Media Engineering and Technology Faculty
German University in Cairo

Incorporating Auxiliary Data for Urban Sound Tagging System (PANN-MAX)

Bachelor Thesis

Author: Ahmed Amin Naem

Supervisors: Dr. Hisham Hassaballah Othman

Submission Date: 12 June, 2022

This is to certify that:

(i) the thesis comprises only my original work toward the Bachelor Degree

(ii) due acknowlegement has been made in the text to all other material used

---

Ahmed Amin Naem
12 June, 2022

Acknowledgments

I would like to thank Dr. Hisham Othman for the efforts he made in supporting us during the work of the thesis project. I also would like to thank Qiuqiang Kong for the pre-trained model. A big thank you to Hossein Gholamalinejad and Hossein Khosravi for their paper that helps us to understand different pooling types. I am very grateful to my friends and family for motivating and supporting me in doing my thesis.

Abstract

The main discussed problem is a multi-label sound tagging problem, and it aims to detect noises in urban areas. In this thesis, we will go through the "incorporating auxiliary data for urban sound tagging system" (IADUSTS) approach, which was a submission for task 5 of DCASE 2020 and apply a modification to it to improve the results. The IADUSTS approach submitted 4 different systems, but we will concentrate on the PANN-MAX system only. Although the 4 systems have a lot of common in the methodology, PANN-MAX is different in terms of the used Convolutional neural network (CNN) network, the used method to map the multiple annotations to one annotation, and the used configuration for the log-mel spectrogram extraction. The main aim of the approach is the usage of auxiliary data to build a sound tagging system. A pre-trained CNN will be used in the discussed approach. A feature vector was constructed using spatiotemporal metadata in parallel with the log-mel spectrogram feature; to use the auxiliary information. Each audio clip may have a multiple different annotations. To overcome the multiple annotations per one clip, taking the element-wise maximum method was used. An adjustment was applied to the pre-trained model in a proposal to improve the results, which is changing the pooling type. The applied modification resulted in better results, and it improved the computational time.

Contents

Chapter 1

Introduction

1.1 Motivation

In this day and age, no one can deny the evolution that is taking place around us, whether it is urban development, evolution in transportation, or communication methods. Despite the progress that humanity is witnessing in the current era, which has played an effective role in helping humans to perform many tangible achievements, there is an ugly side to this evolution; it is negatively affecting humans and the environment. Noise pollution is one of those negative effects that clearly affects humans mentally and physically [18]. As a part of the community, these negative effects compel us to think about a way to handle this problem.

1.2 Problem Statement

Urban areas have a lot of events that are happening at the same time. The particular reason for the circumstance is the humans' needs that synchronize with the development taking place around them, which leads to noise pollution which negatively affects many aspects. As we can see in figure 1.1, one in five people in Europe live in a place that has high noise levels, and it shows that it causes high sleep disturbance, heart disease, and high annoyance. Noise pollution monitoring is a major challenge in urban areas, as the noise exists in a harmful way and needs to be regulated to overcome the negatives that affect us humans and the environment around us. The IADUSTS approach that we are going to discuss is a further step to a solution that will help to overcome noise pollution by detecting and labeling urban noise, and it was a submission for DCASE 2020 task 5 [17].

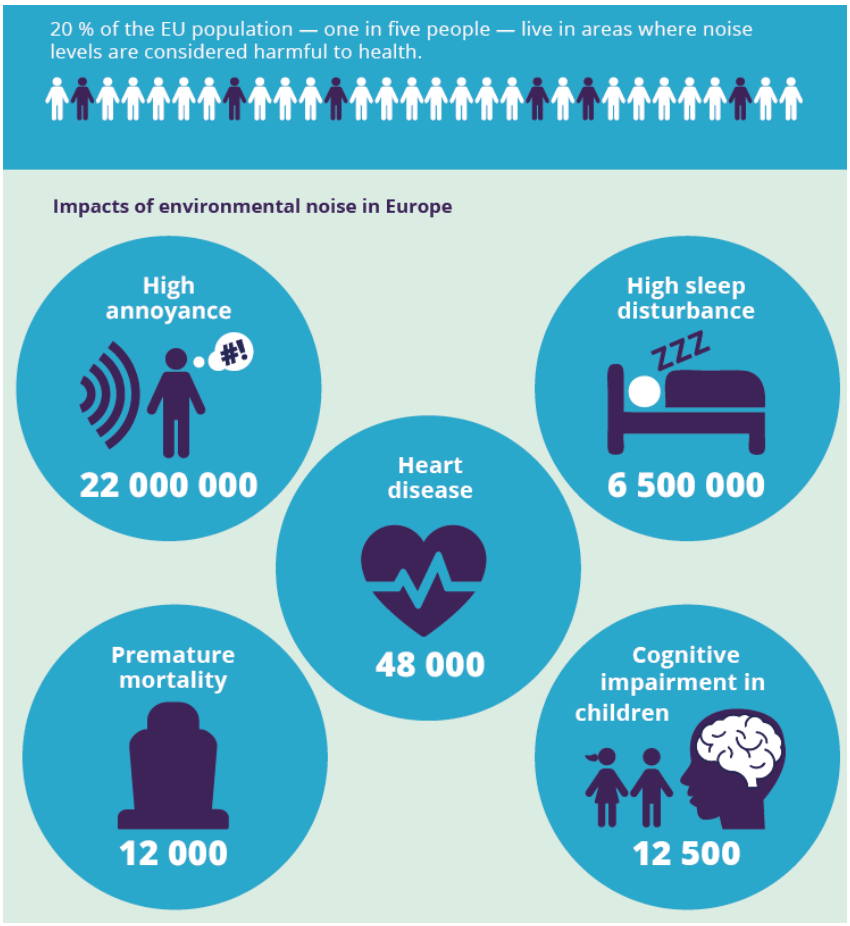

Figure 1.1: Noise pollution impacts on Human at different levels [2].

1.3 Objectives

As mentioned in the problem statement section 1.2, the IADUSTS approach is a submission for DCASE 2020 task 5 1.2, the submission includes 4 different systems, but PANN-MAX will be the discussed one. CNN is going to be used for sound tagging. The network that was chosen for the approach is a pre-trained CNN. Usage of auxiliary data is the main theme of the IADUSTS approach to help in detecting urban noise. To apply the main theme, a feature vector was constructed by using the spatiotemporal metadata that was provided with the dataset and was used along with log-mel spectrogram features that were extracted from the sound files in the dataset. The Sounds of New York City Urban Sound Tagging (SONYC-UST) dataset [8] is the one that will be used in the approach, which was a part of the Sounds of New York City (SONYC) project [6]. The research project SONYC main goal is to mitigate urban noise pollution. Crowdsourced annotations were used in the dataset, each record in the dataset may have a different annotation that annotates what type of sounds are in this record, and there is also metadata with the records dataset that contains some information regarding the audio files. This

metadata is important information that is going to be used for the sound tagging system in the discussed approach. Data augmentation was used as well. Random time-frequency masking was the method used for the data augmentation. There were four basic methods that were used for estimating the 'true' label, as there are multiple different annotations per one recording audio file, and they will be discussed in the label estimation section 3.3. A modification was applied as well to the IADUSTS approach; to enhance the result and the computational time, this is done by changing the type of pooling in the used pre-trained model for the sound tagging system.

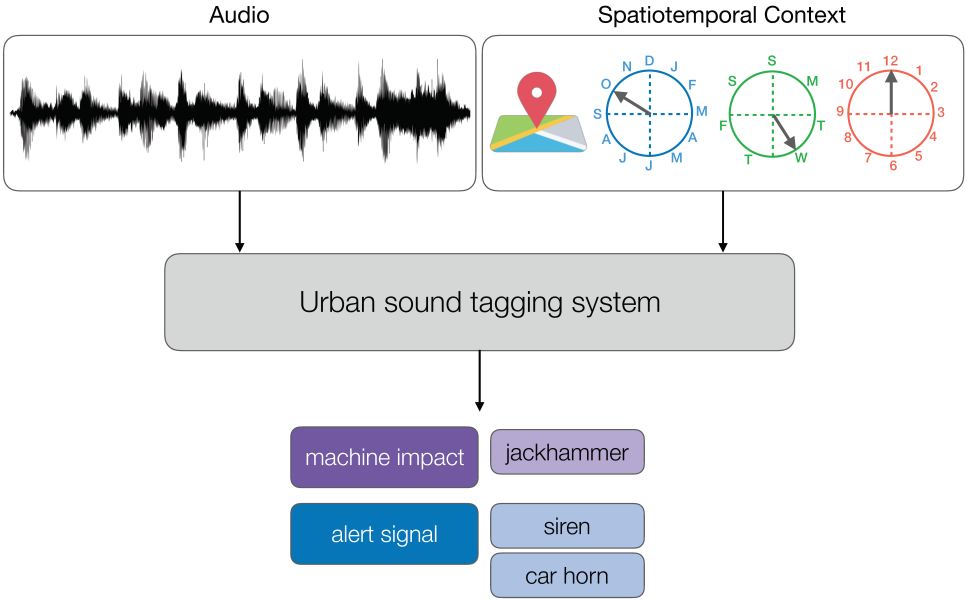

Figure 1.2: An overview of the system for sound tagging using spatiotemporal context [11].

1.4 Thesis outline

This thesis is going to have 5 other chapters than the introduction. We are going to do the literature review in chapter 2, we are going to discuss some other related papers that are related to our thesis, and we are going to introduce some basic concepts about CNN. We will go through our methodology in chapter 3, like the nature of the dataset we will be working with, the models that were used for the approach, and how these models were trained, and discuss as well the applied modification and the reason for choosing it. Our result is going to be in chapter 4, we are going to discuss our result before and after the modification and show how the modification affects it. In chapter 5 we are going to give a conclusion about the IADUSTS approach and the applied modification. Chapter 6 is going to be our future directions.

Chapter 2

Background

2.1 Basic concepts

Machine learning has several types and tools. One of them is deep learning, which is mainly a neural network consisting of three or more layers. These neural networks work like the human brain, and aim to learn from big data. Deep learning is used by many Artificial intelligence (AI) applications and it led to improvements in performing analytical, automation, and physical tasks without human interaction. Deep learning has a lot of real-life applications in different fields around us, like law enforcement, financial services, customer service, and healthcare [13].

CNN is one of the deep learning classes, which play a big role in analyzing and image recognition fields. CNN uses a very special type of method which is called convolution. CNN contains different layers of artificial neurons. Artificial neurons work like the neuron cells that exist in the human brain in passing different input signals and responses. Artificial neurons are mathematical function that is used for calculating the sum of different inputs and result in an output of activation value form. The behavior of the CNN neuron mainly depends on its weight value. When having as an input the values (of the pixel), the different specifications and features of input are recognized by the artificial neurons of a CNN. CNN neurons generally take a patch of pixels as input and multiply the pixels' values by their weight and add them all together and input them through its activation function. The output of the layer passes as an input to the next layer, which will extract some other features of the input of the images, like combinations of edges and corners. The CNN consist of three types of layers, as it is shown in figure 2.1:

(i) Convolutional layers

(ii) Pooling layers

(iii) Fully-connected (FC) layers

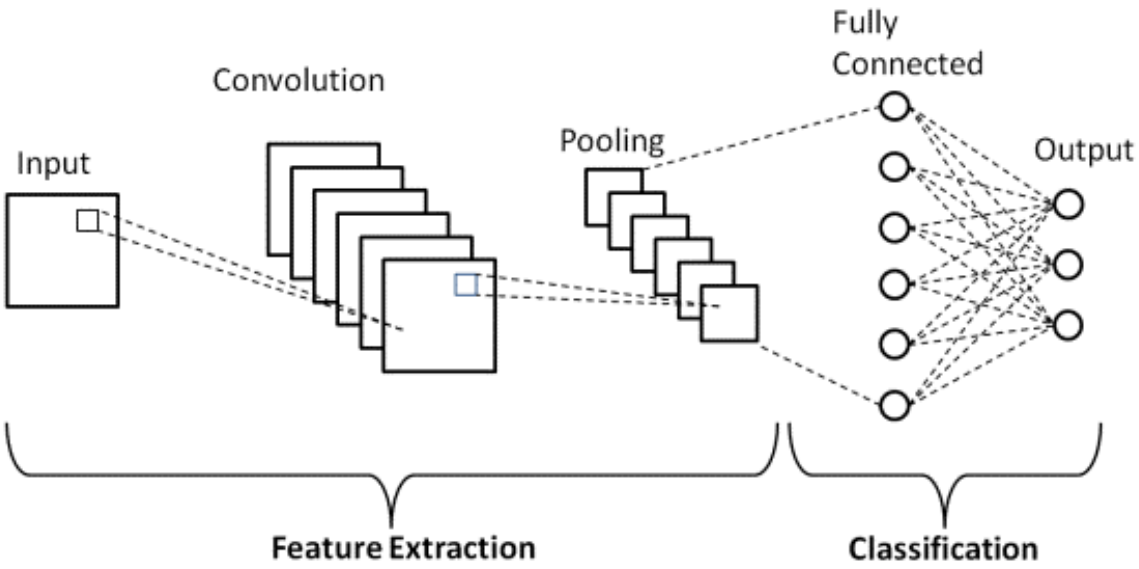

Figure 2.1: The general architecture of CNN [25].

The first layers in CNN are the convolutional layers, and it is used for the feature extraction of the inputs. The usage of the pooling layer is to reduce the features' size. The FC layer comprises the biases and weight together with the neurons and works as a connection between neurons in two separated layers [25].

2.2 Literature review

Iqbal et al. [17] submitted 3 other systems, as was mentioned in the object section 1.3:

(i) GCNN-Pseudo

(ii) PANN-Pseudo

(iii) PANN-Ensemble

The common difference between our system and the other three systems is the used approach for the label estimation. The other three systems used the learning algorithms to estimate the labels, and this was inspired by another submission to DCASE [1]. A two-layer neural network was used for estimating the labels, and it has as an input the mean of the noisy annotations. The inputs were mapped to 128-dimensional embedding by the first FC by using an affine transformation followed by relu as an activation function. This embedding is then used in the second FC layer to get mapped to the estimated annotation, followed by the sigmoid activation function. This pseudo-labelling has a positive effect on the micro AUPRC metric, which is a metric to evaluate the performance

of the neural networks, but not on the other metrics, and this could be noticed between the different results of the two systems, PANN-Max and PANN-Pseudo, in table 4.1. PANN-Ensemble used mean ensemble to combine 4 PANN models. And with this ensemble system, the performance is improved in almost all metrics, as shown in both tables 4.1 and 4.2. GCNN-Pseudo is a single-model system based on the Randomly-initialised gated CNN (GCNN) model [10]. The GCNN model is applied to the log-mel features, it consists of 10 gated convolutional (GC) layers. Batch normalization is applied after each GC layer. Max-pooling was used after every two GC layers; to reduce the feature map size. Average pooling was used following GC layers, to reduce the time and frequency dimensions to a scalar. From tables 4.2 and 4.1, it can be deduced that the PANN systems outperform the GCNN system in all metrics.

A fusion system was submitted by Bai et al. [4] by taking advantage of using different features and data augmentation methods to get better results. 9-layer CNN is used as a primary classifier for urban noise. The training set that the author was using is imbalanced; there are fewer recordings for non-machinery-impact,machinery-impact, powered-saw, dog, and music classes; the author submitted two different data augmentations to solve this problem. The first used augmentation is randomly adding one-class-present recordings of the fewer recording classes into a new recording. Loud sounds such as powered-saw and machinery-impact could mask the non-machinery-impactor or music; to overcome this, amplitude factors are applied. The second used augmentation is mix-up data during the training [29]; this leads to having more training samples without even extra computing resource. For tagging the urban sound, the author explores three different features:

(i) Log-mel

(ii) Log-linear

(iii) Harmonic percussive source separation (HPSS) [14]

Mel filters are used to transform short-time Fourier transform (STFT) spectrogram to mel spectrogram, and linear filters are also applied to generate linear spectrograms. HPSS can split the signal into harmonic and percussive parts [27]. In order to decrease the size of the input, mel filters with 64 bands are used to extract the mel-harmonic spectrogram. To get log spectrograms, log algorithm are used for the three different spectrograms. Both spatial and temporal contexts are concatenated with the feature vector after the convolutional blocks on the frame level, then the concatenated feature is passed to the dense layer. From table 2.1, we can deduce that the used CNN architecture is like the VGG model.

Table 2.1: CNN9 architecture [4].

CNN9		
Features	**Log-mel**	**Log-linear**
Conv1	(3*3@64,BN,Relu)*2	
Pool1	2*2 average pooling	
Conv2	(3*3@128,BN,Relu)*2	
Pool2	2*2 average pooling	
Conv3	(3*3@256,BN,Relu)*2	
Pool3	2*2 average pooling	
Conv4	(3*3@256,BN,Relu)*2	
Pool4	2*2 average pooling	
Dense	TimeDistributed	
Dense	TimeDistributed	
AutoPool	AutoPool1D	

During training to prevent overfitting and to speed up batch normalization is applied. After batch normalization, leaky- Relu is used as an activation function. Average pooling is used to reduce the feature map size. Among the three different spectrograms experimented on the CNN9 model, the log-mel spectrogram outperforms the log-linear spectrogram in human-voice, powered-saw, and non-machinery, especially dogs. Log-linear is better than the log-mel spectrogram by 1% for machinery sound; the explanation for this is that the log-linear spectrogram gives high resolution in the higher-frequency areas. This could be shown in some machinery sounds as it contains high frequency components. Log-mel-h achieves the best score of 69% in the music class, the harmonic components can predict the music sounds highly from other classes. The two augmentation methods were applied based on log-mel and CNN9. The random-adding method resulted in adding 5000 more training recordings. Compared to the no augmentation method, randomly add resulted in decreasing about 4% to 6% over the three metrics; because of the inappropriate amplitude factors, mixup increased the macro-auprc from 68% to 72%. The author experimented the CRNN architecture as well, the feature maps that were extracted from CNN9 were fed to GRU, and the mixup augmentation was applied to the CRNN. CRNN achieved 67% macro-auprc score using mixup and 70% macro-auprc score without using mixup.

Diez et al. [12] propose a sound tagging system that has two different architectures based on CNN. Log-mel spectrogram features are used as an input in the network, and it is operated by two CNN layers. The output is then passed through several FC layers and ends up with the output layer to predict the sound class. The spatiotemporal feature vector is passed as an input to an additional neural network consisting of 2 FC layers. Its output and CNN stack output are concatenated together and pass through the FC output block. The spatiotemporal data consist of spatial information and temporal information. The recording audios are 10 seconds long, and they are fragmented into 1-second frames

with an overlapping of 0.5s; because they are too long to get inside the neural network, the log-Mel spectrogram was extracted using 128 mel bands, and each frame is independently ZScore normalized. The author has evaluated two types of architecture, as we can see from figure 2.2:

(i) Audio-only architecture

(ii) Audio+spatiotemporal context architecture

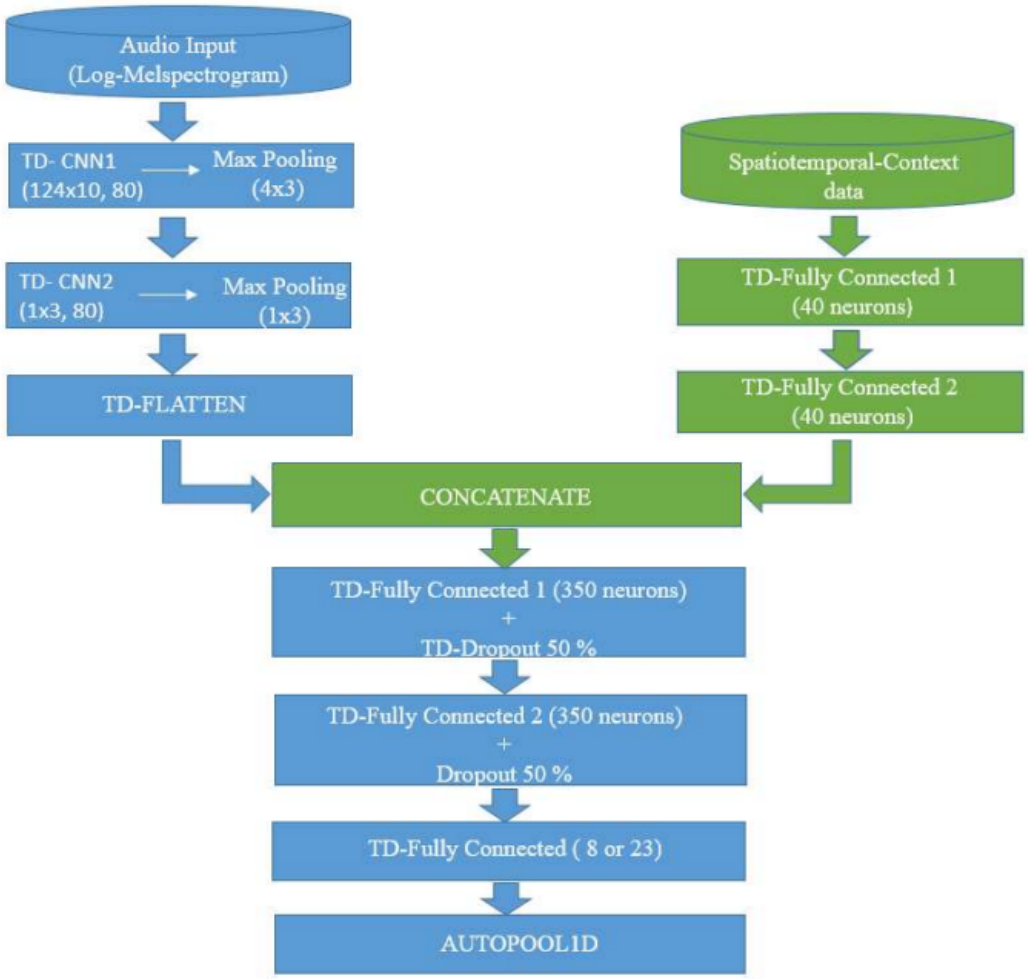

Figure 2.2: The blue shapes represent the only audio architecture. The combination of the blue and green shapes represents audio+spatiotemporal context architecture [12].

The audio-only architecture general core is CNN, which takes as input a 4-D tensor that passes through 2 CNN layers with 80 filters for each of them. The first layer has a kernel with size (124 x10) that is followed by a max-pooling with size (4x3), filter is used that

covers most of the frequency range of the log-mel spectrogram, the second layer has a kernel with size (1x3) that is followed by a max-pooling with size (1x3), the output of the second layer is flattened and passed to a block of two FC layers that have 350 neurons each followed by an output layer that outputs the classification score. The used convolutional and FC layers use relu as an activation function, but the softmax activation function will be used by the output layer, 50% dropout is applied after every FC hidden layer. The Audio+spatiotemporal context architecture is called multi-input architecture; as it has two types of inputs:

(i) Audio data

(ii) Spatiotemporal context data

The audio data branch contains two convolutional layers with the same configuration of the audio-only architecture. The spatiotemporal context branch consists of two FC hidden layers of 40 neurons each and uses relu as an activation function; before the concatenation of the two branch outputs, the audio data branch output should be flattened, and the concatenated output is used as the input of the final classification block that has the same configuration of the audio-only architecture as well. The author submitted 3 different models:

(i) Model 1: the audio-only architecture was used based on the CNN in which the audio files are the only input and not spatiotemporal context data.

(ii) Model 2: the Audio+spatiotemporal context architecture was used, so the inputs will be the audio and spatiotemporal context data.

(iii) Model 3: will be the same as model 2, but the only difference is regarding spatiotemporal context data as it will have in addition the sensor identification, block, and borough.

For coarse-level labels, model 1 which uses audio data only as input without spatiotemporal context data, outperforms the other two systems, and model 3, which uses spatiotemporal context data with the sensor identification, block, and borough as an input, outperforms model 2, but all three systems did not outperform the baseline score. For fine-level labels, model 1 also outperforms the other 2 models, model 3 outperforms model 2, and all models did not outperform the baseline.

Chapter 3

Methodology

3.1 Dataset

The SONYC-UST v2 dataset provides us with audio recordings that were recorded at different places and times in New York City [8]. The dataset includes more than 18000 audio clips through a series of urban sound classes. The Dataset consists of:

(i) Validation set

(ii) Training set

(iii) Test set

The development set for task 5 is both the validation set and the training set. Each audio clip is recorded somewhere in New York at a random time, and it is a recorded 10-second clip. For each clip, it might be noticed multiple urban noises exist; that's why our problem is called multi-label sound tagging. The dataset consists of two types of classes :

(i) coarse-level class

(ii) fine-level class

As shown in figure 3.1, we could deduce that the coarse-level class could be split into one or more fine-level classes. There are 29 fine-level and 8 coarse-level classes in total. The output of the used system will be coarse-level predictions, and it may have as well fine-level predictions. As we mentioned in the introduction 1, that crowdsource annotation was used in the dataset, Zooniverse citizen science platform was used for the crowdsource annotations [26]. In the development set, each recording has at least three annotations. Records could have different annotations regarding what kind of noise exist; crowdsource

nature. But there is an exception with 538 recording clips in the validation set. They are associated with annotations that are agreed to be correct by the SONYC team.

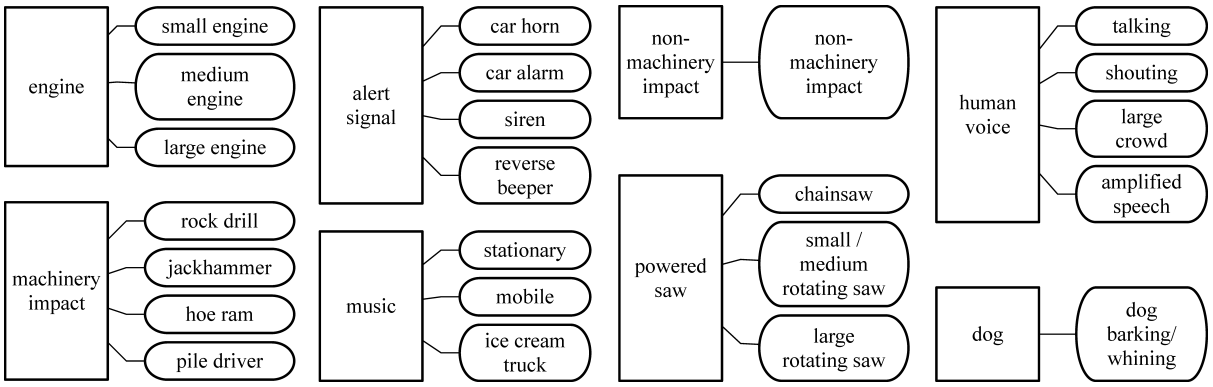

Figure 3.1: Rectangular boxes represent coarse tags, and round boxes represent fine tags [11].

For each recording clip and annotation there is metadata that identifies the following:

(i) Annotator

(ii) Recording sensor

(iii) Perceived proximity

(iv) Time

(v) Location

The time is divided into:

(i) Year

(ii) Week

(iii) Day

(iv) Hour

The location is divided into:

(i) Borough

(ii) Block

(iii) Latitude

(iv) Longitude

Both location and time will be referred to them as the STC, and it will be one of the used features in the sound tagging system.

3.2 Features

We will start to talk about the features that are used for the sound tagging system and how they are extracted. Two types of features were used :

(i) Logarithmic mel-scaled (log-mel) spectrograms

(ii) STC feature vectors

The log-mel spectrograms and the STC feature vectors were extracted using audio files and STC metadata respectively from the dataset.

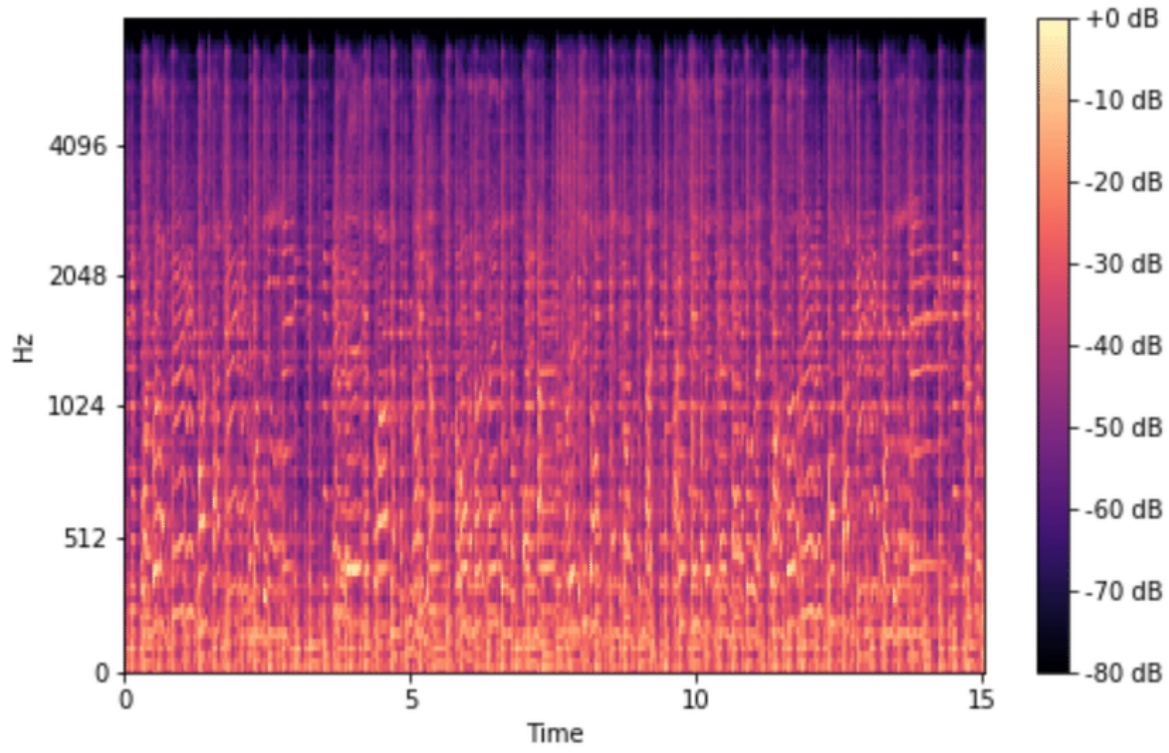

Figure 3.2: How log-mel spectrogram looks like [23].

3.2.1 Log-mel spectrograms

Let us talk about the log-mel spectrograms and the process of the extraction. Firstly and as a prepossessing step, downsampling was applied to the provided audio file, from 48 kHz to 32 kHz, with the goal of reducing the size of the features. The extraction of the log-mel spectrogram happens by applying an STFT on the audio signal, then square the result, the mel filters banks will be used to scale the frequency axis, and the logarithmic

function will be used to scale the magnitude. The log-mel spectrogram should finally look like figure 3.2.

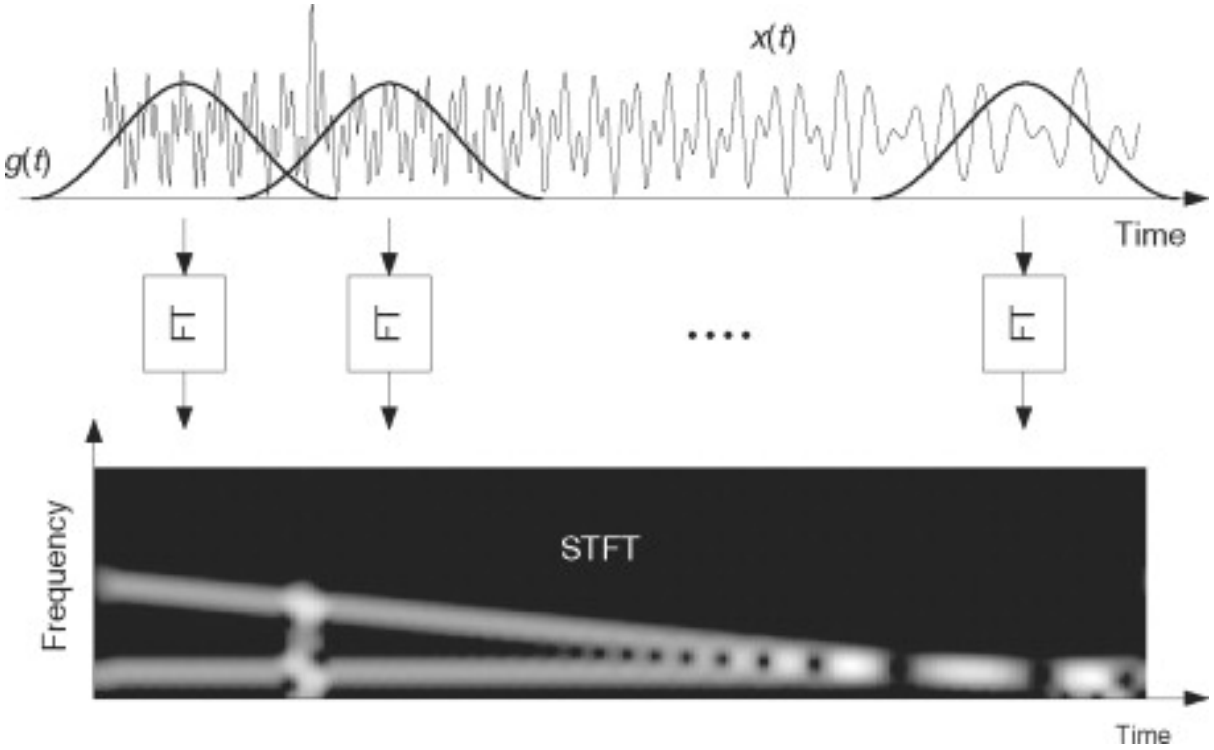

Figure 3.3: How short-time Fourier transform is applied to the signal [3].

As shown in figure 3.3, STFT is a method applied to a long-time signal, by dividing it into small segments which is called windowing, then apply for each segment Fourier transform and adds them all together [3]. STFT has two main parameters :

(i) Window size

(ii) Hop size

Figure 3.4 shows both parameters, window length is the size of the segment, and the number of samples between each successive window is known as hop length. Both Window length and hop length are using samples as measurement units. 64 mel bins were used for the extraction of the spectrogram. The configuration that was used for the window length and hop length of the STFT is:

(i) Window length = 1024 samples

(ii) hop length = 512 samples

The used configuration will result in (626 × 64) log-mel spectrogram.

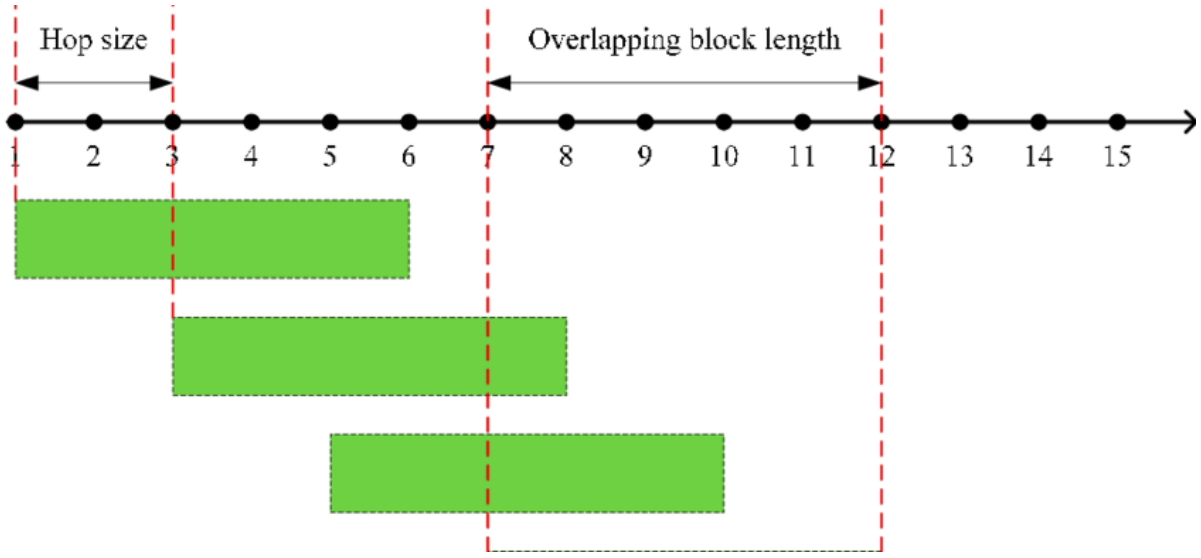

Figure 3.4: The green box represents windows and we can see in the below figure the hop size [7].

3.2.2 STC features

The week, day, and hour were used in the construction of the STC feature vector. The temporal metadata was only used as using the spatial metadata is useless and did not improve the performance of the sound tagging system, so it was omitted. In the dataset, w, d, and t will be referring to week, day, and time respectively. Rather than using those integers values in the dataset to construct the vector, the final feature vector will be a mix of categorical and continuous data and is given by

$$\alpha = [\eta\lfloor w/4 \rfloor, \eta\lfloor t/3 \rfloor, \delta, \theta 26(w), \theta 12(t), d] \tag{3.1}$$

Let us analyze the vector's elements one by one:

(i) The first two elements are one-hot encoding, which is the essential process of converting categorical data to be used by deep learning algorithms, so it will result in improving the predictions as well as classification accuracy of our model [22].

(ii) The third element is used as a tag for indicating if the day is a weekend or not.

(iii) The fourth and fifth elements are triangular functions defined as

$$\theta B(X) = B - \mid B - X \mid \tag{3.2}$$

The reason for using it is to wrap the input to reflect its circular nature. Let us say that we have t = 2 and t = 22, they are far apart as values, but they are so close in the context of time (2 am and 10 pm).

(iv) The sixth element is the day that the audio clip was recorded.

3.3 Label estimation

As was mentioned in the Dataset section 3.1, each audio clip in the development set has at least three annotations that could be different from each other; due to the crowdsource nature. So there were two different approaches used to map the different annotations to a single annotation [17]. One of these approaches is to use a learning algorithm [1], which was not used in our case, as we are using a different system. There were 4 different basic methods that were tried by the author [17]:

(i) Taking the mean

(ii) Taking the mean and then rounding

(iii) Doing a majority vote

(iv) Taking the element-wise maximum

From those 4 basic methods, taking the element-wise maximum proved to give the best result [17].

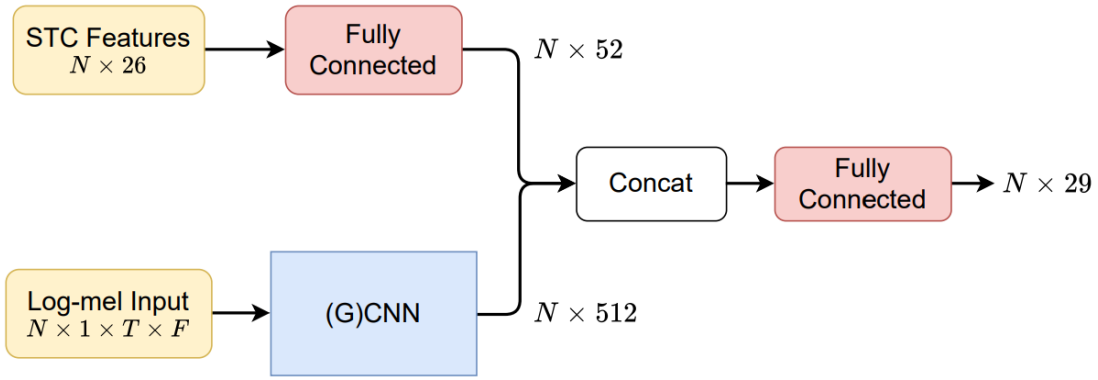

Figure 3.5: The general architecture of the sound tagging system. The number of batches per one epoch is represented by N [17].

3.4 Models

Figure 3.5 shows the general architecture of the urban sound tagging, we can see that there are two branches:

(i) STC feature branch

(ii) log-mel spectrogram branch

STC feature vectors will go through an FC layer that has 52 output features, after applying to it batch normalization [16], followed by the activation function relu that we can see its graph in figure 3.7. Batch normalization is a process of making neural networks faster by adding some more layers that perform normalization and standardization on its input.

The log-mel brach was implemented using a pre-trained CNN. 'CNN10' is the used pre-trained CNN, and it is an audio neural network (PANN) proposed by Kong et al. [21], which was pre-trained on AudioSet. As shown in table 3.1, CNN10 consists of 8 convolution layers, each layer applies batch normalization and has relu as an activation function, and finally, we have two FC layers, the last FC layer will be omitted as it is for mapping to class probabilities.

Table 3.1: The general architecture of the CNN10 pre-trained model [21].

VGGish	CNN6	CNN10	CNN16
Log-mel spectrogram 96 frames x 64 mel bins	Log-mel spectrogram 1000 frames x 64 mel bins		
3 x 3 @ 64 ReLU	5 x 5 @ 64 BN, ReLU	(3 x 3 @ 64 BN, ReLU) x 2	(3 x 3 @ 64 BN, ReLU) x 2
MP 2 x 2	Pooling 2 x 2		
3 x 3 @ 128 ReLU	5 x 5 @ 128 BN, ReLU	(3 x 3 @ 128 BN, ReLU) x 2	(3 x 3 @ 128 BN, ReLU) x 2
MP 2 x2	Pooling 2 x 2		
(3 x 3 @ 256 ReLU) x 2	5 x 5 @ 256 BN, ReLU	(3 x 3 @ 256 BN, ReLU) x 2	(3 x 3 @ 256 BN, ReLU) x 2
MP 2 x 2	Pooling 2 x 2		
(3 x 3 @ 512 ReLU) x 2	5 x 5 @ 512 BN, ReLU	(3 x 3 @ 512 BN, ReLU) x 2	(3 x 3 @ 512 BN, ReLU) x 2
MP 2 x 2 Flatten	Global pooling		Pooling 2 x 2
(FC 4096 ReLU) x 2	FC 512, ReLU		(3 x 3 @ 1024 BN, ReLU) x 2
FC 527, Sigmoid	FC 527, Sigmoid		Pooling 2 x 2
			(3 x 3 @ 2048 BN, ReLU) x 2
			Global pooling
			FC 2048, ReLU
			FC 527, Sigmoid

STC feature branch will result in 52- feature embedding, while the log-mel branch will result in 512-feature embedding. Both results will be concatenated and will output 564 feature embedding. Finally, an FC layer will be applied to the 564 features with a sigmoid non-linearity activation function 3.6 to output the class probabilities.

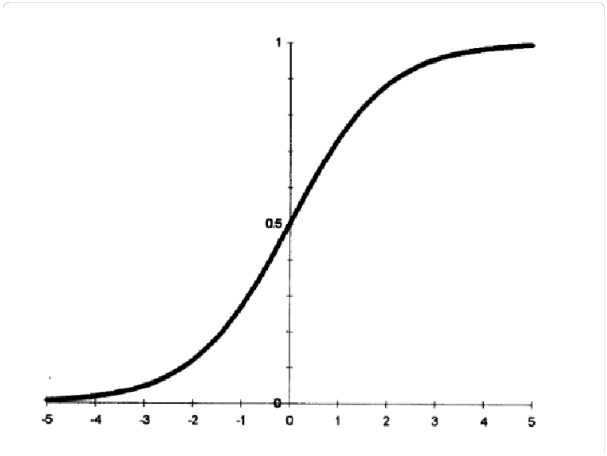

Figure 3.6: Sigmoid non-linearity activation function graph [19].

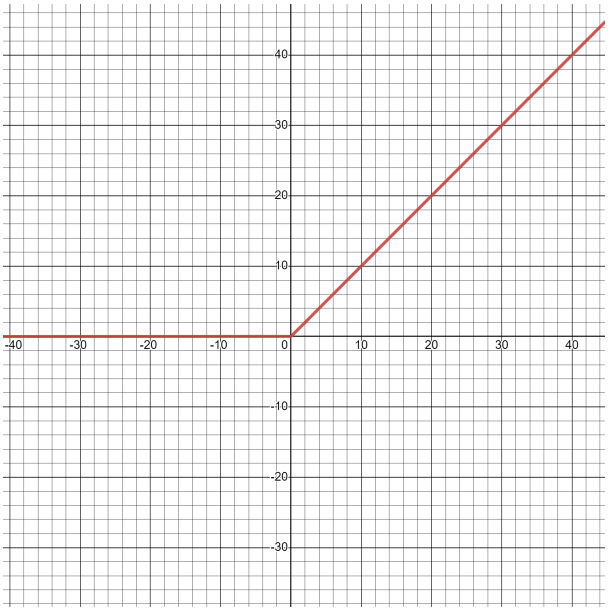

Figure 3.7: Relu activation function graph.

3.4.1 Models training

Binary cross-entropy loss function with the Adam stochastic gradient descent algorithm [20] was used for the model's training. Each model was trained in 25 epochs and batch size 64. Convolutional layers of the PANN were trained by a learning rate of 0.00025; the particular reason for this is that it needs to be fine-tuned, and this results in better performance in the experiments [17]. As a form of data augmentation, SpecAugment [24] was used on the log-mel input. As shown in figure 3.8, SpecAugment has two types of transformations:

(i) Time-frequency masking

(ii) Time warping

Time-frequency was only used for data augmentation, as time warping gives bad results [17]. Up to F = 8 consecutive mel bins were masked, and up to mF = 2 frequency masks were applied on the frequency axis. The number and the location of bins and the number of masks were chosen randomly. For the time axis, up to T = 8 consecutive frames were masked, and up to mT = 8 time masks were applied.

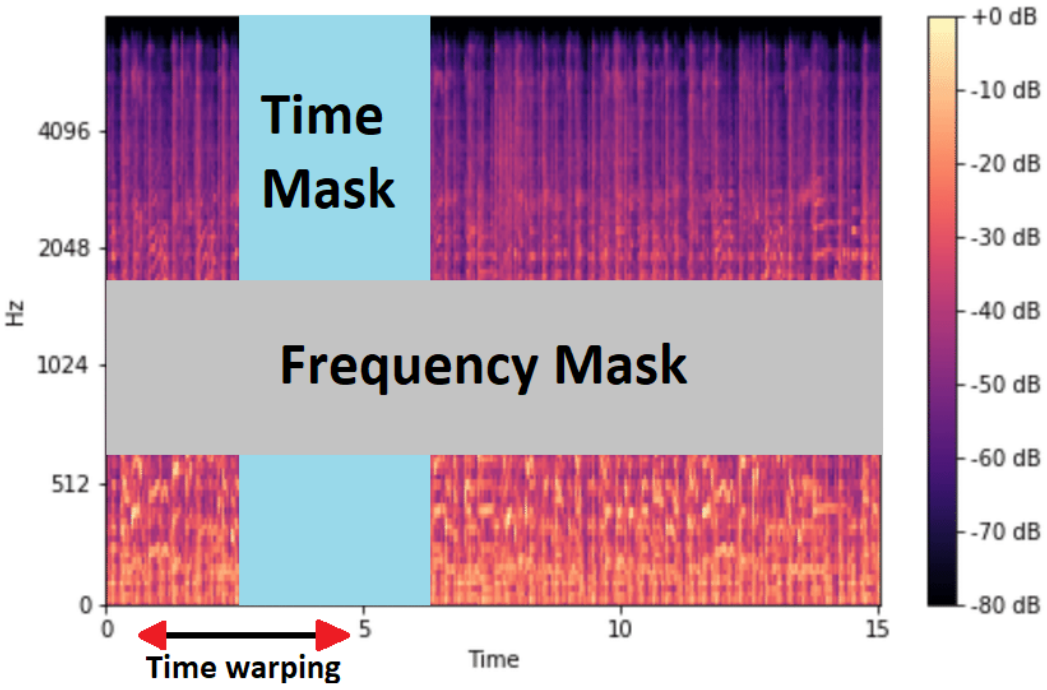

Figure 3.8: Different SpecAugment types and how they are applied to the spectrogram.

3.5 Modification

As shown in table 3.1, after every two convolutional layers, there is an average pooling applied. The pooling layer is a layer that is doing down-sampling on its input that is coming from the previous layer and results in new feature maps with a condensed resolution, and it tends to reduce the spatial dimension of the feature map [15]. There is two main propose for using pooling :

(i) Reduce the numbers of the parameters and weight.

(ii) Overcoming the over-fitting issues in the networks.

There is a lot of different type of pooling. So to choose the right one, you should be focusing on extracting the important information and dropping the irrelevant ones. Two types of pooling are going to be discussed:

(i) Average pooling.

(ii) Max-pooling.

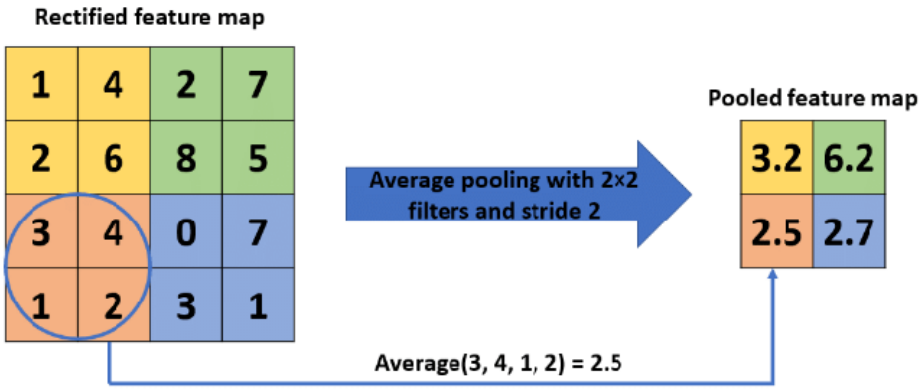

Figure 3.9: How average pooling works [15].

The average pooling layer does the downsampling by dividing the feature map into rectangular pooling regions and calculating the average values of each region, as shown in figure 3.9. The difference with the max-pooling layer is that it is uses the maximum values of each region, as shown in figure 3.10 [15].

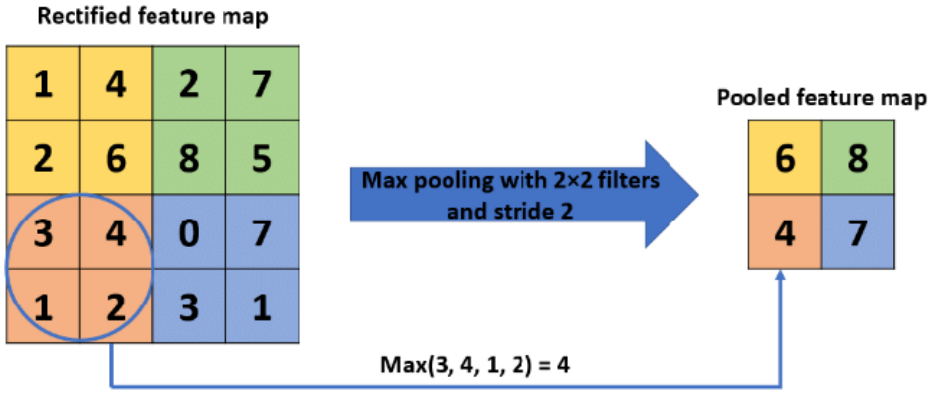

Figure 3.10: How max-pooling works [15].

CNN10 model, which is used on the log-mel inputs as was mentioned in the models section 3.4, uses average pooling. The used modification is to change the type of pooling

to max-pooling; the particular reason for that is that the log-mel spectrogram is a visual representation of sounds, and any image has a pixel representation, as shown in figure 3.11, so the value of pixels' values varies from 0 to 255, and as the value is increasing the pixel get brighter [28]. It can be deduced from figure 3.2 that the log-mel spectrogram is a black background, and each color point that is brighter than the background represents how much frequency is present at a certain time, and this is the important information that we want to extract from the feature map. This modification enhances the result and decreases the time computation as well.

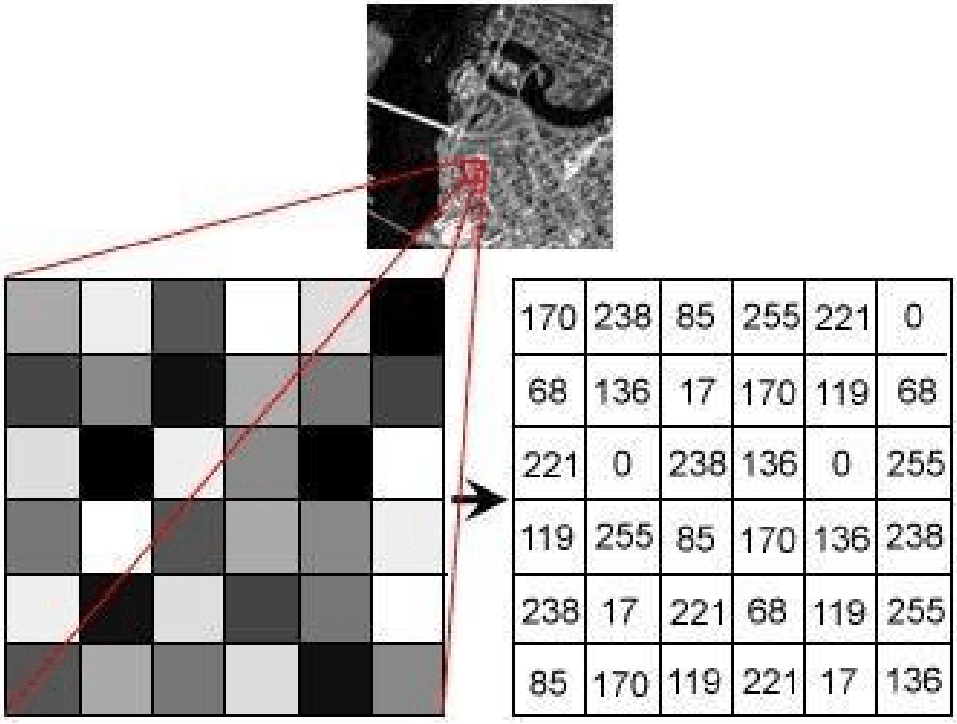

Figure 3.11: Image pixel representation [9].

Chapter 4

Results

4.1 Process and metrics for evaluating sound tagging system

We are going to show our results before the modification, and compare it to the author's results, and the results after applying the modification and show how it improves the performance of the sound tagging system. The used training set will be the verified subset of the validation set; as it is the set with the verified annotations as was mentioned in the dataset section 3.1. The metrics that are used for evaluating the network are proposed by the DCASE team. There are three metrics:

(i) Macro-averaged area under the precision-recall curve (AUPRC).

(ii) Micro-averaged AUPRC.

(iii) Micro-averaged F1.

Macro-averaged AUPRC is the primary metric for evaluating the different submitted sound tagging systems for DCASE.

The dataset has coarse and fine labels as was mentioned in the dataset section 3.1. The results will show the sound tagging system performance in both labels.

4.2 Author's Results

As was mentioned in the object section 1.3, the author submitted 4 different sound tagging systems. The results of the 4 different systems are in table 4.1 and table 4.2, and the baseline result of task 5 that was submitted by the DCASE team is included as well in the table. Table 4.1 represents the result for fine-level labels, while table 4.2 represents the result for coarse-level labels. In both fine and coarse levels PANN-MAX gives better results than the baseline over the three metrics.

Table 4.1: Author's results of the fine-level predictions [17].

	Micro AUPRC	Macro AUPRC	Micro F1
Baseline	0.7329	0.5278	0.6149
GCNN-Pseudo	0.7881	0.5645	0.6845
PANN-Pseudo	0.8111	0.6380	0.7067
PANN-Max	0.8045	0.6324	0.7127
PANN-Ensemble	0.8214	0.6548	0.7174

Table 4.2: Author's results of the coarse-level predictions [17].

	Micro AUPRC	Macro AUPRC	Micro F1
Baseline	0.8391	0.6370	0.6736
GCNN-Pseudo	0.8740	0.6761	0.7506
PANN-Pseudo	0.8918	0.7514	0.7795
PANN-Max	0.8891	0.7515	0.7716
PANN-Ensemble	0.8984	0.7667	0.7776

4.3 Our results

4.3.1 Before modification

Table 4.3 represents our results for fine-level labels, while table 4.4 represents our results for coarse-level labels. The author's results will be our baseline. In the fine-level and coarse-level labels, our results were so close to the author's results and there was not a significant difference over the three different metrics. Our macro AUPRC was less than the author by 0.05% for the fine-level labels, and for coarse-level labels, our is higher by 0.26%.

Table 4.3: Our results of the fine-level predictions.

	Micro AUPRC	Macro AUPRC	Micro F1
PANN-Max(before modification)	0.8008	0.6319	0.7007
PANN-Max(after modification)	0.8048	0.6735	0.7056

Table 4.4: Our results of the coarse-level predictions.

	Micro AUPRC	Macro AUPRC	Micro F1
PANN-Max(before modification)	0.8862	0.7541	0.7621
PANN-Max(after modification)	0.8874	0.7828	0.7723

4.3.2 After modification

After applying the modification, by changing the type of pooling in the 'CNN10' models from average pooling to max-pooling, the results were improved for both fine-level and coarse-level labels. For the fine-level labels, macro AUPRC increased after the modification by 4.16% to our results before modification, and by 4.11% to the author's results. For the coarse-level labels, macro AUPRC increased after the modification by 2.87% to our results before modification, and by 3.13% to the author's results. Also, the computational time was positively impacted by the applied modification, as was mentioned in the models training section 3.4.1, the model was trained for 25 epochs, before the modification, it took on average 8.86 minutes to train one epoch, but after the modification, it decreases to 8.33 minutes.

Chapter 5

Conclusion

To sum up, we went through the IADUSTS approach to develop an urban sound tagging system, which was a submission for DCASE 2020 task5 , and do a modification to enhance the results. We have used PANN-MAX system only, which differs from other systems in some aspects like the used CNN network, and the used approach for the multiple annotations. Spatiotemporal metadata was used for feature vector construction and it was used in parallel with the log-mel feature for training a pre-trained model 'CNN10'. Taking the element-wise maximum method was used to address the multiple annotations to a single one. Data augmentation was applied for the log-mel inputs, time-frequency masking was the used type for the data augmentation. The used modification was to change the pooling type of 'CNN10' from average pooling to max-pooling. The applied modification tends to improve the results and computational time as well. The results were increased by 4.16% for the fine-level labels, and by 2.87% for the coarse-level labels after applying the modification, and the computational time decreased from 8.86 minutes per epoch to 8.33 minutes.

Chapter 6

Future Work

Although the used modification enhances the results, max-pooling is not the most idealistic pool type for the log-mel spectrograms, as some information that could be important is ignored, so we could think about another pooling type that could combine the idea of average pooling and max-pooling. Other activation functions could be used as well rather than using the relu activation function, like mish and swish activation functions that outperform the relu in some very deep networks [5].

Appendix

Appendix A

Lists

IADUSTS	"incorporating auxiliary data for urban sound tagging system"
CNN	Convolutional neural network
AI	Artificial intelligence
SONYC-UST	Sounds of New York City Urban Sound Tagging
SONYC	Sounds of New York City
STC	spatiotemporal context
log-mel	Logarithmic mel-scaled
STFT	short-time Fourier transform
FC	Fully-connected
AUPRC	area under the precision-recall curve
GCNN	Randomly-initialised gated CNN
GC	gated convolutional
HPSS	Harmonic percussive source separation

List of Figures

List of Tables

Bibliography

[1] Sainath Adapa. Urban sound tagging using convolutional neural networks, 09 2019.

[2] EEA(European Environment Agency). Noise pollution is still widespread across europe, but there are ways to reduce the volume, May 2021.

[3] Ali N. Akansu and Richard A. Haddad. Chapter 5 - time-frequency representations. In Ali N. Akansu and Richard A. Haddad, editors, *Multiresolution Signal Decomposition (Second Edition)*, pages 331–390. Academic Press, San Diego, second edition edition, 2001.

[4] Jisheng Bai, Chen Chen, Jianfeng Chen, Mou Wang, Xiaolei Zhang, and Qingli Yan. Data augmentation based system for urban sound tagging. Technical report, DCASE2020 Challenge, October 2020.

[5] Krutika Bapat. Swish vs mish: Latest activation functions, Oct 2019.

[6] 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*, 62(2):68–77, February 2019. Funding Information: This work is supported in part by the National Science Foundation (Award 1544753), NYU's Center for Urban Science and Progress, NYU's Tandon School of Engineering, and the Trans lational Data Analytics Institute at The Ohio State University.

[7] Wahyu Caesarendra and Tegoeh Tjahjowidodo. A review of feature extraction methods in vibration-based condition monitoring and its application for degradation trend estimation of low-speed slew bearing. *Machines*, 5:21, 09 2017.

[8] Mark Cartwright, Jason Cramer, Ana Mendez, Yu Wang, Ho-Hsiang Wu, Vincent Lostanlen, Magdalena Fuentes, Graham Dove, Charlie Mydlarz, Justin Salamon, Oded Nov, and Juan Bello. Sonyc-ust-v2: An urban sound tagging dataset with spatiotemporal context. 09 2020.

[9] Goutam Das, Nurul Anwar, Shovan Chowdhury, and Kazi Rahman. Design and fabrication of an image processing based autonomous weapon. *International Journal of Engineering Research*, ISSN:2319–68902347, 12 2016.

[10] Yann Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. 12 2016.

[11] Dcase. Urban sound tagging with spatiotemporal context, 2020.

[12] Itxasne Diez, Peio Gonzalez, and Ibon Gonzalez. Urban sound classification using convolutional neural networks for DCASE 2020 challenge. Technical report, DCASE2020 Challenge, October 2020.

[13] IBM Cloud Education. What is deep learning?

[14] Derry Fitzgerald. Harmonic/percussive separation using median filtering. *13th International Conference on Digital Audio Effects (DAFx-10)*, 01 2010.

[15] Hossein Gholamalinejad and Hossein Khosravi. Pooling methods in deep neural networks, a review, 09 2020.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 02 2015.

[17] Turab Iqbal, Yin Cao, MarkD . Plumbley, and Wenwu Wang. Incorporating auxiliary data for urban sound tagging technical report. 2020.

[18] Hiral Jariwala, Huma Syed, Minarva Pandya, and Yogesh Gajera. " noise pollution human health: A review ". 03 2017.

[19] Baraka Kichonge. Analysis of tanzanian biomass consumption using artificial neural network. *Journal of Fundamentals of Renewable Energy and Applications*, 05, 01 2015.

[20] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[21] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, and Mark Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition, 12 2019.

[22] Mohita Madaan. Handling categorical variables with one-hot encoding, Apr 2022.

[23] Massoud Massoudi, Siddhant Verma, and Riddhima Jain. Urban sound classification using cnn. pages 583–589, 01 2021.

[24] Daniel Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin Cubuk, and Quoc Le. Specaugment: A simple data augmentation method for automatic speech recognition. pages 2613–2617, 09 2019.

[25] Pranshu Sharma. Basic introduction to convolutional neural network in deep learning, Mar 2022.

[26] Robert Simpson, Kevin Page, and David De Roure. Zooniverse: observing the world's largest citizen science platform. pages 1049–1054, 04 2014.

[27] Hideyuki Tachibana, Nobutaka Ono, and Shigeki Sagayama. Singing voice enhancement in monaural music signals based on two-stage harmonic/percussive sound separation on multiple resolution spectrograms. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22:228–237, 01 2014.

[28] Vipin Tyagi. *Understanding Digital Image Processing*. 09 2018.

[29] Hongyi Zhang, Moustapha Cisse, Yann Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. 10 2017.