
Article:
Tan, J. and Oyekan, J. (2021) Attention Augmented Convolutional Neural Network for acoustics based machine state estimation. Applied Soft Computing, 110. 107630. ISSN 1568-4946

https://doi.org/10.1016/j.asoc.2021.107630

Attention Augmented Convolutional Neural Network for acoustics based machine state estimation

Jiannan Tan and John Oyekan

The Department of Automatic Control and Systems Engineering, The University of Sheffield, United Kingdom, S3 1JD.

Abstract

The rapid development of technology is leading to the emergence of smart factories where the Artificial Intelligence paradigm of deep learning plays a significant role in processing data streams from machines. This paper presents the application of Augmented Attention Blocks embedded in a deep convolutional neural network for the purposes of estimating the state of remote machines using remotely collected acoustic data. An Android application was developed for the purposes of transferring audio data from a remote machine to a base station. At the base station, we propose and developed a deep convolutional neural network called MAABL (**M**obileNetv2 with **A**ugmented **A**ttention **Bl**ock). The structure of the neural network is constructed by combining an inverted residual block of MobileNetv2 with an augmented attention mechanism block. Attention Mechanism is an attempt to selectively concentrate on a few relevant things, while ignoring others in deep neural networks. Due to the presence of audio frames containing silent features not relevant to the task at hand, an Attention Mechanism is particularly important when processing audio data.. The MAABL network proposed in this paper obtains the state of the art results on the accuracy and parameters of three different acoustic data sets. On a relatively large-scale acoustic dataset regarding machine faults, the method proposed in this paper achieves 98% accuracy on the test set. Moreover, after using transfer learning, the model achieved the state of the art accuracy with less training time and fewer training samples.

Keywords: Attention Block, Deep Learning, Estimation, Machine States, MobileNetv2

1.0 Introduction

In recent years, the trend in smart factories and Industry 4.0 has gradually emerged. Some multi-national companies are aiming to better understand the state of their equipment by collecting data from their production machines distributed across the globe. For example, in the fast-moving consumer goods sector, production machines in factories across the global are used to produce parts for a variety of goods including aerosol sprays, shampoos, shower gels to mention a few. Being able to remotely

collect data and understand the production cycles of these machines as well as how they are used will play an essential role in streamlining logistics, efficiency and creating better maintenance programs. Traditional techniques for monitoring the state of machines required installation of sophisticated sensors (such as temperature, humidity, and pressure sensors) which produce data. Information is then obtained from the data through a large amount of human processing. This takes a huge amount of time and does not enable the companies to use the information in real time for competitive advantageous decisions. Furthermore, the aforementioned sensors are prone to damage due to the harsh factory environment and close contact with the machines.

Nevertheless, due to the advancement of remote sensing technology, computer hardware as well as the broad application of deep learning technology in various fields, the use of acoustic data for analysis of factory machines has begun to appear in various studies [1][2][3][4]. Acoustic data collection equipment do not need to be in close contact with machines and the acquired acoustic data usually contains a wealth of information. Due to its robust feature extraction capabilities, deep learning, combined with acoustic data, offers promises of improved remote monitoring of factory machine states.

As a result, the aim of this paper is to estimate the state of remote machines through the application of deep learning algorithms to remotely collected acoustic data. Our contributions are as follows: (1) We developed a technique that enables a readily available, reusable and portable sensor module to collect data from a remote machine and transfer the data to a base station; (2) We developed algorithms in the form of a light-weight neural network model to enable near real-time estimation of remote machine states and (3) We show that transfer learning can be used to reduce the time needed to collect training data for neural networks deployed on manufacturing floors.

The rest of the paper is organised as follows: section 2 presents the relevant literature review on the techniques proposed in this work, section 3 discusses the methodology that was used while section 4 discusses the experiments and results that were carried out and obtained respectively. We conclude in section 5 with discussions on the results and future work.

2.0 Literature Review

Sound plays a vital role in human interaction. The human brain can analyse the content, emotions, and even intent of the talker through acoustic profiles of the human voice in the context of the surrounding environment [2]. Furthermore, an experience factory worker can estimate and predict the states of factory machines by just listening to the sound emitted by the machines. As a result, the success of

future human robot or AI collaborations in factory could also rely on how to make these automated systems understand acoustic data like humans [3][5]. One step in this direction is the recent development of Convolutional Neural Networks (CNN). The application of convolutional neural networks (CNN) and other deep learning paradigms in feature mining of acoustic data has broad application prospects and has led to an increase in research on detection and classification of acoustic data in recent years.

Prior to the emergence of CNNs, many methods were proposed to extract the features of acoustic data towards capturing the information contained in the frequency domain. Among them, the most widely used and still used in the industry today is the Mel-frequency cepstral coefficients [8]. However, these methods are susceptible to low-frequency components and this hinders their accuracy in industrial applications. Driven by the need for increased accuracy requirements, sophisticated industrial applications and the powerful feature extraction capabilities of deep learning algorithms, many studies have started combining the aforementioned methods with deep learning algorithms to achieve better results on classification tasks [6] [9].

CNN is a powerful feature extractor that is modelled on the visual cortex of humans and performs well on image datasets (Figure 1 [13]). As a result, an increasing amount of studies [10] [11] [12] have converted acoustic data into image data through the proposed Mel spectrogram. This approach transfers acoustic problems to relatively mature computer vision problems as well as offers the possibility of applying CNN to acoustic data.

In CNNs, through the addition of convolution layers, more sophisticated filters can be learned and used to extract more complex features. The unique combination and arrangement of these layers has opened up a variety of novel convolution neural network architectures such as DenseNet[15], GoogleNet [16], ResNet [14] to mention a few. Due to the use of skip connections and stacks of residual units, the ResNet architecture is a very powerful architecture that has had a profound impact on CNN network structures that appear later in literature [17][18][19][20]. Nevertheless, due to deep layers, the number of parameters of ResNet are large and this could require a timescale of weeks for training such networks. As a result, this makes it practically infeasible in real-world applications. Therefore, to retain the function of ResNet in feature extraction as much as possible and reduce the number of parameters of CNN, Sandler et al. [17] proposed a light-weight CNN network called the MobileNetV2. The MobileNet V2 can obtain the approximate feature extraction effect of ResNet but with a smaller number of parameters.

Furthermore, since the architecture of convolutional neural networks could have an impact on performance, researchers have started investigating how these architectures could be discovered automatically and how the discovered architectures affect learning performance. This has contributed to the learning to learn paradigm in which a computer constructs a network according to a cost function made up of accuracy, given parameter limits and the basic network structure [21][22][23][24]. Some network structures obtained by this method are not only more capable of feature extraction but also have smaller parameters for deployment on resource constraint platforms such as mobile phones. However, vast computing resources are still often required offline to grid search the hyperparameters of the network structure and this is not conducive for small teams lacking intensive computing resources.

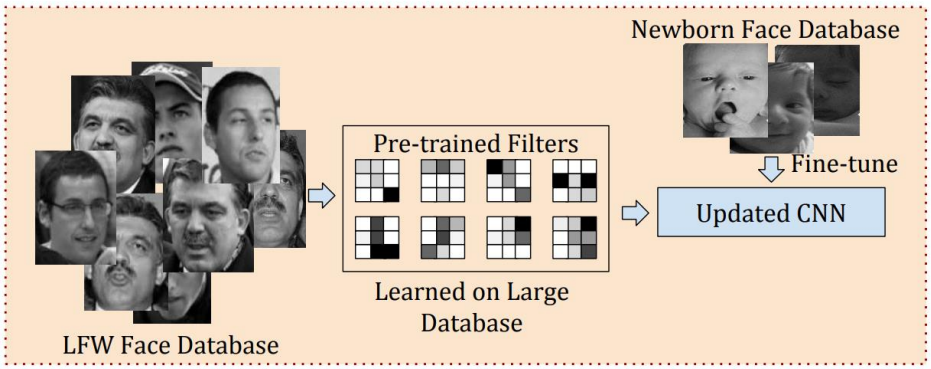


Figure 1. Applying CNN in transfer learning of adult human face recognition to baby face recognition [13]

Nevertheless, MobileNetv2 is a light-weight CNN network that allows very memory-efficient inference on image detection tasks with high accuracy. However, it should be noted that acoustic data is a time series data type and as a result, an element of sequence is needed in the architecture in order to make it suitable for processing audio data successfully. In acoustic data, the previous sequence information and the following sequence information in the audio data are equally essential for the analysis of the entire acoustic data. As a result, many studies [10] [25] have utilised Recurrent neural networks (RNN) to capture the connections between the context and sequence of acoustic data, thereby improving accuracy of detection and recognition. However, when RNNs capture long-distance sequence features, they forget the initial sequence features learned prior. As a result, Hochreiter et al. [26] proposed an improved RNN network called the Long Short-Term Memory (LSTM) that up to a certain extent, overcomes the shortcomings of long-distance sequence features in RNN.

However, this LSTM was unidirectional and did not have the feature of embedding context. Therefore, Graves et al. [27] proposed bidirectional LSTM networks (BiLSTM) to achieve contextual features of sequence data in networks. This has spun off a number of architectures that take time sequence into consideration, such as the Gated Recurrent Unit network, and also solve the challenge of the lack of long-distance dependence of simple RNNs [28][29[30]. Nevertheless, since acoustic data often contain a lot of noise as well as silent audio frames that are not effective for understanding the task itself, a mechanism is needed to ensure that an AI system can focus on vital information and ignore the information that is not important to the current task. Towards solving this, Vaswani et al. [31] proposed the use of an attention mechanism in deep learning structures.

Attention Mechanism is an attempt to selectively concentrate on a few relevant things, while ignoring others in deep neural networks. It was discovered that this had a significant improvement in accuracy compared to just using LSTM. As a result, research has begun to focus on the introduction of attention mechanism to the feature maps in CNNs. For example, Convolutional Block Attention Module was proposed in [32]. However, this method calculated the channel and spatial features separately. Therefore, Bello et al. [33] suggested an attention augmented convolutional network (AA) with a step unit that combines the spatial and feature spaces to optimise the implementation of the attention mechanism. It was discovered that after applying an attention mechanism to the feature map of a chest radiograph image, some of the features in the feature maps were enhanced [34]. In other words, the network was able to capture richer features and thus achieve better accuracy.

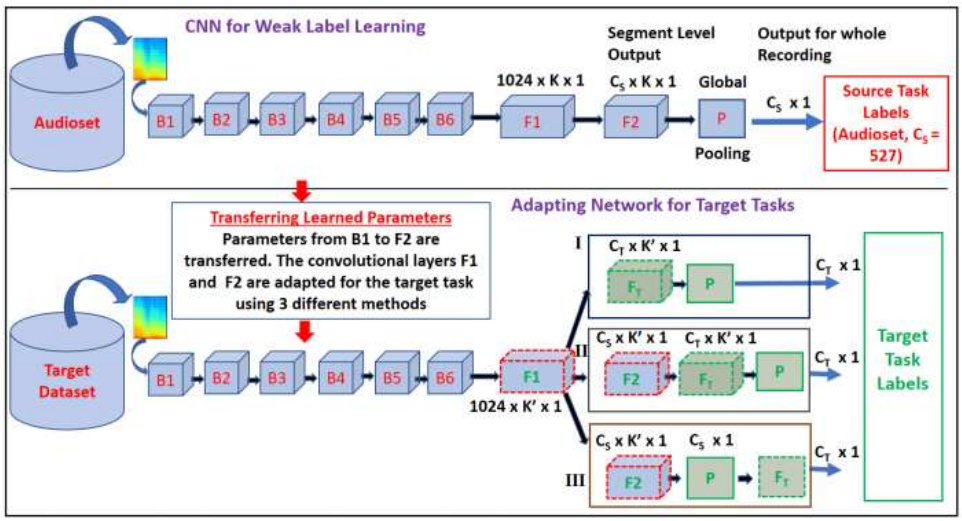


Figure 2. An example of how the parameters of a network can be transferred and how the new network is modified for the target data set in transfer learning for Audiosets [35].

However, even though the above results from literature are impressive, it should be noted that one of the characteristics of traditional deep learning is the need for large-scale data sets to enable models learn sophisticated features. In practical applications in industry, the cost of data collection is high. Therefore, there is a requirement that the model learns new data models with less training data, less training time as well as achieve high accuracy. This has become the focus of research in utilising deep learning in industry [35][36][37]. Towards addressing this, the paradigm of Transfer Learning is increasingly being used.

Transfer learning is a method of sharing extracted features on similar data sets. Researchers in convolutional neural networks have discovered that low-level convolutional layers are universal and represent the lower-level features in images, such as edges and corners. Therefore, the parameters of this part of the neural network can be frozen and the parameters of the higher-level convolutional layers and prediction layers updated (Figure 2). As a result, this can be transferred across domains and significantly speed up training while ensuring accuracy [38].

Figure 3. Noisy environment in the factory [39]

From above, it is proposed in this paper that the combination of CNN, LSTM, attention mechanism, and transfer learning [10][25][40][41] can be applied to the accurate detection and recognition of machine states using acoustic data. Although these neural structures have achieved excellent results on open-source data sets, the actual application effects in the production process are still unknown. This is because as shown in Figure 3, the real industrial scene contains various noise, including human talking and noise from other equipment. Furthermore, an industrial-grade user requires near real-time processing of data which means that the neural network model size needs to be small enough for fast

processing. However, current models contain massive number of parameters, which has an impact on practical applications. The current state of the art method [33] combines the CNN and attention mechanism approaches and achieves a state of the art accuracy in public data sets. However, its skeleton is ResNet, and its weight is about ten orders of magnitude larger than the previously mentioned MobileNetv2 [18]. Furthermore, the model from [33] and [18] were applied to data sets where the input file was not acoustic data but image data instead. This paper aims to improve on the results of [33] while building a light-weight neural network model, based on [18], for the purpose of predicting machine states by acoustic data. Transfer learning was also studied in order to investigate the model's flexibility and adaptability on various industrial-grade acoustic data sets.

3.0 Methodology

In this section, we discuss our proposed system and its components as well as the proposed neural network architecture that we used in this work. Our proposed system was designed and developed with the intention of evaluating remote machine states continuously.

In this work, we followed an incremental process made up of two stages in the development of our algorithms. In the first stage, we constructed a scaled down prototype plant for the purposes of generating data and training our neural network. After building the prototype plant, an application capable of wirelessly transmitting machine acoustic data and GPS location information to a remote server was developed and deployed on a mobile phone. On this server, a neural network model was developed to process the data sent by the application.

In the second stage, we made use of the concept of transfer learning to transfer our trained model for the purposes of estimating the states of industrial machinery using audio data.

3.1 System overview

The architecture of the entire system is shown in Figure 4. Due to the huge improvements in sensors (audio, GPS, temperature etc), telemetry capability, portability and ubiquitousness, we proposed the use of mobile phones to collect data. In our current experiments, the mobile phone sends data collected from remote production machines to a remote server through Bluetooth communication. In the future, this will be replaced by a communication system that uses the phone's telemetry system to send data from the production machines to the remote server.

Once the remote server received the acoustic data, it first converts the acoustic data into RGB image data using the approach discussed in section 3.5. These RGB images were then used as input to the

neural network model proposed in this paper. The neural network extracted the features in the image. According to the extracted feature information, the model predicts the state of different machines using probability. The model finally selects the highest value of the predicted probability as the estimated result.

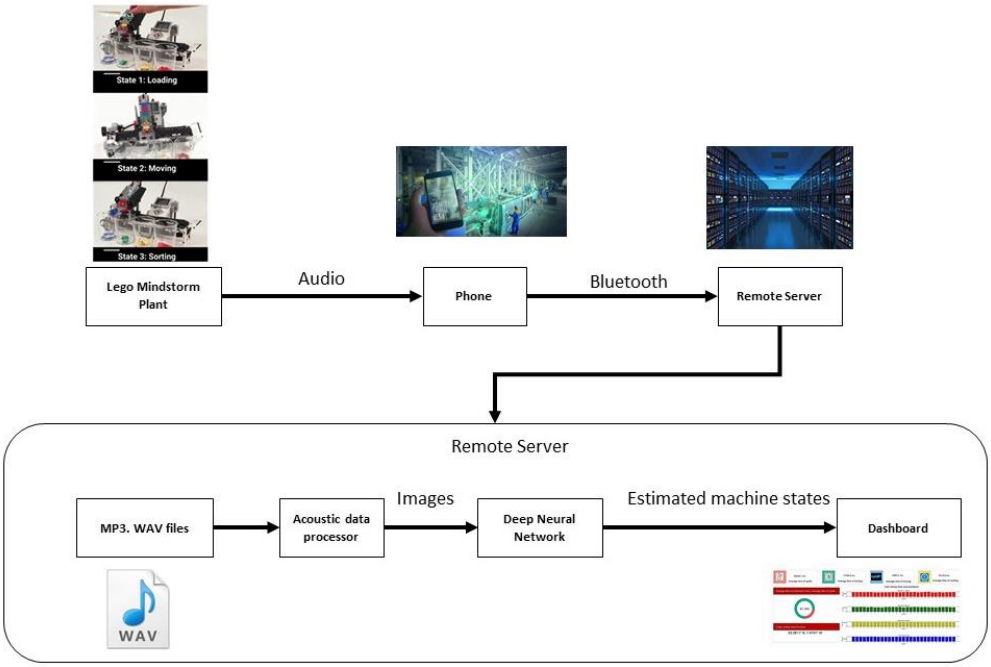


Figure 4. System Overview

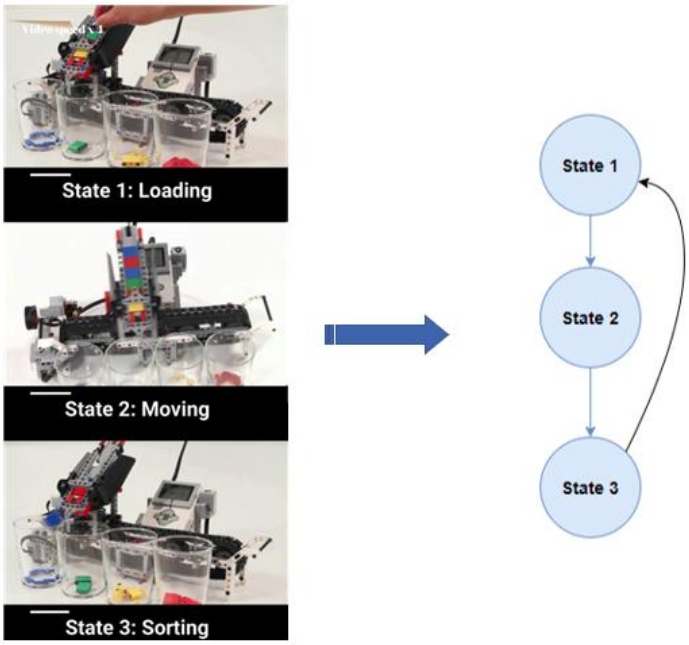


Figure 5. State diagram for Lego Mindstorm Plant

3.2 Lego Mindstorm Plant

As discussed above, a miniature plant based on a Lego Mindstorm EV3 [48] was developed and used to generate an acoustic data set. Compared with other versions of Lego Mindstorm, the EV3 version has more modules. Therefore, a plant with a complex structure and various machine states could be built. This project used this version of Lego Mindstorm to construct a colour sorting robot that sorts color blocks into containers (Figure 5). As shown in Figure 5, the state diagram of this robot consists of three parts, namely loading, moving and sorting. Every cycle starts with loading and ends with sorting. This is repeated continuously until stopped. During experiments, a phone was placed close to the constructed Lego Mindstorm Plant in order to ensure that sounds from other sources in the environment had as little impact as possible on the collected dataset (See Figure 12).

3.3 Phone

The phone used in this project is an Android phone. We developed an Android Application (APP) and installed it on the phone. Figure 6 shows the structure of the APP of this system. This APP consisted of four modules. The four modules were Time Module, Recording Audio Module, GPS Module and Bluetooth Module. At regular intervals, the APP sends a file recording of the acoustic data obtained from the plant. This file is tagged with the GPS location data of the production machine and the time the file was generated to the remote client through the Bluetooth module every minute. The server determines the location of the current device and the corresponding time point according to the GPS information and time information in the file name. The server then begins to analyse the acoustic data generated by the production machine.

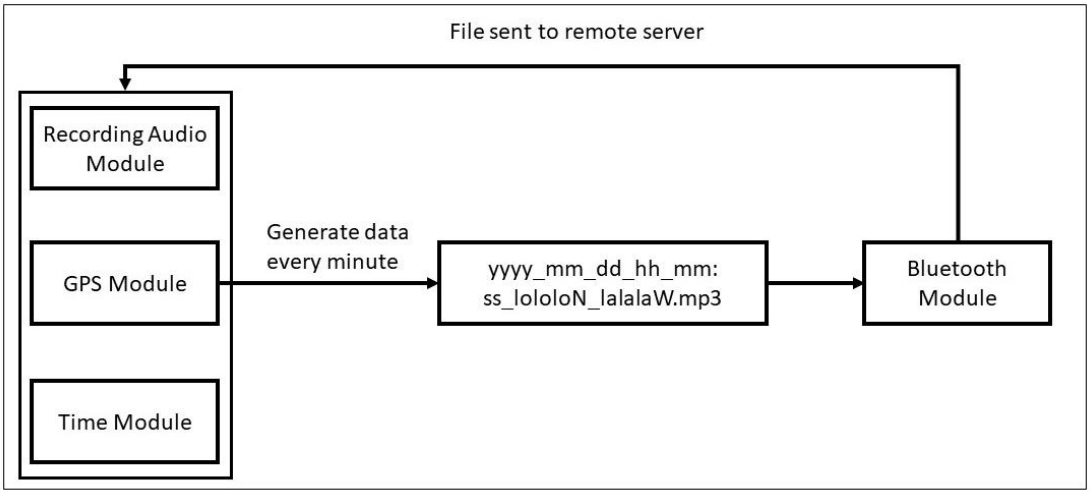


Figure 6. Android APP Architecture

3.4 Server

The server in this work was a computer with a high-performance graphics card. It received audio files from the phone via Bluetooth and stored the files in a data set folder. As shown in Figure 4, this server contains three modules. They are an acoustic data processor, a deep neural network and a dashboard. After the acoustic data passes through these modules, the machine state it represents is obtained and displayed visually to a user. These modules will be described in detail in the following sections.

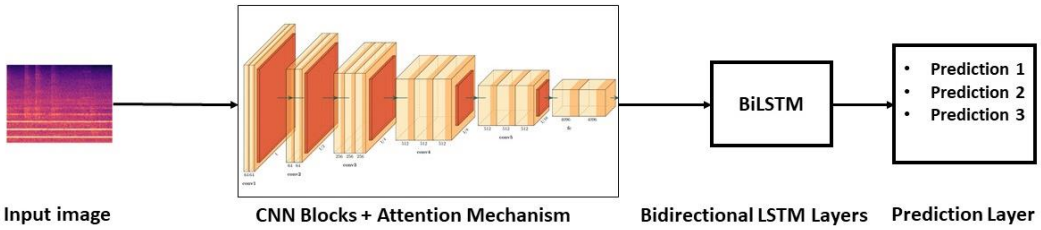
3.5 Acoustic data processor

Received acoustic data was converted into pictures and used as the input to the proposed neural network structure. In order to achieve this, first, the acoustic signal was processed by a short-time Fourier Transform with a window size of 1024 and hop length of 512. The processed signal was a power spectrogram (resampled to 22050 Hz and normalised). Then the Librosa implementation [42] maps the power spectrogram onto the Mel scale with 128 Mel filter bank and obtains the Mel spectrogram. Equation (1) calculates the Log-scaled Mel-spectrograms where the ref is a scalar in this formula. Finally, the two-dimensional Log-scaled Mel-spectrograms cascaded its delta information into a RGB image. The RGB image was used as the input to the neural network model $X \in R^{\mathrm{W}\times\mathrm{H}\times 3}$ where W and H are the width and height of the images respectively. The conversion of sound into a three-dimensional image could be considered as an operation of compiling and obtaining more information in higher dimensions.

$$log\_Mel = 10log\left(\frac{S}{ref}\right) \tag{1}$$

3.6 Network architecture

As shown in Figure 7, the network structure proposed in this paper consists of an input, CNN layers and attention mechanism layers, BiLSTM layers as well as a prediction output layer. The image of the input layer is obtained after processing the acoustic data as discussed in the previous section. The image also underwent normalisation processing. In other words, all pixel values were mapped from the original 0-255 to 0-1 thereby reducing the number of subsequent operations in a vast data set. Then the size of all images is unified to 160 x 160, which is convenient for comparison with other states of the art models at the same input scale.



(a) Showing the input image and the developed neural network architecture

(b) A closer view of the signal going into the neural network architecture and the processing technique used to segment the signal for the neural network architecture

Figure 7: The developed neural network architecture and signal processing technique

In Figure 7b, the Figure on the left shows the Mel spectrogram of the acoustic data in one cycle operation. In the Mel spectrogram, the abscissa was the time axis, the ordinate was the frequency, and the colour depth was the energy level of the signal. There was a certain time interval between each state of the machine. To better separate each state from the background noise, the original image was binarised with a threshold of 0.8, and the middle image in Figure 7b was obtained. This was achieved using the OpenCv library [40]. On the time axis of the binary image, states are separated from each other when the distance of the adjacent white dots exceed 6 pixels. The height is the last white dot in the direction of the vertical axis of each state. By using this strategy, it was possible to segment the states from each other as shown in the rightmost picture of Figure 7b. The segmented states are shown separated by black boxes on the Mel spectrogram and the sub-pictures in each box were used as separate inputs to the proposed neural network's CNN layer for feature extraction. Furthermore, since the convolutional neural network can be viewed as a computational paradigm in which an ensemble of filters is learnt by the computer, this enables us to rely less on humans to perform manual filter

construction in order to denoise the signal. Instead we rely more on the powerful feature extraction properties of the CNN to extract the salient properties in these images and consequently, the salient properties in the audio input.

In this work, the CNN layer uses the MobileNetv2 as a base but with the introduction of an Augmented Attention mechanism as shown in Figure 8. As a result, we denote our proposed network architecture as MAABL (**M**obileNetv2 with **A**ugmented **A**ttention **Bl**ock). We shall now discuss the attention block in more detail.

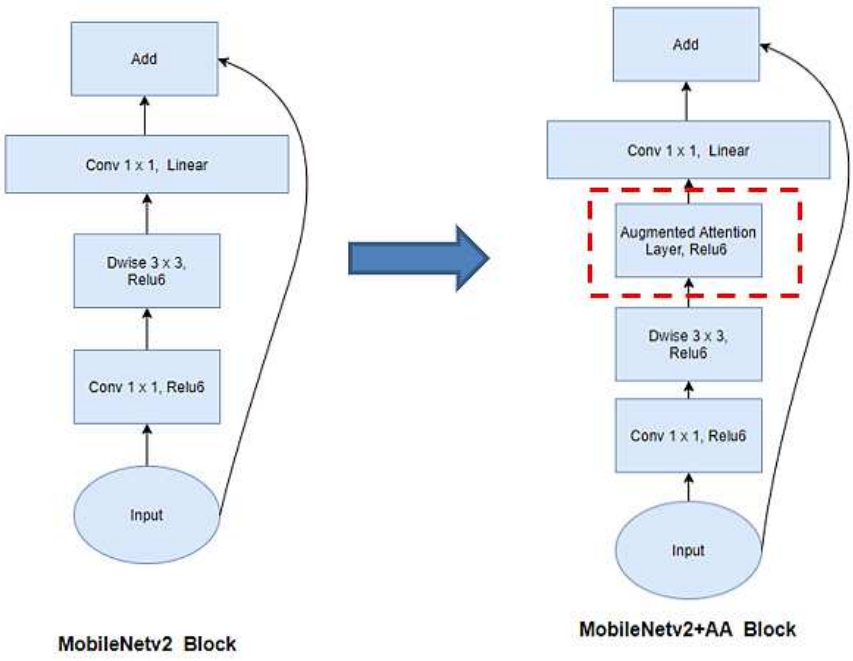


Figure 8. Proposed architecture in this paper combined a MobileNetv2 block with an attention mechanism block as highlighted in red on the left.

3.6.1 Augmented attention block mechanism:

The structure of the attention block is shown in Figure 9. Similarly to [33], we placed the attention block after a feature map. As a result, the input to the attention block is the feature map of the image from the previous layer. Attention maps for each spatial location (W, H) on the feature map are computed by using queries and keys where the queries are constructed from the current input values and the keys constructed from the feature map values. As shown in Figure 9 and using similar notations in [33], N_h refers to the number of heads which in this case is 2 in a multihead-attention (MHA) configuration. The heads contain the weight averages of values which are concatenated, reshaped to the original volume spatial dimensions and then mixed with a pointwise convolution. As shown in

Figure 9, a standard convolution pipeline runs in parallel with the generated attention maps. The two pipelines are merged at the end into an output. We refer the reader to [33] for more information on this.

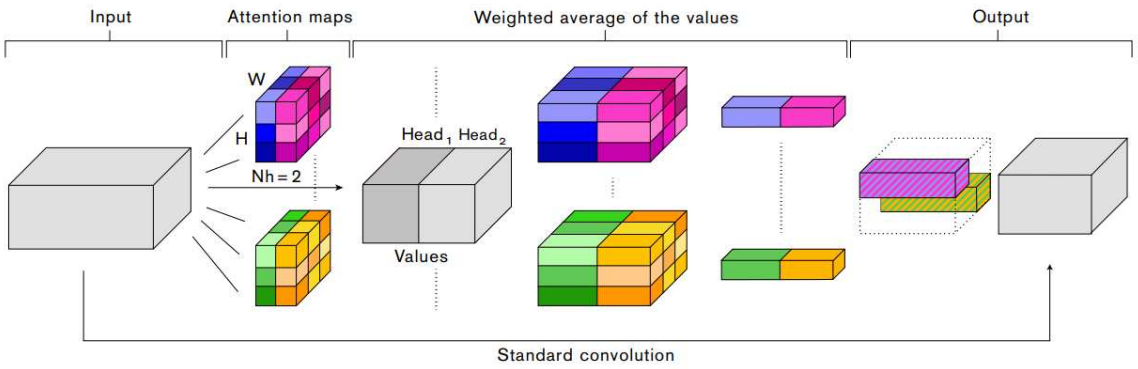


Figure 9. Augmented attention block layer [33]

In order to introduce the attention mechanism to MobileNetv2, MobileNetv2 needed to be modified specifically. As shown in Figure 8, the Augmented Attention Layer is added to the depthwise convolution in the MobileNetv2 block. Equation 2 is the expression of Relu6 in Figure 8. Sandler et al. [18] pointed out that the activation function is a modification of ordinary ReLU. However, the maximum output value is limited to 6, meaning that the output value is clipped. In this way, when the GPU of a mobile device is at a low precision of Float16, an excellent numerical resolution can still be achieved. If the activation range of ReLU is not limited, an output range of 0 to positive infinity is obtained. This results in a broad range distribution and as such, the low-precision Float16 will not be able to accurately describe such an extensive range of values thereby causing a loss of accuracy.

$$y = min(max(0, x), 6) \tag{2}$$

3.6.2 Bidirectional LSTM layers

After the image has gone through the CNN layer and the attention block, a feature map with mined features in space and channel is obtained. Since these image data are not images in the conventional sense but are converted from acoustic signals in the time domain, the next step is to mine the information of the images in the time series. We made use of a Bidirectional Long Short Temporal Memory (BiLSTM) neural network for this purpose. Figure 10 shows the processing of serialised features by BiLSTM. The Forward Layer records the previous information and finds the connections

with the following event. The Backward Layer is used for finding connections between previous events based on what happened later. The outputs of the forward LSTM units and the backward LSTM units are combined in an activation unit to generate an output *y*.

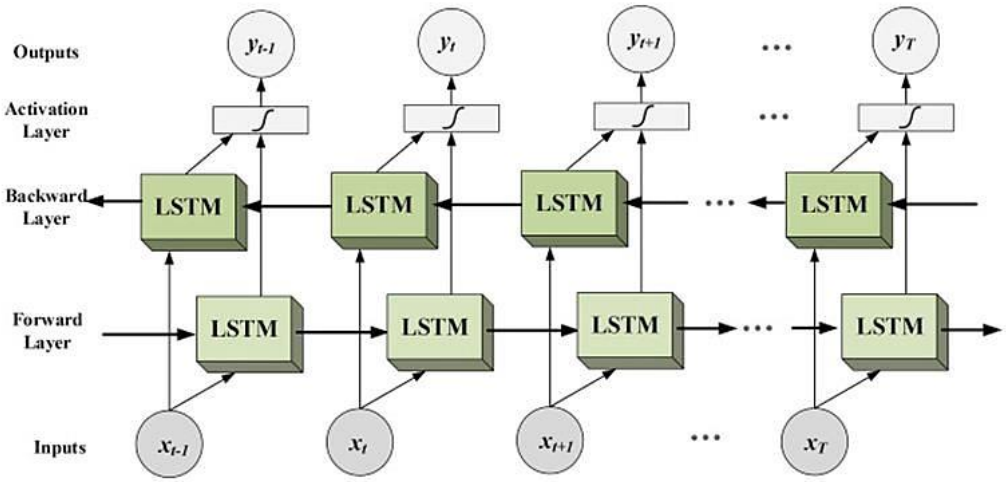


Figure 10. BiLSTM Layer [25]

3.6.3 Prediction layers

According to the number of states to be predicted, a fully connected layer with class, c units and a softmax activation function was designed and constructed as the prediction layer. The features from BiLSTM were converted into c predictions per image. The c outputs of an image in this layer represented a probability distribution in each state category. Therefore, the values of the c outputs added up to 1.

3.7 Dashboard

The dashboard was used to visually present the results of the trained model to a user. The dashboard presented the duration of each detected state in a cycle of the machine running. Each state's operating duration was obtained as a difference in time from the beginning of the state to its end. The duration of one cycle of operation by the machine is from the time a loading operation starts to the end of a sorting operation.

3.8 Multi-label training loss and model performance evaluation

After the prediction layers give a predicted value, the model was evaluated based on a score. This was used to guide the update of the network parameters. The cross-entropy loss [49] [50] is normally used for output evaluation when there are two or more label classes. The cross-entropy loss for each image

is given by Equation (3), where y_c is the target and p_c is the network output for class c. The training loss is the mean of losses overall samples, as shown in Equation (4).

$$l(y_c, p_c) = -\sum_{c=1}^{c} y_c \log(p_c) \quad (3)$$

$$L(\chi, y) = \frac{1}{n} \sum_{n=1}^{n} l(y_c, p_c) \quad (4)$$

Since loss cannot intuitively reflect the performance of classifiers, accuracy, recall and precision are often used to intuitively reflect the quality and robustness of the classifier [51] [52]. Considering that the model is solving a multi-class case, all model evaluation was based on the average score of each class.

The calculation method of accuracy is shown in Equation 5, where N_c is the number of all correctly classified samples, and N_a is the number of samples. Table 1 shows how the confusion matrix for each class state was calculated. The confusion matrix is then used to calculate the precision rate, recall rate, and F1-Score of each class separately, as shown in Equations 6, 7 and 8. Then the average value is taken as the score of the entire model.

Table 1: Confusion matrix

Confusion matrix	Predicted: Positive	Predicted: Negative
Actual: Positive	True Positive: t_p	False Positive: f_p
Actual: Negative	False Negative: f_n	True Negative: t_n

$$\%ACC = 100 * \frac{N_c}{N_a} \quad (5)$$

$$Precision = \frac{t_p}{(t_p + f_p)} \quad (6)$$

$$Recall = \frac{t_p}{(t_p + f_n)} \quad (7)$$

$$F1_{score} = 2 * \frac{precision * recall}{(precision + recall)} \quad (8)$$

4.0 Experiments and Results

An Android APP that uses the Bluetooth module, the Recording module, and a GPS module to record and send acoustic data to a remote server was developed and deployed on a mobile phone. The APP was developed based on Android Studio [43] and ran on Xiaomi Note 2, Android 8.0 system. The mobile phone was placed beside a scaled down prototype manufacturing plant. The scaled down prototype manufacturing plant was built of Lego Mindstorm modules. The Lego Mindstorm Plant (LMP) was based on a sorting robot. Before training the network on a large-scale real factory environment dataset, the LMP was used to generate data in order to optimise the parameters and structure of the proposed model in this paper. We tested our MAABL network on a LMP dataset with no noise (Silent room with only the LMP running) and a noisy room (with the LMP running and a radio tuned to a commentary broadcasting station to simulate people talking in a factory like environment).

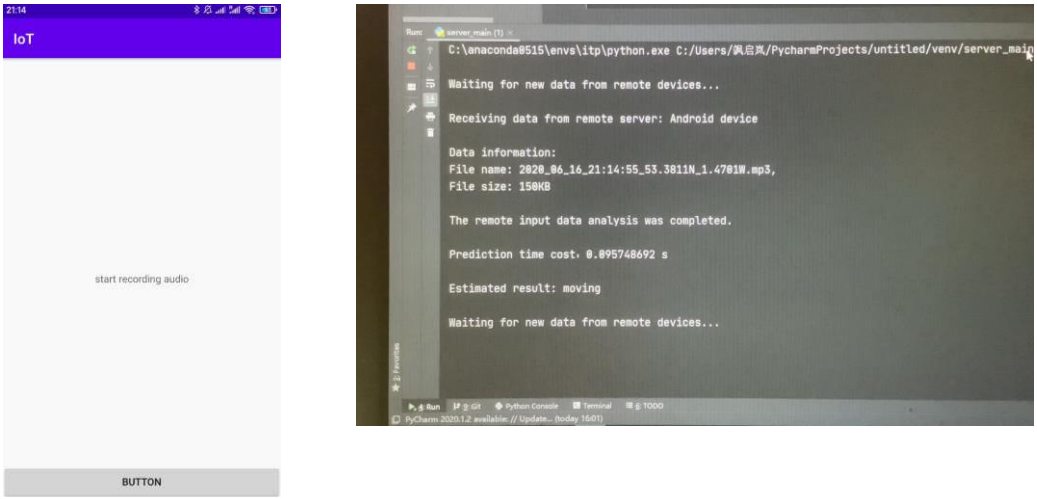
When the model achieved a certain degree of robustness, replicability and accuracy on the LMP datasets, the model was then applied to two large-scale real factory environment data sets to evaluate the performance of the model. This was achieved through transfer learning as well as to observe the adaptability of the model to new datasets. The two large-scale real factory datasets were the MIMII [45] and the ToyADMOS [46] datasets.

Furthermore, we compare the results of our MAABL network with CNN [4], CNN + LSTM [25], CNN + LSTM + attention [40] and MobileNetv2 [18]. These are currently the state of the art convolutional neural network architectures in literature. In order to compare these networks with our MAABL, we downloaded them from their respective Github websites and used the principles of transfer learning to tune the last layers of the network in order to fit the number of machine states we intended to classify. The other parts of the network were left intact. Then we ran them in the same server environment and computer as MAABL. The server was a Hasee with GPU GTX1060 (6G) and CPU i7-7700HQ. MAABL was implemented in python 3.0 in TensorFlow [38] on Window 10 version of Anaconda [44]. Finally, a dashboard was built on the server to evaluate the state of the remote machines in near real-time.

4.1 Android APP

Figure 11 shows the APP for this project. Once the connection to the server is successful, and the button is pressed to confirm the start of the program, the APP will send a file containing acoustic data

to the target server every minute. The file name of the acoustic data contains the GPS location of the production machine recorded by the mobile phone and the time when the acoustic data was sent.



(a) (b)

Figure 11. Our Andriod Application (a) and the Data received by the server (b)

As shown in Figure 11, whenever the server receives an acoustic data file sent by a mobile phone, it records the location information of the device and the time when the acoustic data recording was completed. Then the server passes the acoustic data into the deep neural network proposed in this article to get the estimated machine state.

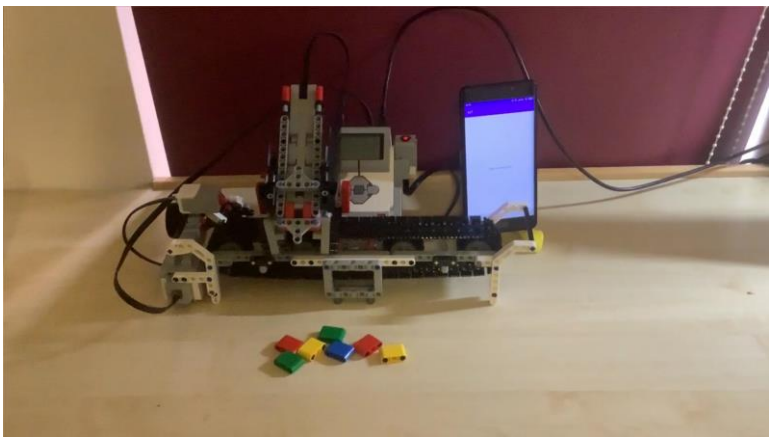


Figure 12. Sorting robot for generating the LMP data. The color blocks in the Figure are used to simulate the parts that need to be sorted. Note that the phone with the microphone is placed close to the LMP in order to reduce the influence of other background sounds during the data collection.

4.2 Lego Mindstorm Plant Datasets Results

Figure 12 shows the use of Lego Mindstorm Plant to build a sorting robot. This robot is used to make acoustic data sets. Datasets generated from Lego Mindstorm Plant are denoted as LMP. It has three states, namely loading, moving, and sorting, as shown in Figure 13. Each state emits a unique sound when it is executing. Therefore, the proposed network model needs to learn the different characteristics of the sound emitted by each state from a number of samples.

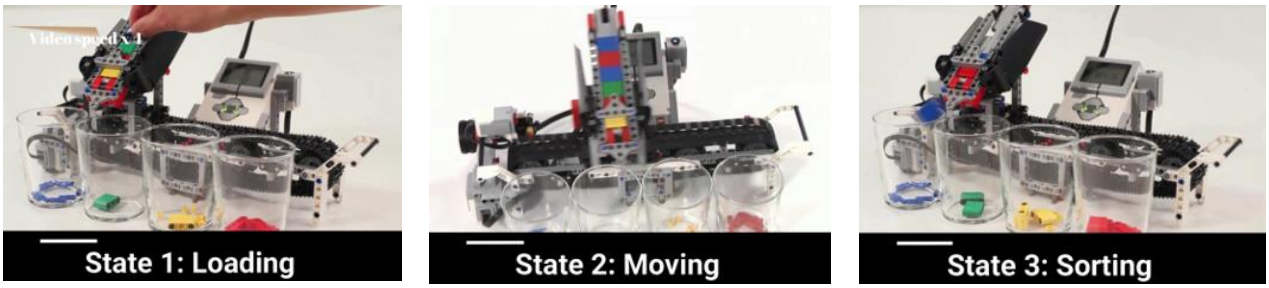


Figure 13. The three states of the LMP sorting robot

The developed Android APP continuously sends the acoustic data from the LMP to the server. When the total duration of the acoustic data set collected was 1 hour, the data set was manually labelled. In the end, 120 MP3 files with a length of 4 seconds each were obtained with 40 samples for each state. Figure 14 shows the conversion of some of the acoustic data into image data using the tool kit of [35].

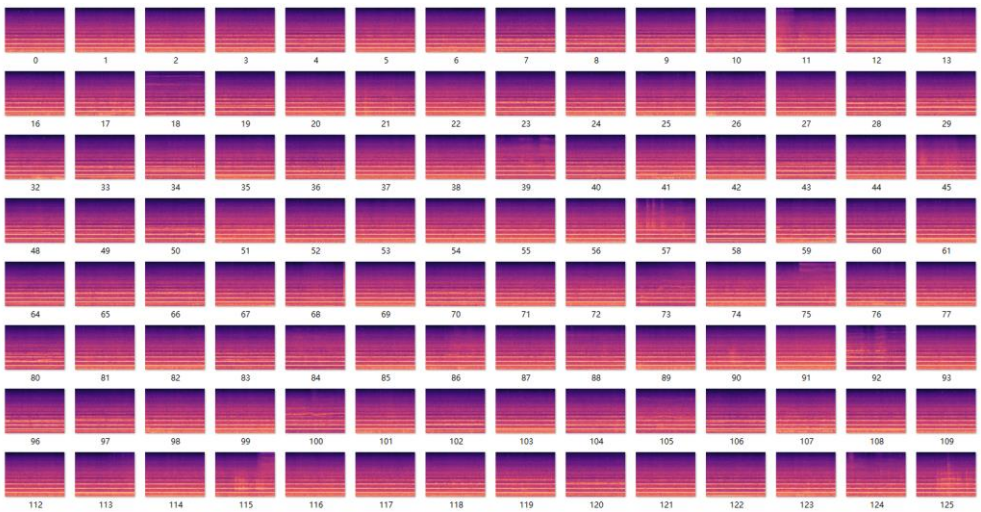


Figure 14. Parts of the Lego Mindstorm Plant data in Log-scaled Mel-spectrograms

Table 2: MAABL network's performance on no noise and noise added LMP datasets when using different training/testing ratios.

Dataset split	**LMP datasets**	**Acc. (%)**
80% for training, 20% for testing	without radio noise	94.29%
	with radio noise	88.57%
70% for training, 30% for testing	without radio noise	94.19%
	with radio noise	88.22%
60% for training, 40% for testing	without radio noise	94.13%
	with radio noise	88.17%

These acoustic data in the form of images are difficult to distinguish between different categories by naked eye observation. In other words, the data can be used to test the feature extraction ability of the network model proposed in this paper to a certain extent. Since the real industrial application scenarios may contain many noises, such as the sound of other equipment or human talking voices, a second dataset of the same size with noticeable noise was produced. The method used to add noise was to play a radio during the acoustic data recording process simultaneously.

The audio files generated for each class were put in separate directories. Once the number of the training and test set is determined, the corresponding number of files is randomly selected from each class's directory. Table 2 shows the best accuracy on the test set of the datasets when using different training and testing ratios of the dataset. Without adding radio noise, the model achieved an accuracy of 94% on the test set, indicating that it can learn the characteristics of different states and also distinguish them with high accuracy. Furthermore, the addition of radio noise did not make the model lose the ability to identify each state. Nevertheless, the accuracy rate on the test set was reduced by approximately 5% compared with no radio noise. Furthermore, different ratios of training and test set did not have a significant impact on the accuracy of the two generated datasets (i.e the dataset with on noise and the dataset with noise). However, an increase in the training set sample size can slightly improve the accuracy of the model as shown in Table 2.

Figure 15 and Figure 16 show the learning curve of the model on this data set and its performance on the test set. The training set accounts for 80%. The test set accounts for 20%, and the test set does not participate in training. Each epoch means that all samples of the training set need to be input once on the model. Each input is called a step or a batch and consists of 16 samples. The weights of the model would be updated at the end of each step. At the end of each epoch, the parameters learned by the model were evaluated by calculating the value of the loss function. The loss function uses the Sparse Categorical Cross entropy provided by TensorFlow for multi-classification problems. This loss function can better evaluate the error between the predicted value and the label value of the multi-classification problem. As shown in Figure 15, the loss function values of the model has converged on both the training set and the validation set, meaning that what the model has learned has stabilised.

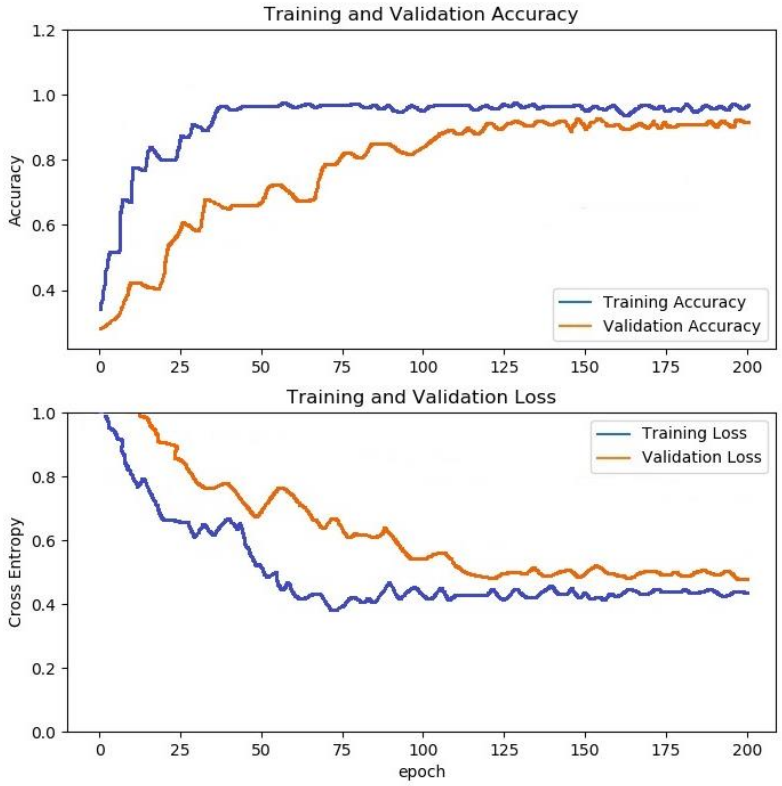


Figure 15. MAABL network's performance without radio noise

Meanwhile, the accuracy of the validation set of the model at this time is already very close to the accuracy of the training set. The accuracy of the training set is as high as 98% at 40 epochs, and the accuracy of the validation set has reached 88% at this time. As the number of iterations further increases, the weights of the model are continuously fitted to the real model, and the accuracy of the final model on the validation set is as high as 94.29%.

To further verify that the accuracy of the model is robust, Table 3 (without radio noise) and Table 4 (with radio noise) also show the performance of the model in other model evaluation metrics. These metrics are recall, precision, and F1-score. Considering that the model solves the multi-class case, all model evaluation methods are based on the average of the score of each class. The model in this paper achieved an average score higher than 0.88 across all classes and metrics in both the datasets with and without radio noise. In other words, the accuracy of the model is robust.

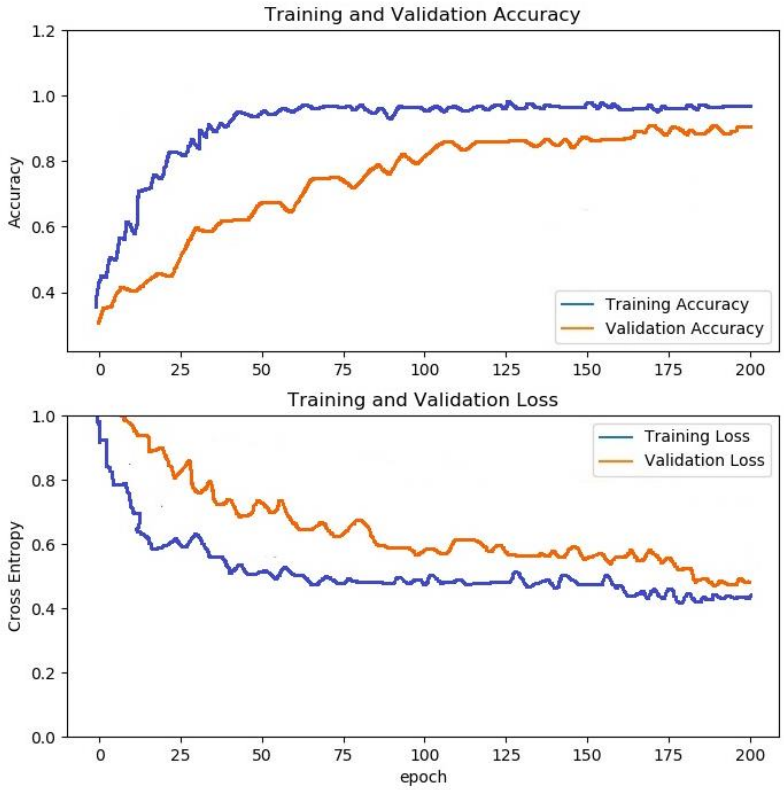


Figure 16. MAABL network's performance with radio noise

4.3 Applying the learnt model to other datasets

Purohit et al. [45] built a sound dataset for malfunctioning industrial machine investigation and inspection (MIMII). In this data set, the sounds produced by four components (valves, fans, pumps and slide rails) of a production machine during its operation were recorded. In order to resemble a real-life scenario, various anomalous and normal sounds were recorded for each component. In this article, we selected three different states from the category Fan. Each state included 2000 6-second WAV files and all training methods as well as model evaluation methods were the same as those used in the previous section.

Table 3. Showing the model evaluation metrics for our MAABL model without radio noise

Labels	**Precision**	**Recall**	**F1-Score**
0	0.9986	0.9932	1.0000
1	0.9891	0.9891	0.9891
2	0.9231	0.9331	0.9422
Average:	0.9703	0.9718	0.9771
		Accuracy:	0.9429

Table 4. Showing the model valuation metrics for our MAABL model with radio noise

Labels	**Precision**	**Recall**	**F1-Score**
0	1.0000	0.8333	0.9891
1	0.8888	0.9231	0.8571
2	0.9000	0.9126	0.9221
Average:	0.9296	0.8897	0.9228
		Accuracy:	0.8857

As discussed at the beginning of this section, we tested MAABL against the neural network architectures of CNN [4], CNN + LSTM [25], CNN + LSTM + attention [40] and MobileNetv2 [18]. Table 5 shows that though our network had a larger number of parameters than MobileNetv2, it achieved the highest accuracy when compared to the other state of the art methods. MAABL had an increase in the number of parameters over MobileNetv2 because of the additions of the attention and the Bidirectional LSTM layers. Nevertheless, such a model with reduced number of parameters is advantageous in real industrial scenarios and reduces the need for a large number of computer resources. Furthermore, the increase in accuracy can help factory managers to supervise remote production machines more efficiently and accurately. Figure 17 shows the accuracy and the loss attained from each epoch during the training when the dataset was split into 80% for training, 20% for testing. Looking at Figure 17 further, the reason for such high accuracy is that the size of the MIMII

data set is much larger than the LMP data set of the previous section. This data set has 6000 state samples, which allowed the model to learn various features thoroughly. Therefore, the model achieved a higher accuracy rate on the test dataset.

Table 5. MAABL's performance on the MIMII dataset compared with other state of the art neural networks.

Dataset split	**Network Architecture**	**Params.**	**Acc. (%)**
80% for training, 20% for testing	CNN [4]	24.2M	97.03%
	MobileNetV2[18]	**4M**	**96.03%**
	CNN+LSTM [25]	26.3M	97.22%
	CNN+LSTM+attention [40]	27.1M	97.26%
	MAABL	**16M**	**98.02%**
70% for training, 30% for testing	CNN [4]	24.2M	96.23%
	MobileNetV2[18]	**4M**	**95.33%**
	CNN+LSTM [25]	26.3M	96.28%
	CNN+LSTM+attention [40]	27.1M	97.02%
	MAABL	**16M**	**97.82%**
60% for training, 40% for testing	CNN [4]	24.2M	96.02%
	MobileNetV2[18]	**4M**	**94.23%**
	CNN+LSTM [25]	26.3M	96.12%
	CNN+LSTM+attention [40]	27.1M	97.15%
	MAABL	**16M**	**97.66%**

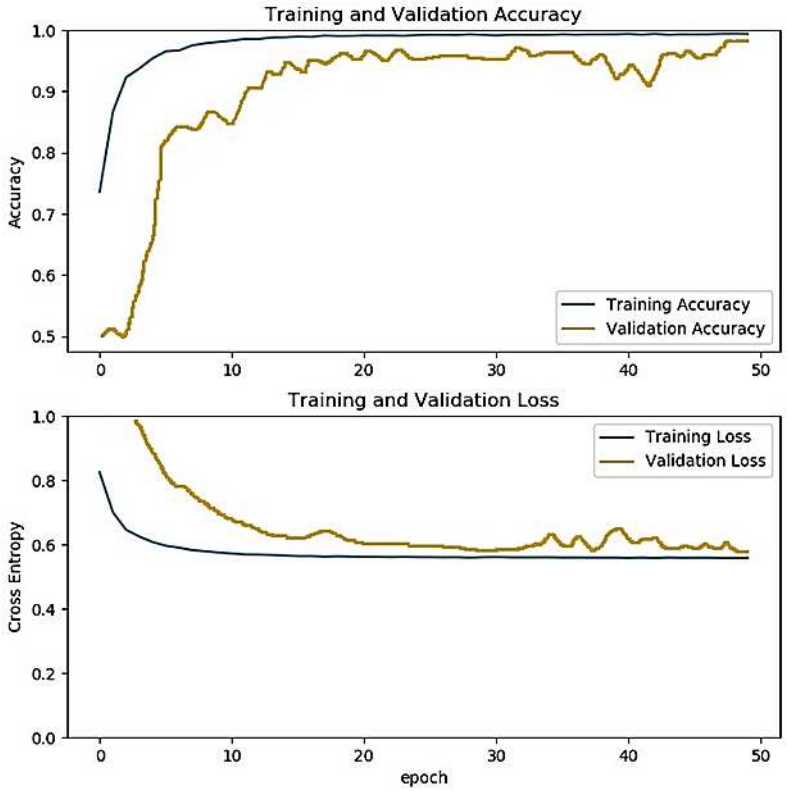


Figure 17. MAABL Network's performance when applied to the MIMII dataset

Table 6 shows the performance of the model on another data set designed for anomaly detection in machine operating sounds (ToyADMOS) [46]. The model proposed in this paper also achieved the highest accuracy on the test set of this data set. From the results, it can be seen that the model proposed in this article can be applied to other acoustic data sets.

4.4 Exploring transfer learning of the MAABL network

This section explores whether the parameters of the already trained model can be applied to a new but similar data category, to reduce the time for retraining and the requirements for the size of the data set. Our model was used in transfer learning to learn the three states in the MIMII dataset that were not previously seen. Our model had a total of 155 layers. During transfer learning, the first 100 layers were retained while the following layers were initialised and retrained. As shown in Figure 17, it took 50 epochs for the model to train a brand new data set from start to finish, while applying the transfer learning method (Figure 18) required five epochs to achieve 98% accuracy. In this experiment, only 1000 WAV samples are taken for each state, which is 50% of the data samples available. In other words, transfer learning allows the model to learn new categories with less time cost and data samples based on previous experience.

Table 6. MAABL's performance on the ToyADMOS dataset compared with other state of the art neural networks.

Network Architecture	**Params.**	**Acc. (%)**
CNN [4]	24.2M	94.13%
MobileNetV2[18]	**4M**	**92.63%**
CNN+LSTM [25]	26.3M	95.62%
CNN+LSTM+attention [40]	27.1M	96.16%
MAABL	**16M**	**97.02%**

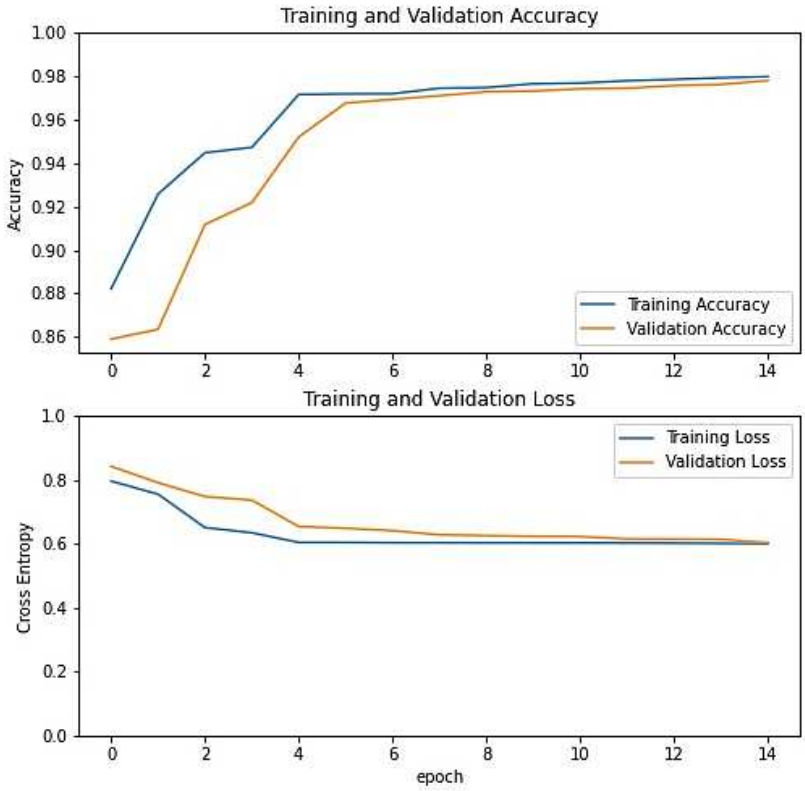


Figure 18. Transfer learning on the MIMII dataset

4.5 Dashboard display

In order to demonstrate near real-time applicability of our approach, a dashboard was designed in order to display results to an operator. The Figure 19 shows the dashboard displaying the cycle time and operation time in each cycle of operation in near real-time. In addition to the estimated cycle times,

the dashboard interface also displayed the average time of each machine state and GPS location of the production machine.

In order to facilitate the comparison of each cycle of operation and to understand if there were variations in the cycle time of each cycle, the order of the color blocks in the loading process were arranged so that the LMP will sort the blocks with the same moving path. Nevertheless, it was discovered that there were variations in the completion time for each cycle of operation. Table 7 presents the cycle time and operation time in 5 cycles of the LMP.

Table 7. Cycle time of states and time spent in each cycle. This table shows an extract of 5 cycles of operation

Cycles ID	**Cycle Time (ms)**	**Loading Time (ms)**	**Moving Time (ms)**	**Sorting Time (ms)**
1	38120	8403	5403	12103
2	38311	8301	5301	12111
3	37819	7809	6702	10809
4	38214	8014	5014	12834
5	38026	8026	5026	12126
Average	**38098.1**	**8110.6**	**5489.2**	**11996.6**
Std.	**168.72**	**214.1**	**625.1**	**656.1**

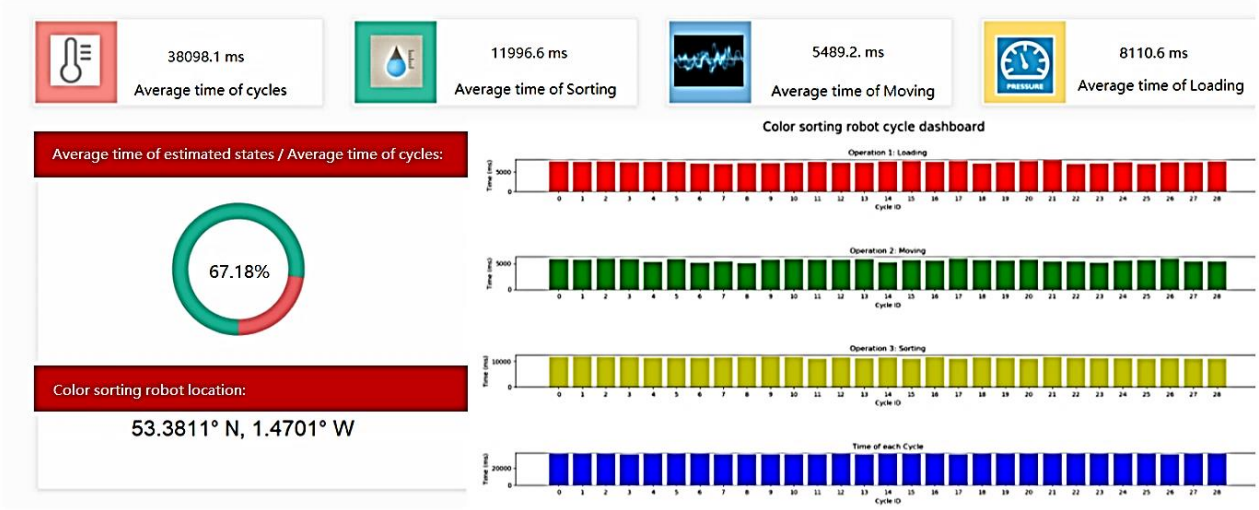


Figure 19. Dashboard for remote state estimation of the LMP sorting robot.

5.0 Conclusion

In this work, we have proposed a new neural network architecture called MAABL that uses audio data for the purposes of monitoring and estimating the states of remote production machines. The architecture was constructed by combining an inverted residual block of MobileNetv2 with an augmented attention mechanism block as well as a bi-directional LSTM layer. The MAABL network proposed in this paper obtained the state of the art results on accuracy on three different acoustic data sets. One of the data sets was generated by a prototype manufacturing plant which we constructed using Lego Mindstorm. The dataset from the Lego Mindstorm Plant (LMP) was generated both in a quiet room (to replicate ideal situations) and in a noisy room (to replicate non-ideal situations). The LMP dataset contained the audio data of the prototype plant operating and cycling through 3 different states of loading, moving and sorting.

The other two datasets were obtained from two open source projects that recorded the states of real world industrial machines and miniature machines. The names of these two projects were Industrial Machine Investigation and Inspection (MIMII) [45] and Miniature-Machine Operating Sounds for Anomalous Sound Detection (ToyADMOS) [46]. MIMII is an acoustic dataset of real industrial machines operating in normal and anomalous states while ToyADMOS is an acoustic dataset that was collected from miniature machines.

The proposed MAABL model achieved 94% accuracy on the test set of the LMP dataset recorded in a quiet room while it achieved 89% accuracy on the test set of the LMP dataset with noise added. Furthermore, the two other open source datasets on acoustic data of production machines were used to train and test the proposed network. When compared with other state of the art networks (such as CNN CNN [4], CNN + LSTM [25], CNN + LSTM + attention [40]), MAABL was able to achieve higher accuracy with less number of parameters in the network. However, since MAABL was based on MobileNetv2, it had a larger number of parameters (**16M**) than MobileNetv2 (**4M**). This was because of the additional attention mechanism block and bi-directional LSTM layer that were added to MobileNetv2. Furthermore, it should be noted that MobileNetv2 was designed for image datasets and not acoustic datasets.

Nevertheless, the neural network proposed in this paper is a lighter-weight network that can be applied in real-time scenarios or various near real-time applications in real industrial environments. This is because of its high accuracy and the small number of parameters it contains when compared to CNN [4], CNN + LSTM [25] and CNN + LSTM + attention [40]. Towards application in real industrial environments, we applied transfer learning to investigate the feasibility of transferring our model between two scenarios. Transfer learning is important because it enables the possibility of deploying models to new scenarios with minimal data required from the destination domain (The domain to which we are transferring the model). In our case, only the last layers of the MAABL network were retrained in order to achieve transfer learning. Our transferred model was tested on the MIMII dataset from production machines and results show that using MAABL in transfer learning can significantly reduce the training time while achieving high accuracy.

However, certain considerations need to be taken into account while applying transfer learning to MAABL. For example, when the destination domain dataset is not acoustic, the parameters learnt by the network do not guarantee that the classification task on the new dataset will achieve a similar accuracy. In this case, the entire network would need to be retrained to learn new features. Furthermore, transfer learning also has certain limitations. Transfer learning requires a high degree of similarity between the target dataset and the pre-trained dataset. Moreover, we do not claim that the current MAABL network structure is optimal. We believe that there is still room for improvement in the design of the depth of the network and optimising its hyperparameters.

In future research, the existing model will be fused with information from other sensors, such as temperature, vibration intensity, humidity and other sensors, to provide more accurate predictions. On the improvement of the network structure, reinforcement learning methods or nature inspired methods

[53][54] could be applied to optimise some of the hyperparameters. Hyperparameters that could be optimised include the number of layers and the depth of the network. In practical applications, the telemetry communication with remote machines needs to be worked upon further and technical issues that come with transferring data across remote geographical sites solved. Issues include interference, bandwidth and so on. Furthermore, technical engineers from factories will need to be trained to use our system and gained knowledge transferred. Also, even though the work conducted in this paper made use of an acoustic dataset from real industrial machinery, future work will look into deploying our system onto an operational factory floor.

REFERENCES

[1] J. Bynum, G. Earle, and D. Lattanzi, “A Convolutional Neural Network Approach to the Semi-Supervised Acoustic Monitoring of Industrial Facilities,” International Conference on Smart Infrastructure and Construction (ICSIC), 2019.

[2]W. Ren, G. Wen, B. Xu, and Z. Zhang, “A novel Convolutional Neural Network base on Time-frequency Spectrogram of Arc Sound and its Application on GTAW Penetration Classification,” IEEE Transactions on Industrial Informatics, pp. 1–1, 2020.

[3] J. Wu, Y. Chua, M. Zhang, H. Li, and K. C. Tan, “A spiking neural network framework for robust sound classification,” Frontiers in Neuroscience, vol. 12, pp. 1–17, 2018.

[4] R.-Y. Yang and R. Rai, “Machine auscultation: enabling machine diagnostics using convolutional neural networks and large-scale machine audio data,” Advances in Manufacturing, vol. 7, no. 2, pp. 174–187, 2019.

[5] A. Vafeiadis, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, and R. Hamzaoui, “Audio content analysis for unobtrusive event detection in smart homes,” Engineering Applications of Artificial Intelligence, vol. 89, p. 103226, 2020.

[6] M. Z. Anwar, Z. Kaleem, and A. Jamalipour, “Machine Learning Inspired Sound-Based Amateur Drone Detection for Public Safety Applications,” IEEE Transactions on Vehicular Technology, vol. 68, no. 3, pp. 2526–2534, 2019.

[7] M.Z. Uddin, and E.G. Nilsson, “Emotion recognition using speech and neural structured learning to facilitate edge intelligence”. Engineering Applications of Artificial Intelligence, 94, p.103775, 2019.

[8] S. Chu, S. Narayanan, and C.-C. J. Kuo, “Environmental Sound Recognition With Time–Frequency Audio Features,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 17, no. 6, pp. 1142–1158, 2009.

[9] P.-J. Chen, T.-L. Wu, T.-J. Lin, Y.-H. Lai, T.-L. Wu, and C. Yeh, “Life Period Estimation of Stamping Process Using Punch Sounds and Deep Neural Network,” 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2019.

[10] X. Li, W. Zhang, and Q. Ding, “Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism,” Signal Processing, vol. 161, pp. 136–154, 2019.

[11] Li, X., Zhang, W., Ding, Q. et al. Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation. Journal of Intelligent Manufacturing, 31, 433–452, 2020.

[12] H. B. Sailor, D. M. Agrawal, and H. A. Patil, “Unsupervised Filterbank Learning Using Convolutional Restricted Boltzmann Machine for Environmental Sound Classification,” Interspeech, 2017.

[13] R. Keshari, M. Vatsa, R. Singh and A. Noore, “Learning structure and strength of CNN filters for small sample size training”. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 9349-9358), 2018.

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[15] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. “Densely connected convolutional networks”. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. “Going deeper with convolutions”. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2015.

[17] Z. Chen, Z. Xie, W. Zhang, and X. Xu, “ResNet and Model Fusion for Automatic Spoofing Detection,” Interspeech, 2017.

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[19] S. Peng, H. Huang, W. Chen, L. Zhang, and W. Fang. More trainable inception-ResNet for face recognition. Neurocomputing, No. 411, pp.9-19, 2020.

[20] S. Karkra, P. Singh, K. Kaur, and R. Sharma, Deep Learning Architectures: A Hierarchy in Convolution Neural Network Technologies. In Advances in Electromechanical Technologies (pp. 439-457). Springer, Singapore, 2020.

[21] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, “MnasNet: Platform-Aware Neural Architecture Search for Mobile,” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[22] Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” In Proceedings of the 36th International Conference on Machine Learning (ICML), pp. 10691–10700, 2019.

[23] A. Howard et al., “Searching for MobileNetV3,” Proceedings of the IEEE International Conference of Computer Vision, pp. 1314–1324, May 2019.

[24] A. Wan, X. Dai, P. Zhang, Z. He, Y. Tian, S. Xie, B. Wu, M. Yu, T. Xu, K. Chen, and P. Vajda. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12965-12974, 2020.

[25] M. A. D. Mauer, T. Behrens, M. Derakhshanmanesh, C. Hansen, and S. Muderack, “Applying Sound-Based Analysis at Porsche Production: Towards Predictive Maintenance of Production Machines Using Deep Learning and Internet-of-Things Technology,” Management for Professionals Digitalization Cases, pp. 79–97, 2018.

[26] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM networks,” In Proceedings of the IEEE International Joint Conference on Neural Networks, 2005.

[28] K. Cho et al., “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,” In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734, 2014.

[29] N. Gruber and A. Jockisch, “Are GRU Cells More Specific and LSTM Cells More Sensitive in Motive Classification of Text?,” Frontiers in Artificial Intelligence, vol. 3, pp. 1–6, 2020.

[30] A. Sherstinsky, “Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network”. Physica D: Nonlinear Phenomena, 404, p.132306, 2020.

[31] S. Zhang, L. Yao, A. Sun and Y. Tay, “Deep learning based recommender system: A survey and new perspectives”. ACM Computing Surveys (CSUR), 52(1), pp.1-38, 2019.

[32] S Woo, J. Park, J.Y. Lee, and I. So Kweon, Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV) (pp. 3-19) 2018.

[33] I. Bello, B. Zoph, Q. Le, A. Vaswani, and J. Shlens, “Attention Augmented Convolutional Networks,” 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[34] “chexpert,” github. [Online]. Available: https://github.com/kamenbliznashki/chexpert. [Accessed: 21-Jul-2020].

[35] A. Kumar, M. Khadkevich, and C. Fugen, “Knowledge Transfer from Weakly Labeled Audio Using Convolutional Neural Network for Sound Events and Scenes,” In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018.

[36] X. Li, W. Zhang, Q. Ding, and X. Li, “Diagnosing Rotating Machines With Weakly Supervised Data Using Deep Transfer Learning,” IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp. 1688–1697, 2020.

[37] T. Koike, K. Qian, Q. Kong, M.D. Plumbley, B.W. Schuller, and Y. Yamamoto, “Audio for Audio is Better? An Investigation on Transfer Learning Models for Heart Sound Classification”. In Proceedings of the 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 74-77, 2020.

[38] TensorFlow. [Online]. Available: https://www.tensorflow.org/. [Accessed: 22-Jul-2020].

[39] “Laser cutting Machine,” Temei Machinery. [Online]. Available: http://en.cntemei.com/. [Accessed: 21-Jul-2020].

[40] Z. Zhang, S. Xu, T. Qiao, S. Zhang, and S. Cao, “Attention Based Convolutional Recurrent Neural Network for Environmental Sound Classification,” Pattern Recognition and Computer Vision Lecture Notes in Computer Science, pp. 261–271, 2019.

[41] X. Li, V. Chebiyyam, and K. Kirchhoff, “Multi-Stream Network with Temporal Attention for Environmental Sound Classification,” Interspeech, 2019.

[42] Librosa. [Online]. Available: https://librosa.org/. [Accessed: 22-Jul-2020].

[43] Android Developers. [Online]. Available: https://developer.android.com/. [Accessed: 22-Jul-2020].

[44] “The World’s Most Popular Data Science Platform,” Anaconda. [Online]. Available: https://www.anaconda.com/. [Accessed: 22-Jul-2020].

[45] H. Purohit, R. Tanabe, T. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, “MIMII Dataset: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection,” Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE2019), 2019.

[46] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, “ToyADMOS: A Dataset of Miniature-Machine Operating Sounds for Anomalous Sound Detection,” 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2019.

[47] J. Sigut, M. Castro, R. Arnay, and M. Sigut, “OpenCV Basics: A Mobile Application to Support the Teaching of Computer Vision Concepts”. IEEE Transactions on Education, pp. 1-8, 2020.

[48] P. Voštinár, “Programming LEGO EV3 in Microsoft MakeCode”. In Proceedings of IEEE Global Engineering Education Conference (EDUCON), pp. 1868-1872, 2020.

[49] I. Goodfellow, Y. Bengio and A. Courville, “Deep Learning”. MIT Press, 2016.

[50] K. P. Murphy, “Machine learning: a probabilistic perspective”. New York: The MIT Press, 2012.

[51] A. Géron, “Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems”. O'Reilly Media, 2019.

[52] D.M.W. Powers, “Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation”. Journal of Machine Learning Technology, 2 (1), pp 37–63, 2011.

[53] J. Oyekan, D. Gu, and H. Hu, “Visual imaging of invisible hazardous substances using bacterial inspiration”. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43(5), pp.1105-1115, 2013.

[54] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, Efficient neural architecture search via parameters sharing. In International Conference on Machine Learning (pp. 4095-4104), 2018.