Technische Universität Berlin
Fakultät I - Geistes- und Bildungswissenschaften
Institut für Sprache und Kommunikation
Fachgebiet Audiokommunikation

Master Thesis
Visualization of feature maps from convolutional neural network for an audio-based patient monitoring system

Submission date: 18.02.2022

Supervised by Prof. Dr. Stefan Weinzierl
Dr. Athanasios Lykartsis

Author:
Yuchen Wang, [redacted]

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt gegenüber der Fakultät I der Technischen Universität Berlin, dass die vorliegende, dieser Erklärung angefügte Arbeit selbstständig und nur unter Zuhilfenahme der im Literaturverzeichnis genannten Quellen und Hilfsmittel angefertigt wurde. Alle Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind kenntlich gemacht. Ich reiche die Arbeit erstmals als Prüfungsleistung ein. Ich versichere, dass diese Arbeit oder wesentliche Teile dieser Arbeit nicht bereits dem Leistungserwerb in einer anderen Lehrveranstaltung zugrunde lagen.

Titel der schriftlichen Arbeit

Visualization of feature maps from convolutional neural network for an audio-based patient monitoring system

Verfasser

Yuchen Wang, Matrikel-Nr: [redacted]

Betreuende Dozenten

Prof. Dr. Stefan Weinzierl,
Dr. Athanasios Lykartsis

Mit meiner Unterschrift bestätige ich, dass ich über fachübliche Zitierregeln unterrichtet worden bin und verstanden habe. Die im betroffenen Fachgebiet üblichen Zitiervorschriften sind eingehalten worden. Eine Überprüfung der Arbeit auf Plagiatemithilfe elektronischer Hilfsmittel darf vorgenommen werden.

[redacted]

.

Abstract

Acoumon is an audio-based acoustic monitoring system, using special hardware and deep learning models to detect emergencies of artificially ventilated patients in intensive care, developed at the TU Berlin. The microphone-array device in the patient's room detects sounds to indicate the emergency to the nursing staff, and the model provides predictions based on continual learning. An initial evaluation of the system shows an accuracy of 80% for a binary, balanced dataset. However, it is still uncertain whether the model can be generalized because a common problem in deep learning is that an accurate interpretation of the model is difficult. In this master thesis, feature maps created by visualization techniques will explain the functionality of a model used in a prototype deep learning system for the acoustic monitoring of intensive care patients and analyze whether it can help us better perform classification. Visualization techniques for neural networks provide a visual explanation of the decisions of a machine learning model. The areas in the feature maps have the most significant contribution to the model decisions by highlighting. These four visualization methods, Vanilla Saliency, Smooth-Grad Grad-CAM, and fast Score-CAM, are the most commonly used in deep learning at the moment. Therefore, they will be compared and evaluated to analyze whether they can provide information about how this specific task is performed and on what basis the networks make decisions. Specifically, layers of a convolutional network will visualize to understand better its function and how it contributes to the network's final decision. This thesis aims to find the most suitable visualization method for the model used in the Acoumon project and find the most useful, most crucial time-frequency information or patterns for this task. In the end, both CAM-based visualization methods performed very strongly. Among them, Fast Score-CAM was the best. The capture time existed mainly for the first 0.5 seconds. The frequency range lived mainly 1-2 kHz, and the pattern around 3.5kHz is additionally marked. The thesis shows that analysis with visualization methods in detecting audio events can be helpful.

Zusammenfassung

Acoumon ist ein an der TU Berlin entwickeltes audiobasiertes akustisches Überwachungssystem, das spezielle Hardware und Deep-Learning-Modelle verwendet, um Notfälle bei künstlich beatmeten Patienten auf der Intensivstation zu erkennen. Das Mikrofon-Array-Gerät erfasst Geräusche im Patientenzimmer, um das Pflegepersonal auf den Notfall hinzuweisen, und das Modell liefert Vorhersagen, die auf kontinuierlichem Lernen basieren. Eine erste Auswertung des Systems zeigt eine Genauigkeit von 80% für einen binären, ausgeglichenen Datensatz. Es ist jedoch noch ungewiss, ob das Modell verallgemeinert werden kann, denn ein häufiges Problem beim Deep Learning ist, dass eine genaue Interpretation des Modells schwierig ist. In dieser Masterarbeit werden mit Hilfe von Visualisierungstechniken erstellte Feature-Maps die Funktionsweise eines Modells erklären, das in einem Prototyp eines Deep-Learning-Systems für die akustische Überwachung von Intensivpatienten verwendet wird, und analysieren, ob es uns helfen kann, eine bessere Klassifizierung durchzuführen. Visualisierungstechniken für neuronale Netze bieten eine visuelle Erklärung der Entscheidungen eines maschinellen Lernmodells. Dabei werden die Bereiche in den Feature-Maps hervorgehoben, die den größten Beitrag zu den Modellentscheidungen leisten. Diese vier Visualisierungsmethoden, Vanilla Saliency, Smooth-Grad Grad-CAM und fast Score-CAM, sind die derzeit am häufigsten verwendeten Methoden im Deep Learning. Daher werden sie verglichen und bewertet, um zu analysieren, ob sie Informationen darüber liefern können, wie diese spezifische Aufgabe ausgeführt wird und auf welcher Grundlage die Netzwerke Entscheidungen treffen. Insbesondere werden die Schichten eines Faltungsnetzwerks visualisiert, um ihre Funktion und ihren Beitrag zur endgültigen Entscheidung des Netzwerks besser zu verstehen. Ziel dieser Arbeit ist es, die am besten geeignete Visualisierungsmethode für das im Acoumon-Projekt verwendete Modell zu finden und die nützlichsten, wichtigsten Zeit-Frequenz-Informationen oder Muster für diese Aufgabe zu finden. Am Ende schnitten beide CAM-basierten Visualisierungsmethoden sehr gut ab, wobei sich Fast Score-CAM als die bessere Methode herausstellte. Die Erfassungszeit bestand hauptsächlich aus den ersten 0,5 Sekunden. Der Frequenzbereich lag hauptsächlich bei 1-2 kHz, und das Muster um 3,5kHz ist zusätzlich markiert. Die Arbeit zeigt, dass die Analyse mit Visualisierungsmethoden bei der Erkennung von Audioereignissen hilfreich sein kann.

Contents

List of Figures

1 Introduction

1.1 Motivation

Machine learning has been a hot topic in recent years. It was always used and had many applications, such as related research on audio events detection and audio evaluation. An important reason for that is that neural networks' performance has improved. With the help of neural networks, many classification and detection tasks can accomplish more efficiently and without manual processing. Generally, an excellent performance could achieve through neural networks, but there is still a challenge that it is always challenging to explain why does the model make such a classification. In other words, what is the proof for the model to make this classification? There are many parameters in the construction of neural network models, these parameters gradually change with the training and learning of the model, but the process of this change is not transparent. So it is difficult to understand the model, and the lack of understanding also leads to uncertainty about whether the model could be generalized. The model can perform better if its reliability can explain. The approach to increase the model's interpretability explores not thoroughly in the audio field, in fact, many convolutional neural networks have been developing in computer vision, and visualization technologies have achieved outstanding performance in image classification.

In this master thesis, many different visualization methods which are used in image classification will apply to acoustic monitoring. These methods will be compared in the actual model and explored which method can better help to improve the interpretability of audio classification. This master thesis selects the most frequently used visualization methods:

Vanilla Saliency, SmoothGrad, Grad-CAM++, and Fast Score-CAM. Detailed comparison and analysis performed in the model task of acoustic monitoring to explore whether these four visualization methods can help the model improve its performance and interpretability and find the most valuable, most critical time-frequency information or patterns for this project.

1.2 Approach and Goals

To better understand why the model made such a classification, the feature map extracted from the middle layer of the neural network needs to be displayed. Different visualization methods use different approaches to remove the feature map. This thesis mainly selects four visualization methods: Vanilla Saliency, SmoothGrad, Grad-CAM++, and Score-CAM. Vanilla Saliency. The first method is Vanilla Saliency, which shows which pixels of a given image contribute the most to the network's decision by estimating the gradients of the class score function at the input [19]. SmoothGrad was chosen as the second method, which attempts to minimize the noise in comparison to Vanilla Saliency by creating somewhat noisy versions of the input picture and averaging the Vanilla Saliency maps of that noisy images [22]. Grad-CAM++, which has the same goal as Vanilla Saliency and SmoothGrad but generates a coarser heatmap from the activation maps of the last convolutional layer by weighting them with coefficients calculated from the estimated gradients of the class score function, is the third visualization technique investigated [1]. The last approach, Faster Score-CAM, is similar to Grad-CAM++, but it calculates the activation map weights differently, without relying on gradients. Instead, it tries to figure out how much each activation map's areas contribute to the total class score [24]. Score-CAM Faster is a variation of the original Score-CAM that can calculate significantly more quickly. All four approaches were chosen because they are commonly used.

The Acoumon project in this master thesis investigated uses equipment placed in the room to detect the patient's condition by recording the patient's voice and transmitting the result to the nurse in time to handle it in time. In this equipment, a convolutional neural network

(CNN) was built in, trained on sound data recorded in inpatient care institutions, and categorized by experienced nursing personnel to recognize this category of noises. Data were obtained from a range of patients in various geographical and auditory circumstances to ensure the model's generalizability. In the end, the trained model must distinguish between critical and non-critical noises. In this thesis, the trained CNN model and datasets will use together to evaluate these four visualization methods.

Besides, many evaluation methods will be used to verify the faithfulness of the visualization method. Average drop and average increase [1] will calculate, which indicates how well visualization tools can highlight only the parts that are most relevant in the decision-making process. Deletion and Insertion Curves [17] are also calculated. This is accomplished by incrementally adding or subtracting pixels from the input picture that the visualization methods identified as the next most critical ones while observing the model's performance. A qualitative method is carried out in terms of Visual Coherence, Class Discriminativity, and Interpretability [22]. The core idea of visual coherence is as significant regarded areas are only on the audio events of the target class and not on other portions of the spectrogram. When many classes of audio events are present, Class Discriminativity means that only the audio events of the target class are noted, and no audio events from other classes are marked. The last qualitative feature is interpretability, which will assess whether the introspection approaches help understand which patterns the network learns and help identify plausible reasons for classification.

1.3 Overview and related work

Deep learning models have recently emerged as powerful alternatives for classification, the field of audio classification is also no exception. Notable examples include [21], where a deep feed-forward network learns to estimate an ideal binary spectrogram mask that represents the spectrogram bins in which the vocal is more prominent than the accompaniment. In [7], the authors employ a deep recurrent architecture to predict soft masks that are multiplied with the original signal to obtain the desired isolated source [8]. The regular operation used in this literature is that the audio signal will always be cut into short time frames slices to perform better, then through the Short-Time Fourier Transformation(STFT) transfer transform to a two-dimensional signal in the frequency domain. It is worth noting that to better analyze the signal from the perspective of the human ear, the converted signal additionally needs to be adjusted using a Mel frequency band scaling [3]. The spectrogram after such processing can be used as the input signal of the convolutional neural network. After processing the above, the audio signal is no longer one-dimensional but a two-dimensional picture signal. The visualization methods often used for image classification seem to be used for audio classification in this case. Despite this, there are still many differences between typical images and audio spectrograms. It mentions that many audio events are calculated using the add-up when they occur at the same time [10], but the object in the typical image is independent. Additionally, the object in the image is generally of a specific size, and the pixels are very concentrated. But the distribution of the captured audio events in the spectrogram may be scattered and not focused. Besides, the abscissa and ordinate of the spectrogram are not in one dimension, one represents the time, and the other represents the frequency, but the length and width units in the image are the same.

The task selected in this thesis is a system based on acoustic monitoring, which can detect an exceptional audio event. Auditory scene analysis and audio event detection are a subset of acoustic monitoring. The majority of publications in this area use CNNs [9] for detecting human, and machine sounds or Convolutional Recurrent Neural Networks (CRNNs) [15] for tasks with a significant temporal correlation between the samples and offering reliable predictions even for datasets with limited labeling. Similar methodologies have been employed for acoustic monitoring for animal species identification [26] or industrial sounds such as rail condition monitoring [27] [4]. There are only a few reports on audio-based medicinal applications. The use of transformer networks and RNNs in a recent study aimed at detecting COVID19 from voice [18] yielded encouraging results. Because of the importance of this topic, comparable COVID19-related studies have developed in recent years, employing RNNs to accurately identify COVID19 from fundamental acoustic parameters such as MFCCs produced from acoustic signals such as coughing or breathing [6]. The acoustic monitoring system used in Acoumon project is similar to that of [16] because the signals, in this case, are a subset of breathing noise to detect many types of abnormal breathing signals.

A widespread problem within machine learning is the lack of large training datasets. It takes much time and human resources to collect data and manually label, so there is a problem of lousy generalization ability. Although the data of the prediction results are impressive, it still cannot 100% prove that the model is reliable. In this aspect, visualization technology can be beneficial. It is robust to adversarial perturbations[1] and, therefore, more faithful to the underlying model and helps achieve model generalization by identifying dataset bias.

There is already a lot of literature available on visualization techniques. Such as [19] have visualized CNN predictions by highlighting ‘important’ pixels (i.e., change in intensities of those pixels which have the most impact on the prediction score). Specifically, Simonyan et al. [20] visualize partial derivatives of predicted class scores w.r.t. pixel intensities, while Guided Backpropagation and Deconvolution make modifications to ‘raw’ gradients that result in qualitative improvements. Despite producing fine-grained visualizations, these methods are not class-discriminative. Other visualization methods synthesize images to activate a network unit or invert a latent representation maximally. Although these can be high-resolution and class-discriminative, they are not specific to a single input image and visualize a model overall. There are more details about the theory of visualization methods in chapter 3.

As the literature review here shows, there are studies researching CNNs used in the field of audio signal processing [7] [8], as well as many articles comparing and summarizing visualization technology [19] [19]. However, no research applies visualization technology to acoustic monitoring, lacking in data sets, and has not been explored much. To the best of our knowledge, it attempts to address this research gap in this thesis.

The presentation of this thesis is split into five sections. The methods of this thesis are described in Chapter 2; this includes the introduction of datasets, the architecture of the

[1] An adversarial example is created by performing worst-case perturbation on the input of a machine learning model.

neural convolutional network used in this thesis, the mathematical foundation of the visualization approaches, and the idea of introspection methods used to evaluate four visualization options. Results and analysis are given in Chapter 3, a discussion and conclusion are reached in Chapter 4. Chapter 5 is an appendix containing many important figures of the results.

2 Methods

This chapter will describe all methods used throughout the thesis, including database collection, the building of convolutional neural network and the training process, the introduction of visualization methods, and evaluation metrics used to analyze visualization results.

2.1 Datasets

Audio Data was recorded in 4 different care homes and ten patients, the whole recording process continued over 12 months, and the memory size is over 10 TB. In addition to the audio recording, there is also a labeled record of the patient's start and end time points of the audio events. According to the detailed written documents, many audio slices were extracted with detailed annotation. It included ambient sounds from the patients' rooms, sounds from the patients, and the raw datasets of different soundscapes were composed of these audio slices.

Each sample in raw datasets was cut to 1 s long to create a balanced dataset. It is also a trade-off between retaining enough acoustic information and ensuring fast processing. Each sample has a precise notation with positive or negative, and it helps audio processing in supervised learning of succedent model training. However, the input to the model cannot be a one-dimensional audio signal, so the audio signal is first converted into a two-dimensional spectral signal through the Short-Time Fourier Transform(STFT), and the Mel scale is added to explore its effect on the human ear more intuitively. The final obtained Mal-spectrogram with the previously marked labels of each audio signal will be used as the input to the model for learning. No further preprocessing was applied.

2.2 Audio Processing

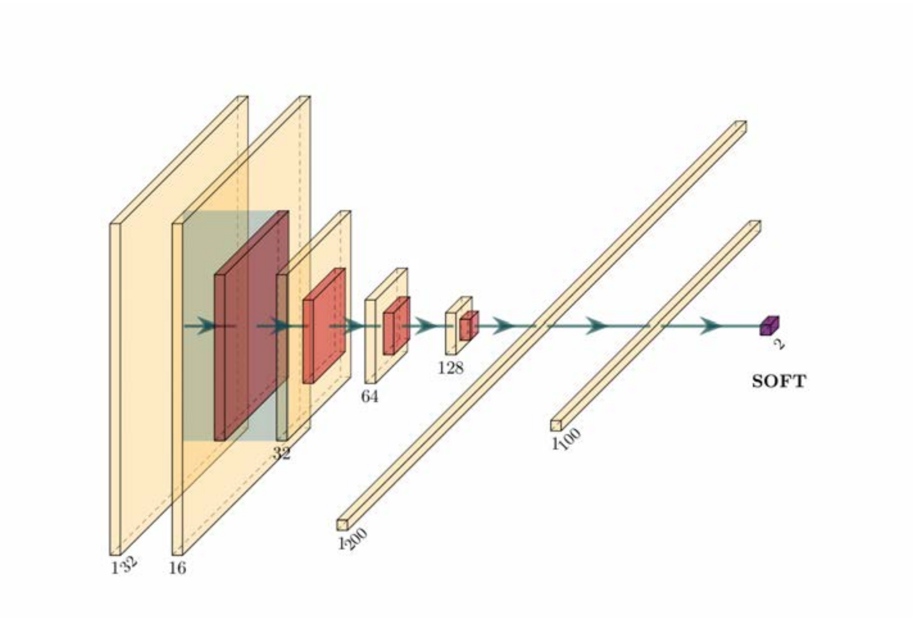


Figure 2.1 Architecture of the used convolutional neural network.
[13]

The processing employed a basic CNN architecture with an increasing number of nodes (16, 32, 64, 128) and filters of size (8, 4) pixels (to accomplish a frequency reduction), followed by two fully connected layers of 200 and 100 nodes, totaling slightly over 1.5M parameters, making for a very lightweight system. After each convolutional layer, batch normalization followed a RELU activation function and a (2, 2) max pooling. Except for the final layer, which employs softmax activation for the two output nodes, there was no dropout, and the activation was RELU for the fully linked layers. Fig. 2.1 depicts the entire construction. The model was trained for a total of 12 epochs at a learning rate of $3 \times 10^{-}5$ until no further change in validation loss could be seen. The 1 s duration was chosen as a compromise between maintaining enough auditory information in the chunk and guaranteeing quick processing. [13]

2.3 Visualizations Methods

The visualizations were created using the open-resource project in Github tf-keras-vis visualization toolbox**??**. All four visualization methods, Vanilla Saliency, SmoothGrad, Grad-CAM++, and Score-CAM, will be introduced in this chapter.

2.3.1 Vanilla Saliency

For a long time, saliency maps have received much attention. They are a common visualization technique for deducing why a deep learning model made a particular choice. Vanilla Gradient is the original saliency map algorithm for supervised deep learning from Simonyan et al. (2013) [19]. Vanilla Saliency calculates the gradients at the input using a local linear approximation of the model, attempting to determine how much each pixel from the input picture contributes to the network's target class output.

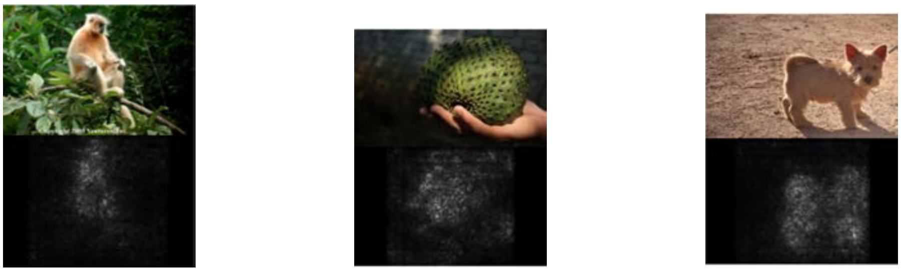


Figure 2.2 Image-specific class saliency maps.
[19]

As mentioned in [19]. The relationship between the class score $S_c(I)$ and the input image I is not a simple linear function. The function can also be infinitely approximated and simulated with the neighborhood of I_0 by the Taylor expansion, w in the equation is the derivative of S_c concerning the image I at the point I_0.

$$S_c(I) \approx w^T I + b \tag{2.1}$$

$$w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0} \tag{2.2}$$

The derivative w is found by back-propagation. One way to think about the magnitude of w is to consider which pixels need to be modified the least to have the most impact on the class score and find the object location in the image by these pixels.

2.3.2 SmoothGrad

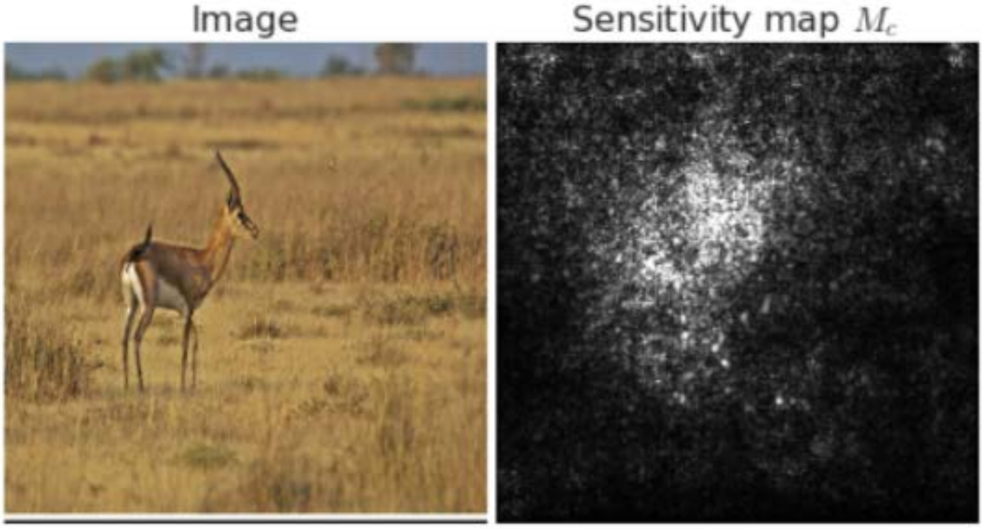


Figure 2.3 A noisy sensitivity map.
[22]

The second visualization method which was used is SmoothGrad. According to the introduction in the previous section, the class score function of Vanilla Saliency $S_c(I)$ is estimated based on the Taylor expansion, which requires the use of a large number of surrounding pixels of I_0. In this technique, sensitivity maps are often visually noisy, which creates difficulty in understanding classification, and results in line with human understanding cannot be given. To solve this problem, SmoothGrad was proposed by

Smilkov [22]. The basic idea is to take an image of interest(as shown on the left in picture 2.3), sample similar images by adding noise to the image(as shown on the right in picture 2.3), then take the average of the resulting sensitivity maps for each sampled image. As shown in picture 2.4, For many inputs, introducing 10%-20% noise seems to produce satisfactory results.

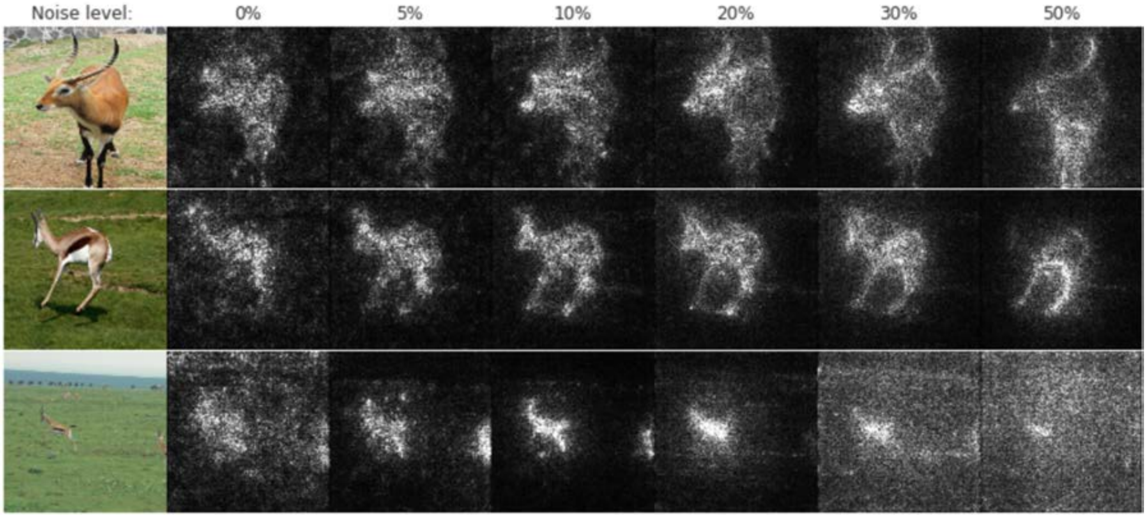


Figure 2.4 Effect of noise level.
[22]

The possible reason for the noise is that the derivative of function S_c has large fluctuations in a small range, so the gradient of S_c at any one pixel is not as significant as a local average of gradient value of this strong fluctuation. By averaging the gradient values, the class score function $S_c(I)$ over-dependence on the surrounding values of I_0 can be reduced. Mathematically, this means calculating

$$\hat{M}_c(I_0) = \frac{1}{n}\sum_{1}^{n} M_c(I_0 + \mathcal{N}(0, \sigma^2)) \tag{2.3}$$

$\mathcal{N}(0, \sigma^2)$ represents Gaussian noise with standard deviation, and n is the number of samples. $\hat{M}_c(I_0)$ is the average of the saliency maps.

2.3.3 Grad-CAM++

Grad-CAM++ generates a visual explanation for the class label under consideration using a weighted mixture of the positive partial derivatives of the final convolutional layer feature maps concerning a specific class score as weights. Compared to class activation mapping (often abbreviated as CAM), Grad-CAM++ better explains target localization and the interpretation of multiple target instances appearing on a single graph [2].

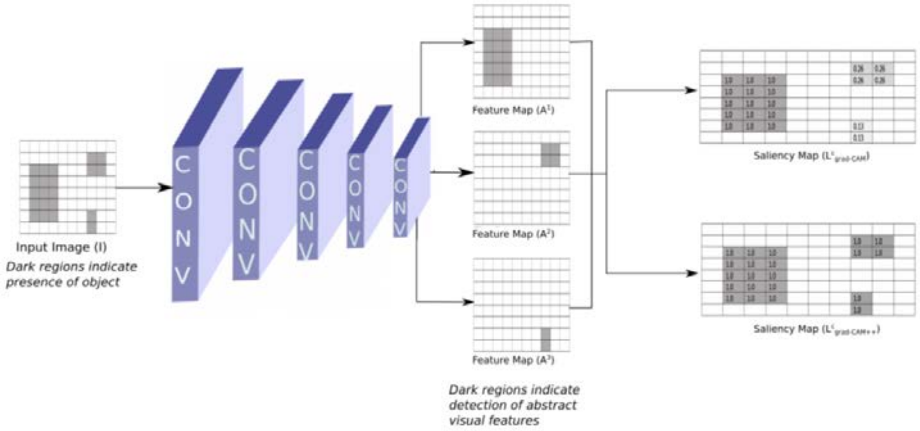

Input Image (I)
Dark regions indicate presence of object
CONV
CONV
CONV
CONV
CONV
Feature Map (A^1)
Feature Map (A^2)
Feature Map (A^3)
Dark regions indicate detection of abstract visual features
Saliency Map ($L^c_{grad-CAM}$)
Saliency Map ($L^c_{grad-CAM++}$)

Figure 2.5 A hypothetical example elucidating the intuition behind grad-CAM++. [2]

Introduced in [1], the central core idea of Grad-CAM++ base on CAM and Grad-CAM, Both CAM and Grad-CAM work on the idea that the final score Y^c of a particular class c can express as a linear combination of the class's global average pooled last convolutional layer feature maps A^k.

$$Y^c = \sum_k w_k^c \cdot \sum_i \sum_j A_{ij}^k \tag{2.4}$$

for each spatial location(i, j), the class-specific saliency map L^c can be calculated as:

$$L_{ij}^c = \sum_k w_k^c \cdot A_{ij}^k \tag{2.5}$$

Like the equation 2.5, L_{ij}^{c} is a visual description of the class predicted by the network since it directly connects with the relevance of a detailed location (i, j) for a specific class c. CAM calculates these weights w_k^c by utilizing the activation maps of the last convolutional layer generated for a given picture to train a linear classifier for each class c. However, this exposes a disadvantage of CAM, which can only be used when the penultimate layer with global average pooling(GAP) is used, and multiple linear classifiers need to be trained for each class.

Grad-CAM was created to solve these problems. In the basic of Eq.2.4, the weights w_k^c can be calculated for a particular feature map A^k and class c as:

$$w_k^c = Z \cdot \frac{\partial Y^c}{\partial A_{ij}^k} \tag{2.6}$$

Z is the number of pixels in the activation map, and it is constant. Grad-CAM allows CAM to generalize to arbitrary CNN models without retraining, making the weight w_c^k independent of the position (i,j). The way to solve this is by taking a global average pooling of the partial derivative $\partial \mathrm{A}_{ij}^k$:

$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \tag{2.7}$$

The shortcomings of Grad-CAM consist in the fact that when an image has several occurrences of the same class objects that cannot be appropriately located in the picture and because of an unweighted average of partial derivatives, the localization does not correspond to the complete object, but only bits or pieces of it. Nevertheless, in the real world, many of the same objects appear in the same image is very common. Complete trust for the human to understand CNN needs credible evidence. Grad-CAM++ improves this.

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} \cdot relu(\frac{\partial Y^c}{\partial A_{ij}^k}) \tag{2.8}$$

The idea behind Grad-CAM++ is to ensure that the weight w_k^c is a weighted average rather than a global average. Thus, the importance of a specific activation map A_k can obtain through the weight. The more detailed mathematical process can be found in [1].

2.3.4 Score-CAM

Equation 2.10 shows the basic core idea of Grad-CAM++, but due to the zero gradient region of the ReLU function, the gradients of networks can be noisy and tend to disappear, for example, the output gradients or internal layer activations concerning the input may be visually noisy. Besides, given two activation maps, the corresponding input region with greater weight is considered more important, but in practice, activation maps with higher weights may contribute less to the network output. Like Fig. 2.6

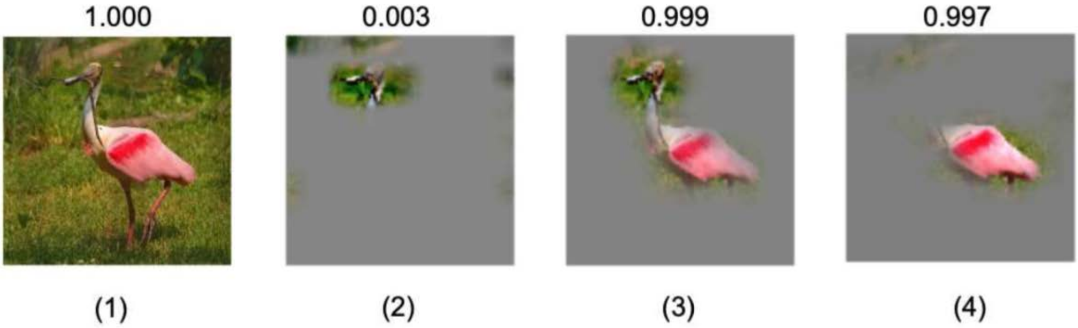


Figure 2.6 Upsampled activation maps with different weights. [25]

This phenomenon may be caused by the operation of global pooling on top of the gradient and the vanishing gradient problem in the network. Score-CAM obtains the weight of each activation map through the forward passing score of each activation map on the target class, thereby removing the dependence on the gradient. It is a solution to the above problems caused by gradients.

In Fig.2.7, phase 1 begins with the extraction of activation maps. Each activation is then used to mask the original picture to calculate the target class's forward-passing score. Phase 2 is repeated N times, with N being the number of activation maps. Finally, a linear combination of score-based weights and activation maps may be used to obtain the saliency

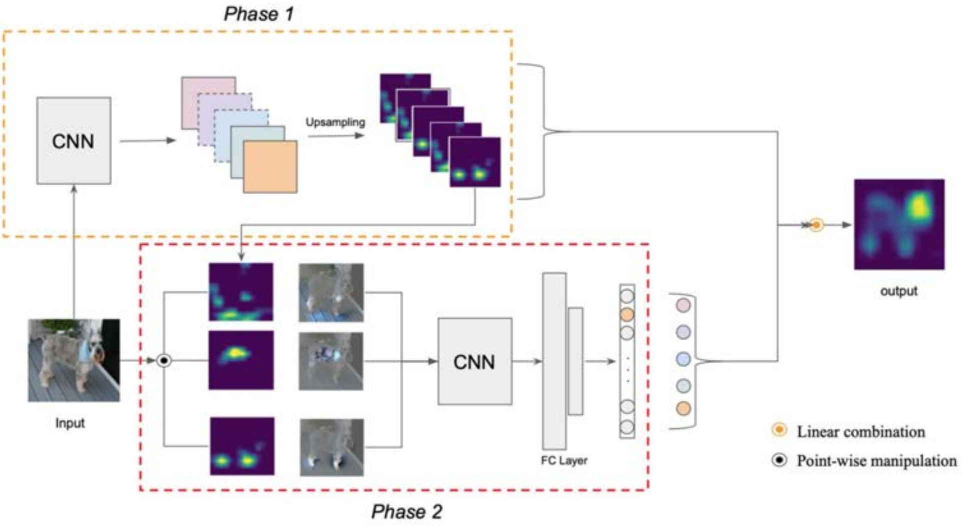


Figure 2.7 Pipeline of score-CAM.
[25]

map. Phase 1 and 2 use the same CNN module as a feature extractor. Because Score-CAM and Grad-CAM++ are based on CAM, the definition equations are also similar. Score-CAM $L^c_{Score-CAM}$ can be defined as

$$L^c_{Score-CAM} = relu(\sum_k \alpha^c_k A^k_l) \tag{2.9}$$

$$\alpha^c_k = C(A^k_l) \tag{2.10}$$

where l is the convolutional layer, f is model, c is the class of interest.

2.4 Evaluation

The comparison and analysis of these four visualization methods more effectively require assessing the faithfulness of the visualization method. The reliability of the visualization methods, the accuracy of the visualization, and the interpretability of the results will be discussed using these evaluation methods. The evaluation methods used in this thesis will be introduced in detail below.

2.4.1 Average Drop and Increase

In [1], two evaluation measures are introduced: *Average Drop* and *Average Increase*. The primary idea behind these two is to see how the predicted probability changes when the input picture is weighted according to the weight obtained after visualization. Average drop and average increase indicate how well visualization tools can highlight only the parts that are most relevant in the decision-making process or even eliminate distracting context [1].

Typically, when parts of an image are removed, the model's predicted probabilities will drop compared to the original input. A more reliable visualization would emphasize the most critical regions for the model's decision-making process, resulting in a minor reduction in predicted probability. The average drop is a measurement of how much the predicted probability decreases and is calculated using the formula 2.11 [11].

$$Average\ Drop(\%) = \frac{100}{N}\sum_{1}^{N}\frac{\max(0, Y_i^c - O_i^c)}{Y_i^c} \tag{2.11}$$

where Y_i^c is the predicted probability of class c on the i^{th} spectrogram, and O_i^c is the predicted probability of class c with the explanation map as input. N denotes the number of spectrograms.

The opposite is the average increase. When the explanation map is employed, it is also likely that the visualization approaches can eliminate distracting context from the original input. If the visualization methods are reliable, the areas critical to the model's decision-making process will be marked, and as unnecessary pixels are deleted, the prediction probability will rise. The equation for calculating average increase is expressed as:

$$Average\ Increase(\%) = \frac{100}{N}\sum_{1}^{N}\mathbb{1}_{Y_i^c < O_i^c} \tag{2.12}$$

where $\mathbb{1}_x$ is a true-or-false indicator function which will returns 1 when the argument is true. Y_i^c and O_i^c are same with the equation 2.11.

2.4.2 Deletion and Insertion Curves

Deletion Curve and Insertion Curve were introduced in [17]. The principle of these two metrics is to understand a visualization as a significant weighting of the pixels from the original input and to track the model output as the most critical highlighted pixels are deleted from the original input or added to a plain or blurred input by step. A similar idea was also mentioned in [23]: Explanation by Preservation and Explanation by Deletion. Explanation by Preservation means that the smallest region of the input that must be preserved to preserve the original model output would be explored. Explanation by Deletion means that the smallest section of the input that needs to be eliminated to affect the model output would be considered. Similar ideas have been used in the comparison of visualization methods in some literature [22] [1] [17].

Deletion Curve and Insertion Curve were chosen to analyze four visualization approaches. Deletion curves demonstrate how the output updates when the highest weighted pixels in the original spectrogram are eliminated one by one and set to zero [17]. The spectrogram is pointwise multiplied using a mask that sets all eliminated pixels to zero and multiplies all remaining pixels by one to achieve the deletion. In this thesis, the existing highest weighted 1% of the pixels are deleted from the 32x32 sized spectrograms at each stage. It would be deemed preferable if the accuracy or predicted probability in a deletion curve fell quicker since the visualization approach discovers sections that are more crucial for the model's correct predictions, and the area can replace an estimate of the downward trend under the curve. Insertion curves work similarly to deletion curves, except that the highest weighted pixels are placed into a clear or blurred image step by step, and the original picture comprises zeros [17]. This thesis adds the highest weighted 1% of the pixels from the 32x32 spectrograms at each stage. It will be preferable if the accuracy or predicted probability in an insertion curve rises quicker since the visualization approach discovered areas that help the network accurately classify.

2.4.3 Qualitative Methods

This thesis chose visual coherence, interpretability, and class discriminativity for evaluation. These three methods came up in [22]. The core idea of visual coherence is as significant regarded areas are only on the audio events of interest and not on other portions of the spectrogram; in other words, it must be determined how well the produced visualizations cover the input spectrogram's energy distribution. Because there is a lot of background noise in negative samples, the only positive class will investigate how closely the visualizations mirror the spectrogram's energy distribution.

As evident by the naming, interpretability means whether visualization approaches help understand the patterns the network learns. One input of the network will be shown for each class. It will be observed how effective visualization tools are in discovering reasons for classification.

If many audio events simultaneously are present, only the audio events that want to be found are marked; it is class discriminated. As an input to the network, layered inputs of two classes will be used for four visualization methods, the spectrograms before layering will be compared to the visualizations for the classes that were used to produce the layered input to assess how effectively they convey just the audio events of the class in concern.

3 Results and Analysis

The results are reported in the following sections: first, a concrete implementation of the visualization method in code, then the results of the CNN training, and the results of the evaluation of the visualization approach. Finally, a table summarizes each of the four visualization methods.

3.1 Implementation of Visualizations

A random selection of 200 samples was taken from the test set and used to plot the graphs and calculate the relevant parameters. Among them, 100 were negative, and 100 were positive.

In the Generation of SmoothGrad, as shown in Fig.2.4, when the added noise is 0, SmoothGrad can be seen as the same as the effect of Vanilla Saliency, and for picture identification tasks, a noise level of 20% seems to perform best. Because alternative noise levels did not show a significant improvement in testing, this noise level was also picked in this thesis. SmoothGrad was programmed with a sample size of 20.

In the Generation of Score-CAM, because Score-CAM is computationally costly, a version was provided in the visualization toolkit $tfkerasvis$ that only computes for channels with significant variations, faster Score-CAM is the name of this implementation. A preliminary test of this experiment discovered that computing all channels did not affect the visualization and its consistency compared to merely calculating the ten channels with the most significant variances. As a result, for this thesis, Faster Score-CAM was selected.

3.2 CNN Performance

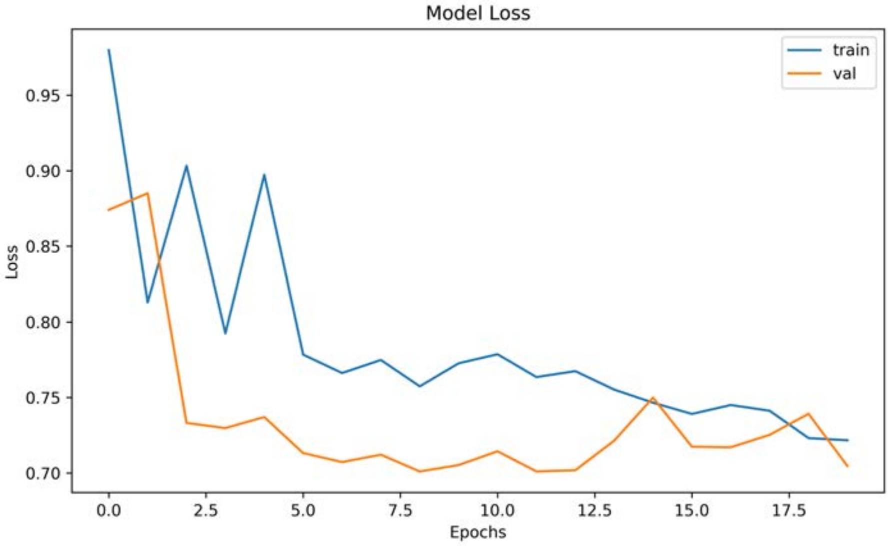


Figure 3.1 Loss function.

Fig.3.1 shows the development of training loss over the epochs. The network generally converges until epoch 10, after which the training loss stayed almost the same. The validation loss has a slight rebound when the epoch is greater than 10, there will be an overfitting phenomenon, and the model used in this thesis is described in [14], the epoch is 10. The final values of training were 0.705. The network accuracy on the training set was 80.2% on the validation set, and a recall of 78.7% and precision of 80.1% was achieved, which shows that the network can also classify audio that was not used during the training process. Fig.3.2 shows the confusion matrix of the test set, each column of the matrix represents a predicted class, while each row represents an actual class. It is so named because this matrix makes it easy to see if the machine is confusing two different classes. In this confusion matrix, There are 297 negative samples, but the system predicts 53 of them as positive samples, and the recognition rate for negative samples after normalization is equivalent to 78.2%; for 300 Positive Samples, 84 of them are predicted as negative samples, after normalization the recognition rate equivalent to positive samples is 71.5%.

The training results are outstanding. According to the accuracy in the validation set, the

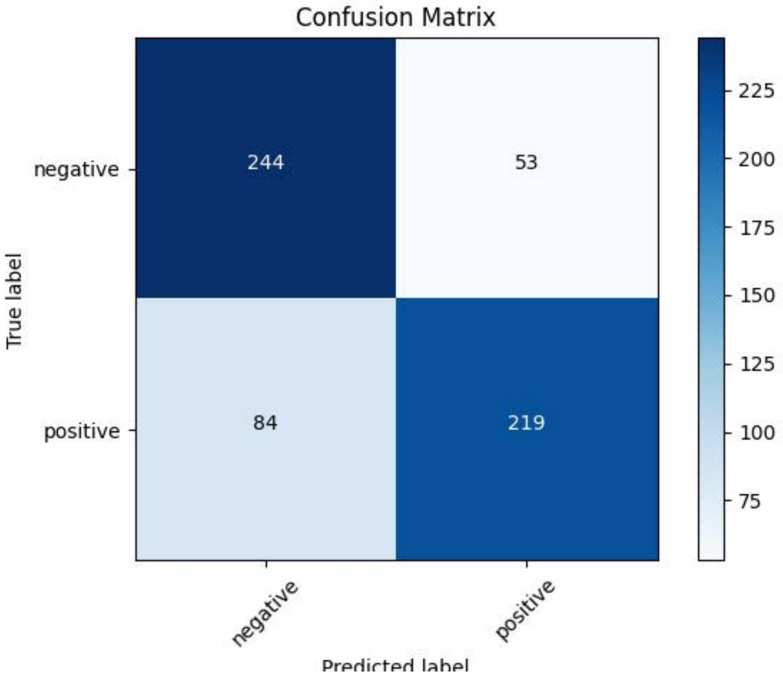


Figure 3.2 Confusion matrix.

model can also generalize well. The loss function within ten epochs shows that the model is not overfitting. It can be seen in the confusion matrix that the most common misclassification is to misclassify positives as negatives, which can be explainable because, in many cases, the wheezing sound is difficult to define as dangerous or not dangerous. The negative class can contain many different spectrograms because it contains human voice and, the sound of machine operation, background noise. So the accuracy of the network for positives might be due to its very distinctive feature maps. Anything inconsistent with it is possibly judged as negative. However, it is still unknown how the network decides when it can decide well, and the visualization methods can show valuable something. It is helpful to understand why the network performs well.

3.3 Average Drop and Average Increase

It can be seen from Equation 2.11 that if the explanation map is identical to the original input, namely $Y_i^c = O_i^c$, this means there is no change between original predicted probabilities and examination map added predicted probabilities. The Average Drop will stay at 0%. Generally, if a visualization has high values, it may be more robust. However, this is not a good sign for assessing the effectiveness of visualization approaches to identify the sections of the essential inputs for the network's decision because it might create an explanation map similar to the original input. Table 3.1 reports the mean average of the weights of the 50 percent highest weighted pixels in the normalized visualizations, and only the top 50% of the pixels are utilized. It can be seen that Grad-CAM++ has the lowest mean average of the four visualization approaches. SmoothGrad and Faster Score-CAM are pretty similar, and both are over 0.2. The visualization technique SmoothGrad has the most excellent mean average of the four. Table 3.1 showing all visualization approaches have found some critical areas for the decision-making process of the model, and the original input and explanation map is already not the same so that an apparent heat map for each visualization could be obtained.

Methods	Mean Average of Weights
Vanilla Saliency	0.14729
SmoothGrad	0.28350
Grad-CAM++	0.04036
Faster Score-CAM	0.22157

Table 3.1 Mean Average of the weights of the top 50% weighted pixels.

When sections of the picture are eliminated, the model's projected probability frequently decreases compared to the original input. A more accurate visualization would emphasize the most critical regions for the model's decision-making process, resulting in a smaller average drop in predicted probability. Because visualization methods may eliminate distracting information from the original input, the model's predicted probability rises when the explanation map is applied. However, these methods which use explanation maps were not employed in this case since it was unclear which samples should be deleted, and it is a time-consuming operation that could be done in the future. Following this principle, a more reliable visualization method is predicted to generate a higher average increase than a less reliable method. The changes of prediction are the same as the average drop. Fewer changes mean less influence on the examination map, which brings to the predicted probability.

The results of the measurement of the average drop, the average increase, and the percentage of the changed probabilities after the explanation map added are shown in table 3.2. The entire range of average drop may be shown to be between 7.48% and 17.33%, with a difference of 9.85% between the highest and lowest values. The minimum value is attained while Faster Score-CAM was utilized, which is 7.48%. The AD values of other visualization methods are very similar, with a gap of nearly 10% from the minimum value.

In the average increase column values, the maximum value is also reached when Faster Score-CAM is used, which is 78%. The minimum value is reached in Grad-CAM++ at 62%. There is a 16% difference between the maximum and minimum values. Interestingly, the AI values of the other three visualization methods except Faster Score-CAM are also very similar, which is very similar to the distribution of AD values. Understandably, Vanilla Saliency is similar to SmoothGrad, because the working principle of SmoothGrad is based on Saliency, and even the SmoothGrad of 0 with added noise can be directly regarded as Vanilla Saliency. However, the performance of Grad-CAM++ and Faster Score-CAM, which both are also based on CAM, is very different. It is speculated here that Faster Score-CAM does not introduce gradient calculation to reduce the influence of gradient, which will be discussed in detail in subsequent discussions.

In the Prediction Changes column, the values of Vanilla Saliency and SmoothGrad are very close, about 30%, the maximum value is Grad-CAM++, which is 37%, and the minimum value is Faster Score-CAM, which is 14.5%. As can be seen from Table 3.1, the reason for the poor performance of Grad-CAM++ may be that its mean average of weights is meager, only 0.04, which is very far from other visualization methods. Combined with the insertion curve of Grad-CAM++ as in Figure 3.3, the accuracy rose sharply after 30% pixel points were added. The speculated reason is that Grad-CAM++ only finds a tiny number of effective pixels and therefore pulls down the average weight when the current 50% of pixel points are averaged. The performance of SmoothGrad with the highest mean average is far less than Faster Score-CAM. To sum up, it is easy to see that Faster Score-CAM achieves the best performance.

In the code implementation of this parameter, the method in [11] is used that only the pixels whose weights are ranked in the top 50% are used for measurement. The aim is to avoid the AD and AI values of visualization methods with high average weights being excellent. At the same time, as shown in the table 3.1, all the mean average of weights are not 0, which will not result in an explanation map that is exactly as same as the input image. However, this value does not directly indicate the quality of the visualization method. It still needs to be observed together with other parameters to compare.

Methods	Average Drop	Average Increase	Prediction Changes
Vanilla Saliency	17.33%	63.50%	29.5%
SmoothGrad	16.53%	65.50%	29%
Grad-CAM++	17.33%	62.00%	37%
Faster Score-CAM	**7.48%**	**78.00%**	**14.5%**

Table 3.2 Average Drop, Average Increase and Prediction Change.

3.4 Insertion and Deletion Curves

3.4.1 with Accuracy

Figure 3.3 shows the accuracy insertion curve, the principle of this curve is to increase the pixels in the picture from 0 until all the pixels are ultimately added. If the curve is steep, there are more pixels found by the visualization methods, which has a significant impact on the output. The insertion curves begin with approximately 55 percent accuracy. When all pixels are added, the accuracy is consistent with the model accuracy.

It is evident from the picture that the four visualization methods almost start at the same point and end at the same point, but the trends in the middle are very different. After about

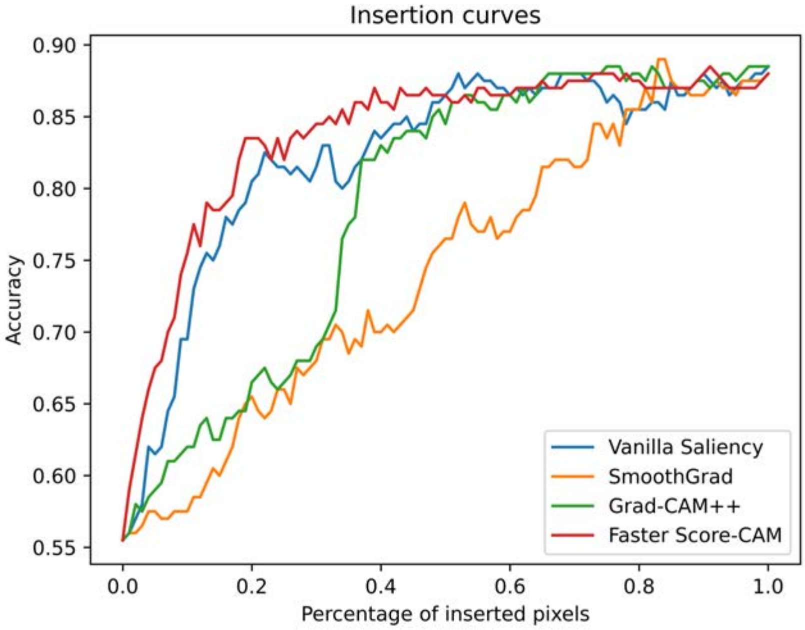


Figure 3.3 Accuracy insertion curve.

35% of the pixels had been added, Fast Score-CAM already got an 85% accuracy, in the subsequent increase of the pixel interval, the accuracy only increased by 2%, and finally stayed at 87%. The overall trend of Vanilla Saliency is similar to that of Faster Score-CAM. After about 45% of the pixels are added, the accuracy has reached 85%. After that, as the pixels continue to be added, the accuracy rate is only improved by about 2%, which is a helpful note that the accuracy drops to around 77% after 70% of the pixels are added, and then back to 87% after 85% of the pixels are added.

From the overall trend, SmoothGrad shows a linear upward trend. After 50% of the pixels are added, the accuracy rate reaches about 75%, which is a middle value. The curve change of Grad-Camm++ is exciting. Its trend in the first 30% of pixels is almost exactly as same as SmoothGrad, but there is an almost linear growth between 30% and 40%, and the trend after 40% is similar to that of Faster Score-CAM, almost the same.

The results of the deletion curves generation are shown in Fig.3.4. Because the visualization approach indicated parts of the input on which the network's proper prediction relied,

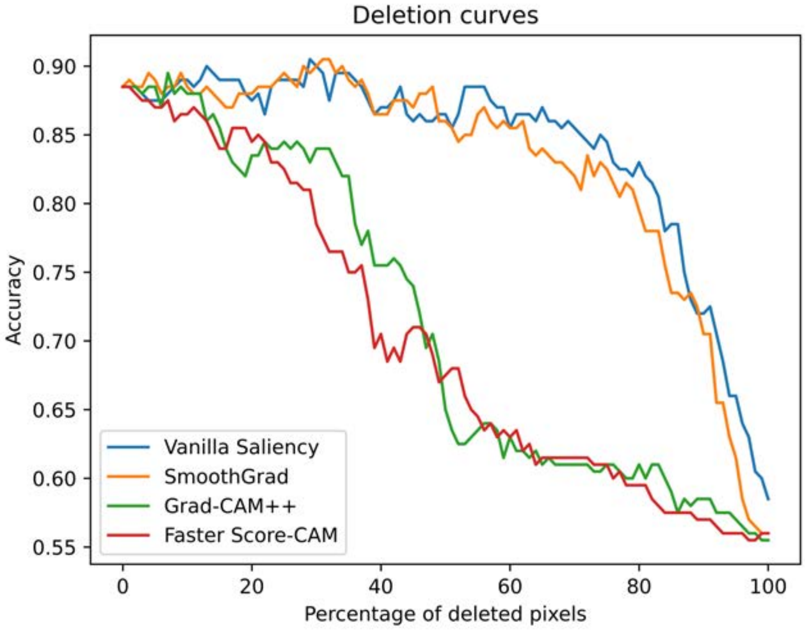


Figure 3.4 Accuracy deletion curve.

a deletion curve that falls quicker is regarded more truthful. The deletion curves begin with the original inputs' accuracy and finish with a 55 percent accuracy after all pixels are eliminated.

In Deletion Curves, the general trend of the two CAM-based methods, namely Grad-CAM++ and Faster Score-CAM, is very similar, after the first 50% of the pixels are removed, the accuracy has dropped by about 70%, but Faster Score-CAM has a slightly steeper downward trend than Grad-CAM++, and the downward curve becomes slower in the second half of more than 50%. The obvious contrast is with SmoothGrad and Vanilla Saliency. The overall trend of these two methods is very similar, but SmoothGrad has a more severe downward trend than Vanilla Saliency in the second half of the development. Both methods have a very slow decline in the first 80% of pixels removed, from about 87% to 80%, but a very steep decline between 80% and 100%, from 80% to 55%.

In addition, the AUC values of the two graphs are listed in Table 3.3. AUC means the area under the curve, which can be used to estimate the trend and steepness of the curve. When

Methods	AUC of Deletion Curve	AUC of Insertion Curve
Vanilla Saliency	0.842	0.821
SmoothGrad	0.830	0.742
Grad-CAM++	0.712	0.787
Faster Score-CAM	**0.701**	**0.837**

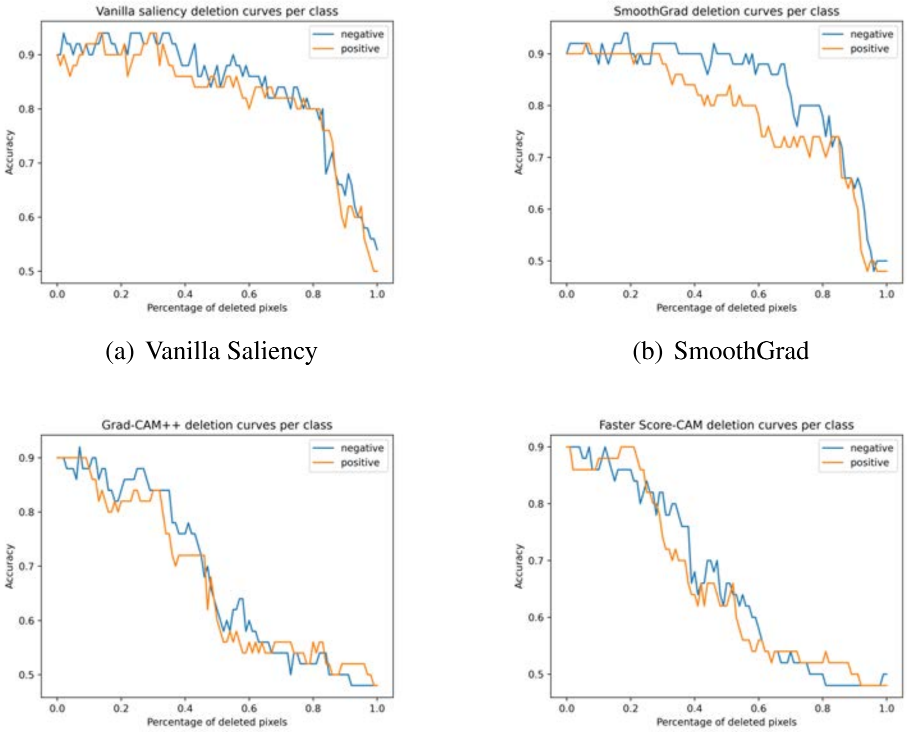
Table 3.3 Area under curve.

the value of AUC is more significant, it means that the curve rises or falls faster. If AUC in the deletion curve is small and the insertion curve is large, it can be regarded as the more reliable model. As can be seen from the table3.3, the overall ranking of AUC is consistent with the ranking obtained by reading the chart. In the Deletion curve, the values of Vanilla Saliency and SmoothGrad are relatively close, the values of Grad-CAM++ and Faster Score-CAM are relatively close, and the minimum value is 0.701 when Faster Score-CAM is used. In Insertion Curve, the maximum value is reached at Faster Score-CAM, and the minimum value is at SmoothGrad. In the early stage of Grad-CAM++, the slope of Grad-CAM++ is consistent with that of SmoothGrad. After reaching a certain pixel point in the later stage, it rises in a straight line, and the trend is similar to that of Faster Score-CAM, resulting in a larger AUC than SmoothGrad.

To sum up, in the Accuracy insertion curve, Faster Score-CAM is the best, followed by Vanilla Saliency, and SmoothGrad has the worst performance because in this case, the linear increase is equivalent to that visualization method does not work. Faster Score-CAM also has the best performance in the Accuracy Deletion curve, followed by Grad-CAM++, and both perform very similarly. In contrast, two Saliency map-based visualization methods, namely Vanilla Saliency and SmoothGrad, perform very similarly, and both perform poorly. It is worth noting that the insertion curve and deletion curve do not reflect the misclassification cases, so the probability deletion curve and probability insertion curve will also be shown later.

3.4.2 per Class

In order to better compare these four visualization methods, in this figure, different samples from the previous section are selected. It is worth observing whether the performance ranking of visualization methods will remain the same when the samples change.



(a) Vanilla Saliency

(b) SmoothGrad

(c) Grad-CAM++

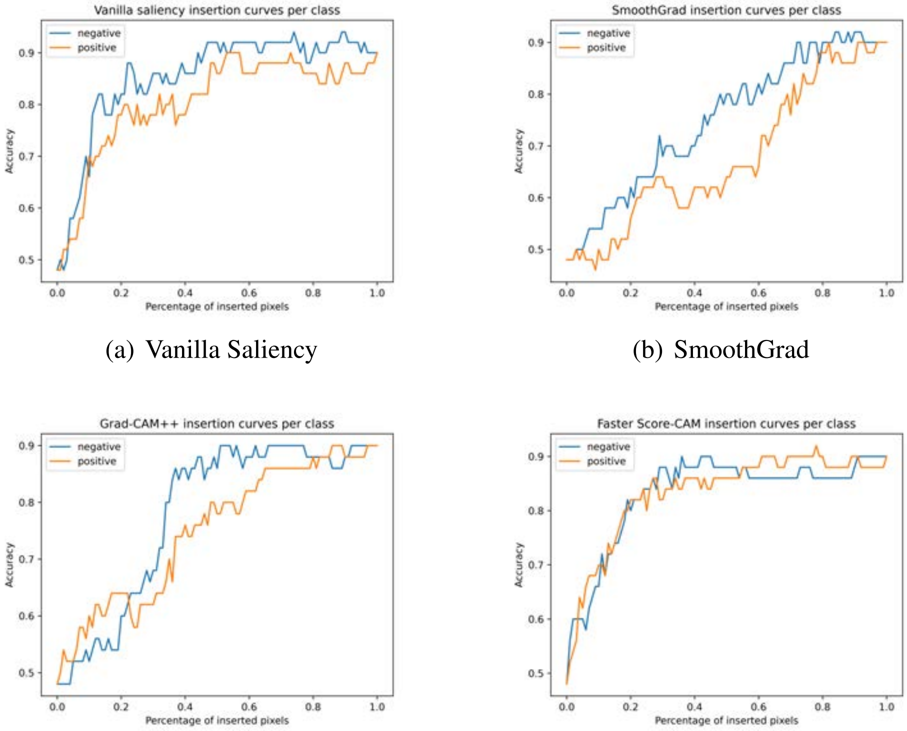
(d) Faster Score-CAM

Figure 3.5 Deletion curve per class-wise.

As shown in Figure 3.5, four different visualization methods are displayed in four small graphs, and classes draw curves. The selected part of the data set was used as input to the neural network for training, so in all visualization methods, the accuracy rates of positive and negative were very high at the beginning, about 90%. After all, pixels are removed, the accuracy rate ends at about 50%, which is the average probability of the two-class model. The trend curves of two Saliency-based methods, namely Vanilla Saliency and SamoothGrad, are very similar. In Vanilla Saliency's picture, the positive curve is slightly lower than the negative curve, but the overall trend is not much different, so it is described together. It has a prolonged decline in the first 80%, only from 90% to 80%, and after more than 80%, accuracy has a sharp decline, from 80% directly to the lowest point of 50%. The negative curve of SmoothGrad is very similar to it. It also starts to drop sharply after about 80% of the pixels are removed, but the inflection point of the positive curve appears earlier. Interestingly, from the beginning to the first 30% of the pixels have been removed, the positive accuracy has remained at 90%, it is evident that the first 30% of the pixels do not play an essential role in the classification the model. After 30% of the pixels are removed, the positive curve slowly declines. This slow decline continues from 30% pixels to 80% pixels, and the accuracy is directly reduced by half. After 80% pixels are removed, there is a sharp drop. Accuracy is directly from 70% to 50%. In other words, the removal of the last 20% of pixels results in an over 20% drop.

The two CAM-based visualization methods are very close. The accuracy of both methods has almost dropped to 50% after 60% pixel removal. It can be seen from the picture of Grad-CAM++ that the negative curve and the positive curve are very similar. As the pixel are removed, the curve gradually decreases, especially between 40%-60%. After 60% of the pixels are removed, the accuracy change is no longer noticeable. The trend of Faster Score-CAM is roughly the same as that of Grad-CAM++, but its decline has been relatively average. It is worth noting that after 20% of the pixels are removed, the positive accuracy does not drop but rises. It shows that the area here cannot help the model identify the positive sample. It hinders good performance. At the beginning of the curve, the overall positive curve is 2% higher than the negative curve. In the middle of the curve, the negative curve exceeds the positive curve. After 80% of the pixels are removed, the negative curve firstly reaches the lowest point of 50%, and the positive curve reaches the lowest point after 90% of the pixels are removed.

Figure 3.6 shows the insertion curves for the four visualization methods, starting at about 45% accuracy and ending at 90% for the four visualization methods. Only the two curves in Faster Score-CAM are almost indistinguishable. The positive and negative curves of other visualization methods are pretty different. It can be seen from the Vanilla Saliency graph that after the first 20% of the pixels are inserted, the growth trends of the positive



(a) Vanilla Saliency

(b) SmoothGrad

(c) Grad-CAM++

(d) Faster Score-CAM

Figure 3.6 Insertion curve per class-wise.

and negative curves are very similar, with a significant increase from 50% to 75%. After 20%, the curve grows slowly until all pixels have been added, then the accuracy reaches 90%. When the added pixels are the same, the negative has higher accuracy than the positive. In other words, the negative curve reaches a higher accuracy earlier than the positive after 20% of the pixels are inserted, this is very similar to SmoothGrad, as shown in the SmoothGrad figure when the first 30% pixels are inserted, the overall trend of negatives and positives is consistent, but the growth rate of positives is lower than that of negatives. After 40% of the pixels are added, the negative is still in a steady upward phase until all the pixels are added. On the contrary, positive, there is a drop in the accuracy of about 10%. This drop continues until 50% of the pixels have been added. After 60% of the pixels are added, the positive curve has a large increase, which lasts until all the pixels are added, and the accuracy increases from 60% to 90%. In other words, the last 40% of the pixels have an essential role in positive classification.

The *c* and *d* in Figure 3.6 are two CAM-based visualization methods. The growth trend is different from the saliency-based method in *a* and *b*. Both Grad-CAM++ and Faster Score-CAM have a critical early stage, and in the second half, with the increase of pixels, the accuracy remains unchanged. However, the significant improvement of Grad-CAM++ occurs later than Faster Score-CAM. As shown in Figure *c*, after the first 20% of pixels are added, the positive curve has a slow rise, and the negative curve follows closely, which also maintains with a not low growth rate, the accuracy ranges from 45% to 60%. After 25% of the pixels are added, the growth rate of the negative curve rises rapidly. After 40% of the pixels are added, the negative curve has reached 85% accuracy. This slow growth lasts until all the pixels are added and accuracy reaches 90% in the end. Although the positive curve also has a rapid rise at 25% of the pixels, the rise stops after 50% of the pixels are added. The accuracy at this time is about 75%. Then the positive curve has a slow rise. This slow rise continues until 65% of the pixels are added, at which point the accuracy is 85%, until after all the pixels are added, the positive and negative curves converge at the same point again. Among the four visualization methods, Faster Score-CAM is the only visualization method in which the positive curve and the negative curve are highly overlapped. After the first 30% of the pixels are added, both reach an accuracy of 85%, then with the pixel points, the increase in the accuracy is only 5%, the absolute accuracy is 90%.

3.4.3 with Probability

Based on the deletion curve and insertion curve, after each deletion or addition of a pixel, the new predicted probability output by the model is recalculated once again, which is the deletion curve with probability and insertion curve with probability.

Figure 3.7 shows the probability deletion curve, with a maximum value of about 0.78 and a minimum value of about 0.55. Compared with the accuracy deletion curve (Fig.3.4), the decreasing trends of the four curves which represent the four visualization methods are similar, the fastest decrease is Faster Score-CAM, the decrease in Vanilla Saliency

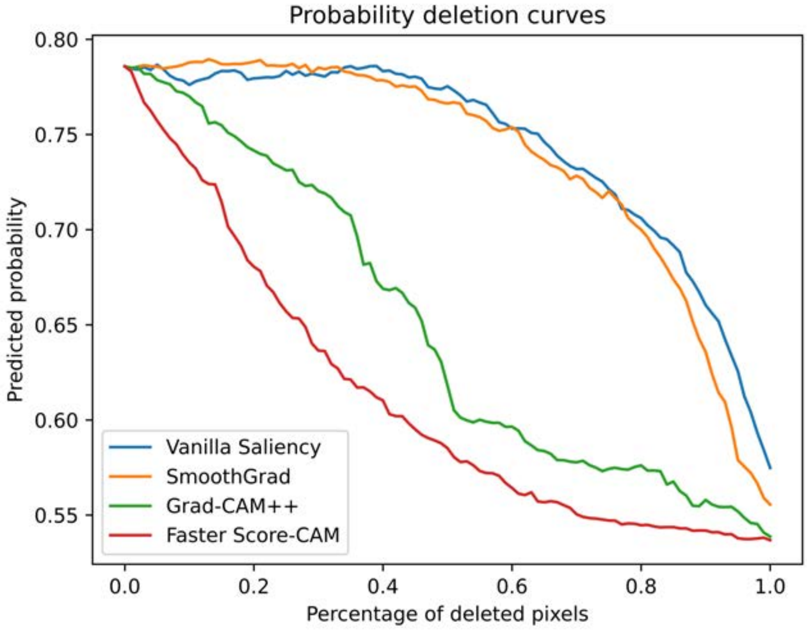


Figure 3.7 Probability deletion curve.

and SmoothGrad are very slowly, Grad-CAM++ is somewhere in between. Table 3.4 lists the area under the curve of each method. The exact number can help better for comparison. The AUC of the overall probability deletion curve is lower than that of the accuracy deletion curve.

Figure 3.8 shows the probability insertion curve. The lowest point is around 0.55, and the highest point is 0.8. Its trend and growth rate are similar to the accuracy insertion curve (Fig.3.3). Faster Score-CAM has the best performance and the highest growth rate. SmoothGrad is still underperforming, with a linear upward trend throughout, which means that it has not found a part that plays an important role in classification. The trend of Grad-CAM++ is still a significant increase after 30% pixel inserted. It achieves the same probability as Vanilla Saliency, which is in second place after 40% pixel inserted. It can be seen from the table 3.4 that the overall ranking is consistent with the accuracy insertion curve, but the value is small.

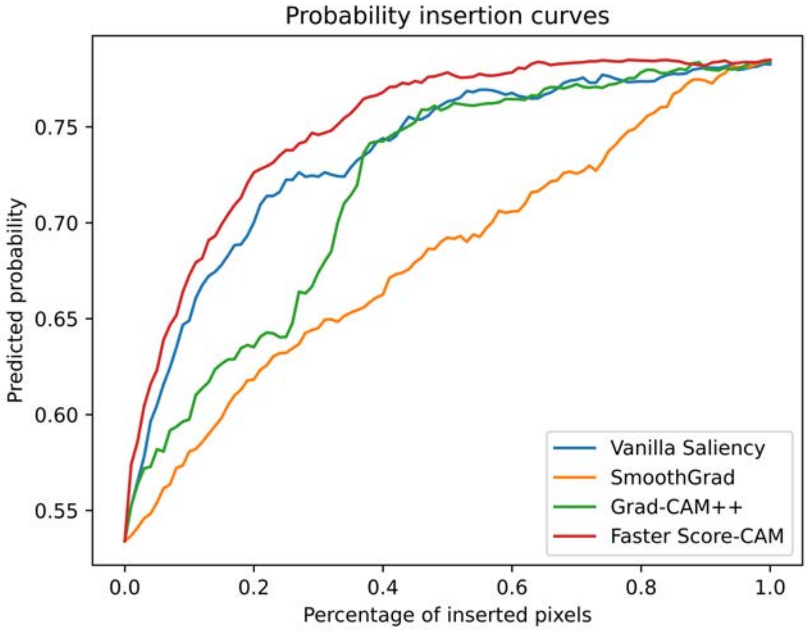


Figure 3.8 Probability insertion curve.

Methods	AUC of Deletion Curve	AUC of Insertion Curve
Vanilla Saliency	0.745	0.735
SmoothGrad	0.736	0.682
Grad-CAM++	0.649	0.718
Faster Score-CAM	**0.611**	**0.750**

Table 3.4 Area under curve of probability deletion curve and insertion curve.

3.5 Visual Coherence

Three positive examples and three negative examples and their resulting heatmaps using different visualization methods are shown in Figures 3.9 and 3.10. The difference between several visualization methods can be more intuitively felt through the heat map. The core idea of visual coherence is as significant regarded areas are only on the audio events of interest and not on other portions of the spectrogram. In other words, it must be determined how well the produced visualizations cover the input spectrogram's energy distribution.

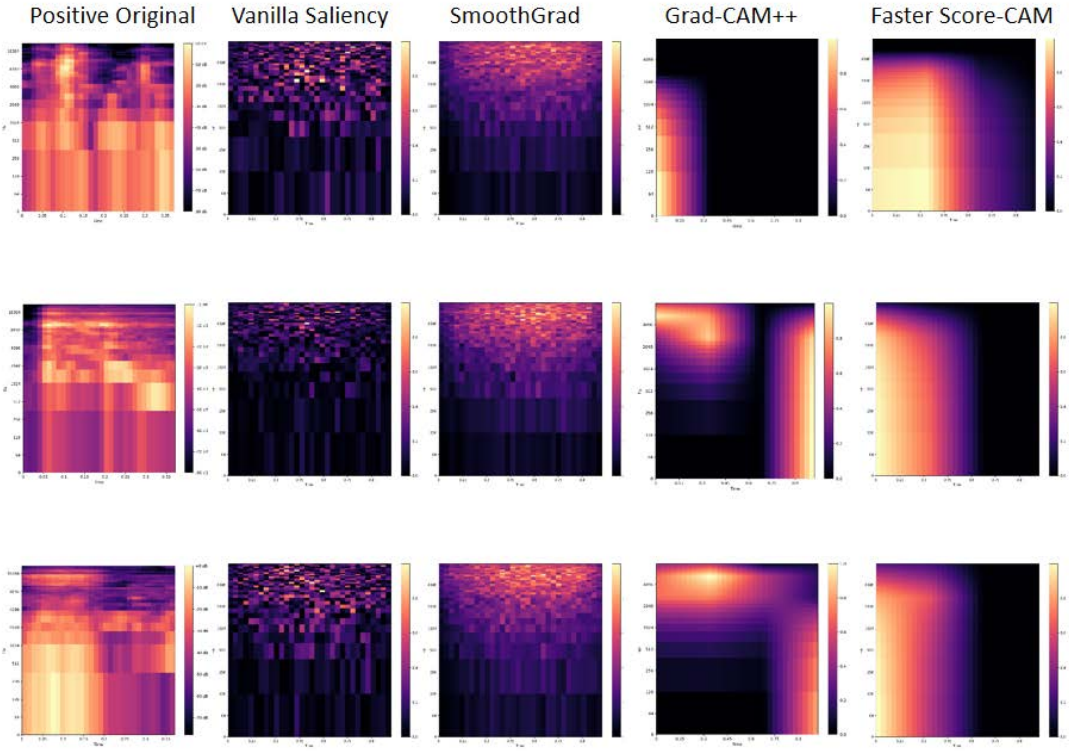


Figure 3.9 Qualitative evaluation of different methods for positive samples.

The examples show that the distribution of visualizations based on CAM covers the actual input spectrogram considerably more precise than the saliency methods.

In Figure 3.9, the leftmost is the spectrogram of the three selected positive samples, and from left to right are the heatmaps of different visualization methods. The Vanilla Saliency method marks a lot of small pixels in the high-frequency band, the pixels marked by each sample are different, but because each pixel is too small and the distribution is too scattered, it is impossible to interpret a pattern or even a specific position of the pixels. In addition, there are no audio events in the original spectrogram at some marked pixel positions. In Figure 3.10, the heatmaps which were generated by the three negative samples using vanilla saliency are similar to that of positive samples. The pixels are too scattered to interpret the pattern. It can only be roughly interpreted that the highlighted part is mostly higher than 2kHz.

The result of SmoothGrad is also very similar to Vanilla Saliency. The highlights of important areas were also marked by pixel, but it is denser than Vanilla Saliency. The disadvantage is the same as Vanilla Saliency. The distribution of marked pixels between classes is very similar, so it is challenging to distinguish positive and negative samples by relying on heatmaps. At the same time, there are also many labeled points in areas where audio events do not exist. The only conclusion that can be interpreted is that the distribution of markers in the high-frequency band is relatively dense, but as shown in the first example in Figure 3.9, the energy of the original spectrogram in the high-frequency band is not dense.

In the heatmaps which Grad-CAM++ visualized, it can be clearly seen that it has made significant progress compared with the Saliency-based method. The marked high-weight area is no longer a pixel point but a continuous area. It can be read from Figure 3.9, which are mainly marked around 0-2kHz and 4kHz. 0-500Hz is the vocal frequency range, while 4kHz speculate that this may be an essential band for determining wheezing sound. The patterns of the second and third examples are very similar from observation. The high-frequency area of about 5kHz in the first 0.6 seconds and the frequency band area between 0Hz-1kHz at the last 0.1 seconds are marked, but in the middle period, almost all of the entire frequencies are marked as 0. When compared with the original spectrogram, it can also be easily seen that the energy distribution of the area marked by Grad-CAM++ is also very high in the original spectrogram. As can be seen from Figure 3.10, the Grad-CAM++ method still marks a very high substantial weight, and the pattern is evident and readable. However, there is a lot of background noise in negative samples, so the only positive class will investigate how closely the visualizations mirror the spectrogram's energy distribution.

The performance of Faster Score-CAM is very similar to that of Grad-CAM++. The marked areas are visible and readable. The pixel effect of Faster Score-CAM will be weaker, and the transition area will be smoother. From the three positive samples in Figure 3.9, it can be concluded that the patterns obtained by Faster Score-CAM are very similar, the marked area is the first half of the time axis, namely within 0.5 seconds, and the frequencies band are about 0-2kHz. In contrast, in Figure 3.10, the three negative faster

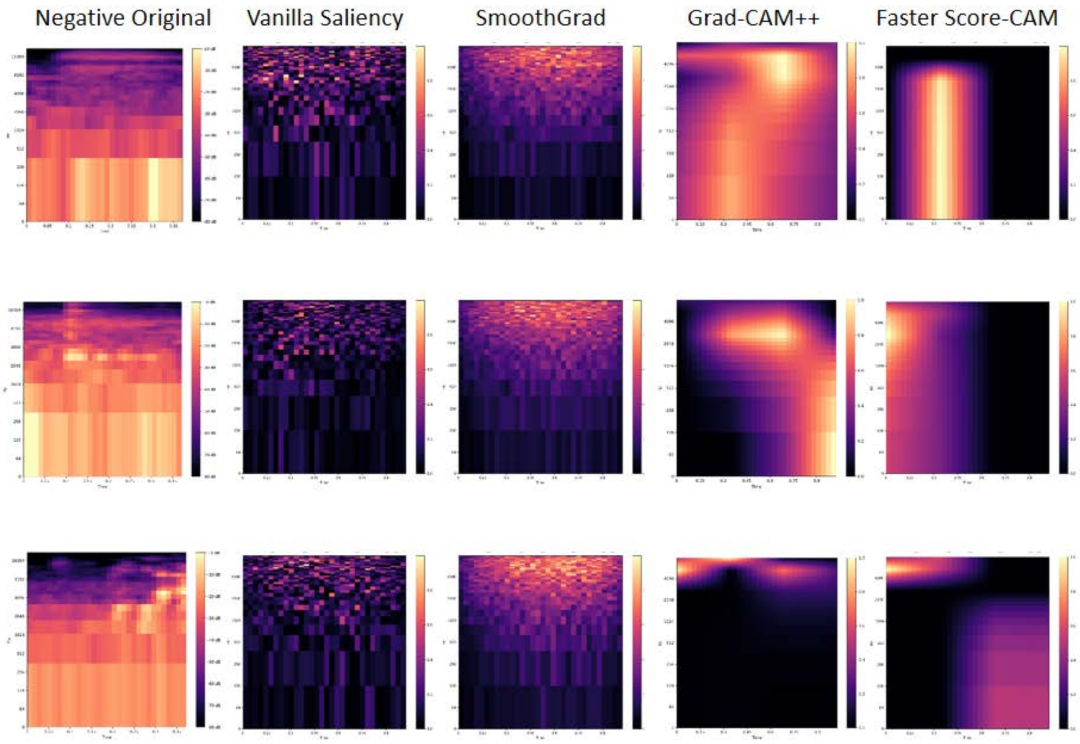


Figure 3.10 Qualitative evaluation of different methods for negative samples.

Score-CAM marked images do not have the same readable patterns. The three examples present three distinct highlight regions.

More examples of visual coherence can be found in the appendix from figure 5.1 to figure 5.4.

3.6 Class Discriminativity

If many audio events at the same time are present, only the audio events that want to be found are marked. It is class discriminated. As shown in Figure 3.11, the first and second rows in the first column on the left are the original spectrograms of the selected negative

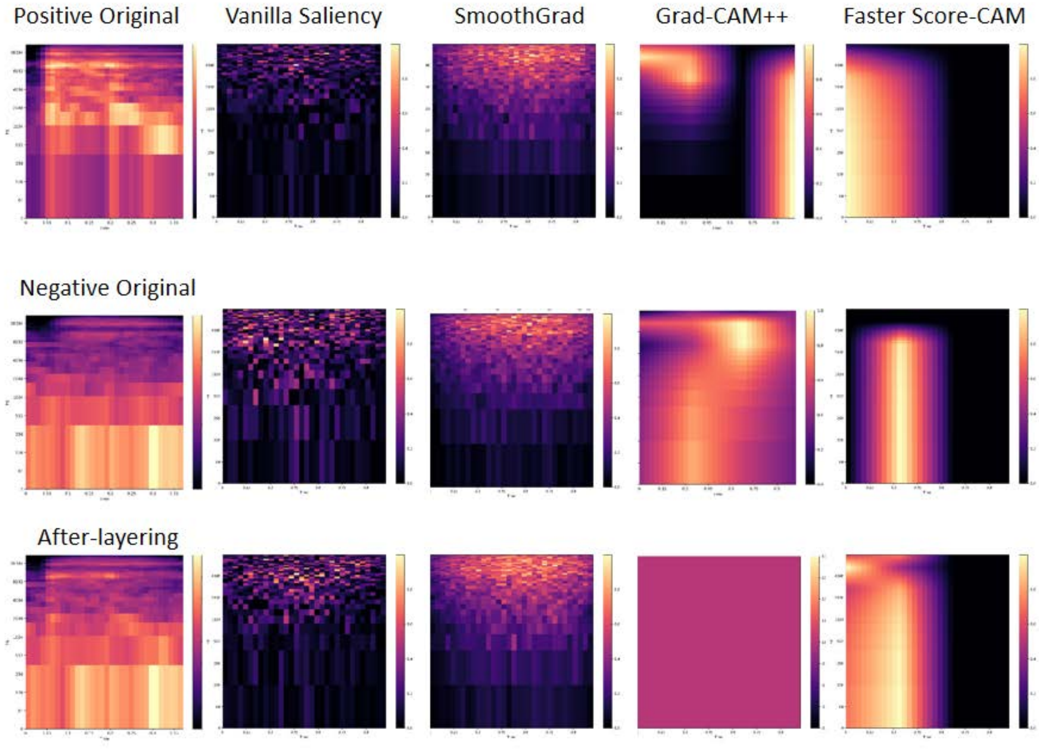


Figure 3.11 Class discriminativity example.

sample and selected positive sample before mixing, and the third row is the spectrogram after mixing. The following columns show their respective heatmaps generated by different visualization methods.

In the figure 3.11, the predicted probability of the mixed sample determined by the model is negative. In addition to other mixing attempts, when the background noise of the negative sample is not severe, the mixed samples are mainly predicted to be positive. When the sound pressure level of vocal or musical recording of the negative sample is high, the mixed samples are mostly predicted negative. In Figure 3.11, the results of the mixed sample, which are using Vanilla Saliency and SmoothGrad, are similar to the results in the previous section. Most of the pixels with high weight are distributed in the high-frequency band, and their distribution is neither concentrated nor continuous, resulting in their labeled pixels not providing helpful information to help identify classes.

The Grad-CAM++ method, which is used in the after-layering sample, presents with an entire frequency band weight of 0 as shown in the figure 3.11, it cannot identify a specific type of highlight area. This shows that Grad-CAM++ cannot provide vital information to identify the class for the mixed sample in this example. However, it can be read from the heatmaps of Grad-CAM++ before the overlap, and the energy distribution has marked almost the entire frequency band. It is speculated that the possible reason is that when the two overlap, marking the entire frequency band is approximately equal to not marking. The best performance in class discriminativity is still Faster Score-CAM because the mixed sample is predicted to be negative. Therefore, the main focus is on comparing the negative heatmap before mixing and the heatmap after mixing. The pattern marked in the negative sample is still there after mixing but affected by the positive sample. There is also a transition area in the 0Hz-2kHz band before 0.3 seconds. In addition, the high-weight region at 5kHz within the first 0.3s identified by Grad-CAM++ is also marked in the method of Faster Score-CAM. This shows that it identifies the fixed pattern of a specific class after mixing and marks the specific pattern obtained by other visualization methods. In fact, the negative samples are easier to predict because they are more heterogeneous and activate more areas in the spectrogram. When the network sees many areas activated, it tends to decide negatively, as faster score cam shows, this can be used to design training datasets for further training better.

More examples of class discriminativity can be found in the appendix from figure 5.7 to figure 5.10.

3.7 Interpretability

Interpretability means whether visualization approaches help understand the patterns the network learns. It can be seen from the excerpted samples of Figure 3.9 and Figure 3.10 that although the Saliency-based visualization method could mark many pixels of a part of a high-frequency band, it still cannot draw a concrete pattern to provide information to help to understand. After the audio events are mixed, as shown in Figure 3.11, there is no focus on a specific audio event, so a concrete explanation cannot be obtained through its noticed weight. Compared with Saliency-based methods, the performance of the CAM-based methods is better. Before the audio events do not overlap, as shown in Figure 3.9 and 3.10, both Grad-CAM and Faster Score-CAM can mark the most featured regions. For the prediction of positive samples, the region of 1-2kHz within the first 0.1s is significant. For predicting negative samples, the region of 2k-3kHz within the first 0.3s is crucial. However, because the negative samples contain a wide variety of background sounds, human voices, musical sounds, machine running noise, and environmental noise, the patterns in the negative samples are more diverse, and it is difficult to summarize a fixed rule. However, this does not mean that the pattern on the negative sample is not essential. When many patterns on the negative consistent with the pattern exist in the positive, it indicates that the frequency band in that region plays a vital role for classification. The signal was not detected in these frequency bands of the samples, and therefore, it was predicted as negative. After positive and negative audio events are mixed, Grad-CAM++ presents a spectrogram with an entire frequency band of 0, which means that it cannot find specific audio events after the audio events overlap. In comparison, Faster Score-CAM performs very well, marking the distinct regions of a specific audio event and marking the featured regions marked by other visualization methods when a single audio event was inputted.

In the case of misclassification, an interpretable visualization approach should also demonstrate the cause of the false prediction.

The visualization results of the three negative examples shown in Figure 3.10, in which the model predicted result of the second sample is positive. As in the previous analysis, the two visualization methods based on Saliency do not give any useful information about the case of misclassification. In contrast, Grad-CAM++ performs well, it marks the area of 3kHz in the middle of the time axis and the area of 0-1kHz in the last 0.1 seconds, in the positive visualize heatmap, it also marks the area of about 4kHz in the first half of time axis and the area of 0-2kHz in the last 0.1 seconds, it is obviously that the two patterns are very similar. Although the subtle frequency values are different, the horizontal and vertical positions and the relative positions of the two patterns do explain the reason for the misclassification of the model from a certain perspective. The results of Faster Score-CAM visualization are more intuitive, it is marked in the first 0.1 seconds 2kHz region in the visualization of misclassified samples, and there is a long transition zone from 0Hz to 2kHz, which is similar to the resulting feature maps of positive sample. As analyzed above, both CAM-based methods can provide understanding support for misclassification.

More examples of misclassification can be found in the appendix under figure 5.5 and figure 5.6. A table about overview of all results is shown in table 3.5.

Methods Results	Vanilla Saliency	SmoothGrad	Grad CAM++	Fast Score-CAM
Mean Average of Weights	0.14729	0.28350	0.04036	0.22157
Average Incrase	63.50%	65.50%	62.00%	**78.00%**
Average Drop	17.33%	16.53%	17.33%	**7.48%**
AUC of Insertion Curve (Accuracy)	0.821	0.742	0.787	**0.837**
AUc of Deletion Curve (Accuracy)	0.842	0.830	0.712	**0.701**
AUC of Insertion Curve (Probability)	0.735	0.682	0.718	**0.750**
AUc of Deletion Curve (Probability)	0.745	0.736	0.649	**0.611**
Visual Coherence	scattered pixels form high frequency	similar with Vanilla Saliency but denser	specific readable pattern	specific readable pattern
Class Discriminativity	indistinguishable	indistinguishable	partial distinguishable	mostly distinguishable
Interpretability	incomprehensible	incomprehensible	understandable	understandable

Table 3.5 Overview of all results for the visualization methods.

4 Discussion and Conclusion

Table 3.5 shows all results of the methods together. When all criteria are taken into consideration, we observe that Fast Score-CAM has a good performance in each introspection method. The different visualization methods are discussed below in the context of each parameter. Chapter 4.5 is a conclusion of the complete thesis.

4.1 Average Increase and Drop

In the average increase and average drop, except Fast Score-CAM, the results of the other three visualization approaches are very similar. Although SmoothGrad's mean average weight is high, its average increase and drop performance are not very high, with SmoothGrad showing the lowest performance for insertion curves. It is speculated that this is probably because the noise introduced by SmoothGrad has an unfavorable impact. The data of Vanilla Saliency is a good illustration of this when the noise added by SmoothGrad is 0. Therefore, when the influence of noise is removed, Vanilla Saliency still achieves the almost identical AI and AD values as SmoothGrad under the condition of the low mean average of weights, even outperforming SmoothGrad in the comparison of insertion curves. This shows that visualization in the presence of noise is more difficult. SmoothGrad's poor performance is probably based on Vanilla Saliency, where noise is added to reduce the local fluctuations caused by class score functions. With the addition of noise, SmoothGrad makes noticed pixels more conspicuous. Moreover, as can be seen from the Vanilla Saliency heat map, the Saliency-based visualization approaches would have performed poorly in this project.

4.2 Insertion and Deletion Curve

The performance of the insertion curve and deletion curve of different classes might be different because, for different classes, the position of essential pixel points may be different, so the position where the slope of the curve changes will also be different. For example, in Figure 3.5, for the SmoothGrad method, the positive curve falls a little earlier than the negative curve. If the visualized feature pattern is scattered, it will perform poorly in the overall insertion and deletion curve because it cannot find pixels that are the most critical area for correct classification. SmoothGrad's upward trend of the accuracy insertion curve is non-specific, almost similar to a linear increase, which is shown in Figure 3.3 and Figure 3.8. It might be because the marked patterns are evenly distributed on the entire spectrogram. It is worth noting that the average weight of SmoothGrad is very high. However, its performance is not better than Vanilla Saliency. Vanilla Saliency has a meager average weight, but in the comparison of accuracy insertion curves, as shown in Figure 3.3, it is still the best visualization method except for Fast Score-CAM. All in all, it can be said that the two CAM-based methods are better than the two Saliency-based methods. Although the average weight of Grad CAM++ is deficient, other parameters still show its excellent performance. In other words, this method is the most efficient visualization method. The AUC value verifies the results of the curve analysis. The final ranking is consistent with the AD and AI values. Vanilla Saliency performs better among the two Saliency-based methods, and among the two CAM-based methods, Fast Score-CAM is better. It shows that SmoothGrad cannot help reduce the influence of local fluctuations of the class score functions in a specific situation. As mentioned in [22], Smoothgrad has been shown to work well for images with complicated but well-located characteristic parts for a class. Still, it is not easy to find such well-defined areas in the spectrogram in this experiment. Therefore, other methods, such as Faster Score-CAM, seem to be working better for the Acoumon project. The results of the probability insertion and deletion curve are very similar to the accuracy insertion and deletion curve. The better results of Faster Score-CAM than Grad-CAM++ show that gradient-free computation of weights improves the reliability of visualization methods.

As shown in Figure 3.5, there is not much difference between the two types of curves in the class-wise deletion curve. It is consistent with the performance in the deletion curve. The two Saliency-based methods decline later, and the inflection point of the two CAM-based methods appears earlier. This shows that the two CAM-based methods have found important regions that two Saliency-based methods do not find. In the SmoothGrad method, the drop point of the positive curve appears earlier than the negative curve. It is speculated that the noise introduced by SmoothGrad has a negative impact on the negative sample. The situation of the class-wise insertion curve is slightly different, as shown in Figure 3.6. Faster Score-CAM still has the highest AUC value, but the performance of Grad-CAM++ in the first half of the time axis is not as good as Vanilla Saliency. Combined with the following heatmap analysis, Grad-CAM++ marks more middle frequency bands in the middle time axis, so when the low-frequency region at the beginning of the time is deleted, it does not bring a helpful improvement. In particular, the positive curve in the first half of the time axis is higher than the negative curve. Then the negative curve exceeds the positive curve after a certain point, which shows that the patterns of the two classes marked by Grad-CAM++ are located in different positions, which makes deleting pixels at different positions have different effects on the two classes, this is also a hint that both classes in our dataset have very heterogeneous spectrogram manifestations.

4.3 Visualized Results

Numeric results and curves are mainly helpful for comparing the performance of different visualization approaches, but they can not offer a faithful ranking to the best technique. In the post-visualization analysis, it can be seen that the two Saliency-based methods mark many high weighted pixels, which are densely distributed in the high-frequency band. Although SmoothGrad does not achieve good results from the numerical results, after visualization, SmoothGrad is similar to Vanilla Saliency's enhancement, it marks the pixels more strongly and makes them more discernible. But its shortcomings are also apparent; namely, the differences between samples are too small to identify, there is no fixed pattern, and it is not easy to explain in a coherent way by applying audio-domain knowledge.

The two CAM-based visualization methods are more clearly visualized. Both Grad-CAM++ and Faster Score-CAM can mark specific patterns which are interpretable. There are many types of sound in negative samples; for example, it might include vocals, musical, and noisy sounds. So the marked patterns in negative samples are not uniform. However, this does not mean that the pattern on the negative sample is not essential. When many patterns on the negative samples are similar to those existing in the positive samples, it indicates that the frequency band in that region plays a vital role in classification. The signal was not detected in these frequency bands of the samples, and therefore, it was predicted as negative. Many similar patterns can be found with the two CAM-based methods in positive samples; the marked frequency is mainly distributed around 0-2kHz and 3.5kHz, which is very understandable because the frequency range of the wheezing sound is about 80-1.6kHz [5], and the sounds of about 3.5KHz is probably a high frequency that is not easily heard by the human ear with the wheezing sound. After two classes overlap, Grad-CAM++ loses the ability to discriminate specific audio events. It is speculated that marking the entire frequency band is approximately equal to not marking when two classes overlap. In the aspect of class discriminativity, Fast Score-CAM still performs very well, not only recognizing patterns of specific audio events but even recognizing new patterns that were not recognized before the overlap.

4.4 General Comparison

To sum up, Vanilla Saliency can identify pixels of interest, it performs well regarding insertion and deletion, but it does not help increase explainability. SmoothGrad is the worst of the four visualization methods. In theory, SmoothGrad is a noise improved version of Vanilla Saliency, but its results are not as good as expected in this project. However, the color of pixels is more vivid, making the marked area more conspicuous. Still, numeric results and visualized spectrogram analysis prove that its performance is inferior, which means that the introduction of noise will have a negative impact. Both CAM-based visualization methods have achieved acceptable performance in this experiment. However, Grad-CAM++ is not particularly outstanding in some numeric results, such as AI, AD values, and some AUC values, probably due to its low average weight value. Grad-CAM++ can find uniform patterns on the visualized spectrogram. Most of these patterns are easy to read, explainable, and relevant, but when audio events of two different classes overlap, Grad-CAM++ cannot identify a specific audio event. Despite this, the great performance of the Grad-CAM can still find the frequency band for audio features when a single audio event was inputted. By comparison, Faster Score-CAM performs very well in every aspect. It illustrates the favorable impact of not using gradient computations. It can identify specific audio events and always gets first place in comparing various numeric indicators among the four methods. The visualized spectrogram is readable and can find logic to understand, providing solid data support and guarantee for the model. Nevertheless, the transition area marked on the spectrogram by Faster Score-CAM is too broad, making it impossible to pinpoint accurate frequency values.

4.5 Summary

The main purpose of this graduation thesis is to evaluate several visualization methods through various metrics, find the most suitable visualization method for an audio-based acoustic monitoring system, and find the frequency basis for the model to rely on and provide, if possible, a suitable explanation. The results show that the two Saliency-based methods cannot give reliable information to interpret the visualization. In contrast, both CAM-based methods can find a uniform pattern, and the labeled frequency bands are meaningful and can be explained how the network works. The greatest difficulty in the experiment is that the recording segment usually contains the full frequency band, and only specific frequency positions will be highlighted. Hence, its performance on the spectrogram is only the color differences in specific areas calculated by the dB calculation formula; that said, it is not possible to pinpoint objects with very distinct boundaries, as in the case of image recognition. Before visualization, model training and dataset processing were performed as described in [14]. The visualization implementation was created using the open-resource project in Github $tf-keras-vis$ visualization toolbox [12]. Several evaluation methods after visualization are based on the evaluation methods used in [1], [17] and [22], including average drop, average increase as well as deletion and insertion curves, visual and aural coherence, class discriminativity, and interpretability.

The final results show that it is similar to results obtained in image classification. The CAM-based method outperformed the Saliency-based method, which provides more and more consistent interpretations, reliable results, and well performance [22] [19]. The performance of two Saliency-based methods in this project is not satisfactory; it is speculated that audio events are too widely distributed on the spectrogram and cannot be precisely positioned. In the numerical results, Vanilla Saliency performs better than SmoothGrad, which indicates that noise cannot reduce the influence of local fluctuations of the class score functions in this project. Among the two CAM methods, Grad-CAM++ can provide a particular feature map and label the frequency bands important for the classification, but when the two classes overlap, it can no longer label the critical regions. Fast Score-CAM is better in detail; it has excellent performance on all metrics, especially AI and AD values, far higher than several other visualization methods. However, as mentioned above, audio events are spread too widely on the spectrogram, resulting in extensive transition areas of their marked patterns, so precise frequency values cannot be obtained. Most of the marked areas are located at 0-2kHz and about 3.5kHz, mainly in the first half of the time axis, consistent with the wheezing sound's frequency range. In daily recordings, it could be seen that when 3.5kHz is too high, a sound like a cough will occur.

A summary of each of the four visualization methods is placed in 3 dimensions

- In terms of the ability to mark pixel points, all four visualization methods can mark significant pixel points, meaning that implementing visualization methods on spectrograms is feasible and meaningful.
- In terms of interpretation, both CAM-based methods perform well, probably because they are based on CAM. The basic idea is to extract the complete abstract feature map learned from the higher convolutional layers and then maximize the feature map by upsampling and deconvolution. This allows the spatial position of the convolution kernel to be maximally preserved with the corresponding position of the input image. The two Saliency-based methods only calculate the gradient of the

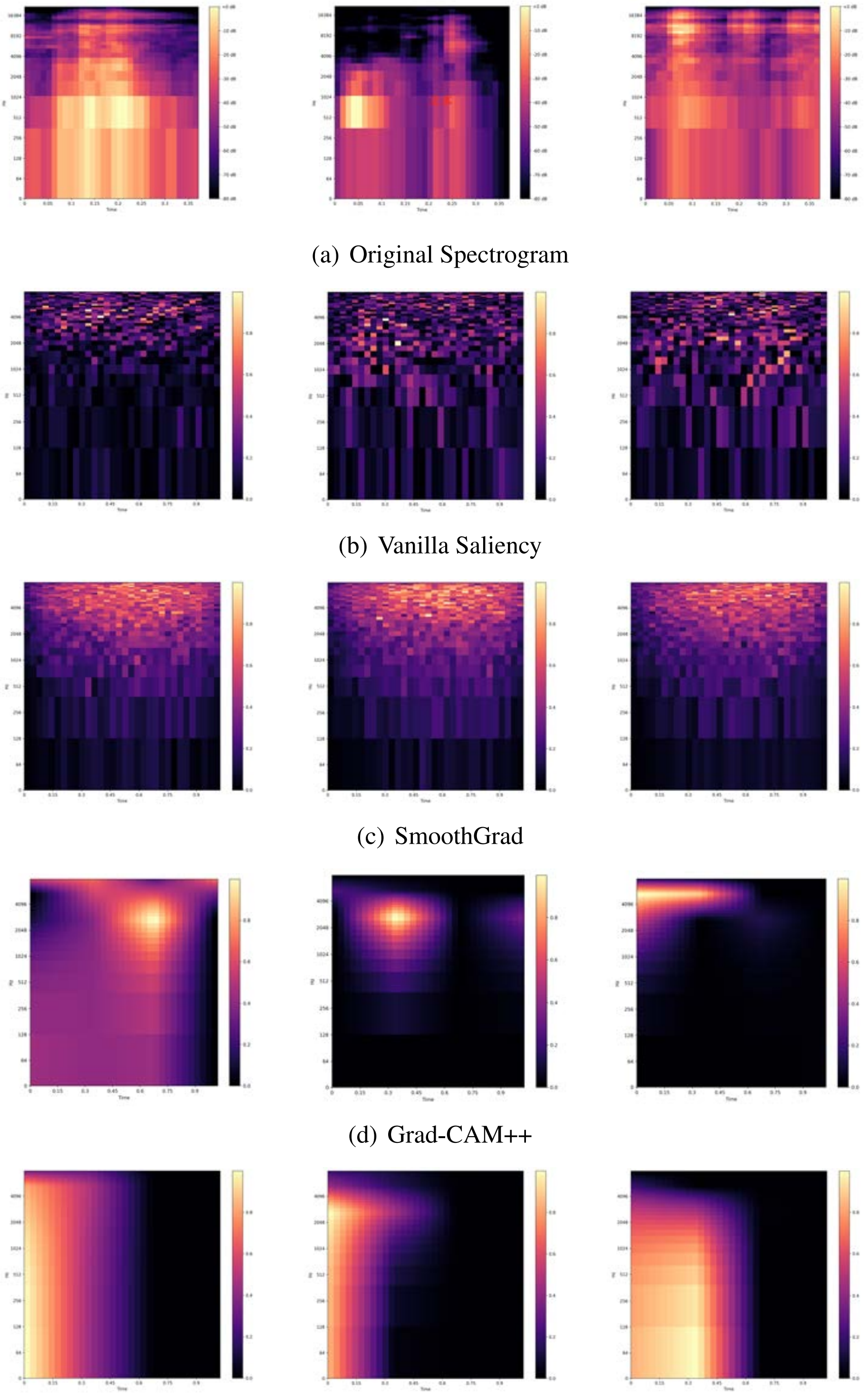
class score function for a particular input image but do not perform gradient ascent operations, which may distort the spatial location of the feature map.

- In terms of class discriminativity, Fast Score-CAM performs best, recognizing specific audio events well in the case of mixed audio events. It might be because Fast Score-CAM avoids the use of gradients to obtain feature weights, thus avoiding a series of adverse effects due to gradients, which did give satisfactory results in this experiment with gradient-free.

In contrast to image recognition, audio spectrograms cannot offer objects with highly defined bounds. Besides, the overlap of audio events is calculated using the add-up, unlike objects in images that are separate. Therefore, the application of visualization methods on audio spectrograms needs to consider the ability of good class discriminativity especially. These are the areas where audio recognition differs from image recognition, so these visualization methods set up for image recognition are not completely adapted to acoustic monitoring. For example, proving 20% noise in [22] optimizes the performance of SmoothGrad, but this may not necessarily apply to audio events detection. It might be possible to optimize the visualization method for audio detection in the future. In addition to this, the acoustic scene is very complex. The recording file contains not only direct sound but also impulse features of the room. The room's acoustic characteristics will cause serious interference in recognizing the audio events. Better results may be obtained if a filter is first applied after making the recording to focus on the frequency band of interest.

Overall, the experimental results obtained in this paper prove that the application of visualization methods in the audio field is still reasonable. But, there is still much room for future research in this field.

5 Appendix



(a) Original Spectrogram

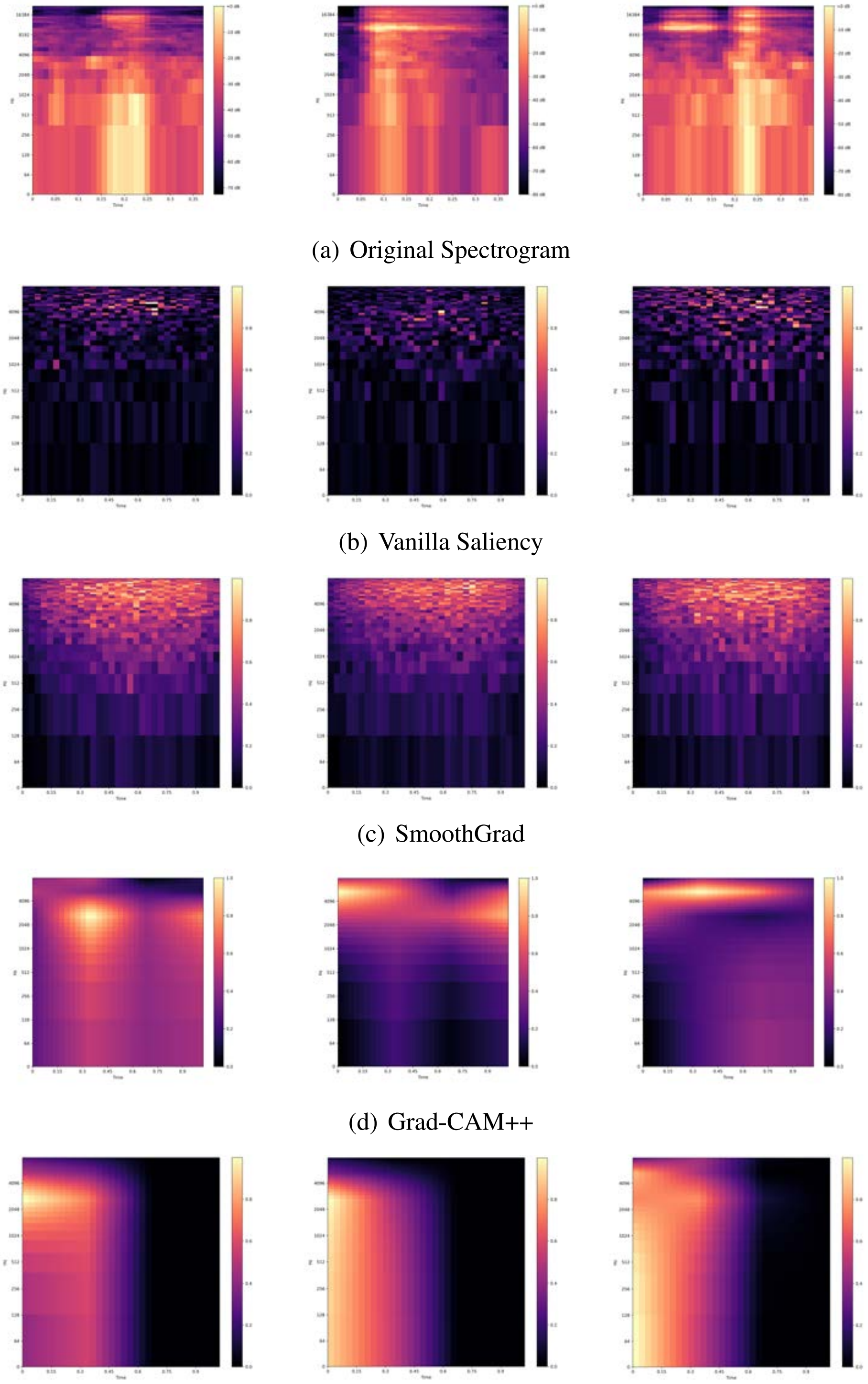
(b) Vanilla Saliency

(c) SmoothGrad

(d) Grad-CAM++

(e) Faster Score-CAM

Figure 5.1 Plots of all four visualization methods for the class positive.



(a) Original Spectrogram

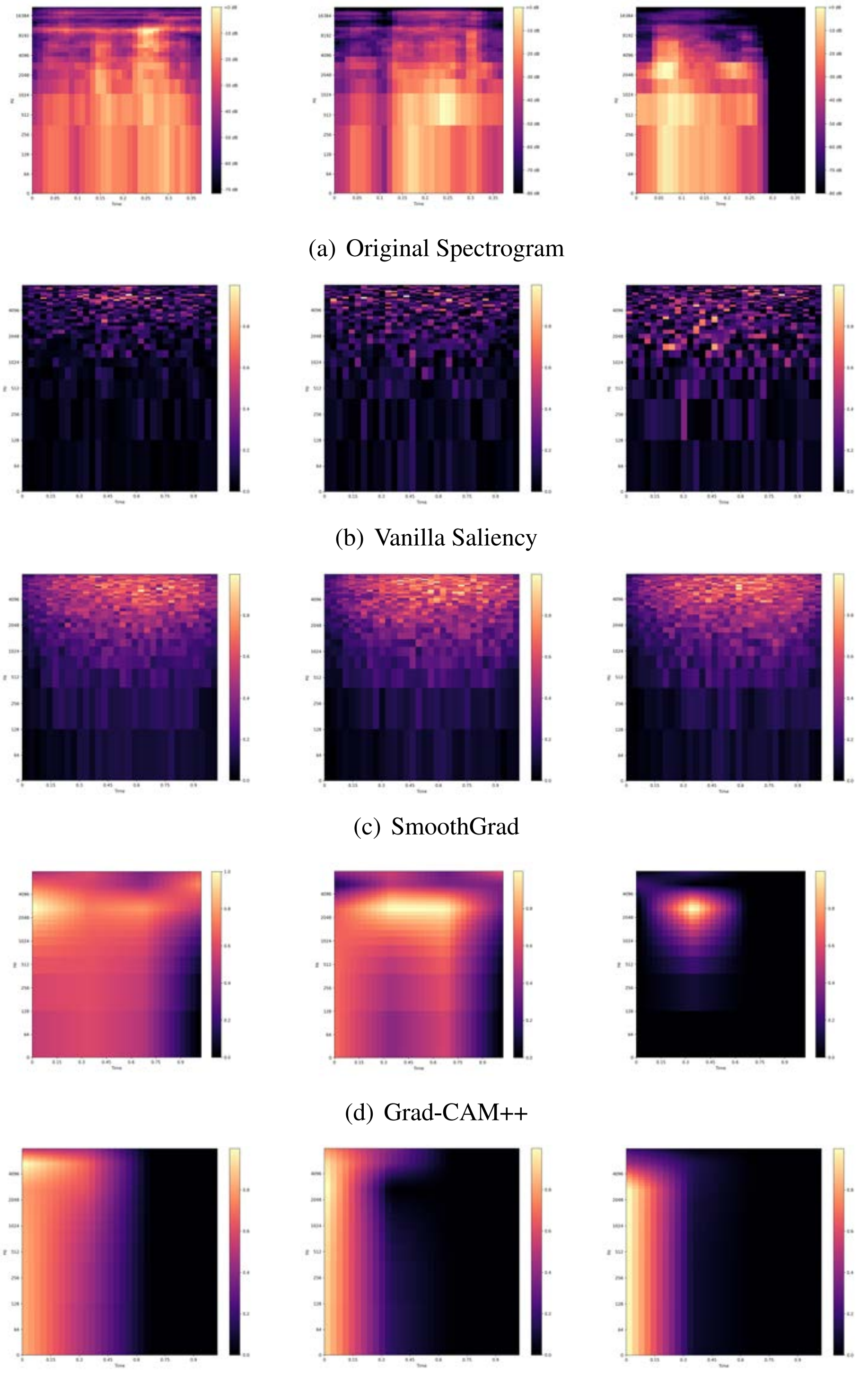
(b) Vanilla Saliency

(c) SmoothGrad

(d) Grad-CAM++

(e) Faster Score-CAM

Figure 5.2 Plots of all four visualization methods for the class positive.



(a) Original Spectrogram

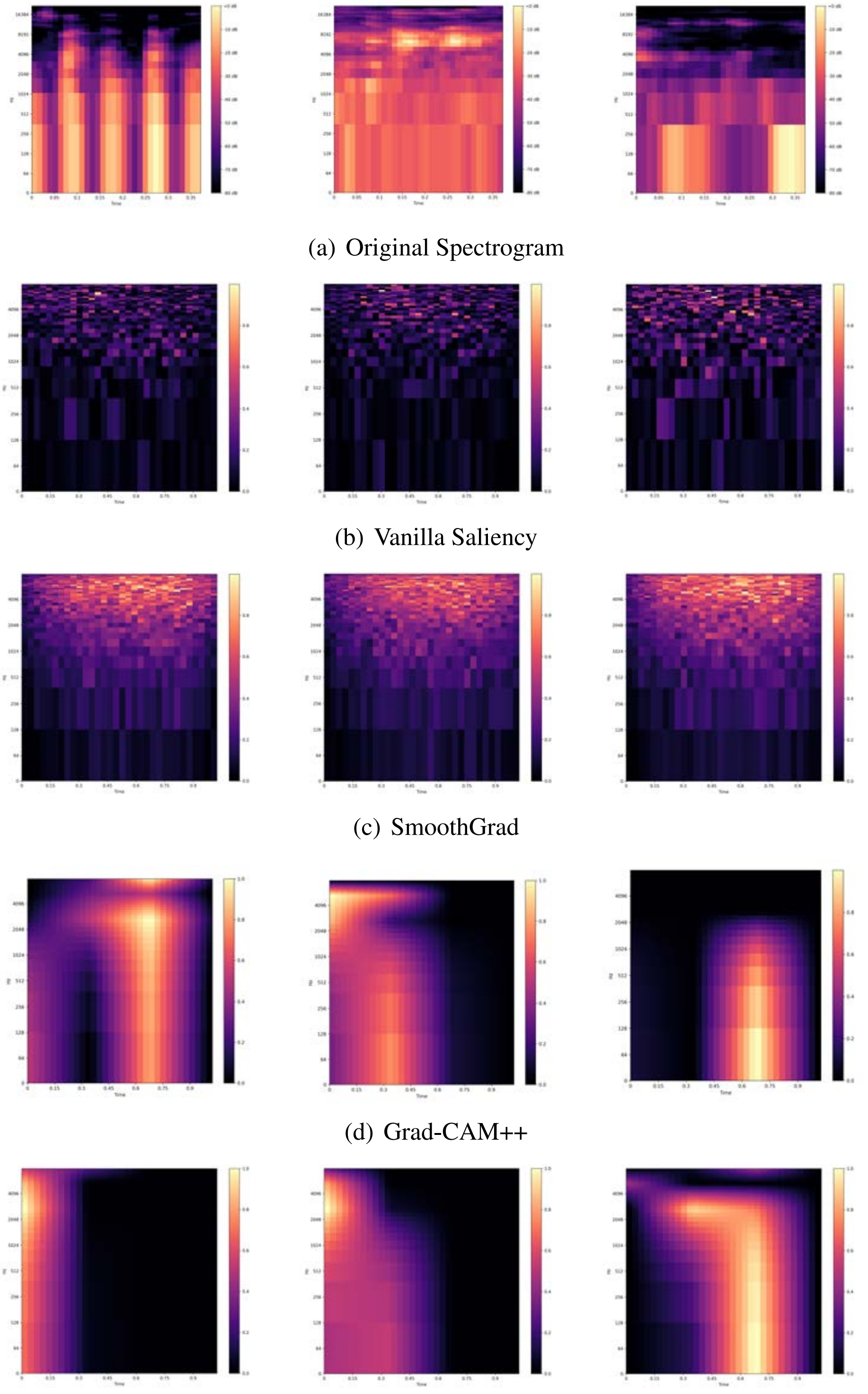
(b) Vanilla Saliency

(c) SmoothGrad

(d) Grad-CAM++

(e) Faster Score-CAM

Figure 5.3 Plots of all four visualization methods for the class positive.



(a) Original Spectrogram

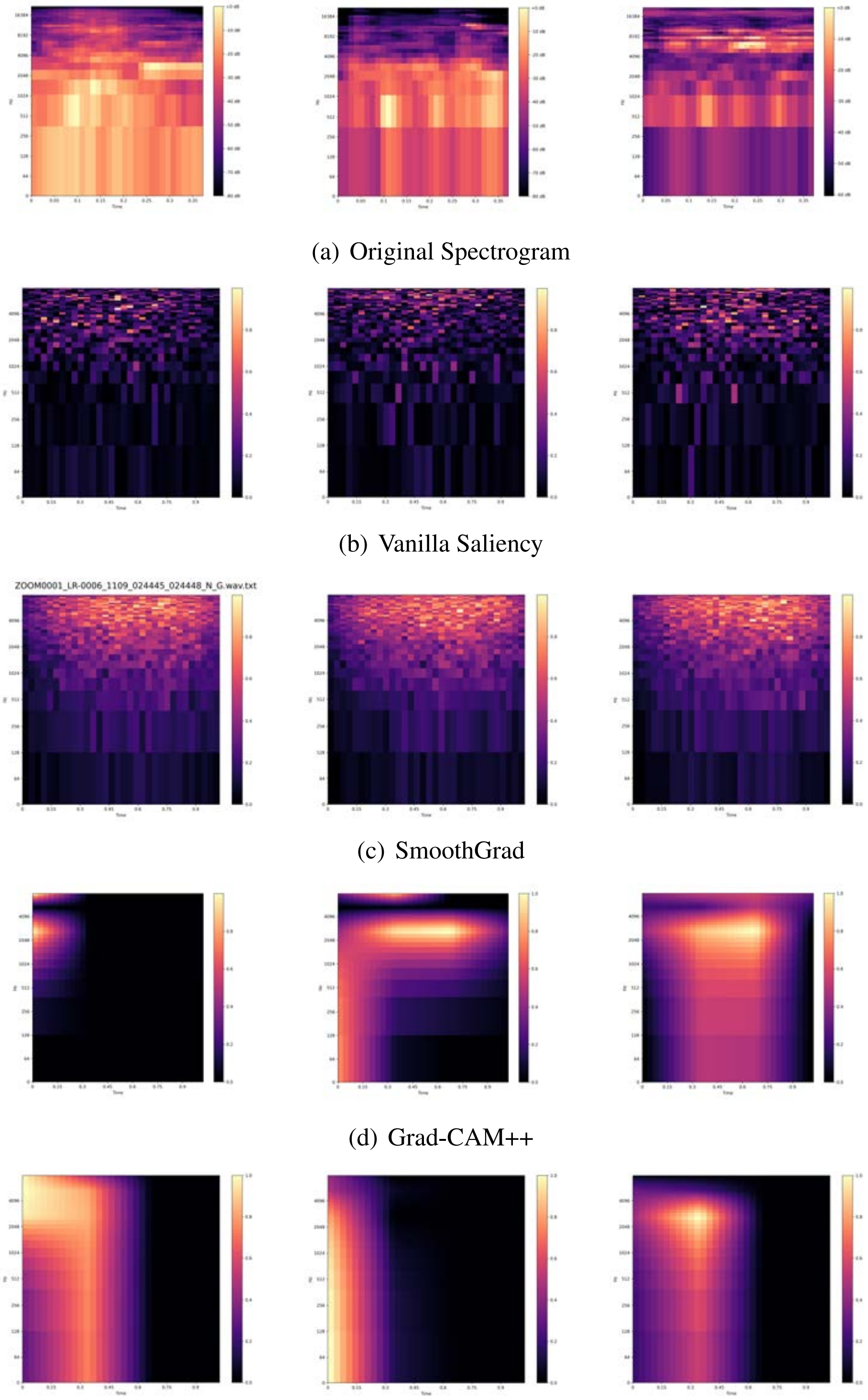
(b) Vanilla Saliency

(c) SmoothGrad

(d) Grad-CAM++

(e) Faster Score-CAM

Figure 5.4 Plots of all four visualization methods for the class negative.



(a) Original Spectrogram

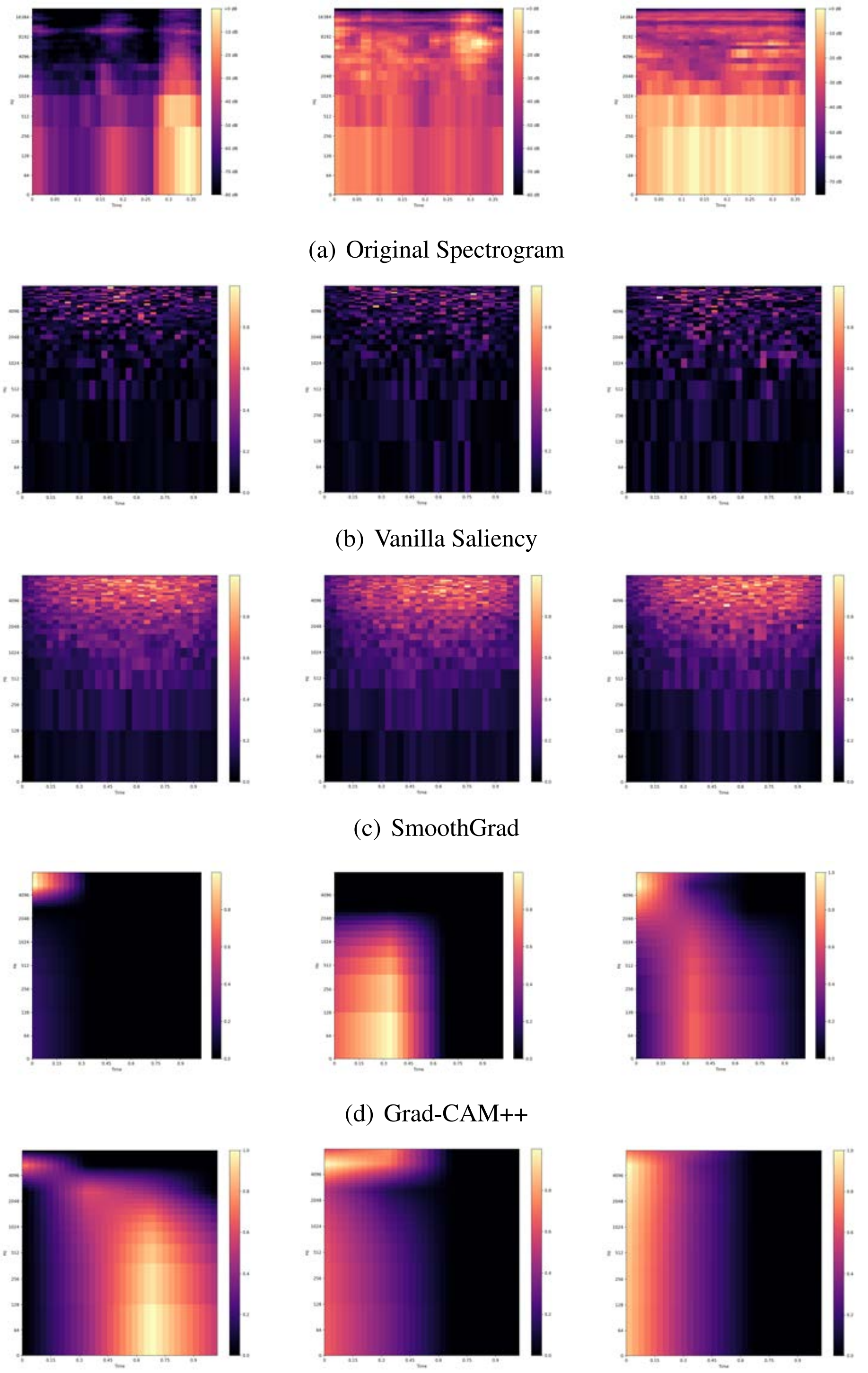
(b) Vanilla Saliency

(c) SmoothGrad

(d) Grad-CAM++

(e) Faster Score-CAM

Figure 5.5 Labeled negative but predicted as positive.



(a) Original Spectrogram

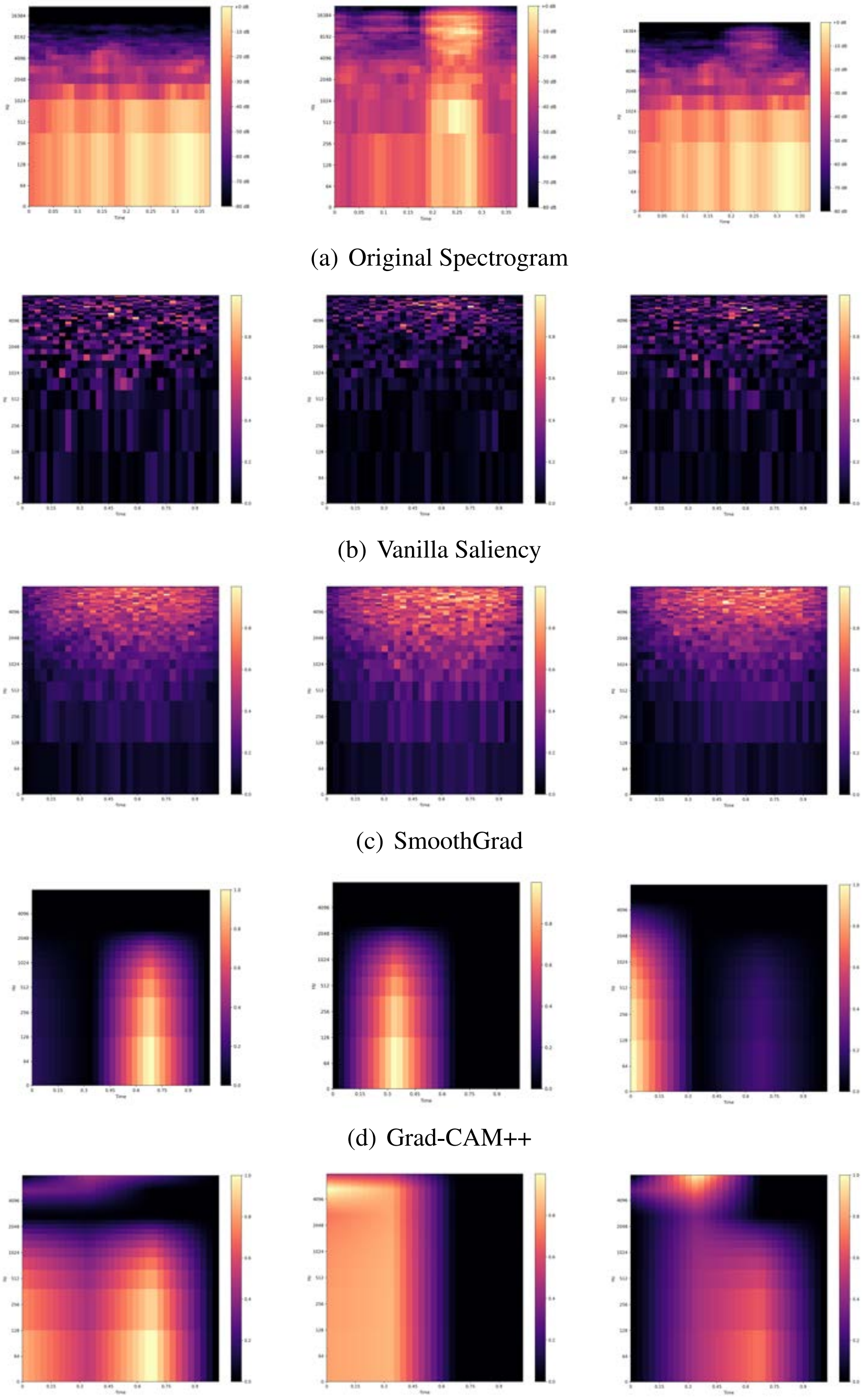
(b) Vanilla Saliency

(c) SmoothGrad

(d) Grad-CAM++

(e) Faster Score-CAM

Figure 5.6 Labeled positive but predicted as negative.



(a) Original Spectrogram

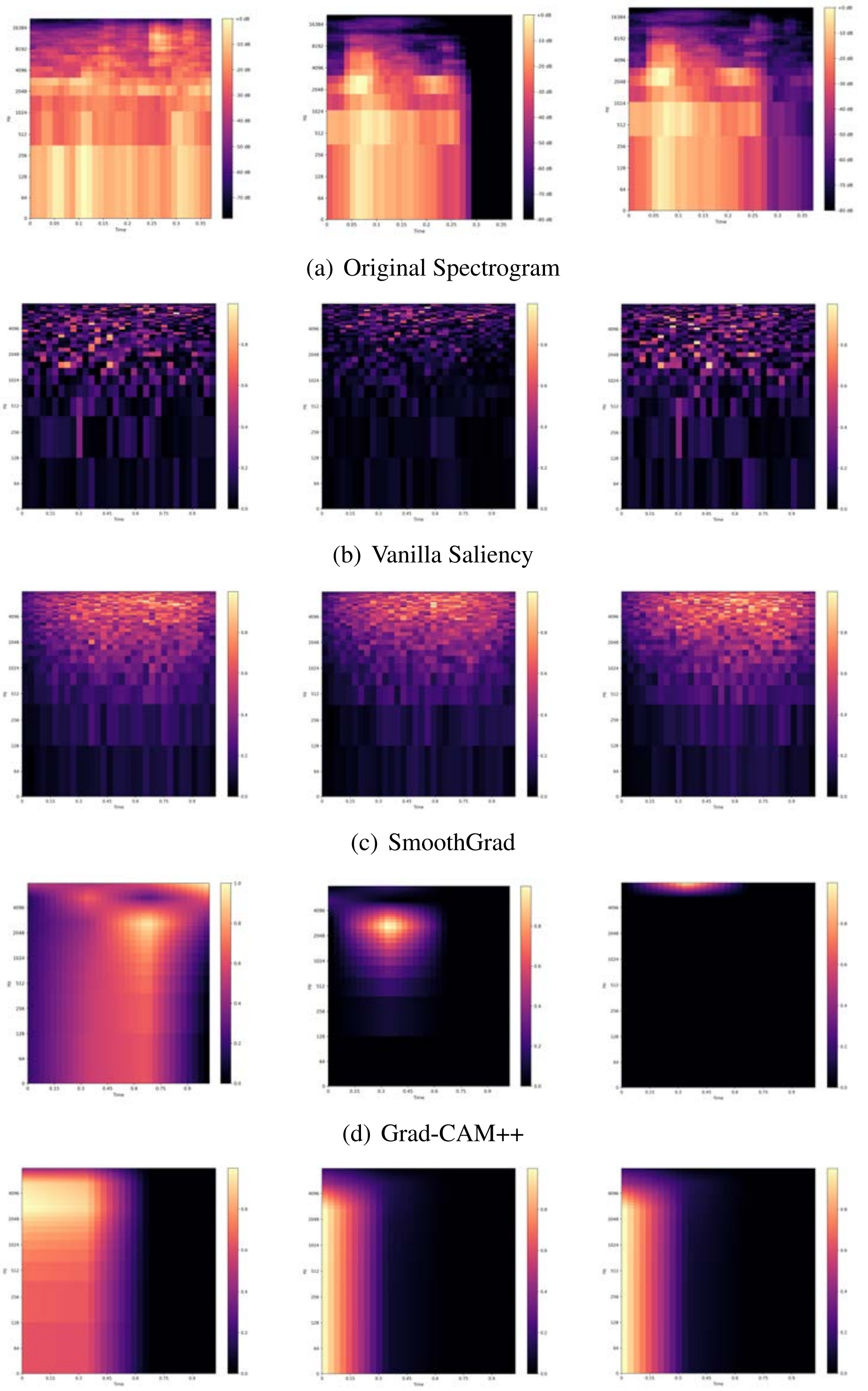
(b) Vanilla Saliency

(c) SmoothGrad

(d) Grad-CAM++

(e) Faster Score-CAM

Figure 5.7 Class discriminativity. Negative on the left, positive in the middle, mixed on the right.



(a) Original Spectrogram

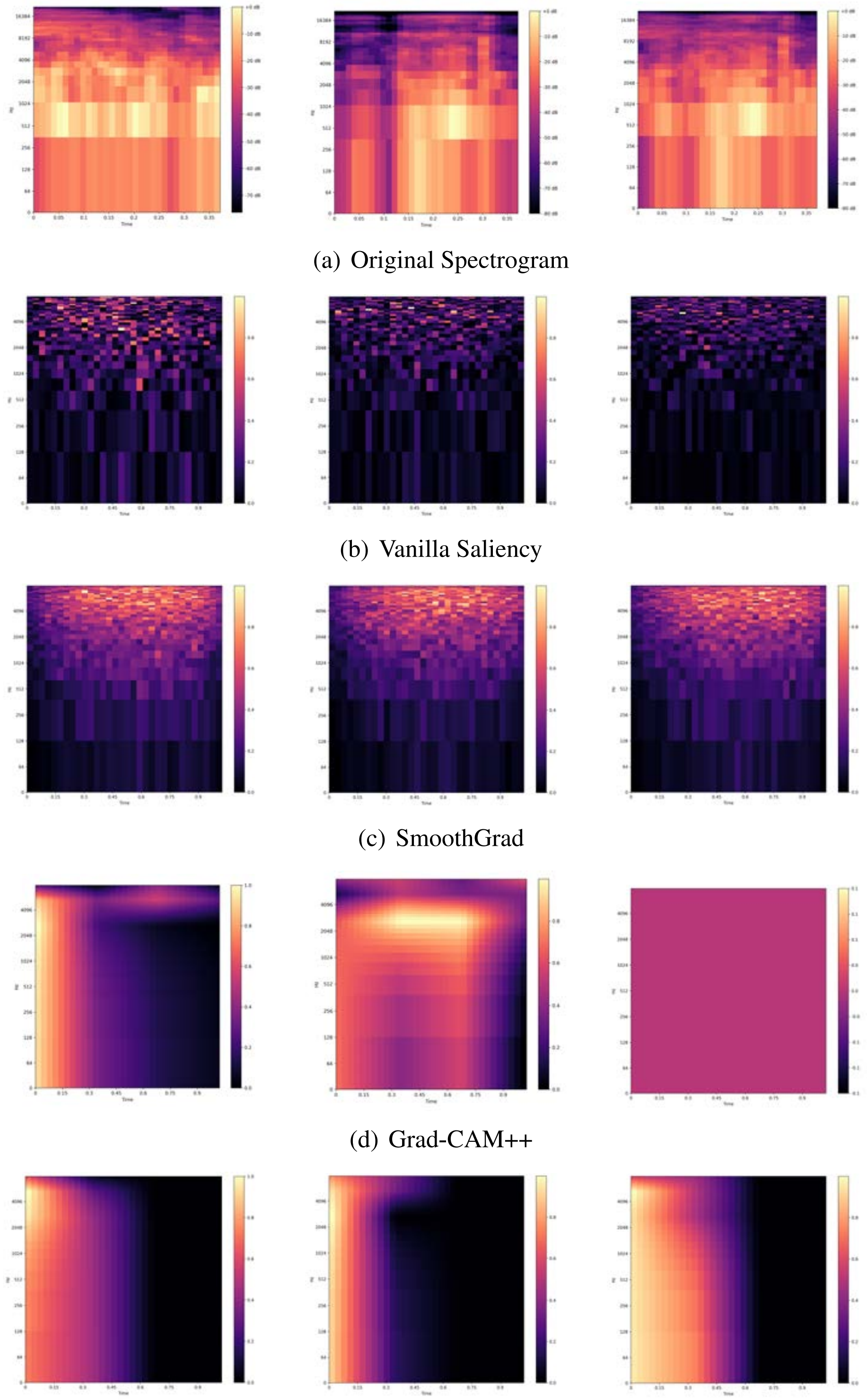
(b) Vanilla Saliency

(c) SmoothGrad

(d) Grad-CAM++

(e) Faster Score-CAM

Figure 5.8 Class discriminativity. Negative on the left, positive in the middle, mixed on the right.



(a) Original Spectrogram

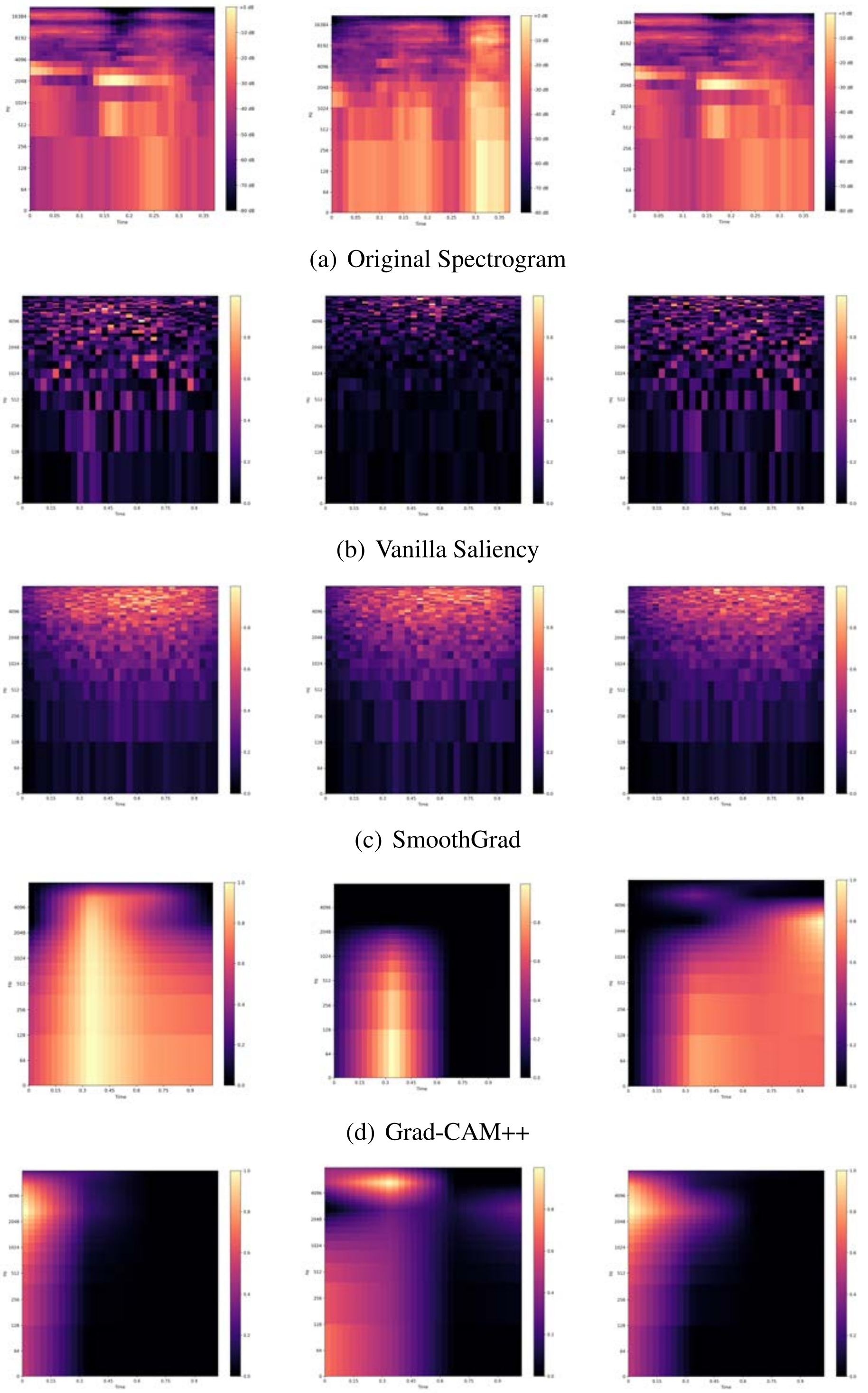
(b) Vanilla Saliency

(c) SmoothGrad

(d) Grad-CAM++

(e) Faster Score-CAM

Figure 5.9 Class discriminativity. Negative on the left, positive in the middle, mixed on the right.



(a) Original Spectrogram

(b) Vanilla Saliency

(c) SmoothGrad

(d) Grad-CAM++

(e) Faster Score-CAM

Figure 5.10 Class discriminativity. Negative on the left, positive in the middle, mixed on the right.

Bibliography

[1] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018.

[2] A Chattopadhyay, A Sarkar, P Howlader, and VN Balasubramanian. Grad-cam++: Improved visual explanations for deep convolutional networks. arxiv 2017. *arXiv preprint arXiv:1710.11063*.

[3] Keunwoo Choi, George Fazekas, Mark Sandler, and Jeonghee Kim. Auralisation of deep convolutional neural networks: Listening to learned features. In *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR*, pages 26–30, 2015.

[4] Gabriel Dernbach, Athanasios Lykartsis, Leon Sievers, and Stefan Weinzierl. Acoustic identification of flat spots on wheels using different machine learning techniques. 2020.

[5] Noam Gavriely, Yoram Palti, Gideon Alroy, and James B Grotberg. Measurement and theory of wheezing breath sounds. *Journal of Applied Physiology*, 57(2):481–492, 1984.

[6] Abdelfatah Hassan, Ismail Shahin, and Mohamed Bader Alsabek. Covid-19 detection system using recurrent neural networks. In *2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, pages 1–5. IEEE, 2020.

[7] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. Singing-voice separation from monaural recordings using deep recurrent neural networks. In *ISMIR*, pages 477–482, 2014.

[8] Andreas Jansson, Eric Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, and Tillman Weyde. Singing voice separation with deep u-net convolutional networks. 2017.

[9] Il-Young Jeong, Subin Lee, Yoonchang Han, and Kyogu Lee. Audio event detection using multiple-input convolutional neural network. *Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2017.

[10] Andreas Krug and Sebastian Stober. Introspection for convolutional automatic speech recognition. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 187–199, 2018.

[11] Anton Krusche. Visualization and auralization of features learned by neural networks for musical instrument recognition. 2021.

[12] Yasuhiro Kubota. tf-keras-vis. `https://github.com/keisen/tf-keras-vis`, 2021.

[13] Athanasios Lykartsis, Markus Hädrich, and Stefan Weinzierl. A prototype deep learning system for the acoustic monitoring of intensive care patients. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 980–984. IEEE, 2021.

[14] Hadrich Lykartsis and Weinzierln. A prototype deep learning system for the acoustic monitoring of intensive care patients. 2021.

[15] Veronica Morfi and Dan Stowell. Deep learning for audio event detection and tagging on low-resource datasets. *Applied Sciences*, 8(8):1397, 2018.

[16] Venkata Srikanth Nallanthighal and H Strik. Deep sensing of breathing signal during conversational speech. 2019.

[17] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[18] Gadi Pinkas, Yarden Karny, Aviad Malachi, Galia Barkai, Gideon Bachar, and Vered Aharonson. Sars-cov-2 detection from voice. *IEEE Open Journal of Engineering in Medicine and Biology*, 1:268–274, 2020.

[19] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] Andrew JR Simpson, Gerard Roma, and Mark D Plumbley. Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 429–436. Springer, 2015.

[22] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[23] Jorg Wagner, Jan Mathias Kohler, Tobias Gindele, Leon Hetzel, Jakob Thaddaus Wiedemer, and Sven Behnke. Interpretable and fine-grained visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9097–9107, 2019.

[24] Haofan Wang, Mengnan Du, Fan Yang, and Zijian Zhang. Score-cam: Improved visual explanations via score-weighted class activation mapping. *arXiv preprint arXiv:1910.01279*, 2019.

[25] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 24–25, 2020.

[26] Weitao Xu, Xiang Zhang, Lina Yao, Wanli Xue, and Bo Wei. A multi-view cnn-based acoustic classification system for automatic animal species identification. *Ad*

Hoc Networks, 102:102115, 2020.

[27] Xin Zhang, Kangwei Wang, Yan Wang, Yi Shen, and Hengshan Hu. An improved method of rail health monitoring based on cnn and multiple acoustic emission events. In *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6. IEEE, 2017.