO)

The second approach followed, is the basic train of the original TUT dataset (44.1kHz and MONO) using DAFP.

The model trained for 71 epochs and produced the results below:

- F1-Score = 81% (testing amidst of 20% of the samples during training)
- F1-Score = 76% (testing on the unseen development dataset)

With the following confusion matrix:

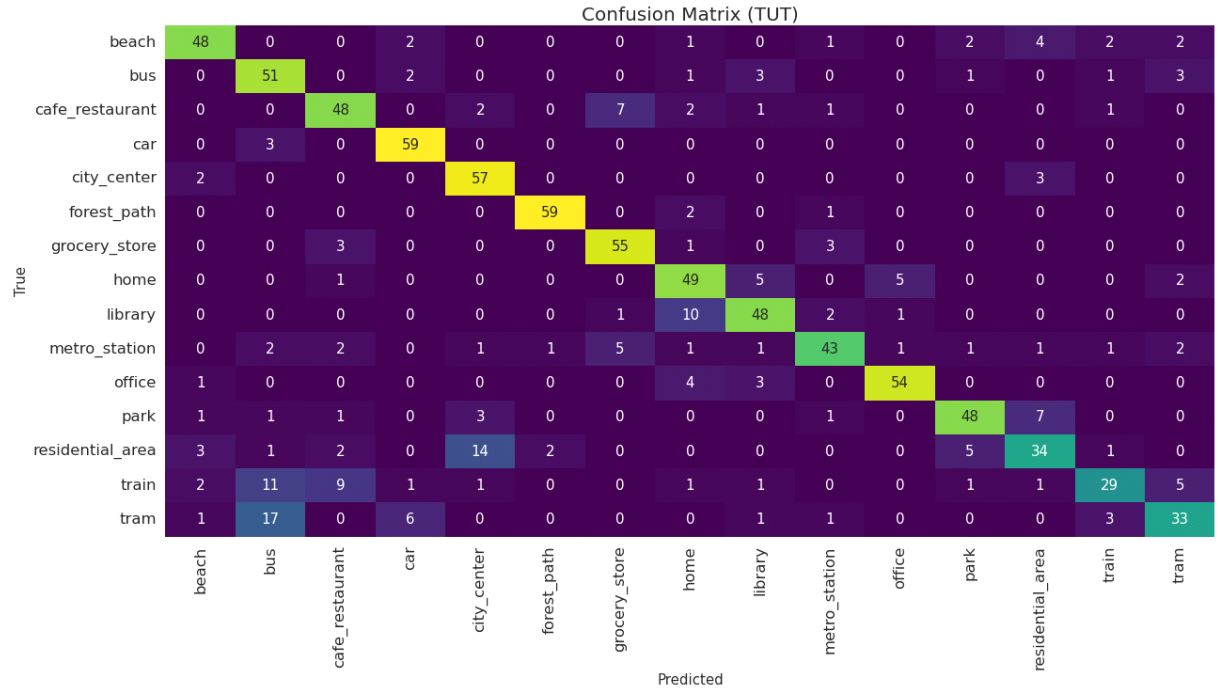


Figure 4.5: Confusion Matrix of TUT 2017 basic training using DAFP (in 44.1kHz and MONO)

Finally, the third approach is to train the TUT dataset (8kHz and MONO) in 1 sec segmentation. To this end, the original TUT (8kHz, 1sec segmentation and MONO) dataset was trained again used DAFP.

The model trained for 54 epochs and produced the results below:

- F1-Score = 94% (testing amidst of 20% of the samples during training)
- F1-Score = 91% (testing on the unseen development dataset)

With the following confusion matrix:

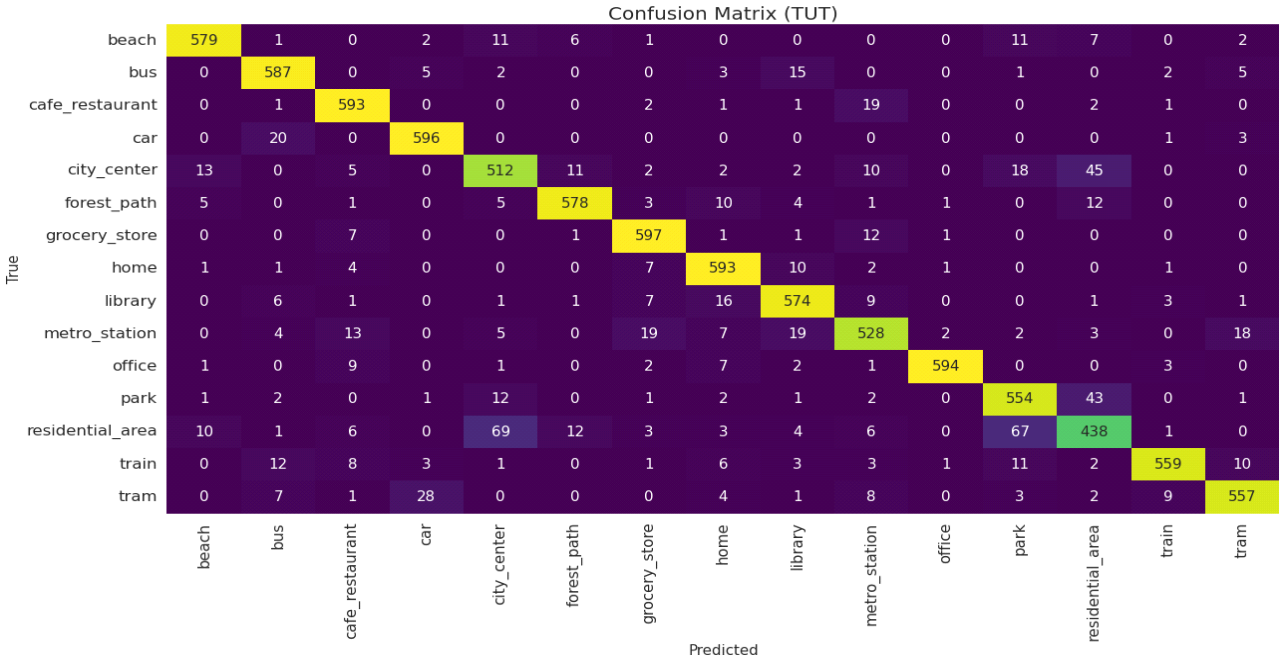


Figure 4.6: Confusion Matrix of TUT 2017 basic training using DAFP (in 8kHz, MONO and 1 sec segmentation)

4.4. Transfer Learning using DAFP (From TUT to ATHUS)

4.4.1. Transfer Learning with Strategy 0

In this experiment the TUT dataset has been utilized in order to provide TL techniques to ATHUS dataset. The Strategy used was 0 which means the model does not perform any freezing to CNN layers [all layers (both CNN and linear) are used to train and finetune the model].

In this part of the experiment three (3) different approaches are proposed (as in previous sections discussed), as follows:

- From TUT to ATHUS (8kHz and MONO)
- From TUT to ATHUS (44.1kHz and MONO)
- From TUT to ATHUS (8kHZ, MONO and 1 sec segmentation).

As the first approached implies, the TL model trained for 48 epochs and produced the results below:

- F1-Score = 37% (testing amidst of 20% of the samples during training)
- F1-Score = 36% (testing on the unseen development dataset)

With the following confusion matrix:

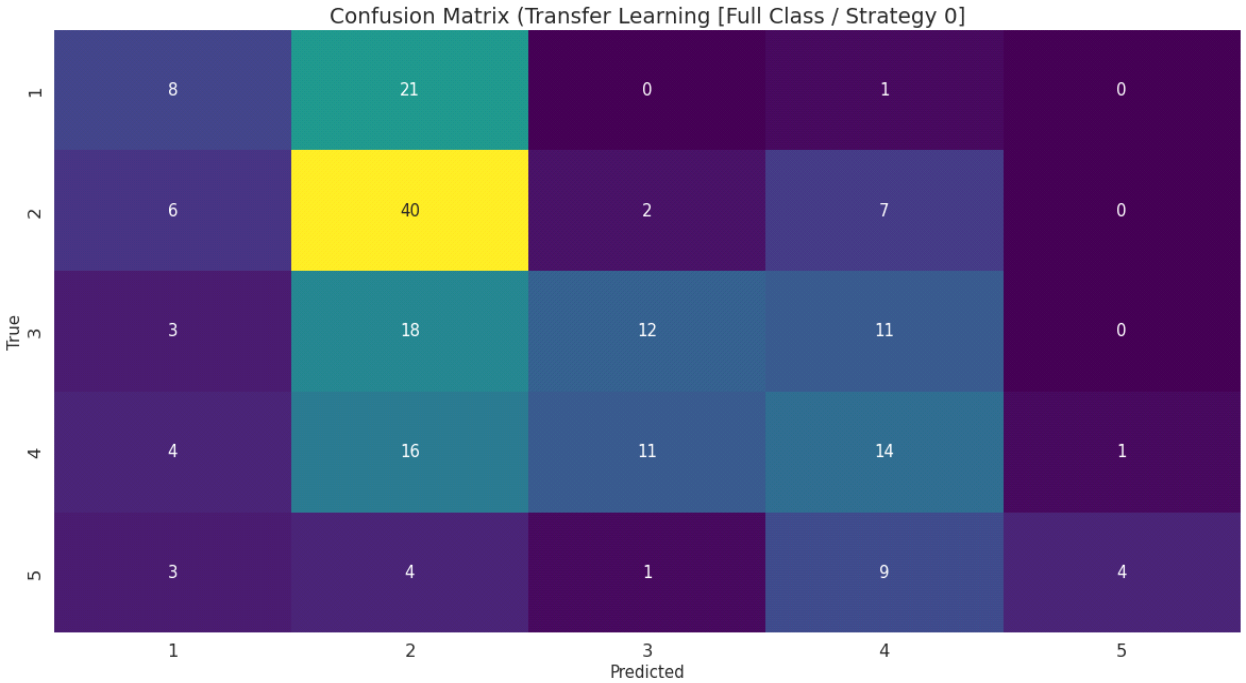


Figure 4.7: Confusion Matrix of TL using DAFP and Strategy 0 [From TUT to ATHUS (8kHz, MONO)]

In the second experiment the original TUT (44.1kHz and MONO) dataset has been utilized in order to provide TL techniques to ATHUS dataset. The TL model trained for 48 epochs and produced the results below:

- F1-Score = 38% (testing amidst of 20% of the samples during training)
- F1-Score = 38% (testing on the unseen development dataset)

With the following confusion matrix:

Figure 4.8: Confusion Matrix of TL using DAFP and Strategy 0 [From TUT to ATHUS (44.1kHz, MONO)]

In the last (third approach) experiment the TUT (8Hz, 1sec segmentation and MONO) dataset has been utilized in order to provide TL techniques to ATHUS dataset. The TL model trained for 36 epochs and produced the results below:

- F1-Score = 63% (testing amidst of 20% of the samples during training)
- F1-Score = 42% (testing on the unseen development dataset)

With the following confusion matrix:

Figure 4.9: Confusion Matrix of TL using DAFP and Strategy 0 [From TUT to ATHUS (8kHz, MONO and 1 sec segmentation)]

4.4.2. Transfer Learning with Strategy 1

In this experiment the TUT dataset has been utilized in order to provide TL techniques to ATHUS dataset. The Strategy used was 1 which means the model performs freezing to CNN layers [only linear layers are used to train and finetune the model].

Again and as discussed in the previous section, three (3) different approaches are proposed (as in previous sections discussed), as follows:

- From TUT to ATHUS (8kHz and MONO)
- From TUT to ATHUS (44.1kHz and MONO)
- From TUT to ATHUS (8kHZ, MONO and 1 sec segmentation).

As the first approached implies, the TL model trained for 24 epochs and produced the results below:

- F1-Score = 37% (testing amidst of 20% of the samples during training)
- F1-Score = 36% (testing on the unseen development dataset)

With the following confusion matrix:

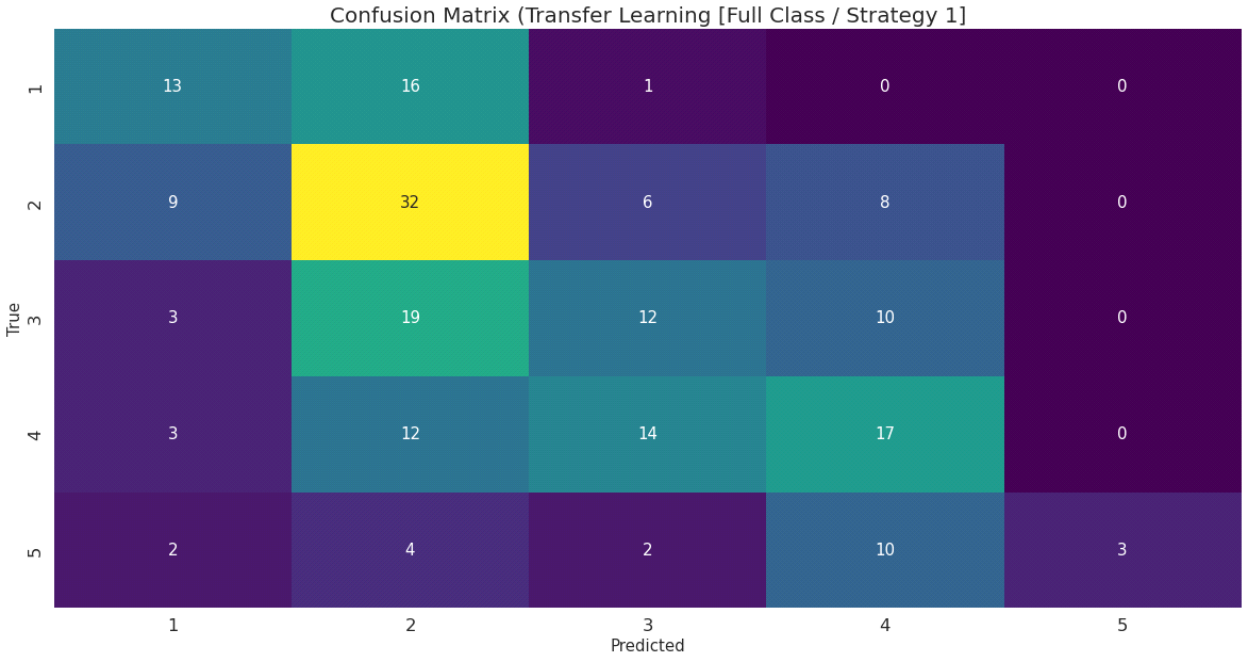


Figure 4.10: Confusion Matrix of TL using DAFP and Strategy 1 [From TUT to ATHUS (8kHz, MONO)]

In the second experiment the TUT dataset has been utilized in order to provide TL techniques to ATHUS dataset. The TL model trained for 27 epochs and produced the results below:

- F1-Score = 37% (testing amidst of 20% of the samples during training)
- F1-Score = 31% (testing on the unseen development dataset)

With the following confusion matrix:

Figure 4.11: Confusion Matrix of TL using DAFP and Strategy 1 [From TUT to ATHUS (44.1kHz, MONO)]

In the last approach the TUT dataset has been utilized in order to provide TL techniques to ATHUS dataset (8kHz, 1 sec segmentation and MONO sound).

The TL model trained for 47 epochs and produced the results below:

- F1-Score = 59% (testing amidst of 20% of the samples during training)
- F1-Score = 40% (testing on the unseen development dataset)

With the following confusion matrix:

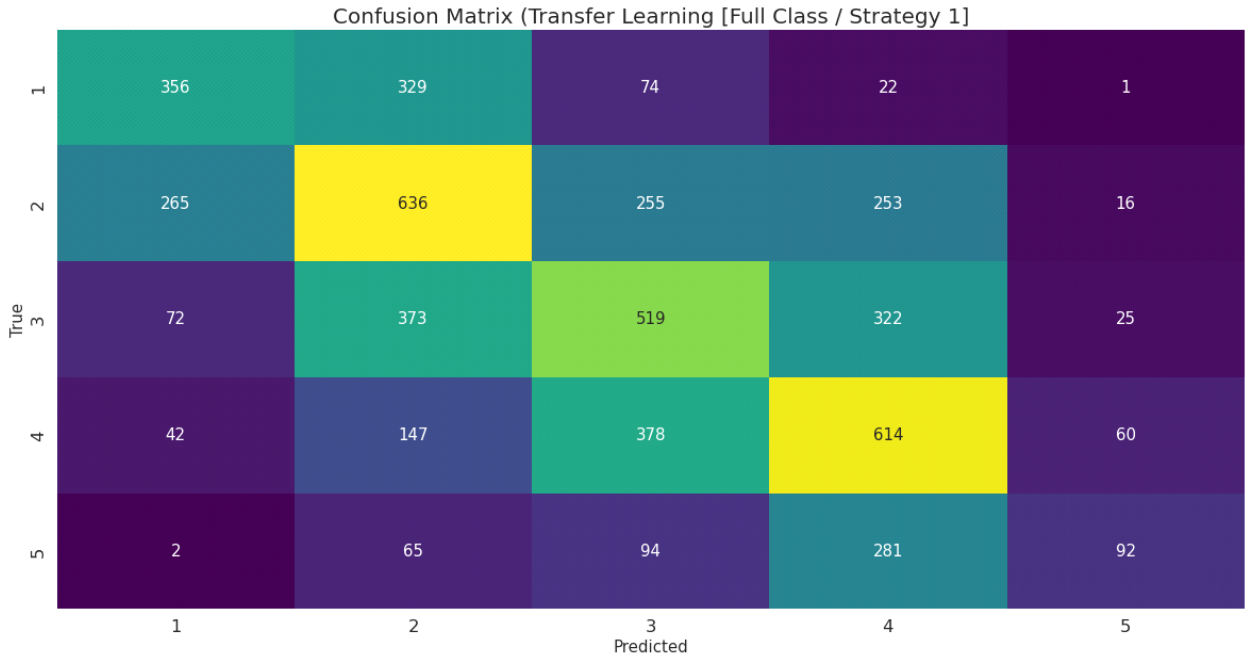


Figure 4.12: Confusion Matrix of TL using DAFP and Strategy 1 [From TUT to ATHUS (8kHz, MONO and 1 sec segmentation)]

4.5. TUT 2017 Basic Training using DAFP: A 5 class approach

In order to apply highly optimized TL techniques from the TUT 2017 to the ATHUS dataset another approach is employed. The TUT 2017 dataset retrained with only 5 classes. For this purpose and after many different experiments, the classes were chosen in a way that describes the ATHUS problem best.

To this end, the new classes that have been constructed are the following:

- City Center,
- Metro Station,
- Park,
- Forest Path,
- Library

that fit best to the ATHUS classes (from 1 to 5 respectively).

Three (3) different approaches are proposed, as follows:

- From TUT to ATHUS [5 class approach (8kHz and MONO)]
- From TUT to ATHUS [5 class approach (44.1kHz and MONO)]
- From TUT to ATHUS [5 class approach (8kHZ, MONO and 1 sec segmentation)].

As the first approached implies, the model trained for 48 epochs and produced the results below:

- F1-Score = 86% (testing amidst of 20% of the samples during training)
- F1-Score = 86% (testing on the unseen development dataset).

With the following confusion matrix:

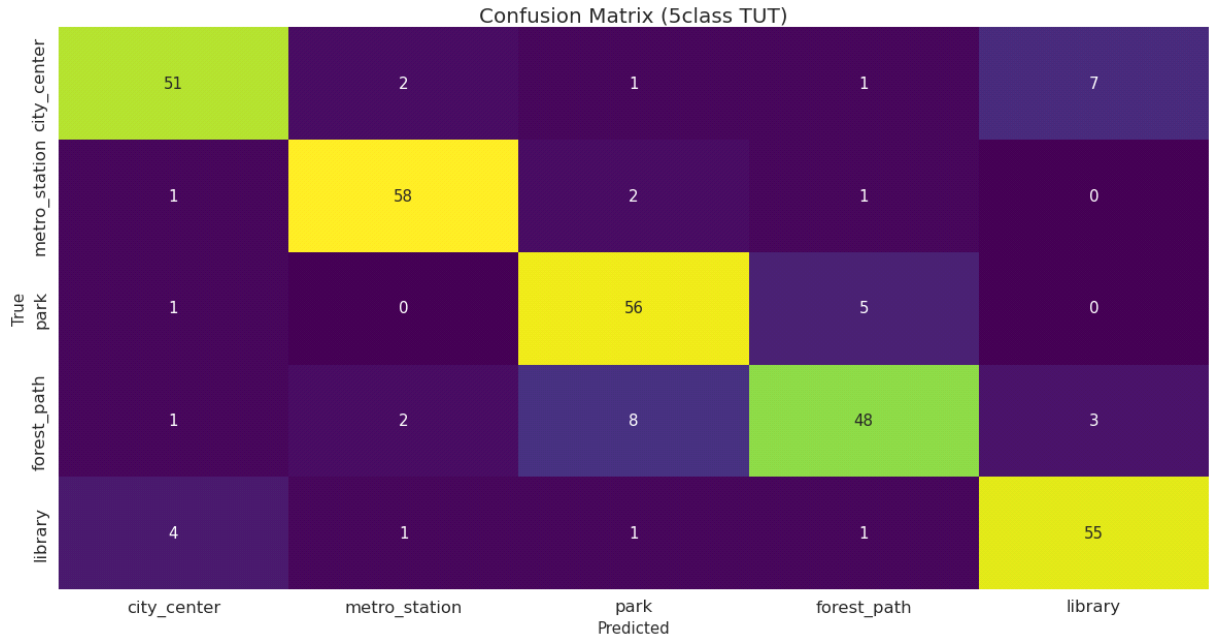


Figure 4.13: Confusion Matrix of TUT 2017 basic training using DAFP (5 class approach, 8kHz and MONO)

The second approach reveals that the model trained for 44 epochs and produced the results below:

- F1-Score = 84% (testing amidst of 20% of the samples during training)
- F1-Score = 85% (testing on the unseen development dataset).

With the following confusion matrix:

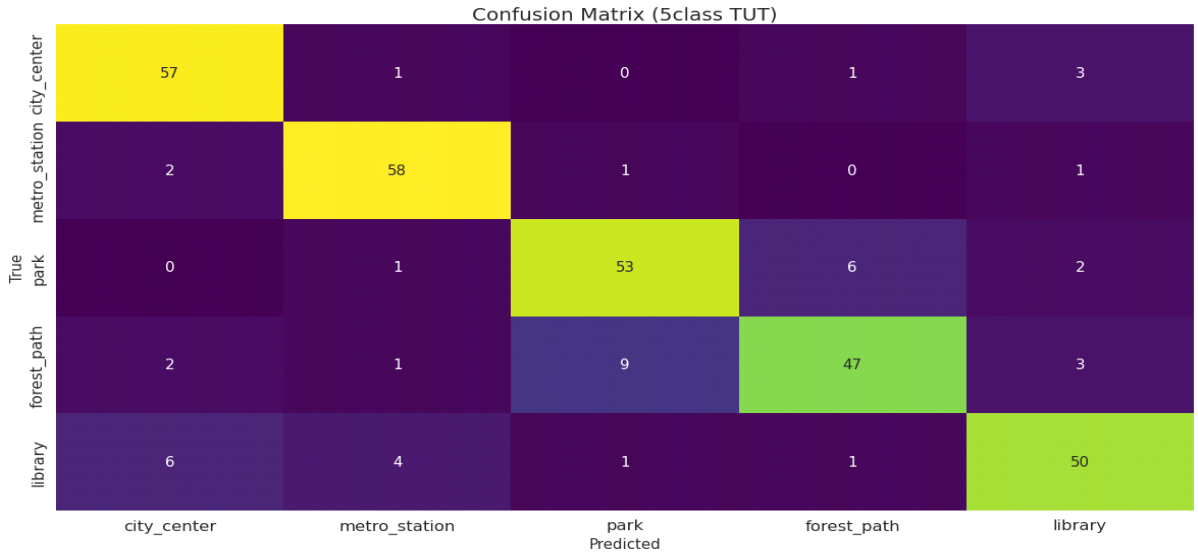


Figure 4.14: Confusion Matrix of TUT 2017 basic training using DAFP (5 class approach, 44.1kHz and MONO)

According to the last approach, the model trained for 40 epochs and produced the results below:

- F1-Score = 96% (testing amidst of 20% of the samples during training)
- F1-Score = 94% (testing on the unseen development dataset).

With the following confusion matrix:

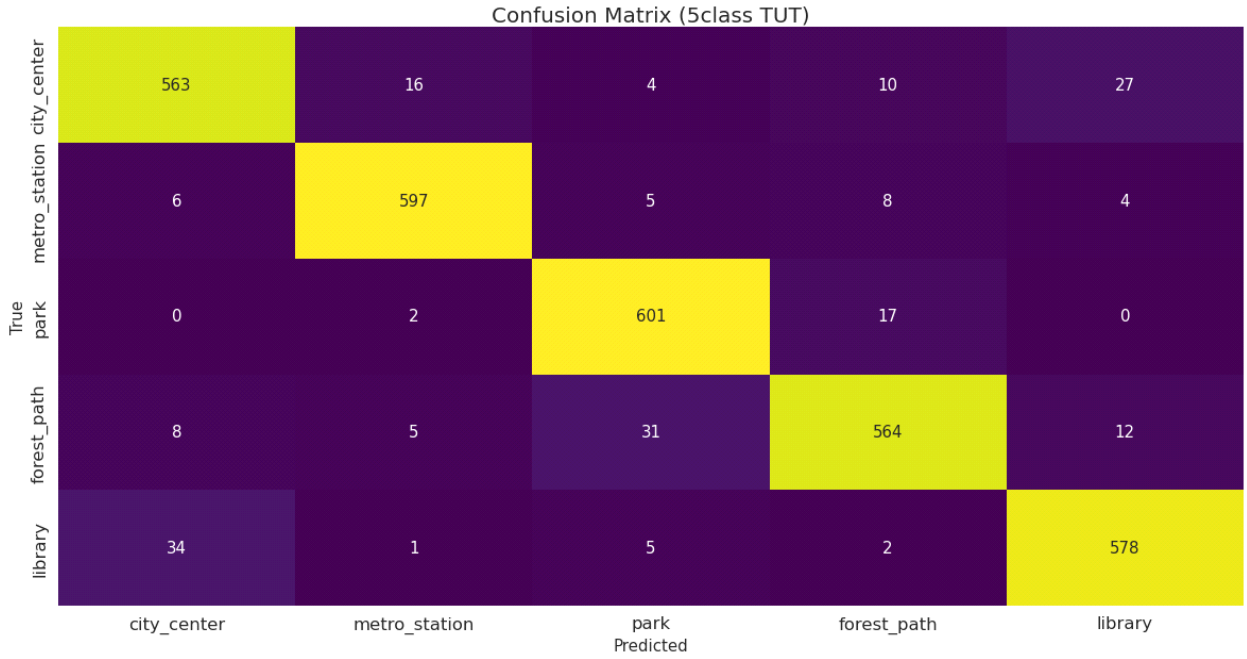


Figure 4.15: Confusion Matrix of TUT 2017 basic training using DAFP (5 class approach, 44.1kHz, MONO and 1 sec segmentation)

4.6. Transfer Learning using DAFP (From TUT to ATHUS – 5class approach)

4.6.1. Transfer Learning with Strategy 0

In this experiment the 5 class approach of the TUT dataset has been utilized in order to provide TL techniques to ATHUS dataset. The Strategy used was 0 which means the model does not perform any freezing to CNN layers [all layers (both CNN and linear) are used to train and finetune the model].

In the first approach the TL model trained for 36 epochs and produced the results below:

- F1-Score = 36% (testing amidst of 20% of the samples during training)
- F1-Score = 33% (testing on the unseen development dataset)

With the following confusion matrix:

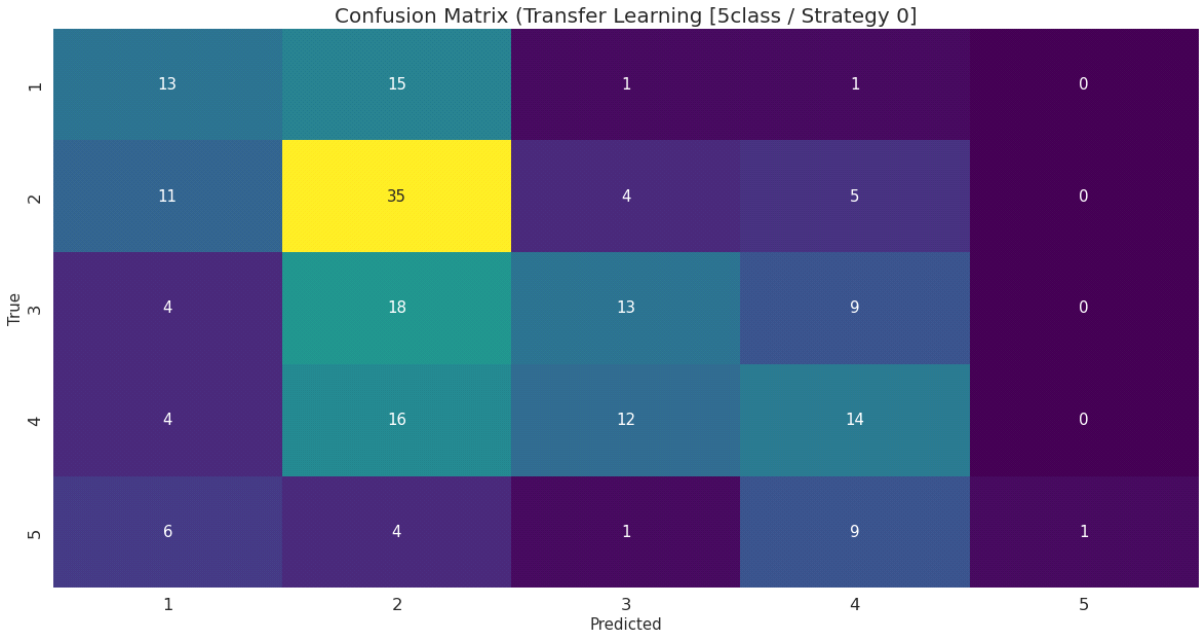


Figure 4.16: Confusion Matrix of TL using DAFP and Strategy 0 [From 5 class TUT to ATHUS (8kHz and MONO)]

In the second experiment, the TL model trained for 21 epochs and produced the results below:

- F1-Score = 41% (testing amidst of 20% of the samples during training)
- F1-Score = 49% (testing on the unseen development dataset)

With the following confusion matrix:

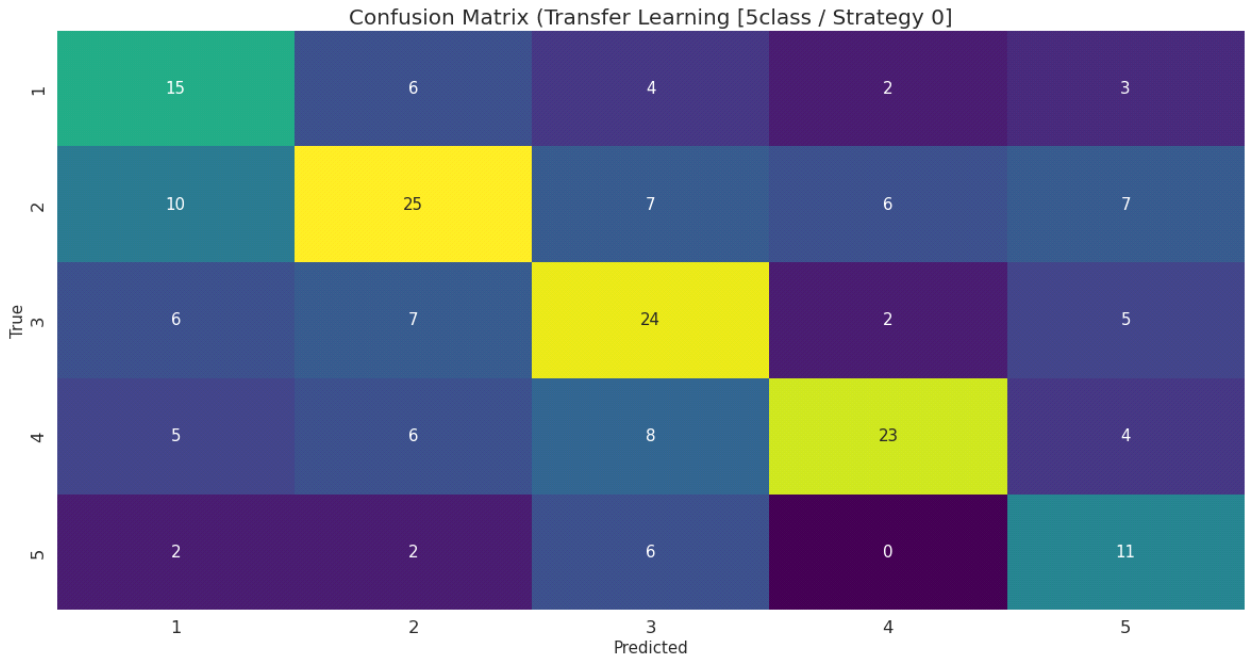


Figure 4.17: Confusion Matrix of TL using DAFP and Strategy 0 [From 5 class TUT to ATHUS (44.1kHz and MONO)]

In the last experiment, the TL model trained for 34 epochs and produced the results below:

- F1-Score = 62% (testing amidst of 20% of the samples during training)
- F1-Score = 42% (testing on the unseen development dataset)

With the following confusion matrix:

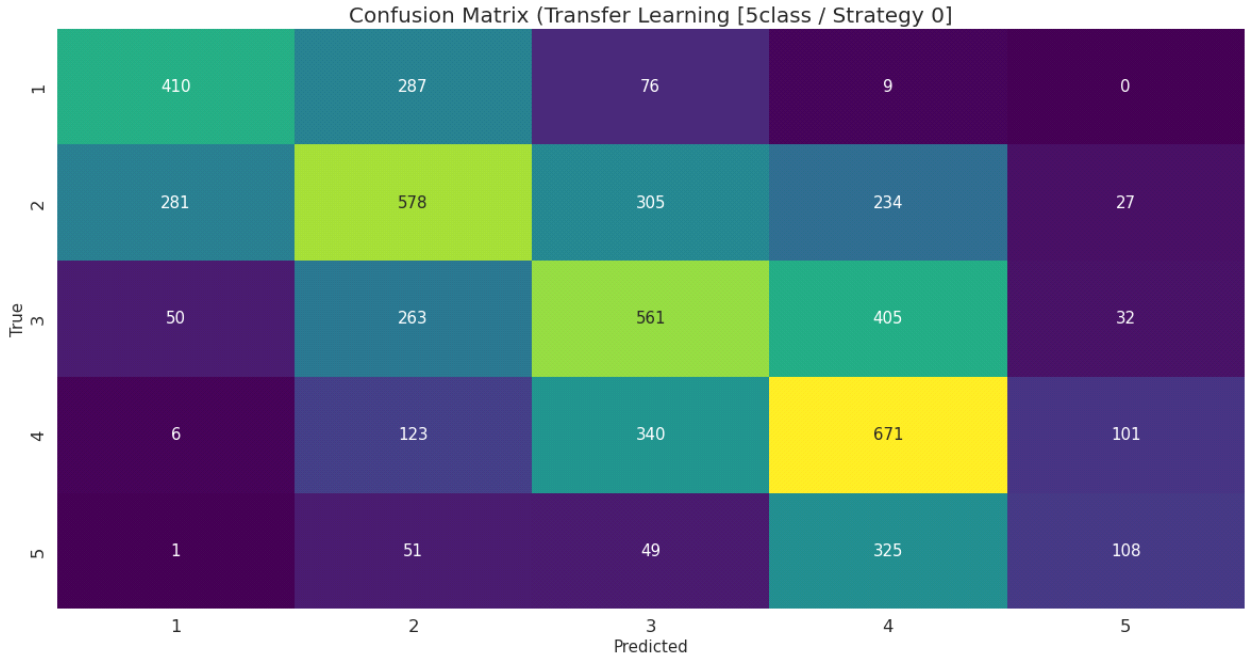


Figure 4.18: Confusion Matrix of TL using DAFP and Strategy 0 [From 5 class TUT to ATHUS (44.1kHz, MONO and 1 sec segmentation)]

4.6.2. Transfer Learning with Strategy 1

In this experiment the TUT dataset with the 5 class approach has been utilized in order to provide TL techniques to ATHUS dataset. The Strategy used was 1 which means the model performs freezing to CNN layers [only linear layers are used to train and finetune the model].

In the first approach the TL model trained for 33 epochs and produced the results below:

- F1-Score = 37% (testing amidst of 20% of the samples during training)
- F1-Score = 36% (testing on the unseen development dataset)

With the following confusion matrix:

Figure 4.19: Confusion Matrix of TL using DAFP and Strategy 1 [From 5 class TUT to ATHUS (8kHz and MONO)]

In the second experiment the TUT dataset with the 5 class approach has been utilized in order to provide TL techniques to ATHUS dataset (44.1kHz and MONO).

The TL model trained for 24 epochs and produced the results below:

- F1-Score = 39% (testing amidst of 20% of the samples during training)
- F1-Score = 47% (testing on the unseen development dataset)

With the following confusion matrix:

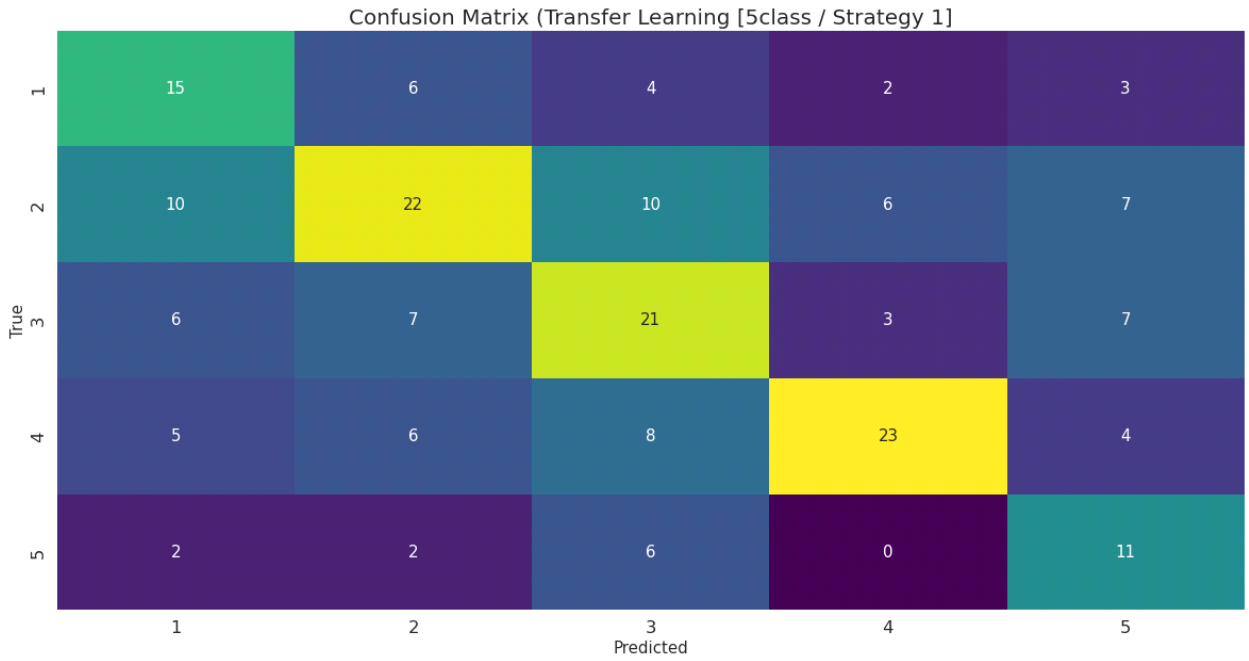


Figure 4.20: Confusion Matrix of TL using DAFP and Strategy 1 [From 5 class TUT to ATHUS (44.1kHz and MONO)]

In the third experiment the TUT dataset with the 5 class approach has been utilized in order to provide TL techniques to ATHUS dataset (8kHz, 1 sec segmentation and MONO). The Strategy used was 1 which means the model performs freezing to CNN layers [only linear layers are used to train and finetune the model].

The TL model trained for 53 epochs and produced the results below:

- F1-Score = 57% (testing amidst of 20% of the samples during training)
- F1-Score = 39% (testing on the unseen development dataset)

With the following confusion matrix:

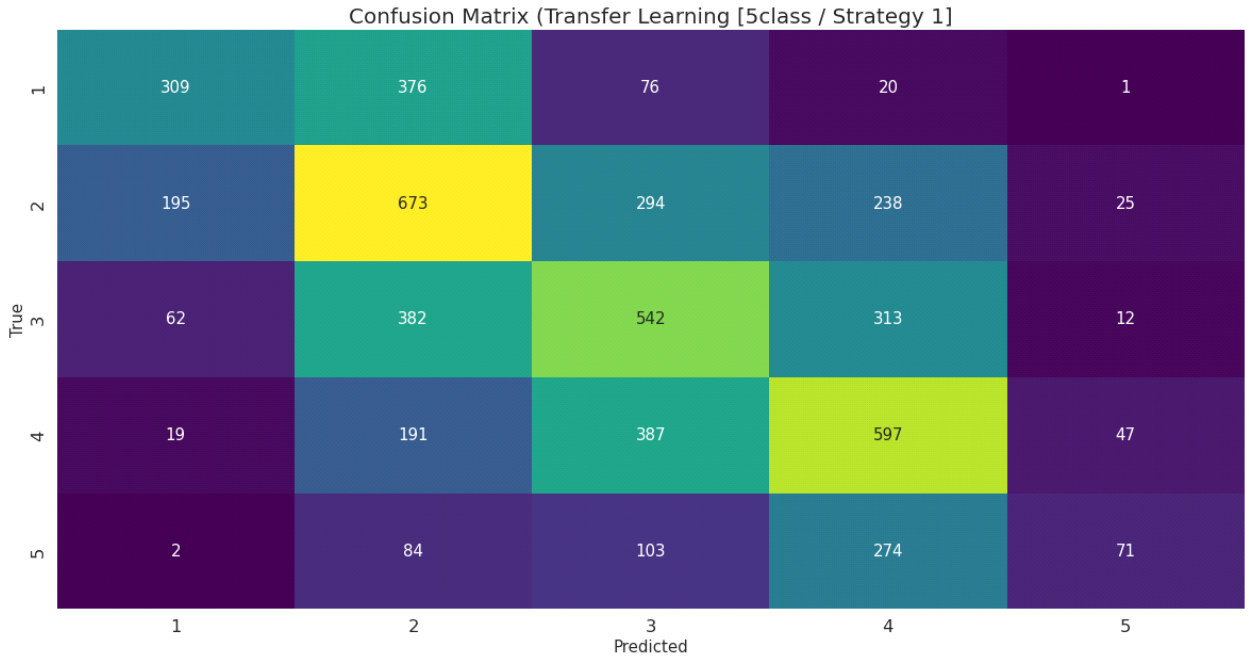


Figure 4.21: Confusion Matrix of TL using DAFP and Strategy 1 [From 5 class TUT to ATHUS (8kHz, MONO and 1 sec segmentation)]

4.7. TUT 2017 Basic Training using DAFP: A 3 class approach

In this section another approach is employed; In order to apply more optimized TL techniques from the TUT 2017 to the ATHUS, the TUT 2017 dataset retrained with only 3 classes. In this experiment all the TUT classes merged into 3.

To this end, the new classes that have been constructed are the following:

- **Bad**
  This category contains all the initial classes of the TUT dataset that are deemed as low quality soundscape environments. These are the following:
  - City Centre,
  - Metro Station,
  - Train,
  - Tram and
  - Bus
- **Mid**
  This category contains all the initial classes of the TUT dataset that are deemed as mid range soundscape environments in terms of quality. These are the following:
  - Café / Restaurant,
  - Grocery Store,
  - Beach,
  - Residential Area,
  - Car
- **Good**
  This category contains all the other initial classes that are not among the above. This category contains all the initial classes that are deemed as high quality soundscape environments and contains the following:
  - Forest Path,
  - Home,
  - Library,
  - Office,
  - Park

It is stated that the above categorization has been made after many experiments and many different combinations in order to come up with this specific solution mentioned above.

Three (3) different approaches are proposed, as follows:

- TUT [3 class approach (8kHz and MONO)]
- TUT [3 class approach (44.1kHz and MONO)]
- TUT [3 class approach (8kHZ, MONO and 1 sec segmentation)].

As the first approached implies, the model trained for 59 epochs and produced the results below:

- F1-Score = 79% (testing amidst of 20% of the samples during training)
- F1-Score = 77% (testing on the unseen development dataset)

With the following confusion matrix:

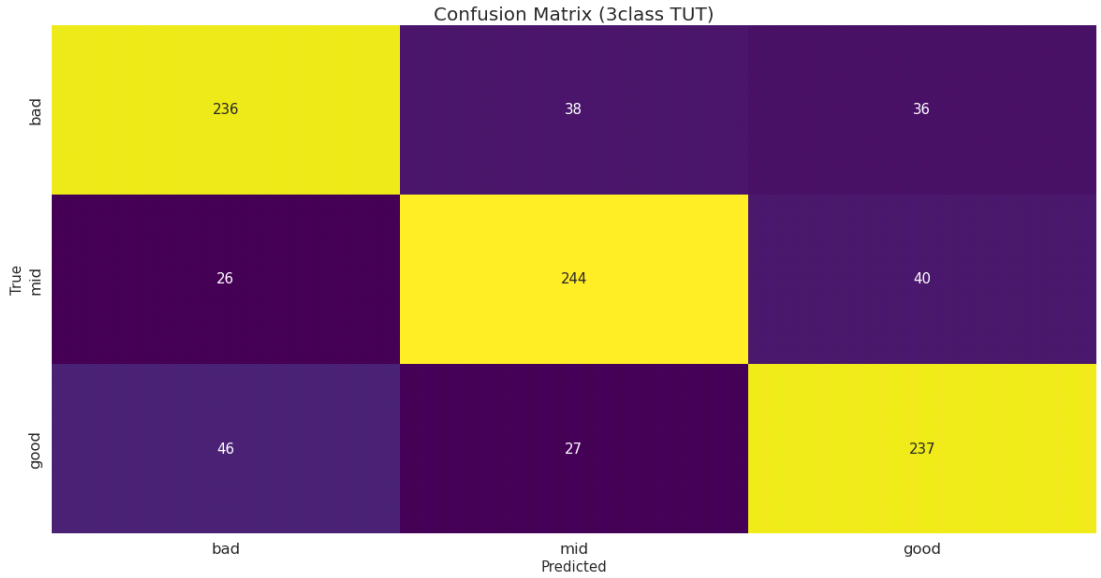


Figure 4.22: Confusion Matrix of TUT 2017 basic training using DAFP (3 class approach, 8kHz and MONO)

In the second approach, the model trained for 59 epochs and produced the results below:

- F1-Score = 89% (testing amidst of 20% of the samples during training)
- F1-Score = 87% (testing on the unseen development dataset)

With the following confusion matrix:

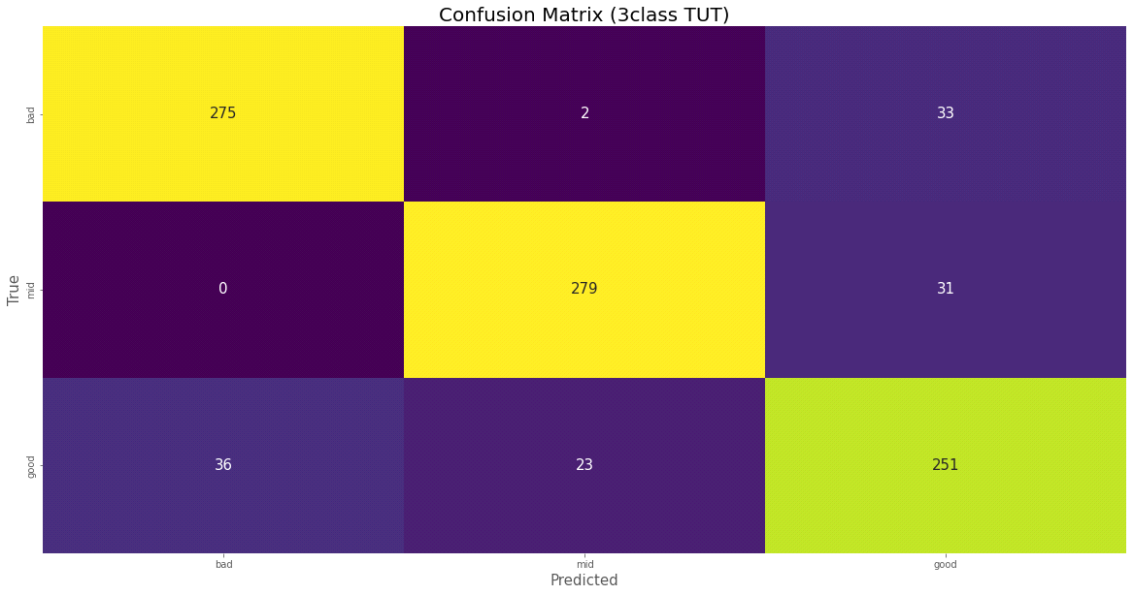


Figure 4.23: Confusion Matrix of TUT 2017 basic training using DAFP (3 class approach, 44.1kHz and MONO)

In the last, third approach, the model trained for 70 epochs and produced the results below:

- F1-Score = 93% (testing amidst of 20% of the samples during training)
- F1-Score = 92% (testing on the unseen development dataset)

With the following confusion matrix:

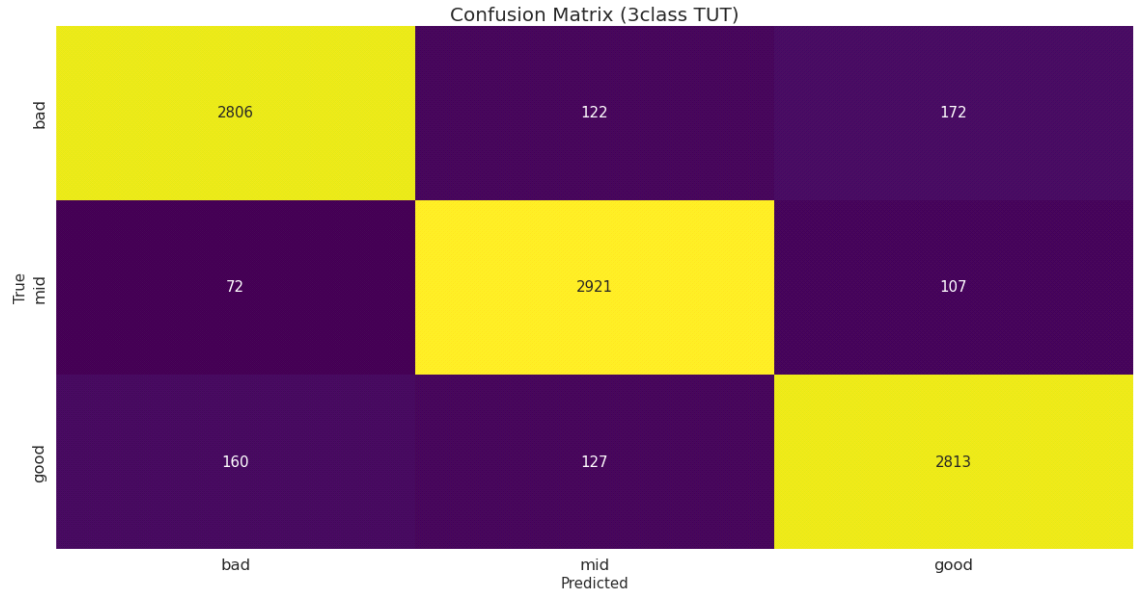


Figure 4.24: Confusion Matrix of TUT 2017 basic training using DAFP (3 class approach, 8kHz, MONO and 1 sec segmentation)

4.8. Transfer Learning using DAFP (From TUT to ATHUS – 3 class approach)

4.8.1. Transfer Learning with Strategy 0

In this experiment the TUT dataset (3 class approach) has been utilized in order to provide TL techniques to ATHUS dataset. The Strategy used was 0 which means the model does not perform any freezing to CNN layers [all layers (both CNN and linear) are used to train and finetune the model].

As expected, three (3) different approaches are proposed, as follows:

- From TUT to ATHUS [3 class approach (8kHz and MONO)]
- From TUT to ATHUS [3 class approach (44.1kHz and MONO)]
- From TUT to ATHUS [3 class approach (8kHZ, MONO and 1 sec segmentation)].

In the first approach, the TL model trained for 48 epochs and produced the results below:

- F1-Score = 33% (testing amidst of 20% of the samples during training)
- F1-Score = 28% (testing on the unseen development dataset)

With the following confusion matrix:

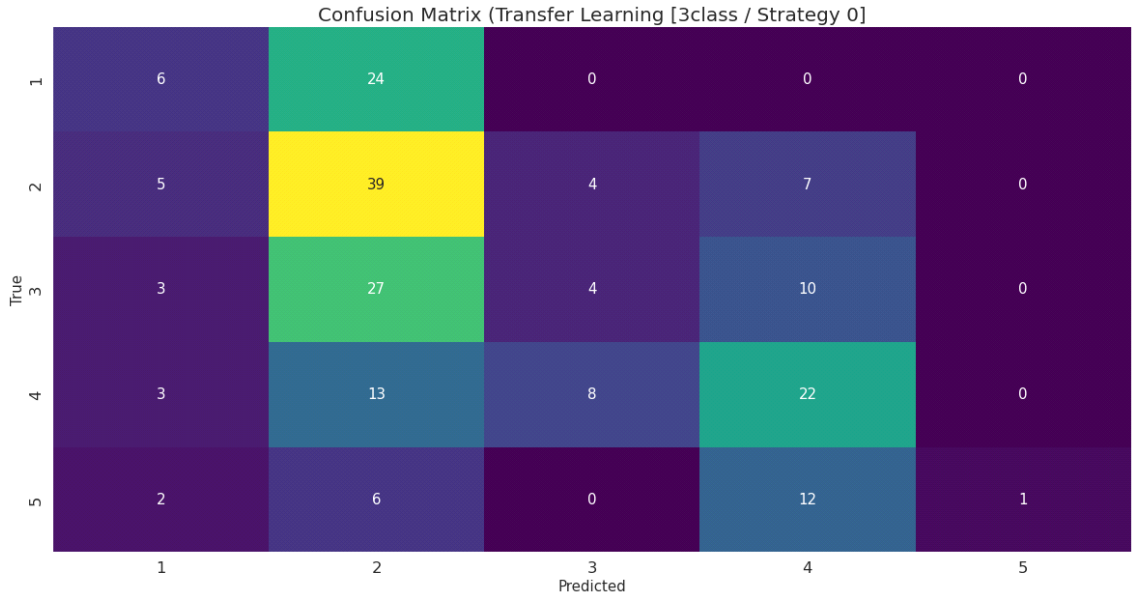


Figure 4.25: Confusion Matrix of TL using DAFP and Strategy 0 (3 class approach, 8kHz, MONO)

In the second approach, the TL model trained for 20 epochs and produced the results below:

- F1-Score = 36% (testing amidst of 20% of the samples during training)
- F1-Score = 48% (testing on the unseen development dataset)

With the following confusion matrix:

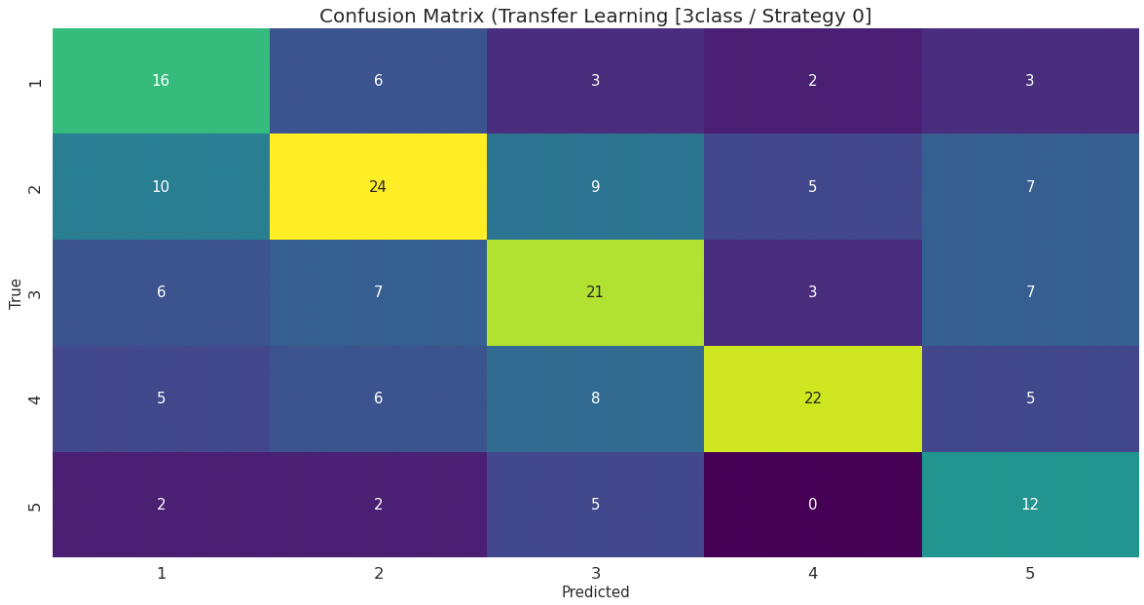


Figure 4.26: Confusion Matrix of TL using DAFP and Strategy 0 (3 class approach, 44.1kHz, MONO)

In the last, third approach, the TL model trained for 20 epochs and produced the results below:

- F1-Score = 62% (testing amidst of 20% of the samples during training)
- F1-Score = 44% (testing on the unseen development dataset)

With the following confusion matrix:

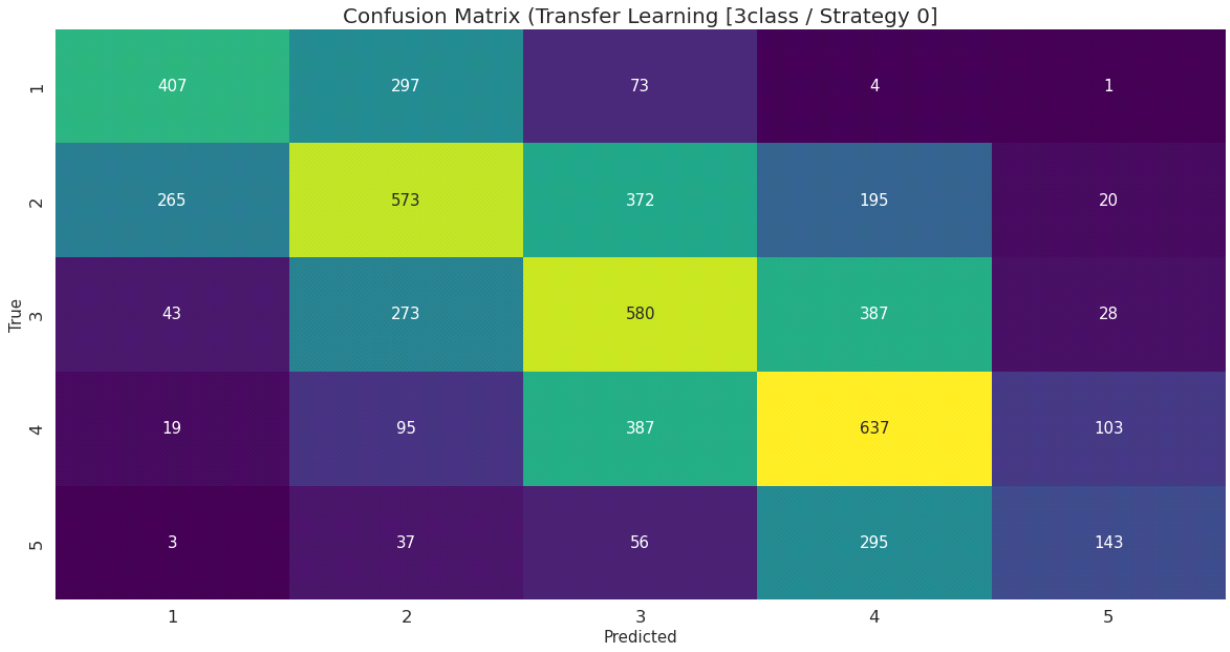


Figure 4.27: Confusion Matrix of TL using DAFP and Strategy 0 (3 class approach, 8kHz, MONO and 1 sec segmentation)

4.8.2. Transfer Learning with Strategy 1

In this experiment the TUT dataset with the 3 class approach has been utilized in order to provide TL techniques to ATHUS dataset. The Strategy used was 1 which means the model performs freezing to CNN layers [only linear layers are used to train and finetune the model].

Three (3) different approaches are proposed, as follows:

- From TUT to ATHUS [3 class approach (8kHz and MONO)]
- From TUT to ATHUS [3 class approach (44.1kHz and MONO)]
- From TUT to ATHUS [3 class approach (8kHZ, MONO and 1 sec segmentation)].

During the first approach, the TL model trained for 48 epochs and produced the results below:

- F1-Score = 40% (testing amidst of 20% of the samples during training)
- F1-Score = 32% (testing on the unseen development dataset)

With the following confusion matrix:

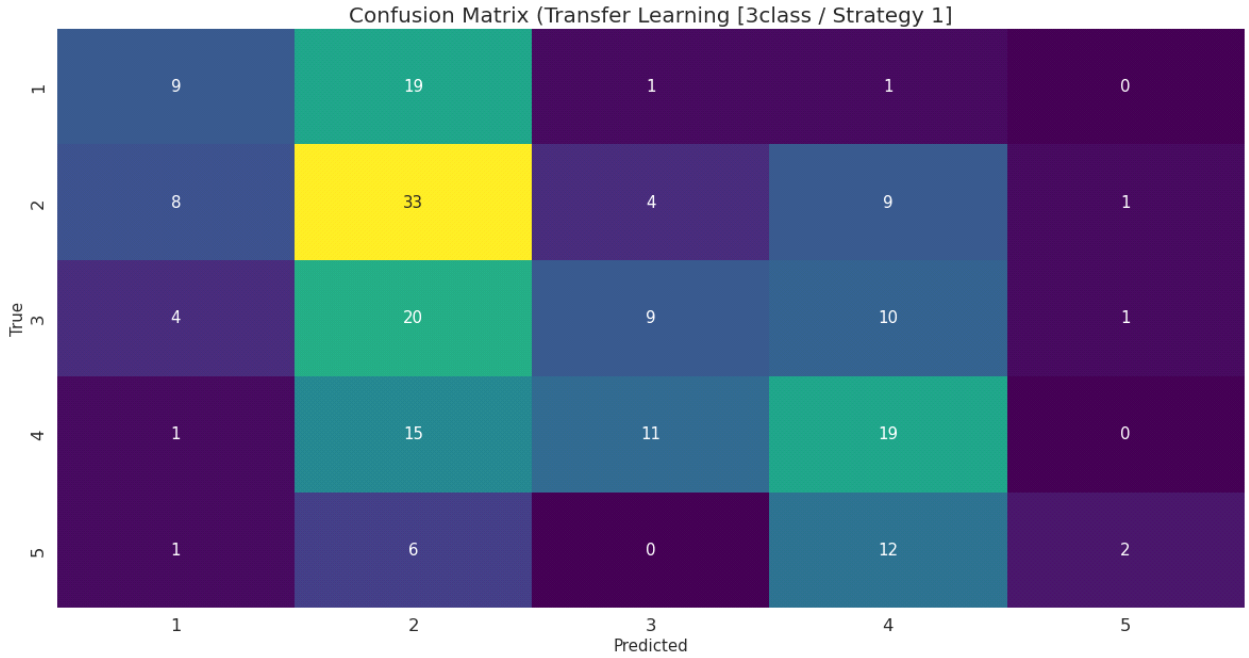


Figure 4.28: Confusion Matrix of TL using DAFP and Strategy 1 (3 class approach, 8kHz, MONO)

During the second approach, the TL model trained for 20 epochs and produced the results below:

- F1-Score = 50% (testing amidst of 20% of the samples during training)
- F1-Score = 49% (testing on the unseen development dataset)

With the following confusion matrix:

Figure 4.29: Confusion Matrix of TL using DAFP and Strategy 1 (3 class approach, 44.1kHz, MONO)

In the third approach, the TL model trained for 43 epochs and produced the results below:

- F1-Score = 55% (testing amidst of 20% of the samples during training)
- F1-Score = 41% (testing on the unseen development dataset)

With the following confusion matrix:

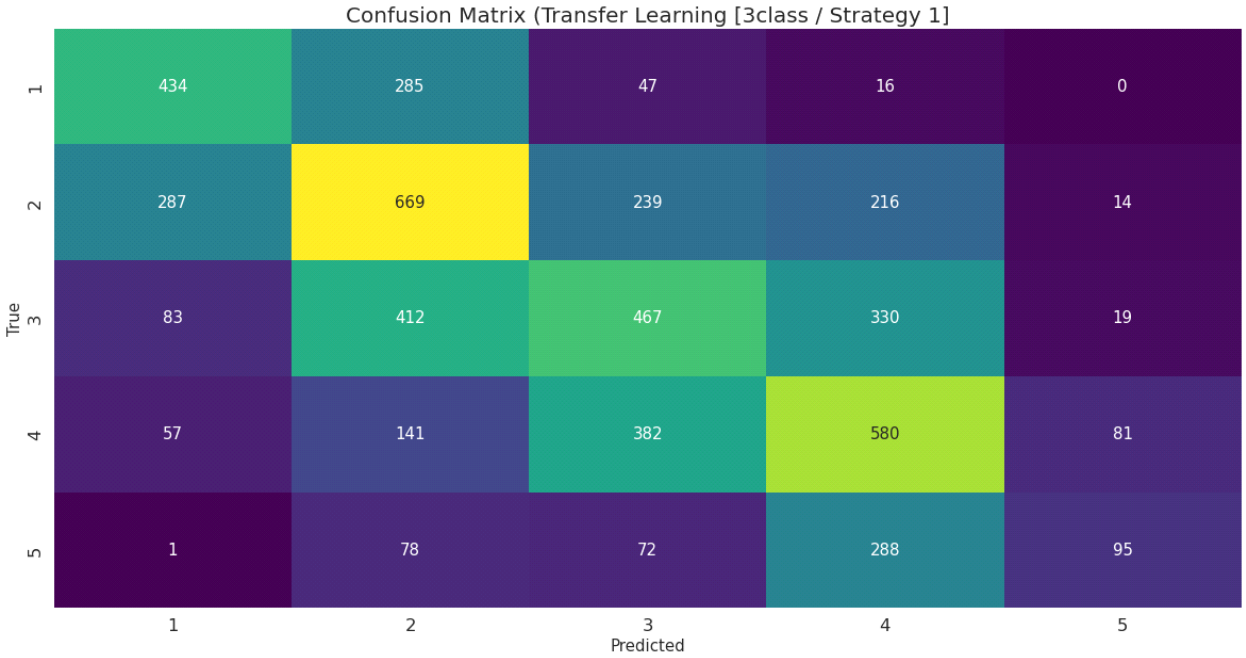


Figure 4.30: Confusion Matrix of TL using DAFP and Strategy 1 (3 class approach, 8kHz, MONO and 1 sec segmentation)

4.9. Transfer Learning with Medium Freeze and Class Weighting Balance

In this Section, the top models[20] acquired trained again with a different freeze method. In all the above cases, freezing either all or only linear layers provided really worth mentioning results.

In this attempt the top 3 models trained again by freezing 3 out of 4 CNN layers. The models trained for this purpose are:

1. TL from TUT source model to ATHUS (original sampling at 44.1kHz) in the 5 class approach
2. TL from TUT source model to ATHUS (resampled at 8kHz) in the 3 class approach
3. TL from TUT source model to ATHUS (resampled at 8kHz with 1 sec segmentation) in the 3 class approach.

Thus, the architecture of the model remained for the training:

- 1 CNN
- 3 Linear Layers

The first model trained for 33 epochs and produced the results below:

- F1-Score = 0.40% (testing amidst of 20% of the samples during training)
- F1-Score = 0.33% (testing on the unseen development dataset)

With the following confusion matrix:

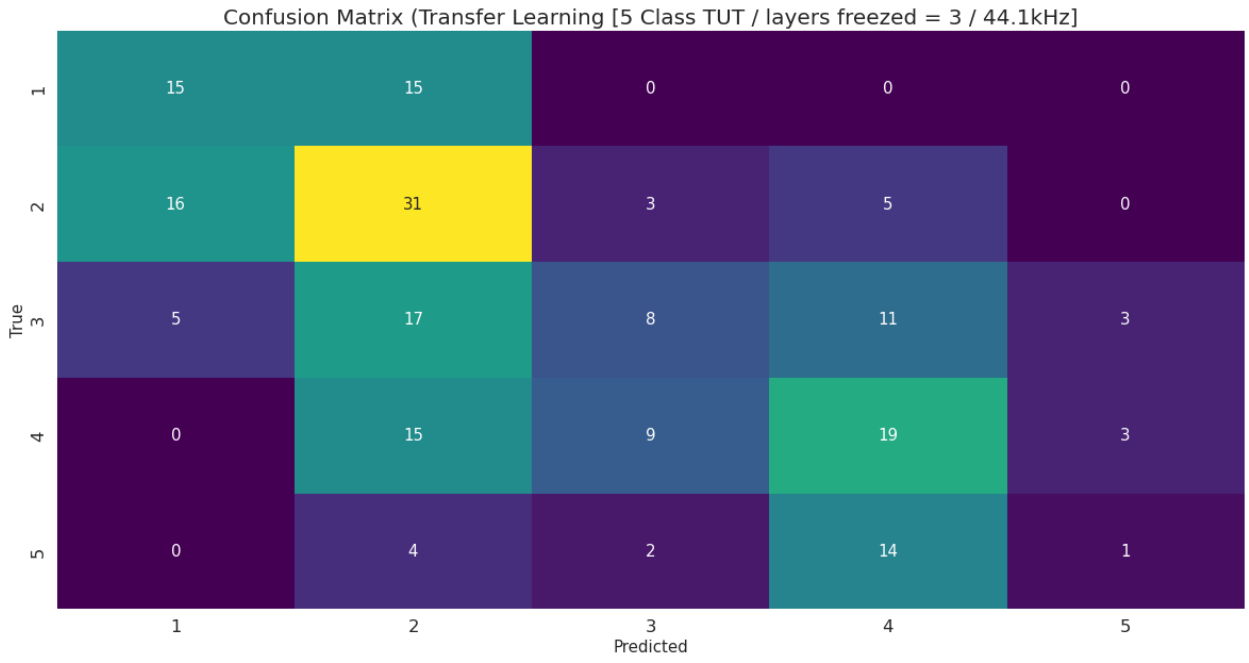


Figure 4.31: Confusion Matrix of TL using DAFP and 3 layers frozen (5 class approach, 44.1kHz, MONO)

[20] In terms of f1 score on unseen / test data from each of the categories (original, 8kHz and 1 sec segmentation)

The second model trained for 21 epochs and produced the results below:

- F1-Score = 0.45% (testing amidst of 20% of the samples during training)
- F1-Score = 0.38% (testing on the unseen development dataset)

With the following confusion matrix:

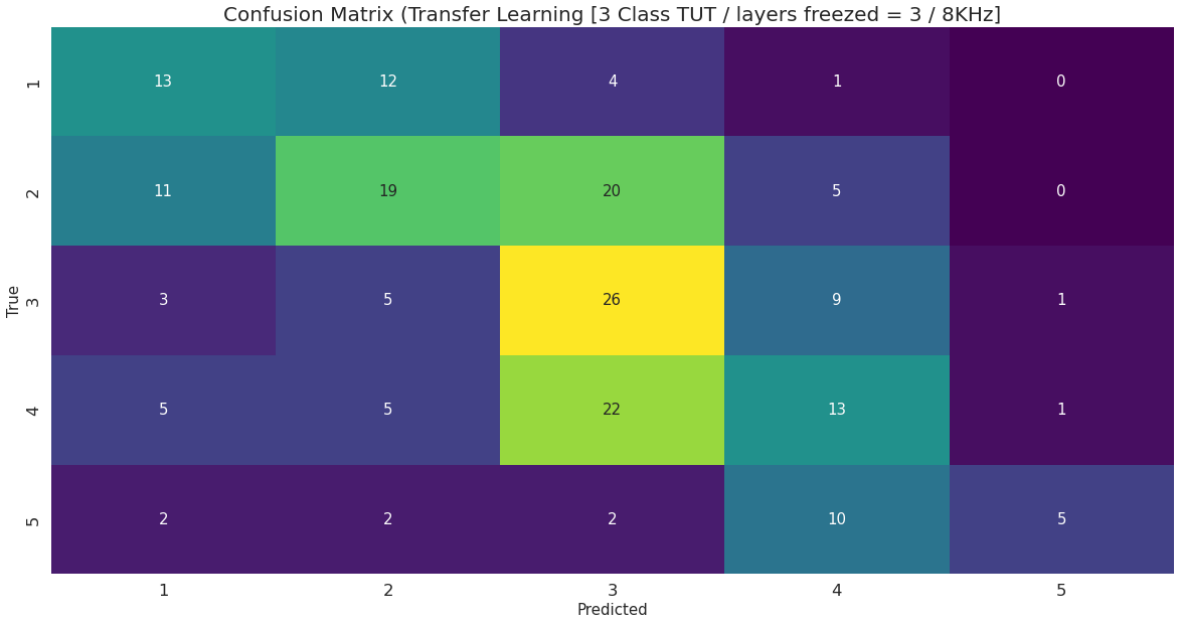


Figure 4.32: Confusion Matrix of TL using DAFP and 3 layers frozen (3 class approach, 8kHz, MONO)

The last third model trained for 51 epochs and produced the results below:

- F1-Score = 0.51% (testing amidst of 20% of the samples during training)
- F1-Score = 0.42% (testing on the unseen development dataset)

With the following confusion matrix:

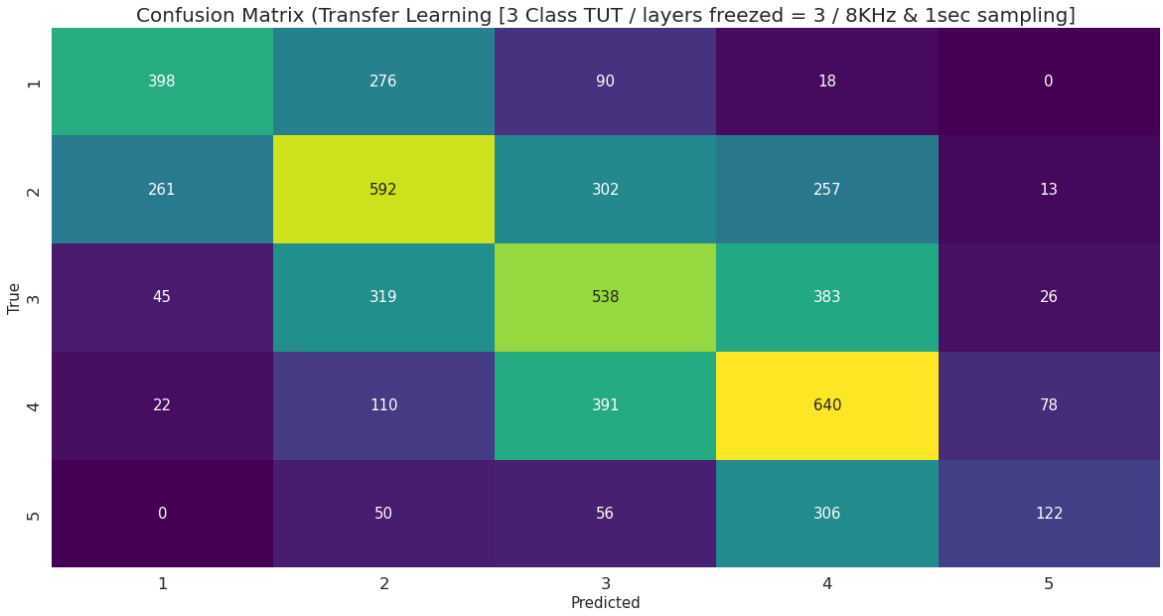


Figure 4.33: Confusion Matrix of TL using DAFP and 3 layers frozen (3 class approach, 8kHz, MONO with 1 sec segmentation)

The top model as shown in Table 5.1 is considered to be the TL training of the ATHUS dataset, using the TUT as source model into a 3 class approach when all CNN are frozen (only linear layers in the training phase).

For this overall top model class weighting balance has been performed in order to acquire even better results. So the top model trained again and provided with the following:

The top model trained for 29 epochs and produced the results below:

- F1-Score = 0.48% (testing amidst of 20% of the samples during training)
- F1-Score = 0.42% (testing on the unseen development dataset)

With the following confusion matrix:

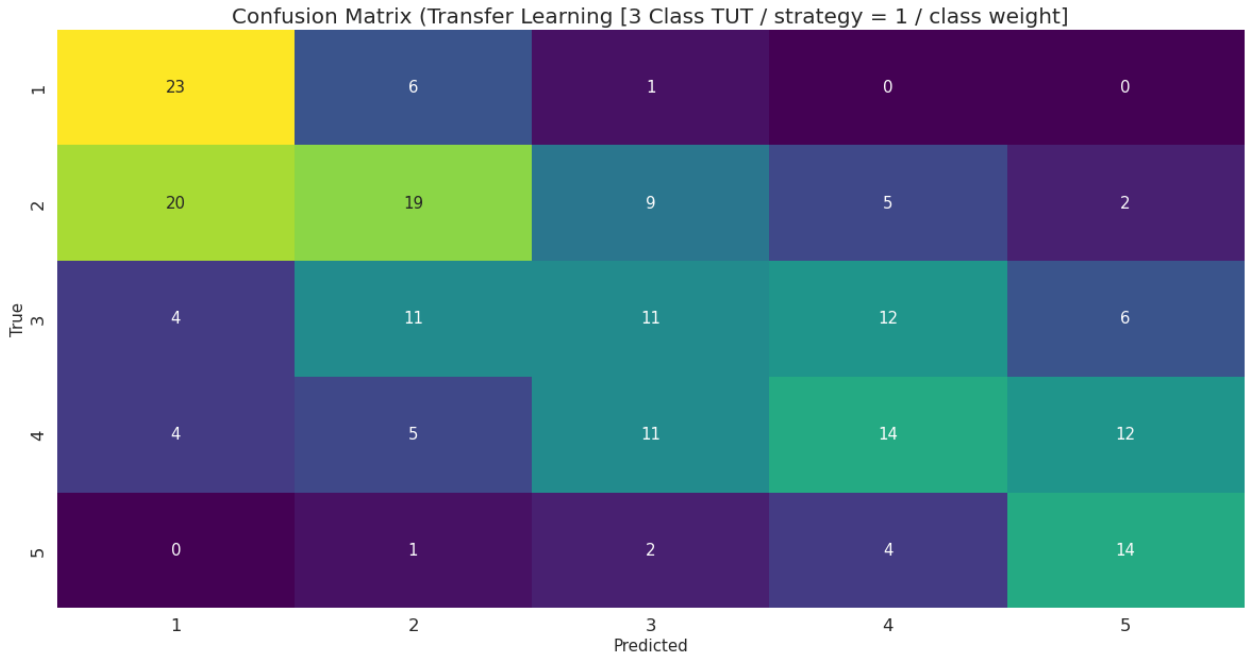


Figure 4.34: Confusion Matrix of TL using DAFP and class weighting (3 class approach, 44.1kHz, MONO with strategy 1)

Table 4.2: Metrics for all models

	MODEL	F1 SCORE	
		TRAIN	TEST
8kHz and MONO	BASIC TRAINING OF SOUNDSCAPE	0.33	0.31
	TL (TUT TO SOUNDSCAPE, STRAT 0)	0.37	0.36
	TL (TUT TO SOUNDSCAPE, STRAT 1)	0.37	0.31
	TL (5CLASS TUT TO SOUNDSCAPE, STRAT 0)	0.36	0.33
	TL (5CLASS TUT TO SOUNDSCAPE, STRAT 1)	0.38	0.36

	TL (3CLASS TUT TO SOUNDSCAPE, STRAT 0)	0.33	0.28
	TL (3CLASS TUT TO SOUNDSCAPE, STRAT 1)	0.40	0.32
	TL (3CLASS TUT TO SOUNDSCAPE, layers freeze 3)	0.45	0.38
	SOURCE MODEL FOR TL	**F1 SCORE**	
		TRAIN	**TEST**
	BASIC TRAINING OF TUT	0.79	0.77
	BASIC TRAINING OF 5 CLASS TUT	0.86	0.86
	BASIC TRAINING OF 3 CLASS TUT	0.79	0.77
44.1kHz and MONO	**MODEL**	**F1 SCORE**	
		TRAIN	**TEST**
	BASIC TRAINING OF SOUNDSCAPE	0.35	0.30
	TL (TUT TO SOUNDSCAPE, STRAT 0)	0.38	0.38
	TL (TUT TO SOUNDSCAPE, STRAT 1)	0.37	0.31
	TL (5CLASS TUT TO SOUNDSCAPE, STRAT 0)	0.41	0.49
	TL (5CLASS TUT TO SOUNDSCAPE, STRAT 1)	0.39	0.47
	TL (3CLASS TUT TO SOUNDSCAPE, STRAT 0)	0.36	0.48
	TL (3CLASS TUT TO SOUNDSCAPE, STRAT 1)	0.50	0.49
	TL (3CLASS TUT TO SOUNDSCAPE, STRAT 1 with class weighting balance)	0.48	0.42
	TL (5CLASS TUT TO SOUNDSCAPE, layers freeze 3)	0.40	0.33
	SOURCE MODEL FOR TL	**F1 SCORE**	
		TRAIN	**TEST**
	BASIC TRAINING OF TUT	0.81	0.76

	BASIC TRAINING OF 5 CLASS TUT	0.84	0.85
	BASIC TRAINING OF 3 CLASS TUT	0.89	0.87
8kHz, MONO and 1sec segmentation	**MODEL**	**F1 SCORE**	
		TRAIN	**TEST**
	BASIC TRAINING OF SOUNDSCAPE	0.55	0.41
	TL (TUT TO SOUNDSCAPE, STRAT 0)	0.63	0.42
	TL (TUT TO SOUNDSCAPE, STRAT 1)	0.59	0.40
	TL (5CLASS TUT TO SOUNDSCAPE, STRAT 0)	0.62	0.42
	TL (5CLASS TUT TO SOUNDSCAPE, STRAT 1)	0.57	0.39
	TL (3CLASS TUT TO SOUNDSCAPE, STRAT 0)	0.62	0.44
	TL (3CLASS TUT TO SOUNDSCAPE, STRAT 1)	0.55	0.41
	TL (3CLASS TUT TO SOUNDSCAPE, layers freeze 3)	0.51	0.42
	SOURCE MODEL FOR TL	**F1 SCORE**	
		TRAIN	**TEST**
	BASIC TRAINING OF TUT	0.94	0.91
	BASIC TRAINING OF 5 CLASS TUT	0.96	0.94
	BASIC TRAINING OF 3 CLASS TUT	0.93	0.92

The above table 4.2 concludes all the metrics acquired from all the experiments.

Discussion and final conclusion will be made in the following Chapter.

5. CONCLUSION – FUTURE WORK

5.1. Conclusion

In the present Thesis a meticulous overview of an end-to-end approach of Auditory Scene Analysis has been made.

As mentioned from the beginning, earth is full of different acoustic environments that includes from quite and calm to very busy and noisy soundscapes.

The present Master Thesis was inspired by the problem of soundscape quality and sound recognition and it is addressed in the context of which sound / audio gathered from different environments and corresponds to either a bad or good quality, in terms of sound pollution.

It is very crucial and because of a very big intervention of human in the planet, to be able to measure and control this noise pollution, especially in very hectic and busy downtown areas. Thus, it would be very beneficial for the researcher to be able to analyze sound clips from different environments and take measures accordingly for enhancing and improving humans' life. Undoubtedly, many technologies can help this effort. By using sensors and IoT infrastructures this attempt can be done more easily. Monitoring city areas with the use of audio sensors can prevent overwhelming noise among cities especially during siesta hours. Also, this can detect almost simultaneously events that can possibly harm human consistency. Such activities could possibly refer to sudden explosions, fires and all events that are related to sound event detection.

To this end and because of the significance of sound analysis especially amidst cities, environmental audio scene and sound event recognition is proposed and analyzed in detail in the present Thesis.

Modern Deep Learning and Transfer Learning techniques were applied in order to analyze and study the quality of different environments. Without doubt this could not have been achieved if there were no appropriate and real life data. A real world dataset that collected and annotated by human annotators has been used in order to imitate and train the best possible DL models for this approach.

In order to solve the audio classification problem, one top tier and method has been presented and analyzed in high detail among the present Thesis. This method takes the audio clip as input, transforms the audio part into MEL Spectrogram and then applies CNN and DL techniques in order to classify the sound clips to different classes according to their quality.

The method divided in three (3) main sub-approaches.

- In the first sub-approach, the dataset was transformed many times in order to be trained and fitted in many different models. The first approach was to transform the audio clips into 8KHz and MONO. The same engineering also performed into the TUT dataset (dataset used for the TL part). In this approach the ATHUS dataset trained into three (3) different permutations:

  - ✓ Original training phase with the whole dataset was included in the training phase,

  - ✓ A 5 class case where only 5 of the classes of the TUT dataset used to for the TL process and

- ✓ A 3 class case where all the classes of the TUT dataset used have been merged into 3 (bad, mid, good) to for the TL process.

- In the second sub-approach, the target was to train the dataset as it is and without any other user intervention (44kHz and MONO). The same engineering also performed into the TUT dataset, used for the TL evaluation. In this approach again the ATHUS dataset trained into three (3) different permutations as follows:

  - ✓ Original training phase with the whole dataset was included in the training phase,

  - ✓ A 5 class case where only 5 of the classes of the TUT dataset used to for the TL process and

  - ✓ A 3 class case where all the classes of the TUT dataset used have been merged into 3 (bad, mid, good) to for the TL process.

- Finally, during the third approach the main goal was to train the dataset by transforming it into 8kHz and MONO. In addition, 1 sec segmentation was applied to all datasets (1 sec segmentation performed to every single audio clip) used (in both ATHUS and TUT 2017). In the last approach again the ATHUS dataset trained into three (3) different permutations as follows:

  - ✓ Original training phase with the whole dataset was included in the training pahse,

  - ✓ A 5 class case where only 5 of the classes of the TUT dataset used to for the TL process and

  - ✓ A 3 class case where all the classes of the TUT dataset used have been merged into 3 (bad, mid, good) to for the TL process.

From the all above models the one that overcomes all the others in terms of f1-score and overall accuracy is the application of **the TL techniques on soundscape ATHUS dataset when the TUT source model was categorized into three (3) main categories.** Also in this model the training **phase frozen the CNN layers and used only the linear for the training phase**. The f1 score acquired was equal to 0.49 on unseen dataset, as summarized in the following Tables 5.2 – 5.4.

Table 5.1: Metrics for models (8kHz and MONO)

MODEL (8kHz, MONO)	**F1 SCORE**	
	TRAIN	**TEST**
BASIC TRAINING OF SOUNDSCAPE	0.33	0.31
TL (TUT TO SOUNDSCAPE, STRAT 0)	0.37	0.36
TL (TUT TO SOUNDSCAPE, STRAT 1)	0.37	0.31
TL (5CLASS TUT TO SOUNDSCAPE, STRAT 0)	0.36	0.33
TL (5CLASS TUT TO SOUNDSCAPE, STRAT 1)	0.38	0.36
TL (3CLASS TUT TO SOUNDSCAPE, STRAT 0)	0.33	0.28
TL (3CLASS TUT TO SOUNDSCAPE, layers freeze 3)	***0.45***	***0.38***
TL (3CLASS TUT TO SOUNDSCAPE, STRAT 1)	0.40	0.32
SOURCE MODEL FOR TL (8kHz, MONO)	**F1 SCORE**	
	TRAIN	**TEST**
BASIC TRAINING OF TUT	0.79	0.77
BASIC TRAINING OF 5 CLASS TUT	0.86	0.86
BASIC TRAINING OF 3 CLASS TUT	0.79	0.77

The above Table 5.1, depicts the results (basic DL and TL training phases) for the ATHUS dataset when both have been resampled to 8kHz and MONO.

Transformation of the TUT source model into a 3 class problem overcomes the baseline model that involves simple DL techniques by 22.5%

Table 5.2: Metrics for models (44.1kHz and MONO)

MODEL (44.1kHz, MONO)	**F1 SCORE**	
	TRAIN	**TEST**
BASIC TRAINING OF SOUNDSCAPE	0.35	0.30
TL (TUT TO SOUNDSCAPE, STRAT 0)	0.38	0.38
TL (TUT TO SOUNDSCAPE, STRAT 1)	0.37	0.31
TL (5CLASS TUT TO SOUNDSCAPE, STRAT 0)	0.41	0.49
TL (5CLASS TUT TO SOUNDSCAPE, STRAT 1)	0.39	0.47
TL (3CLASS TUT TO SOUNDSCAPE, STRAT 0)	0.36	0.48
TL (5CLASS TUT TO SOUNDSCAPE, layers freeze 3)	0.40	0.33
TL (3CLASS TUT TO SOUNDSCAPE, STRAT 1 with class weighting balance)	0.48	0.42
TL (3CLASS TUT TO SOUNDSCAPE, STRAT 1)	***0.50***	***0.49***
SOURCE MODEL FOR TL (44.1kHz, MONO)	**F1 SCORE**	
	TRAIN	**TEST**
BASIC TRAINING OF TUT	0.81	0.76
BASIC TRAINING OF 5 CLASS TUT	0.84	0.85
BASIC TRAINING OF 3 CLASS TUT	0.89	0.87

The above Table 5.2, depicts the results (basic DL and TL training phases) for the ATHUS dataset in their original form (44.1kHz and MONO)

Transformation of the TUT source model into a 3 class problem overcomes by far the baseline model that involves simple DL techniques by 63% on unseen data.

Table 5.4: Metrics for models (8kHz, MONO and 1sec segmentation)

MODEL (8kHz, MONO and 1sec segmentation)	**F1 SCORE**	
	TRAIN	**TEST**
BASIC TRAINING OF SOUNDSCAPE	0.55	0.41
TL (TUT TO SOUNDSCAPE, STRAT 0)	0.63	0.42
TL (TUT TO SOUNDSCAPE, STRAT 1)	0.59	0.40
TL (5CLASS TUT TO SOUNDSCAPE, STRAT 0)	0.62	0.42
TL (5CLASS TUT TO SOUNDSCAPE, STRAT 1)	0.57	0.39
TL (3CLASS TUT TO SOUNDSCAPE, STRAT 0)	***0.62***	***0.44***
TL (3CLASS TUT TO SOUNDSCAPE, layers freeze 3)	0.51	0.42
TL (3CLASS TUT TO SOUNDSCAPE, STRAT 1)	0.55	0.41
SOURCE MODEL FOR TL (8kHz, MONO and 1sec segmentation)	**F1 SCORE**	
	TRAIN	**TEST**
BASIC TRAINING OF TUT	0.94	0.91
BASIC TRAINING OF 5 CLASS TUT	0.96	0.94
BASIC TRAINING OF 3 CLASS TUT	0.93	0.92

The above Table 5.4, depicts the results (basic DL and TL training phases) for the ATHUS dataset in 8kHz, MONO and with 1 sec resampling.

Transformation of the TUT source model into a 3 class problem overcomes the baseline model that involves simple DL techniques by 7% on unseen data.

Undoubtedly Transfer Learning Techniques to all the different approaches studied and presented in the present Thesis overcame all the baseline models that refer to simple application of CNN. TL combines previous knowledge of a source model and a new one in order to fine tune the problem as much as possible.

5.2. Implementation Issues

Despite all TL models returned higher scores than the baseline models, the original dataset (no other resampling or segmentation) provided the best possible solution with the highest f1 score. Nevertheless, applying TL techniques require extra time and effort in order to wrangle and clean the source model efficiently. To this end and for such problems, it is highly recommended to use segmentations and train the date with the baseline model for accurate and trustworthy results. This will add more data to the

training pool (in circumstances that there is a lack of many data) and give a fast and worth mentioning solutions for almost all the cases.

5.3. Future Work

In the present Thesis the soundscape classification problem studied meticulously and covered almost all the possible aspects that involve different, top-tier DL and NN architectures.

This effort undoubtedly can be enhanced in the future by using and developing other types of architectures.

Wav to Vec architectures, that are considered state of the art in speech recognition can be parameterized accordingly in order to provide stable and very accurate results for the Soundscape analysis problem. This approach achieved the best published result to date on the popular WSJ benchmark while using two orders of magnitude less labeled training data than a comparable system. The algorithm works with existing ASR systems and uses raw audio as training data, without the need for written transcriptions, demonstrating that self-supervision can make even high-performing speech recognition models more effective.

Also, model agnostic meta learning and learning to learn techniques can be used in the future for such problems. Such systems are trained by being exposed to a large number of tasks and are then tested in their ability to learn new tasks; a good example of a task is classifying a new image within 5 possible classes, given one example of each class, or learning to efficiently navigate a new maze with only one traversal through the maze. This differs from many standard machine learning techniques, which involve training on a single task and testing on held-out examples from that task. So this technique can be applied to the soundscape classification problem and make the model learn with very low supervision and with less data available.

ACRONYMS

AES	Audio Engineering Society
AI	Artificial Intelligence
ANN	Artificial Neural Networks
ASA	Auditory Scene Analysis
ATHUS	ATHens Urban Soundscape
CAP	Credit Assignment Path
CM	Confusion Matrix
CNN	Convolutional Neural Networks
CS	Computer Science
DAFP	Deep Audio Features Package
DBSCAN	Density-Based Spatial Clustering Of Applications With Noise
DL	Deep Learning
DL	Deep Learning
DNN	Deep Neural Networks
DS	Data Science
FWT	Fast Wavelet Transform
MAE	Mean Absolute Error
MFCC	Mel Frequency Cepstral Coefficients
ML	Machine Learning
MSE	Mean Squared Error
NLP	Natural Language Processing
NN	Neural Network
OCR	Optical Character Recognition
PCA	Principal Component Analysis
PDF	Probability Density Function
PLP	Perceptual Linear Prediction
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Networks
SED	Sound Event Detection
STFT	Short-time Fourier Transform
SVD	Singular Value Decomposition

SVM	Support Vector Machines
TFR	Time-frequency Representation
TL	Transfer Learning
WVD	Wigner-Ville Distribution

APPENDIX A

A.1. TUT Acoustic Scenes 2017 Dataset

TUT Acoustic Scenes dataset has been used for TL techniques in the original ATHUS dataset of previous Chapter.

The dataset consists of audio files from various cases, with all of them having distinct recording locations in Finland. For each recording location, 3-5 minute long audio recording was captured. The original recordings were then split into segments with a length of 10 seconds. These audio segments are provided in individual files. The classes of the dataset depending on the content of the recording are the following:

- Bus
- Cafe / Restaurant
- Car
- City center
- Forest path
- Grocery store
- Home
- Beach
- Library
- Metro station
- Office
- Residential Area
- Train
- Tram
- Park

As proposed in related work [28], 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 bit 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.

Postprocessing of the recorded data involves aspects related to privacy of recorded individuals. For audio material recorded in private places, written consent was obtained from all people involved. Material recorded in public places does not require such consent, but was screened for content, and privacy infringing segments were eliminated. Microphone failure and audio distortions were annotated, and the annotations are provided with the data. Based on experiments in DCASE 2016, eliminating the error regions in training does not influence the final classification accuracy. The evaluation set does not contain any such audio errors.

APPENDIX B

B.1. Deep Audio Features (Python Package)

As Theodoros Giannakopoulos proposes in

https://github.com/tyiannak/deep_audio_features,

Deep Audio Features is a Python library for training Convolutional Neural Netowrks as audio classifiers using MEL SPECTROGRAMS. The library provides wrappers to pytorch for training CNNs on audio classification tasks, and using the CNNs as feature extractors.

This is the main Python library utilized for solving most of the problems in the present Thesis.

The input is folders with audio files in different classes. The package uses the folder names as classnames, extracts spectrogram representations from the respective sounds, trains and validates the CNN and saves the trained model.

Finally, deep audio features package is capable of performing TL techniques from different models as far as the testing of the results.

The main NN architecture consists of seven layers; 4 CNN and 3 Linear

The library also can provide different strategies for the TL problems as follows:

- **Strategy 0**: Use all the seven layers when transferring knowledge from a source model to another
- **Strategy 1**: Use only the three linear layers when transferring knowledge from a source model to another
- **Layers Freeze**: Choose how many layers to freeze in order to transfer knowledge from a source model to another
- **Class Weighting**: Deals with unbalanced classes

In the present Thesis all the available techniques have been utilized in order to provide the best and most accurate results.

REFERENCES

[1] Stuart Rusell, Peter Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Edition, 2010

[2] *Artificial Intelligence*, Wikipedia (https://en.wikipedia.org/wiki/Artificial_intelligence)

[3] *A complete guide to reinforcement learning*, deepsense.ai, Big Data Science (https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/)

[4] *What is AI,* TechTarget, (https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence)

[5] *Deep Neural Networks*, Anastasia Kyrykovych, Listlink, (https://www.kdnuggets.com/2020/02/deep-neural-networks.html)

[6] *What Is Deep Learning AI? A Simple Guide With 8 Practical Examples, (https://www.kdnuggets.com/2020/02/deep-neural-networks.html)*

[7] *How to Learn Mathematics For Machine Learning? What Concepts do You Need to Master in Data Science?* (https://www.analyticsvidhya.com/blog/2021/06/how-to-learn-mathematics-for-machine-learning-what-concepts-do-you-need-to-master-in-data-science/)

[8] *20 Popular Machine Learning Metrics. Part 1: Classification & Regression Evaluation Metrics* (https://towardsdatascience.com/20-popular-machine-learning-metrics-part-1-classification-regression-evaluation-metrics-1ca3e282a2ce)

[9] *Loss and Loss Functions for Training Deep Learning Neural Networks* (https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/)

[10] *12 Types of Neural Network Activation Functions: How to Choose?* (https://www.v7labs.com/blog/neural-networks-activation-functions)

[11] *Artificial Intelligence (AI)* (https://www.ibm.com/cloud/learn/what-is-artificial-intelligence)

[12] Thomas Davenport, Jeff Loucks, and David Schatsky, *Bullish on the Business Value of Cognitive (Deloitte, 2017)* (https://www2.deloitte.com/us/en/pages/deloitte-analytics/articles/cognitive-technology-adoption-survey.html)

[13] *What is Transfer Learning? (https://jpt.spe.org/what-is-transfer-learning?gclid=Cj0KCQiAu62QBhC7ARIsALXijXTvvQFvt3rUp-paL-ZRFUZVtJo0PFROZzIraK9Omg_vy-1XCuoz00MaAkrJEALw_wcB)*

[14] *What is Machine Listening? (Part 1)* (https://medium.com/cochl/what-is-machine-listening-part-1-6fbdf2a3d892)

[15] Juan P. Bello, Claudio Silva, Oded Nov, R. Luke Dubois, Anish Arora, Justin Salamon, Charles Mydlarz, And Harish Doraiswamy, *SONYC: A System for Monitoring, Analyzing, and Mitigating Urban Noise Pollution,* Communications of the ACM, vol. 62, no. 2, February 2019

[16] S. Chandrakala, S. L. Jayalakshmi, *Environmental Audio Scene and Sound Event Recognition for Autonomous Surveillance: A Survey and Comparative Studies,* ACM Computing Surveys, Vol. 52, No. 3, Article 63. Publication date: June 2019

[17] Rao B. Visvesvara, Rajeshwari & Rao, *Signals and Systems ,* PHI Learning Pvt. Ltd., 2009

[18] Theodoros Giannakopoulos Aggelos Pikrakis, *Introduction to AUDIO ANALYSIS: A MATLAB Approach,* Academic Press, 2014

[19] John G. Proakis, Dimitris K. Manolakis, *Digital Signal Processing*, fourth ed., Pearson Education, 2009.

[20] *Mono vs Stereo Sound: What's the Big Difference?* (https://www.rowkin.com/blogs/rowkin/mono-vs-stereo-sound-whats-the-big-difference)

[21] *What is a Spectrogram, (https://pnsn.org/spectrograms/what-is-a-spectrogram)*

[22] Dan Lavry, *The Optimal Sample Rate for Quality Audio,* Lavry Engineering Inc. May 3, 2012

[23] Albert S. Bregman, Auditory Scene Analysis: The Perceptual Organization of Sound, The MIT Press, 1990

[24] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, Michael Auli, *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,* v1, 2020

[25] *Sound Waves,* (https://www.pasco.com/products/guides/sound-waves)

[26] S. Chandrakala, S. L. Jayalakshmi, *Environmental Audio Scene and Sound Event Recognition for Autonomous Surveillance: A Survey and Comparative Studies,* ACM Computing Surveys, Vol. 52, No. 3, Article 63, June 2019

[27] Theodoros Giannakopoulos, Margarita Orfanidi and Stavros Perantonis, *Athens Urban Soundscape (ATHUS): A dataset for urban soundscape quality recognition,* MultiMedia Modeling (pp.338-348), 2019

[28] Acoustic Scene Classification (https://dcase.community/challenge2017/task-acoustic-scene-classification)