Estimating Classification Accuracy for Unlabeled Datasets Based on Block Scaling

Shingchern D. You*, Kai-Rong Lin, Chien-Hung Liu

Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei, Taiwan, ROC

Received 15 May 2023; received in revised form 30 July 2023; accepted 09 August 2023

DOI: https://doi.org/10.46604/ijeti.2023.11975

Abstract

This paper proposes an approach called block scaling quality (BSQ) for estimating the prediction accuracy of a deep network model. The basic operation perturbs the input spectrogram by multiplying all values within a block by α, where α is equal to 0 in the experiments. The ratio of perturbed spectrograms that have different prediction labels than the original spectrogram to the total number of perturbed spectrograms indicates how much of the spectrogram is crucial for the prediction. Thus, this ratio is inversely correlated with the accuracy of the dataset. The BSQ approach demonstrates satisfactory estimation accuracy in experiments when compared with various other approaches. When using only the Jamendo and FMA datasets, the estimation accuracy experiences an average error of 4.9% and 1.8%, respectively. Moreover, the BSQ approach holds advantages over some of the comparison counterparts. Overall, it presents a promising approach for estimating the accuracy of a deep network model.

Keywords: prediction accuracy estimation, unlabeled dataset, machine learning, convolutional neural network, vocal detection

1. Introduction

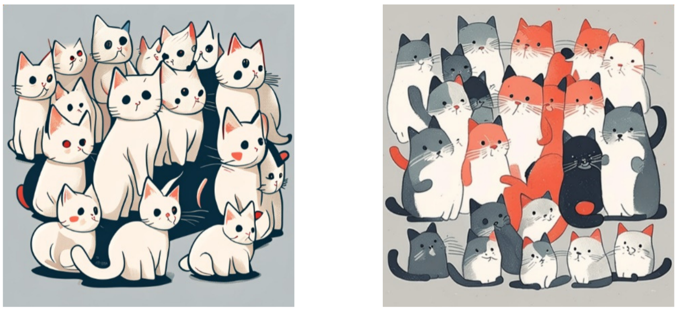
One of the major machine learning areas is supervised learning, which can be used to predict the labels (also known as classes) of unknown samples in real life. In this kind of application, it is vital to know the performance of the model. Such performance is typically measured based on accuracy. However, alternative metrics like the F1 measure are also widely used. In a typical experimental setting, the samples in the available (labeled) training dataset are split into training, validation, and testing sets. A model is trained with the training dataset, and its hyperparameters are tuned based on the validation results. Finally, the testing set is used to evaluate the model performance, such as accuracy, when encountered with new samples. It is a common practice to assume that the real, unknown samples have a similar distribution as the training dataset. With this assumption, the testing accuracy of the model obtained during model training is used to estimate the actual accuracy in real applications. In the following, true accuracy is defined as the ratio of correctly classified samples to the total number of samples, and estimation accuracy is defined as the accuracy inferred by other methods.

The concept of in-distribution and out-of-distribution is depicted in Fig. 1, where two sets of cats are presented. In Fig. 1(a), the cats consist solely of white cats, while in Fig. 1(b), cats of various colors are shown. Therefore, a white cat is an "in-distribution" sample in Fig. 1(a), but a gray cat is an "out-distribution" sample. Conversely, both white and gray cats are in-distribution samples in Fig. 1(b). From a machine-learning perspective, if a model is trained with only white cats, the model

[*] Corresponding author. E-mail address: scyou@ntut.edu.tw

English language proofreader: Yen-Chun Hsieh

may have difficulties identifying a gray cat as another instance of a cat. Unfortunately, in reality, it is usually unknown whether a test sample is similar to the training samples or not. Therefore, the accuracy obtained based on the dataset of Fig. 1(a) may not reflect the actual accuracy when predicting cats in Fig. 1(b).



(a) A set of white cats (b) A set of various colors of cats

Fig. 1 Illustration of in-distribution and out-of-distribution

In reality, it is difficult to infer whether two datasets have a similar distribution or not. To support this argument, previous studies [1-2] from the authors are presented. Two datasets were used in the studies, the Jamendo dataset [3] and the free music archive (FMA) dataset source from the FMA website [4]. Both datasets collect soundtracks of Western pop music. If a deep network model is trained with the Jamendo audio clips, the model reaches an accuracy of 94.1% when predicting the Jamendo test set. However, when predicting the FMA test dataset, the accuracy drops to only 82.7%. On the contrary, if the model is trained with the FMA audio clips, the accuracy is 92.5% for predicting the FMA test set, and 89.8% for predicting the Jamendo test set [2]. From the accuracy discrepancy, it is concluded that the FMA dataset consists of out-of-distribution samples of the Jamendo dataset. However, the Jamendo samples are within the distribution of the FMA dataset. Therefore, even though both datasets contain Western popular music, it is still difficult to answer whether the two datasets are "in-distribution" or "out-of-distribution". Consequently, it is questionable whether the test accuracy is a good indicator for unlabelled datasets or not.

It is practically useful if one can estimate the accuracy of a model when datasets with unknown distribution are presented, such as deciding whether to use the existing model or not. If the expected accuracy is sufficiently high (based on prior knowledge), the existing model can be continuously used. Otherwise, training a new model would be necessary by labeling some samples from the new dataset. In the vocal detection problem mentioned previously, the acceptable accuracy is around 90% as most vocal detection methods developed lately achieve an accuracy of 90% or higher [5]. Therefore, if the existing model is unable to reach such accuracy for a dataset, training a new model should be seriously considered.

Estimating the prediction accuracy of a model is a practical and important problem, but it seems to be overlooked by the research community. Bhaskaruni et al. [6] stated that they did not find any paper directly addressing this topic in 2018. In contrast to estimating the prediction accuracy of a model, many researchers focus on the problem of evaluating prediction uncertainty when the incoming samples are actually "out-of-distribution," and have developed several methods to evaluate whether the prediction of the model is trustworthy or not [7-9]. However, knowing the prediction uncertainty of a single sample is not sufficient to infer the prediction accuracy of the entire dataset as accuracy is calculated based on all samples instead of a single sample. At least, a conversion procedure is necessary (to be discussed in Section 4.3) [10].

In this paper, the singing voice (vocal) detection problem is used to carry out the concept of the block scaling quality (BSQ) approach. Singing voice detection is a crucial pre-processing step for many applications. For example, to remove the vocal sound from a singing soundtrack for karaoke singers, the audio segments with singing voices need to be located [11]. To summarize a Western popular song or extract its melody, the verse part that usually contains singing performance needs to be identified [12-13]. To recognize the singing performer in a music work, the singing segments need to be isolated before applying recognition techniques [14]. To convert lyrics to melody, the singing segments need to be known as well [15]. These examples show the importance of singing voice detection for various tasks.

The main contributions of this paper are:

(1) A new method called BSQ is introduced to estimate the accuracy of an unlabeled dataset. BSQ only needs one model and has advantages over the previous method proposed by You et al. [10], which will be discussed in Section 4.6.

(2) The paper compares BSQ with the following methods: reversed testing qualities (RTQ), multiple classifier qualities (MCQ), ensemble average qualities (EAQ), dropout qualities (DOQ), and last-layer Bayesian qualities (LBQ), some of which are originally developed for estimating the uncertainty of a sample instead of accuracy estimation. This paper also briefly explains how these methods are adapted to the problem of accuracy estimation in Section 4.3.

2. Related Work

Bhaskaruni et al. [6] initially studied the problem of estimating the prediction qualities, including classification error, F1-score, conditional prediction disparities, and area under curve (AUC) for datasets without ground truth in 2018. Following the idea of the reverse testing framework [16], their approach initially trains a model, called Model I, using a labeled dataset denoted as P. Next, Model I is used to predict (classify) samples of an unlabeled dataset Q. The predicted labels of samples from Q are treated as ground truth to train another model, called Model II. By using Model II to predict dataset P again, the measures, such as accuracy, can be estimated as dataset P has labels. Bhaskaruni et al. [6] called the obtained measures RTQ, and used RTQ to estimate the accuracy of Model I when predicting dataset Q. Although the RTQ approach is intuitively easy to understand, its performance seems not comparable to other approaches [10]. Nevertheless, this approach is chosen as a comparison counterpart in the experiments.

Previously, You et al. [10] proposed an approach, called MCQ to estimate the prediction accuracy by using multiple identical convolutional neural network (CNN) models with voting. By computing the ratio of majority votes to all votes on a given sample, one can compute the confidence value of the given sample. The average of the confidence values over the entire dataset is then used, with additional steps, to estimate the prediction accuracy of the model. This approach will be compared in the experiments in Sections 4.4 and 4.5.

Similar to the MCQ approach, Lakshminarayanan et al. [7] also used multiple identical neural networks in their approach, called deep ensembles. In contrast to the MCQ approach, however, their goal is to estimate the confidence level of a sample. As this approach also uses multiple identical models, it will be used as a comparison counterpart after a conversion procedure.

To estimate the confidence level of an unlabeled sample, Gal and Ghahramani [8] applied the Monte-Carlo dropout with rate p to a trained model. The dropout mechanism randomly removes some processing units and associated weights from the model. Therefore, multiple models can be generated if the dropout process is applied multiple times. Although this approach was not proposed for estimating the prediction accuracy, one can use the same conversion procedure to estimate the accuracy based on the generated multiple models.

The Bayesian network [9] differs from a conventional neural network. In the former, each of the connection weights is a probability distribution, whereas in the latter, it is a trainable but deterministic value. Since the weights in a Bayesian network are probability distributions, the prediction results vary when predicting a sample multiple times. This, in a sense, is equivalent to having multiple models. In the experiments, a variant of the Bayesian network, which contains only the Bayesian network on the last-layer (last-layer Bayesian) [17], will be used as a comparison counterpart.

Ovadia et al. [17] studied the relative performance of several approaches, including deep ensembles, Bayesian networks, vanilla, etc., when presenting out-of-distribution samples. They concluded that deep ensembles (described previously) seem to perform the best for various metrics with dataset shifts. In addition, they found that the deep ensemble approach may just need five models.

The prediction problem to be studied in the experiments is a vocal detection problem. In the present case, an audio clip of 2 seconds is presented to the model, and the model determines whether the vocal sound is present in the audio clip. Based on previous studies [1-2] conducted by the authors, it was found that separating the feature extraction step from the deep network model had better prediction accuracy. Overall, it is concluded that the "spectrogram plus CNN" model was better. Thus, this model will also be used in the experiments.

3. The BSQ Approach and the CNN Model Used in the Experiments

This section describes the BSQ approach. The first subsection describes the chosen CNN model. This model was hand-crafted to optimize the prediction accuracy [2]. The second subsection covers the proposed BSQ model, starting from the conceptual description, and then the algorithm of the BSQ approach.

3.1. SCNN model

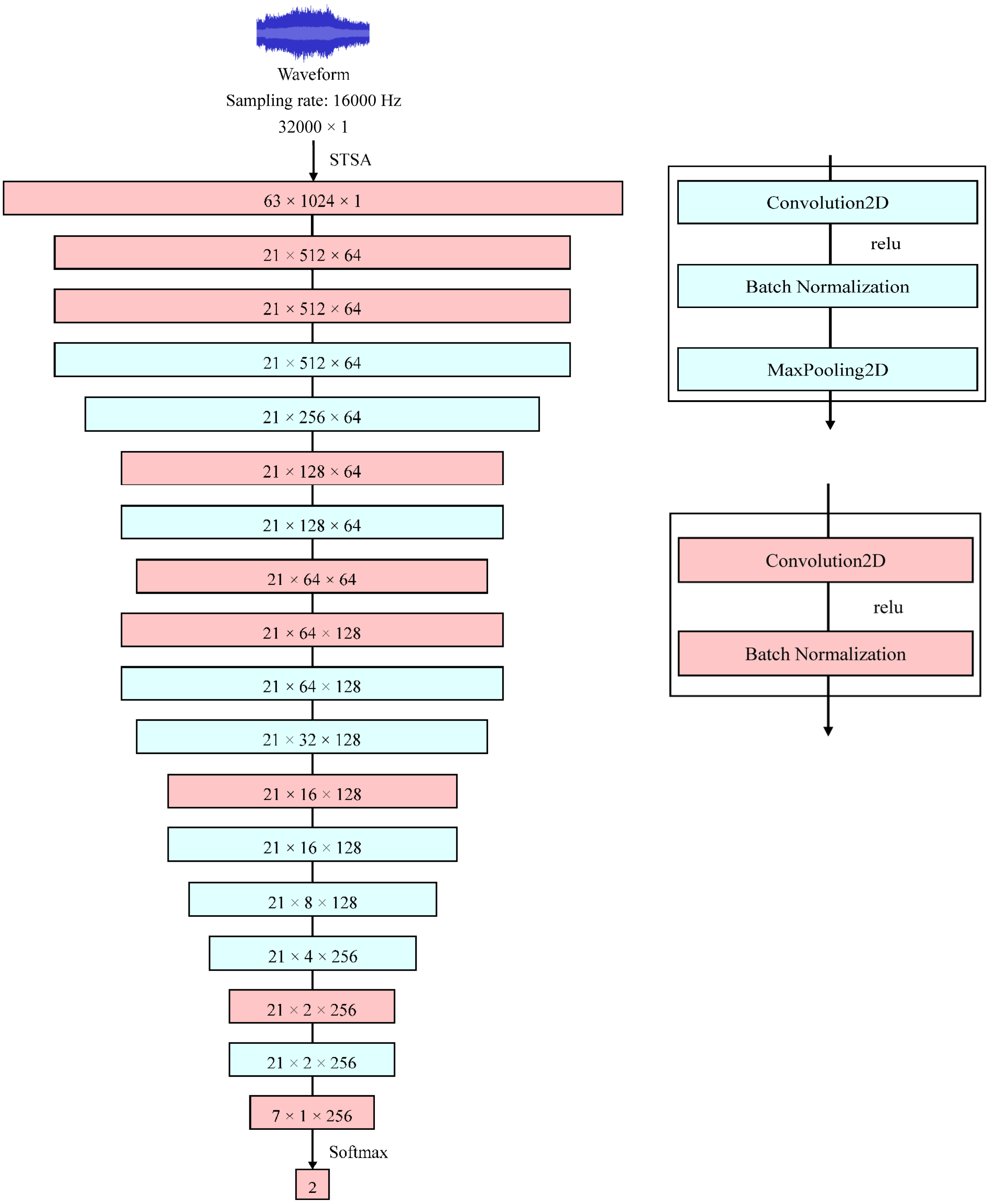


Fig. 2 The SCNN model used in the experiments

The experimental model employed is a CNN model. CNN models are a special type of neural network widely used in many image-related problems, such as for brain tumor detection [18] and tea image verification [19]. It is noteworthy that conventional approaches have also demonstrated a high AUC in breast cancer detection [20]. In the present study, the incoming audio clip is sent to a non-trainable network performing the time-to-frequency conversion. The output from this network is a spectrogram, used as the input to the CNN model. In the following, this model is denoted as a Spectrogram + CNN (SCNN) model. It is hand-crafted and optimized for the studied vocal detection problem [2]. In the experiments, all of the approaches employed are implemented based on this model, with possible modifications. If a particular approach needs more than one model, models with different initial weights are employed.

The experimental model is shown in Fig. 2 [2]. The input to the SCNN model is a 2-second audio clip, sampled at a rate of 16 ks/s. The audio samples are multiplied with 63 Hamming windows, each consisting of 512 coefficients, with a hop length of 512 samples. A pre-processing network performing the fast Fourier transformation (FFT) operation is utilized to convert these windowed samples to spectral coefficients. The resulting spectrogram is constructed on a time-frequency grid, where each cell is the energy of a spectral coefficient in the one-time instance. The dimension of the spectrogram is 63 × 1024, where "63" represents the number of time instances and "1024" represents the frequency bins (spectral values).

The FFT network is not explicitly given in Fig. 2, but only a mark of short-time spectral analysis (STSA). Therefore, the first box with numbers 63 × 1024 × 1 is the computed spectrogram. The spectrogram is processed by the next box labeled as "21 × 512 × 64", meaning that this layer accepts input feature maps of sizes of 21 × 512 and there are 64 kernels in this layer. In the figure, a box with a blue background includes the combination of a max-pooling layer, a convolutional layer, and a batch normalization layer, whereas a box with an orange background does not have the max-pooling layer. To be complete, Table 1 lists the used hyperparameters of this model. The nodes in all layers, except the output layer, use rectified linear unit (ReLU) as the activation function, and nodes in the output layer use softmax. The SCNN model used in this paper is designed for vocal sound detection on 2-second audio clips. This is a different task from detecting all vocal segments in a full-length soundtrack [5], which can also be done with the proposed SCNN model and some visualization tools [21]. However, this is beyond the scope of this paper and will not be discussed further.

Table 1 Hyperparameters of the model

Layer	No. of kernels	Kernel size	Padding	Max pooling	Stride
1	64	(3,2)	No	No	(3,2)
2	64	(1,2)	Yes	No	(1,1)
3	64	(1,2)	Yes	No	(1,1)
4	64	(1,2)	Yes	(1,2)	(1,1)
5	64	(1,2)	Yes	(1,2)	(1,1)
6	64	(1,2)	Yes	No	(1,1)
7	64	(1,2)	Yes	(1,2)	(1,1)
8	128	(1,2)	Yes	No	(1,1)
9	128	(1,2)	Yes	No	(1,1)
10	128	(1,2)	Yes	(1,2)	(1,1)
11	128	(1,2)	Yes	(1,2)	(1,1)
12	128	(1,2)	Yes	No	(1,1)
13	128	(1,2)	Yes	(1,2)	(1,1)
14	256	(1,2)	Yes	(1,2)	(1,1)
15	256	(1,2)	Yes	(1,2)	(1,1)
16	256	(1,2)	Yes	No	(1,1)
17	256	(3,2)	Yes	(3,2)	(1,1)
18	256	(1,1)	Yes	No	(1,1)

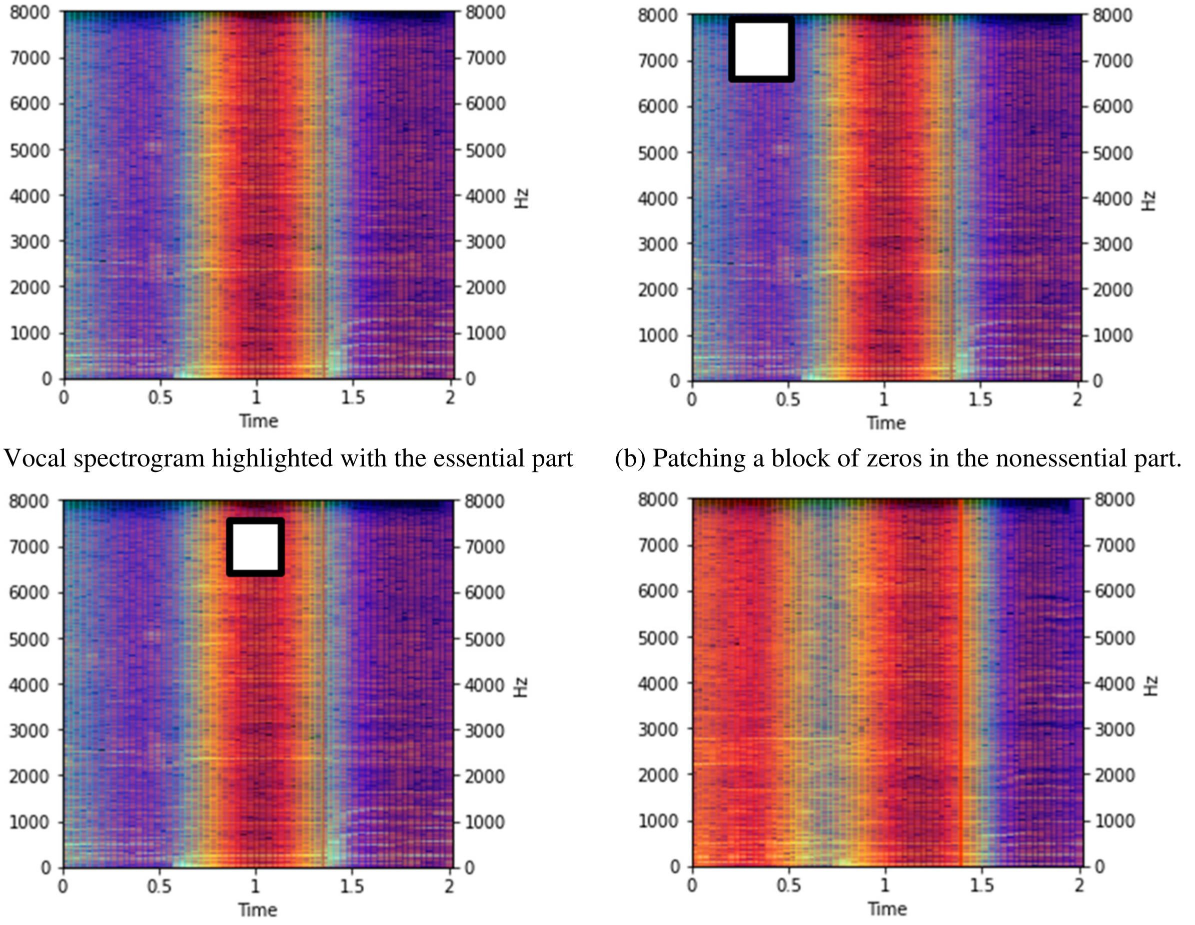
3.2. BSQ approach

The proposed block scaling approach is inspired by the concept of "local interpretable model-agnostic explanations" (LIME) [22]. The LIME approach has been successfully applied to applications such as chronic heart disease detection [23].

While the LIME is used to know which parts of the input image are essential in prediction, the BSQ approach uses this concept to measure the "relative robustness" of a model when the input image is altered. Specifically, the BSQ approach applies a window to the input image (in the experiments, the spectrogram), where the covered pixels are multiplied by a constant, such as 0 or 2, and the altered spectrogram is predicted again. By sliding the window from left to right and up to down, one can obtain many altered images with their corresponding prediction results. The number of prediction alternations over the total number of the block-scaled images is an indicator, called BSQ, measuring the robustness of the model in predicting incoming images.

If a sample has a low BSQ, it means that the prediction is mainly based on only a few spots in the image. This situation usually occurs when the model catches the essential parts of the image and uses them for prediction. On the other hand, if the model has a high BSQ for a sample, the model is very sensitive to any alternation of the image. Usually, it is an indication that the model does not capture the important portion of the sample. Therefore, the BSQ approach is intuitively understandable.

The idea of essential and nonessential parts of a spectrogram is illustrated by a vocal segment in Fig. 3(a), where the red and black colors highlight the essential part. This is the region that determines the label of the segment. In Fig. 3(b), the nonessential part is covered with a white block (equivalent to setting spectrogram coefficients to zeros), and the label prediction remains unchanged. However, in Fig. 3(c), the essential part is covered with a zero block, and the label prediction is likely to change. Thus, the number of changes reflects the size of the essential part. A small essential part implies a high confidence in the prediction, as shown in Fig. 3(a). A large essential part indicates a low confidence in the prediction, as shown in Fig. 3(d). Therefore, by counting the number of changes, the confidence level of each sample can be estimated and the average confidence can be used to measure the accuracy of the dataset. Based on this idea, other types of signals (such as obtained from machine vibration) or even general images could also use the approach mentioned in this paper.



(a) Vocal spectrogram highlighted with the essential part

(b) Patching a block of zeros in the nonessential part.

(c) Patching a block of zeros in the essential part.

(d) A large size of the essential part in a vocal segment.

Fig. 3 Spectrograms to illustrate the BSQ approach.

In summary, the BSQ approach is outlined below. It is presented in a format similar to an algorithm, but with more words to explain the steps.

(1) Train a SCNN model using the training dataset.

(2) Pick a sample from the unlabeled dataset S, denoted as $S(k)$, and predict its label with the trained model.

(3) Set $x = 0$ and $y = 0$. Set the scaling factor of the window as α. In the experiments, $\alpha = 0$. Set change-counter $c = 0$ and iteration counter $c_M = 0$.

(4) Set the window location to (x, y). Multiply the spectrogram coefficients covered by the window α.

(5) Use the trained model in step 1 to predict the label of the spectrogram in step 4 and record the result. If the new prediction result is different from the result in step 2, increase c.

(6) Hop the window according to hop lengths in the x-direction and y-direction.

(7) Increase c_M repeat steps 4 to 6 until the window slides through the entire spectrogram.

(8) Compute $r(k) = c/c_M$.

(9) Repeat steps 2 to 8 for all samples in S.

(10) Compute the average change rate

$$R = \frac{\sum_{k=1}^{T} r(k)}{T} \tag{1}$$

where T is the number of samples in S.

(11) Use R to build a linear model (given below), and use the linear model to predict the accuracy of the dataset.

In step 8, $r(k)$ is the ratio of altered spectrograms (from step 4) that have different prediction labels than the original spectrogram to the total number of altered spectrograms. This ratio reflects how much of the spectrogram is "essential" for the prediction. A large $r(k)$ means that the spectrogram of sample k is sensitive to random perturbation, and thus the prediction result is less reliable. Therefore, the average $r(k)$ is inversely related to the accuracy of the dataset.

In actual applications, a linear equation can be used to compute the estimated accuracy $\hat{A}$ from R, i.e.,

$$\hat{A} = \alpha_0 R + b_0 \tag{2}$$

where a_0 is the slope of the line and b_0 is the intercept of the Y-axis.

In case several labeled datasets are available, a linear regression algorithm [24] can be employed. Based on the calculated R and accuracy $\hat{A}$, it is easy to obtain the values of a_0 and b_0. If, unfortunately, there is only one labeled dataset, the R and $\hat{A}$ from the training and testing sets can be used to compute a_0 and b_0.

Before conducting the experiments, the authors perform a preliminary experiment to find a suitable window size and a hop length based on the fitness errors of the regression line (see Fig. 4). According to the preliminary experiment, it is observed that the proposed BSQ is not sensitive to the window size and the hop length. However, the estimation accuracy is highly sensitive to the chosen scaling factor α. The optimal values are 0 or 2. With the preliminary results, the following parameters are used in the experiments: Window size is$x \times y$, with $x = 30$ and $y = 40$, the hop length is 5 in the x direction and is 40 in the y direction, and the scaling factor $\alpha = 0$. Recall that the spectrogram has a size of 63 × 1024. The optimal parameters, such as window size, hop length, and α, may vary depending on the size of the spectrogram and need to be determined experimentally. To have a visual observation of the effectiveness of the BSQ approach, a simple regression over ten datasets is depicted in Fig. 4. The vertical direction in the figure is R and the horizontal direction is the actual accuracy. It shows that a linear model fits the data points (R, A) well.

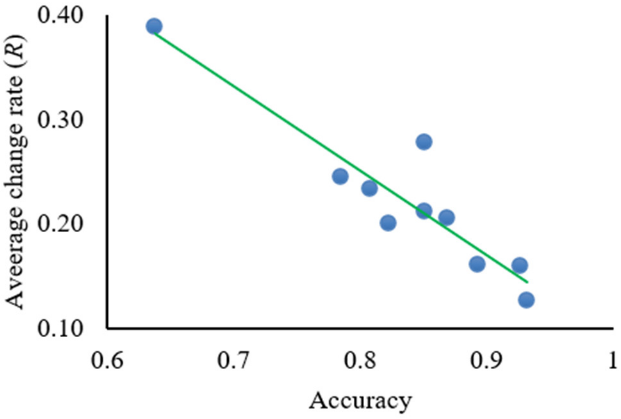


Fig. 4 Linear regression of the BSQ approach over ten datasets. The value R is from (1)

4. Experiments and Results

This section contains experimental datasets, settings, procedures, and results. The first subsection is the experimental datasets. It is followed by a description of experimental settings in the second subsection. The third subsection describes the comparison counterparts. The next two subsections present experimental results in two cases: (a) Using multiple labeled datasets to obtain the values of a_0 and b_0, as well as leaving only one dataset for validation, and (b) Using only one dataset to obtain the values of a_0 and b_0. The final subsection is the discussion and future work.

4.1. Experimental datasets

Table 2 Experimental datasets with descriptions

Dataset	Vocal clips	Non-vocal clips	Jamendo Train Accuracy	Comments / Descriptions
Jamendo train	6,981	6,376	-	Including training and validation soundtracks
Jamendo test	1,487	1,499	94.03%	Including only Jamendo test soundtracks
FMA-C-1 train	5,007	7,247	83.88%	From the FMA website, without genre balance
FMA-C-1 test	1,669	2,416	82.56%	-
FMA-C-2 train	5,277	8,475	87.33%	From the FMA website, with genre balance
FMA-C-2 test	1,759	2,824	85.79%	-
Test hard	4,746	3,545	66.02%	Collections of samples misclassified by a simple CNN model [18]
A-Cappella	7,922	0	94.77%	Pure vocal sound, no instruments
Instrumental	0	7,516	80.20%	Pure instruments, no vocal sound
KTV train	6,370	164	95.42%	Popular Karaoke songs from in-house video
KTV test	1,332	35	94.03%	-
MIR-1K	2,817	2,817	86.96%	From the MIR-1K dataset [19]
Chinese-CD train	3,060	2,163	90.70%	Popular Mandarin songs from CD soundtracks
Chinese-CD test	1,280	943	91.05%	-
Taiwanese-CD train	760	489	90.15%	Popular Taiwanese songs from CD soundtracks
Taiwanese-CD test	314	213	88.98%	-
Taiwanese-stream train	2,753	1,396	83.78%	Popular Taiwanese songs from Internet streaming (lossy compressed format)
Taiwanese-stream test	1,188	586	82.60%	-
Classical train	2,007	3,726	83.27%	Vocal segments from operas and non-vocal segments from orchestra
Classical test	847	1,597	82.06%	-

The datasets utilized are the ones employed in the previously published paper [9]. Datasets and source codes are available to the public (at https://github.com/NTUT-LabASPL). To be self-contained, a list of the datasets is given in Table 2 along with some descriptions. Among the datasets, the chosen training datasets are from Jamendo and FMA-C-1 with the suffix "train". The rest datasets are used for evaluating the estimation performance.

The following briefly describes the listed datasets. Jamendo train and Jamendo test datasets are excerpted from the Jamendo dataset [3]. FMA-C-1 and FMA-C-2 are obtained from the FMA dataset [5], but FMA-C-2 has a balanced number of samples for each broad music genre. Test-hard is an artificial dataset that consists of samples that a simple 4-layer CNN classifier [25] misclassified. A-Cappella is a collection of vocal-only works from the Internet, while Instrumental is a collection of audio segments with only instrumental sounds.

Karaoke TV (KTV) is a dataset extracted from Karaoke videos, and music information retrieval 1k (MIR-1K) is a dataset originally used for the MIR contest (available at https://sites.google.com/site/unvoicedsoundseparation/mir-1k). The experimental dataset contains audio segments from this dataset. Chinese-CD and Taiwanese-CD are datasets of segments from various Chinese or Taiwanese CD titles, mostly popular music. Taiwanese-stream is a dataset of segments of Taiwanese songs from the Internet. Classical is a dataset of classical music only, with non-vocal samples from orchestra works and vocal samples from solo or chorus in opera performances. In the experiments, only the Jamendo or FMA datasets are used for training, but the trained models are applied to many music genres to simulate the actual applications.

4.2. Experimental settings

The experimental platform is developed based on the Tensorflow framework (available at https://www.tensorflow.org/) and the Keras library (available at https://keras.io/). The dropout mechanism is used during training with a rate of 0.5. The chosen optimizer is the ADADELTA [26]. The number of training epochs is 200. The experiments are carried out using three computers, where each has three NVIDIA graphic cards. The used software versions and computer hardware are listed in Tables 3 and 4.

Table 3 Experimental software and versions

Software	Version
CUDA	11.2
Tensorflow	2.3.0
Python	3.6.9

Table 4 Computers used in the experiments

-	Processor	Memory	Graphic card	OS
Computer 1	Intel Core i7-6850K	32 GB	GeForce GTX 1080ti × 3	Ubuntu 20.04
Computer 2	Intel Core i9-7900X	40 GB	GeForce GTX 1080ti × 3	Ubuntu 20.04
Computer 3	Intel Core i7-6850K	32 GB	GeForce RTX 2070 × 3	Ubuntu 18.04

4.3. Comparison targets

To evaluate the performance of the BSQ approach, the following approaches are selected as the comparison targets: RTQ, MCQ, EAQ, DOQ, and LBQ. As the RTQ and the MCQ approaches have been described in Section 2, their implementation details are omitted here. The EAQ value is derived from the deep ensembles approach, the DOQ is from the Monte Carlo dropout approach, and the LBQ is from the last-layer Bayesian network approach. The deep ensemble approach uses multiple models. In the Monte Carlo dropout approach, one can generate multiple models by applying the dropout on the model. In the last-layer Bayesian network approach, one can predict the same sample multiple times to receive multiple prediction results, similar to the presence of multiple models.

As the EAQ, DOQ, and LBQ approaches are originally designed to measure prediction uncertainty, a procedure is necessary to calculate estimated accuracy from prediction uncertainty. To this end, You et al. [9] follow the procedure mentioned in the previous work. Let T denote the number of samples in the dataset, M the number of models, $p_{i,j}$ the softmax output of the vocal class for model i and sample j. The estimated quality is calculated as

$$EAQ,\ DOQ,\ or\ LBQ = \frac{1}{T}\sum_{j=1}^{T}\max\left(\bar{p}_j, 1-\bar{p}_j\right) \tag{3}$$

where

$$\bar{p}_j = \frac{1}{M}\sum_{i=1}^{M} p_{i,j} \tag{4}$$

In the experiments, the value of $M = 5$ used as 5 models might be sufficient for the deep ensembles [12].

4.4. Experiment results for leave-one-out cross-validation

As stated at the beginning of Section 4, there are two cases in the experiments. This subsection presents the results for case (a): using multiple labeled datasets to obtain the values of a_0 and b_0. In this case, one dataset is left out to compute the estimation accuracy, which is called leave-one-out cross-validation (CV) in the following.

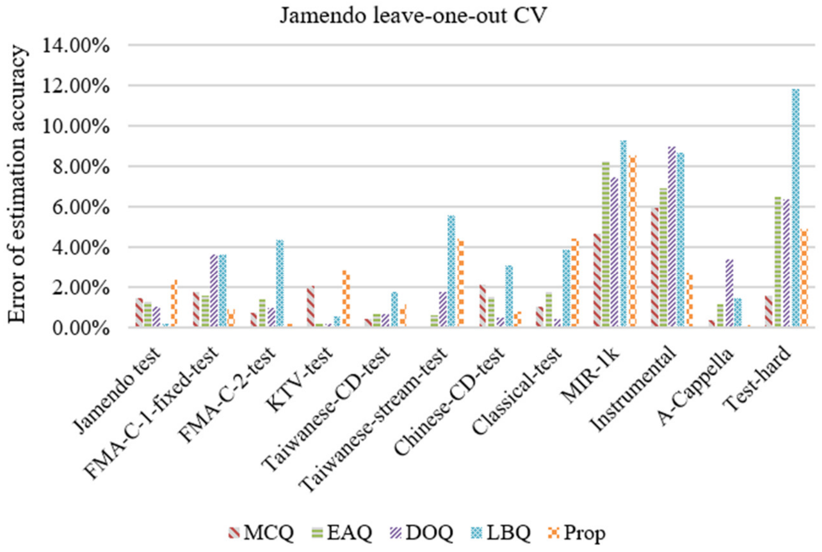
The next subsection reports the results for case (b), i.e., using one dataset (either Jamendo or FMA-C-1) to determine a_0 and b_0. The remaining datasets are used to compute the estimation accuracy. Case (b) is referred to as the “one dataset” in the following. The experiment in this subsection shows the best performance of the BSQ approach for the tested datasets, whereas the experiment in the next subsection demonstrates a more realistic scenario.

In the following, the estimation error for dataset S is calculated as

$$e_S = \left| A_S - \hat{A}_S \right| \tag{5}$$

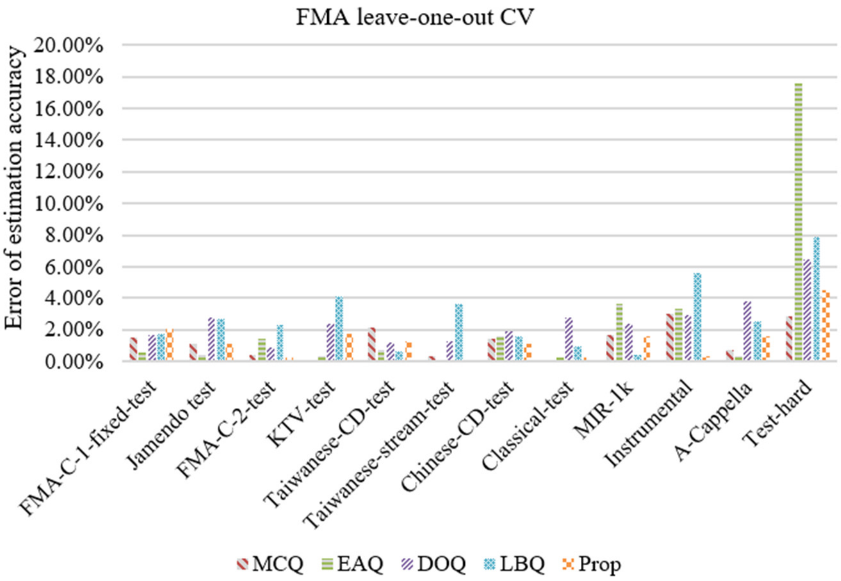
where A_S is the true accuracy (or averaged true accuracy for MCQ and EAQ) and the $\hat{A}_S$ is the estimated accuracy from Eq. (2).

Fig. 5 shows the estimation errors of leave-one-out CV for the models trained on Jamendo and FMA datasets. As the RTQ has very large estimation errors, this approach is not shown in the figure. Fig. 5 reveals that the Jamendo-trained model performs well on most datasets for the BSQ approach with errors of up to 5%. Only the MIR-1K dataset has an estimation error of 8%. However, most of the other approaches do not perform well on this dataset either.



(a) Results for Jamendo leave-one-out CV experiment

Fig. 5 Estimation errors of leave-one-out CV for various approaches



(b) Results for FMA-C-1 leave-one-out CV experiment

Fig. 5 Estimation errors of leave-one-out CV for various approaches (continued)

The BSQ approach performs even better with the FMA-trained model than with the Jamendo-trained model. The test-hard dataset has the worst performance, but the error around 4% is still acceptable. For all other datasets, the estimation errors of the BSQ approach are less than 2%.

Table 5 shows the average estimation errors of the datasets for different approaches. The BSQ approach has small estimation errors, less than 2.8% for the Jamendo-trained model and 1.3% for the FMA-trained model. The MCQ approach is the only one that performs better than the BSQ approach for both types of trained models. However, the MCQ approach requires multiple models, and the errors are based on the average accuracy. A discussion regarding the relative merits of the approaches will be given in Subsection 4.6.

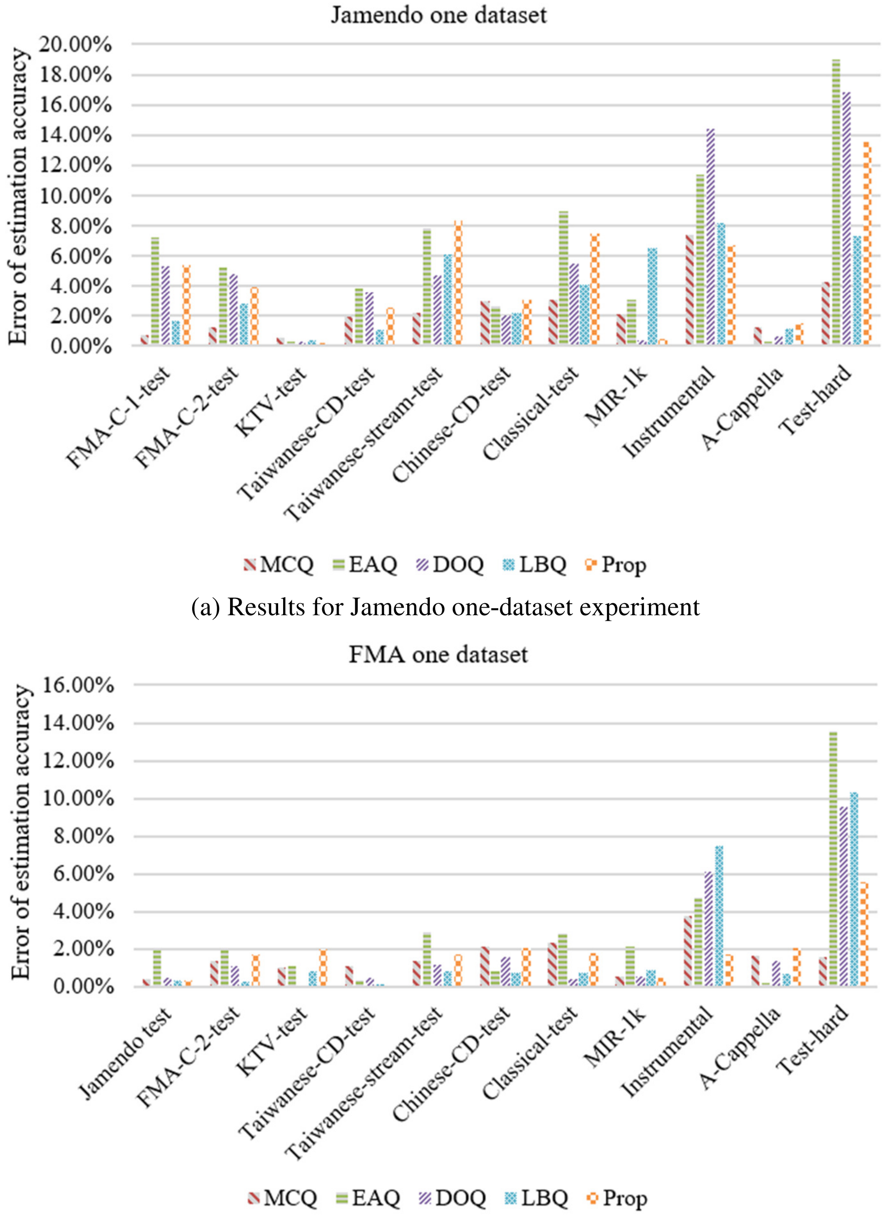
Table 5 Average estimation errors in various approaches for leave-one-out CV

Approach	Jamendo train	FMA-C-1 train
RTQ	7.02%	4.52%
MCQ	1.85%	1.27%
EAQ	2.64%	2.51%
DOQ	2.95%	2.53%
LBQ	4.52%	2.84%
Proposed	2.79%	1.31%

4.5. Experimental results for using one dataset

Fig. 6 shows the results of using only one dataset to fit the line in Eq. (2). The estimation errors are relatively larger with this method. The BSQ approach has estimation errors higher than 6% for some datasets with Jamendo-trained models. Only the MCQ and LBQ approaches have lower estimation errors in this training dataset. However, the BSQ approach performs much better with FMA-trained models. This time, the MCQ approach is the only one that outperforms the BSQ approach.

Table 6 shows the average estimation errors of the compared approaches. Notably, the BSQ approach has only 1.75% of the average estimation error for FMA-trained models. This level of estimation error is practically useful. Although the average estimation error in the Jamendo-trained models is higher, the error is mainly due to the test-hard dataset, as shown in Fig. 6(a). Recall that the test-hard dataset has a true accuracy of 66% (given in Table 2), which is far away from the typical accuracy level of 80% to 90%. The accuracy of this dataset is certainly more difficult to estimate. Overall, the performance of the BSQ approach is slightly worse than the MCQ approach. Other than the MCQ approach, the BSQ approach is better than others in the FMA training set and is only slightly worse than the LBQ approach in the Jamendo training set.



(a) Results for Jamendo one-dataset experiment

(b) Results for FMA-C-1 one-dataset experiment

Fig. 6 Estimation errors of using one dataset for various approaches

Table 6 Average estimation errors in various approaches for using only one dataset

Approach	Jamendo train	FMA-C-1 train
RTQ	12.60%	21.7%
MCQ	2.51%	1.55%
EAQ	6.34%	2.94%
DOQ	5.32%	2.06%
LBQ	3.78%	2.10%
Proposed	4.85%	1.75%

4.6. Discussion and future work

It is clear from Subsections 4.4 and 4.5 that the BSQ approach is slightly worse than the MCQ approach but better than other approaches in terms of average estimation errors. However, there are some shortcomings in the MCQ (and also EAQ) approach. Firstly, the MCQ requires 21 trained models, therefore the training time is much longer. Secondly, as 21 models are used in the estimation process, the predicted value is the average accuracy of the 21 used models. In other words, it is not possible to predict the accuracy of a specific model by using the MCQ approach.

The average accuracy used in the calculation of estimation errors introduces some uncertainty for each model, and the performance (accuracy) differences among the 21 models should not be overlooked. When examining the actual accuracy of the MCQ models in the experiments, one finds that the accuracy ranges from 91.3% to 94.8%, with a standard deviation of 0.71%, for Jamendo-trained models to predict the Jamendo test set. With this observation, the advantage of lower average estimation error from the MCQ approach may not hold if only one model is in use. In contrast, the BSQ approach estimates the accuracy based solely on the model of interest. Note that the BSQ approach is better than these other approaches requiring only one model, such as RTQ, DOQ, and LBQ. Overall, the BSQ approach is a good tool for accuracy estimation in the presented case.

The results in Table 5 and Table 6 show that using more than one dataset to calculate the regression line has lower estimation errors. To further improve the estimation performance with only one dataset, it is beneficial to artificially generate multiple datasets based on the only available dataset. Using the data augmentation technique is an approach to achieve this goal. This technique is widely used in conventional images as well as in medical images [27]. By performing random cropping or color distortion, additional training images can be obtained, and thus the trained model can achieve better accuracy. Following the idea of data augmentation, the authors plan to artificially generate multiple datasets with a linear-scale change, a pitch-scale change, or a signal-to-noise ratio (SNR) change. Hopefully, with the augmented datasets, the estimation errors can be further reduced.

5. Conclusions

This paper presents the BSQ approach to estimate the prediction accuracy of a deep network-based model. The approach computes the confidence level of each sample based on altered spectrograms and then uses the average confidence level as a parameter to estimate the accuracy of the dataset. Several experiments are conducted to compare the performance of the BSQ approach with others. Experimental results show that the BSQ approach is the second best among the evaluated approaches. Although the MCQ approach has smaller estimation errors than the BSQ, it only estimates the average accuracy of 21 models. Therefore, the MCQ approach is inappropriate for situations where only a specific model is of interest. Conversely, the BSQ approach has lower estimation errors than other approaches that use only one model. The experimental results demonstrate that the BSQ approach has low average prediction errors. Specifically, the errors are under 2.8% for the Jamendo-trained model and 1.3% for the FMA-trained model in the CV experiments. Furthermore, in the one-dataset experiments, the corresponding errors are 4.9% and 1.8%, respectively. Thus, it is a practical and effective approach.

In future work, the application of data augmentation methods is required to further enhance the estimation accuracy of the BSQ approach with only one labeled dataset. Although the BSQ approach is validated on audio segments, it might also be applied to image problems. Extending the BSQ approach to images is a future work.

Funding

This work was supported by the Ministry of Science and Technology, Taiwan through research grant MOST 111-2221-E-027-119.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] S. D. You, C. H. Liu, and W. K. Chen, “Comparative Study of Singing Voice Detection Based on Deep Neural Networks and Ensemble Learning,” Human-Centric Computing and Information Sciences, vol. 8, no. 1, article no. 34,

December 2018.

[2] S. D. You, C. H. Liu, and J. W. Lin, “Improvement of Vocal Detection Accuracy Using Convolutional Neural Networks,” KSII Transactions on Internet and Information Systems, vol. 15, no. 2, pp. 729-748, February 2021.

[3] M. Ramona, G. Richard, and B. David, “Vocal Detection in Music with Support Vector Machines,” IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1885-1888, March-April 2008.

[4] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A Dataset for Music Analysis,” https://doi.org/10.48550/arXiv.1612.01840, September 2017.

[5] X. Zhang, Y. Yu, Y. Gao, X. Chen, and W. Li, “Research on Singing Voice Detection Based on a Long-Term Recurrent Convolutional Network with Vocal Separation and Temporal Smoothing,” Electronics, vol. 9, no. 9, article no. 1458, September 2020.

[6] D. Bhaskaruni, F. P. Moss, and C. Lan, “Estimating Prediction Qualities without Ground Truth: A Revisit of the Reverse Testing Framework,” 24th International Conference on Pattern Recognition, pp. 49-54, August 2018.

[7] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles,” Advances in Neural Information Processing Systems, vol. 30, pp. 6405-6416, December 2017.

[8] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning,” Proceedings of the 33rd International Conference on International Conference on Machine Learning, vol. 48, pp. 1050-1059, June 2016.

[9] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight Uncertainty in Neural Network,” Proceedings of the 32nd International Conference on International Conference on Machine Learning, vol. 37, pp. 1613-1622, July 2015.

[10] S. D. You, H. C. Liu, and C. H. Liu, “Predicting Classification Accuracy of Unlabeled Datasets Using Multiple Deep Neural Networks,” IEEE Access, vol. 10, pp. 44627-44637, 2022.

[11] C. L. Hsu, D. Wang, J. S. R. Jang, and K. Hu, “A Tandem Algorithm for Singing Pitch Extraction and Voice Separation from Music Accompaniment,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 20, no. 5, pp. 1482-1491, July 2012.

[12] B. Logan and S. Chu, “Music Summarization Using Key Phrases,” IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100), vol. 2, pp. II749- II752, June 2000.

[13] J. Salamon, E. Gomez, D. P. W. Ellis, and G. Richard, “Melody Extraction from Polyphonic Music Signals: Approaches, Applications, and Challenges,” IEEE Signal Processing Magazine, vol. 31, no. 2, pp. 118-134, March 2014.

[14] Y. E. Kim and B. Whitman, “Singer Identification in Popular Music Recordings Using Voice Coding Features,” Proceedings of the 3rd International Conference on Music Information Retrieval, article no. 17, October 2002.

[15] Z. Ju, P. Lu, X. Tan, R. Wang, C. Zhang, S. Wu, et al., “TeleMelody: Lyric-to-Melody Generation with a Template-Based Two-Stage Method,” https://doi.org/10.48550/arXiv.2109.09617, November 2022.

[16] W. Fan and I. Davidson, “Reverse Testing: An Efficient Framework to Select amongst Classifiers under Sample Selection Bias,” Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 147-156, August 2006.

[17] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, et al., “Can You Trust Your Model’s Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift,” Advances in Neural Information Processing Systems, vol. 32, pp. 13991-14002, December 2019.

[18] S. Mohsen, W. M. F. Abdel-Rehim, A. Emam, and H. M. Kasem, “A Convolutional Neural Network for Automatic Brain Tumor Detection”, Proceedings of Engineering and Technology Innovation, vol. 24, pp. 15-21, August 2023.

[19] K. Y. Chen, C. Y. Chang, Z. R. Tsai, C. T. Lee, and Z. Y. Shae, “Tea Verification Using Triplet Loss Convolutional Network,” Advance in Technology Innovation, vol. 6, no. 4, pp. 199-212, October. 2021.

[20] S. J. Sushma, S. C. Prasanna Kumar, and T. A. Assegie, “A Cost-Sensitive Logistic Regression Model for Breast Cancer Detection,” The Imaging Science Journal, vol. 70, no. 1, pp. 10-18, 2022.

[21] S. D. You, H. C. Cho, and C. H. Liu, “Vocal Detection Using Convolution Neural Networks with Visualization Tools,” IEEE International Conference on Consumer Electronics-Asia, pp. 1-2, October 2022.

[22] M. T. Ribeiro, S. Singh, and C. Guestrin “Why Should I Trust You?”: Explaining the Predictions of Any Classifier,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135-1144, August 2016.

[23] T. A. Assegie, “Evaluation of Local Interpretable Model-Agnostic Explanation and Shapley Additive Explanation for Chronic Heart Disease Detection,” Proceedings of Engineering and Technology Innovation, vol. 23, pp. 48-59, January 2023.

[24] K. Kumari and S. Yadav, “Linear Regression Analysis Study,” Journal of the Practice of Cardiovascular Sciences, vol. 4, no. 1, pp. 33-36, January-April 2018.

[25] H. M. Huang, W. K. Chen, C. H. Liu, and S. D. You, “Singing Voice Detection Based on Convolutional Neural Networks,” 7th International Symposium on Next Generation Electronics, pp. 1-4, May 2018.

[26] M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method,” https://doi.org/10.48550/arXiv.1212.5701, December 2012.

[27] S. C. Pravin, S. P. K. Sabapathy, S. Selvakumar, S. Jayaraman, and S. V. Subramani, “An Efficient DenseNet for Diabetic Retinopathy Screening,” International Journal of Engineering and Technology Innovation, vol. 13, no. 2, pp. 125-136, April 2023.