Accurate Estimation of Leaf Wetness Duration using Deep Learning Based Time-Resolution Technique

by

MAYANK I PATEL
202011045

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

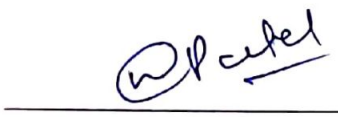
July, 2022

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

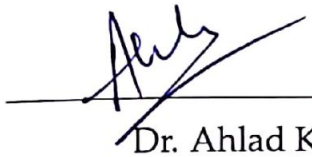
ii) due acknowledgment has been made in the text to all the reference material used.



Mayank I. Patel

Certificate

This is to certify that the thesis work entitled "Accurate Estimation of Leaf Wetness Duration using Deep Learning Based Time-Resolution Technique" has been carried out by **Mayank I Patel (202011045)** for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my supervision.



Dr. Ahlad Kumar
Thesis Supervisor

Acknowledgments

First and foremost, I would like to express my gratitude to Dr. Ahlad Kumar, my research supervisor, for consistently guiding my research work, providing me with useful ideas, and correcting my errors, all of which helped me much during my research. I'd also want to express my gratitude to Prof. Vinay S Palaparthy for his support in developing the research's basic idea and providing a deeper understanding of various IOT-based domain expertise. I'd want to convey my gratitude to Riya Saini, a DAIICT JRF member, for providing continuous guidance for professional writing and novel ideas for manuscript writing.

Furthermore, I would like to thank the Dhirubhai Ambani Institute of Information and Communication Technology for giving me this opportunity in the competitive research environment. Finally, I want to thank my family for their constant support during this challenging time.

Contents

Abstract

Plant diseases are one of the leading causes of loss of crops. Monitoring the plant growth conditions could help prevent the spread of diseases and ensure healthy agricultural productivity. A leaf wetness sensor (LWS) is used to detect wetness on the surface of the leaves. Leaf wetness duration (LWD) is an essential parameter that governs the possibility of the occurrence of diseases in plants. The sensor system used in this work for monitoring the environmental conditions transmits data every 30 minutes due to power consumption considerations. To improve the time-resolution of the signal and accurate estimation of LWD, a deep learning-based architecture is proposed. The data from LWS has been used to train the architecture and the results obtained have been compared with the existing methods to analyze its performance. Using the proposed architecture, the signal from LWS was bench-marked with the commercially available Phythos-31. The time-resolution of data has been improved from ± 30 minutes to ± 4 minutes. The average SNR values for three test signals with improved time-resolution have increased from 19.36dB to 22.16dB. Similarly, the average RMSE of the estimated LWD values after time-resolution improved from 0.55 to 0.20. During the experimental analysis, it has been observed that the proposed architecture estimates the LWD values better compared to the other time-resolution techniques.

List of Tables

List of Figures

CHAPTER 1
Introduction

Many crops are damaged due to diseases, weeds, and pests [1]. This leads to a loss of livelihood for the population that solely relies on farming for a living. The subpar management and supervision of crops during their growth is the primary cause of undetected diseases and subsequent loss of crops. To abate the potential wastage of crops due to diseases, an early prediction mechanism should be adopted as a common practice that entails regular supervision of plant growth conditions. The advancements in sensor technology have enabled the development of various electronic tools that could be employed for this purpose. These sensors could monitor plant growth parameters like leaf wetness duration (LWD), ambient temperature, and ambient humidity, which are the most significant parameters contributing to the spread of diseases in plants [2, 3, 4]. Fig. 1.1 is the

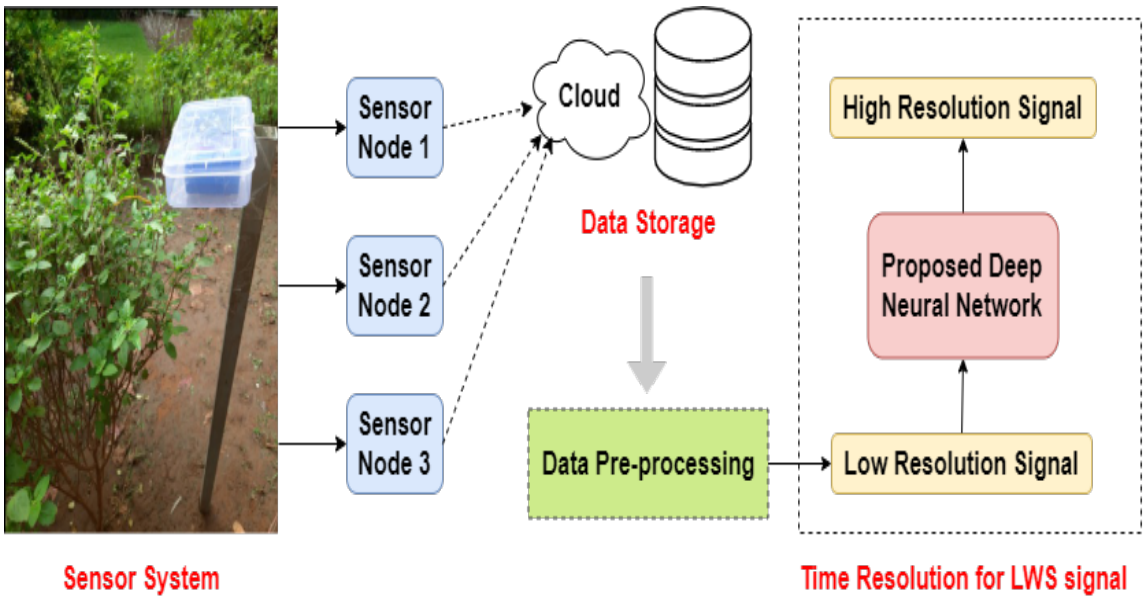


Figure 1.1: Schematic representation of time-resolution of LWS signals that uses signals collected from sensor's nodes with data pre-processing for training proposed deep neural network

representation of the whole process of time-resolution for the LWS signal. The signal from the sensor node is stored in the database. Data pre-processing is used to create a pair of low and high-resolution signals for the training dataset.

The quality of LWD estimation using the original signal is constrained by the time-resolution of the sensor signal. The absolute minimum error in the estimation of LWD is thirty minutes which limits the accuracy of the LWD measurements [5]. To overcome this limitation, the sampling frequency of the sensor could be increased. However, doing so will demand higher power leading to a higher cost for the sensor system. Thus, a more feasible approach is to increase the time-resolution of the signal using computational methods. An IoT-based Leaf wetness sensor gives us a signal in which each sequential value is in 30min duration. So, in order to detect the value of a signal within 4 minutes, we must predict a high-resolution signal using a scale of 8 time-resolution methods. So, for LWD estimation, we're employing the time-resolution technique to improve signal quality from low to high-resolution. Time-resolution is a similar process to the super-resolution technique of audio signal and image.

1.1 Super resolution technique

Super resolution is the method of upsampling or improving details with fill up missing data points in image and audio signals. In this process, fill up the required data point using observing the existing patterns of image and audio signals. Image super-resolution is a technique for creating high-quality images from low-resolution images. Let's take one example if we have a 32×32 size image we need to find 64×64 which means we need to add one more pixel value between two-pixel values in both dimensions. So image super resolution process fills up the missing place with image pixel value by observing the pattern of the existing image. Here we can call a scale of 2 image super resolution. Similarly, audio super-resolution is used to improve the quality of an audio signal by converting it from a low-resolution to a high-resolution audio signal. In order to improve signal quality, we must predict signal values between the values of each sample in time resolution. To do so, we'll need to employ the deep learning principle to observe the existing patterns and predict the missing place of image and signal.

1.2 Deep Learning strategy

Deep learning is used to observe patterns in a dataset and attempt to learn the current pattern for future estimation. We used a supervised learning strategy in our thesis. Supervised learning is a machine learning technique that makes use of a labeled dataset. For the training model, the labeled dataset contains the right ground truth answer. So the learning model in this machine learning technique is based on prior experience. Supervised learning improves the efficiency of deep learning models by using labeled input and output. The main benefit of supervised learning is that it aids in model optimization by utilizing prior experience, which will be fully utilized for accurate output prediction. The deep neural network is used to train the model with the training dataset and to interpret the signal patterns of the training dataset. We employ a proposed deep neural network to understand the pattern of the LWS signal and provide a high-resolution signal as an output, which is then used for LWD estimation. The key aspect of the thesis is mentioned below:

- The deep neural network with a progressive approach is proposed for the time-resolution of the LWS signal. By using a supervised learning approach, the proposed model is trained and estimated high-resolution LWS signal from low-resolution LWS signal.
- Residual building block is used for feature extraction and observing the pattern of LWS signals.
- The proposed architecture has two branches; the upper feature extraction branch is used for feature extraction and observing patterns with each scale of 2,4 and 8. Similarly for each scale of 2,4 and 8, the lower branch is used for the reconstruction of the LWS signal by adding the feature extraction output of the signal.
- In the proposed progressive architecture bicubic interpolation method is used for the upsampling signal.

CHAPTER 2
Related Work

2.1 Previous work of leaf wetness sensors

Wetness on the surface of leaves under specific circumstances could lead to the germination of fungal diseases in the crops [6]. Therefore, leaf wetness sensors (LWS) detect the LWD values. A wide array of LWS is commercially available for use. A lot of research on LWS for in-situ measurement of LWD values has been reported in the literature [7, 8, 9]. From the operational point of view, the studies using commercial LWS are not reliable because of the lack of standard procedure. The discrepancy in sensor weight and size, orientation and placement, and the effect of diurnal temperature variation contribute to ambiguity in sensor measurements [10, 11, 12]. Earlier reported work has introduced the novel flexible LWS used to monitor the wetness on the surface of leaves [5]. To conserve power, our sensors transmit a signal every thirty minutes through a gateway and then enter sleep mode. The signal from LWS is analyzed to extract LWD information [5].

2.2 Previous work on time resolution

Time-resolution is a widely used technique for improving signal quality, especially to create high-resolution signals from low-resolution signals [13]. A common way to improve time-resolution is by using the interpolation technique. It is a numerical method that estimates the value of unknown data points in accordance with the already known data points. The interpolated points are estimated using a function of any chosen degree. An appropriate function is used to estimate these values. Several approaches for enhancing signal time-resolution have been described in the past, including nearest neighbour [14], bi-linear interpolation [15], and bi-cubic interpolation [15]. The nearest value is picked for interpolating between existing data points in the nearest neighbour approach. For categorical

data, the nearest neighbour is more useful. Nearest neighbours produce a blocky solution for continuous data. Bi-linear interpolation utilizes a weighted average of four closest values for interpolated data points. The new interpolated value is always within the input range. For continuous data, bi-linear is beneficial, but not so much for categorical data. For the interpolation process, bi-cubic interpolation uses a weighted average of 16 closest values, which results in higher smoothness. In our work, we have used bi-cubic interpolation as the baseline method for comparison because of its efficient interpolation ability. It interpolates the signal values using the *interpolate* sub-library available in the *scipy* library [16].

The drawback of using conventional interpolation techniques for time-resolution tasks is that they do not interpolate between the samples efficiently. One approach to address this shortcoming is by using deep neural networks. The popularity of deep learning architectures has sparked interest in using deep learning models to improve time-resolution. The convolutional neural network (CNN) architecture is utilized in deep learning algorithms for feature extraction [17]. Advanced networks like ResNet [18] and Dense Net [19] uses residual block which contains convolution layers as a sub-module. Because of its feature extraction capabilities, the residual network has been employed in image processing-based deep learning architectures [20, 21, 22].

2.2.1 Previous work on audio time resolution

Kuleshov *et al*. proposed the AudioUNet[13] network, which is utilized to improve the quality of the signal. In this paper, the author introduced a new approach for increasing the sampling rate of the audio signal by using a deep neural network. Furthermore, generative adversarial networks (GANs) are a prominent artificial data generation approach for producing high-resolution signal data. For audio super-resolution problems, Kim *et al*. introduced MU-GAN[23], a GAN-based architecture incorporating CNN. A conditional GAN-based model [24] for voice enhancement of speech audio bandwidth expansion was presented. SRCNN[25] and SRGAN[25] were proposed for improving the quality of lossy compressed music. AudioUNetGAN[26], a network is also used to improve audio signal quality. In this paper, AudioUnet[13] is used as a generator in GAN-based architecture.

2.2.2 Previous work on sensor signal time resolution

SRPNet[27] is used for time-resolution tasks of sensor data for reconstructing high-quality sensor data from low-quality sensor data in industrial systems. For the smart grid state estimation system, Liang *et al*. propose SRPNSE[28] to overcome the problem of data completeness for high-frequency data from low-frequency data. Other time-resolution industrial system models, such as SRPCNN[29] and M-SRPCNN[30] are used to reconstruct high-quality smart meter data and to recover missing historical hourly data from monthly energy consumption data.

CHAPTER 3
Motivation

In the previous section 2 we discussed the time resolution related work based on some similar techniques like image super resolution and signal super resolution. In this thesis, we are using IoT-based sensor signals for improved time-resolution tasks. IoT-based Leaf wetness sensor gives us a signal which contains a frequency value that will use for estimation of the LWD. In our work, we have proposed a deep learning architecture that can improve the time-resolution of the LWS signal, which is motivated by the concept of time-resolution used in signal processing which is discussed in the previous section 2 .

3.1 Progressive approach

In deep learning architecture, for efficient feature extraction and pattern observation we can use a different type of upsampling model framework strategy as follow:

1. Pre-upsampling framework.
2. Post-upsampling framework
3. Progressive upsampling framework

The input signal is first upsampled in a pre-upsampling framework, and then feature extraction and pattern recognition are used. In post-upsampling, feature extraction is applied to the input signal first, followed by upsampling. We used a progressive strategy for upsampling in our thesis at all scales, including 2,4, and 8. The main benefit of this strategy is that we can observe the pattern on an individual scale and achieve efficient learning.

In the proposed network, we are using residual building blocks as sub-modules and use a feature extraction process with every level in a progressive manner. We

have used the IoT-Enabled leaf wetness sensor (LWS), which provides leaf wetness duration (LWD) information. Using the proposed architecture the signal's time resolution is improved by reducing the time delay between consecutive measurements. The number of data points inserted between the original sensor data depends on the scale chosen for time resolution. In our sensor system, the sensor transmits the signal to the server every 30 minutes. By choosing a scale of 8, the time-resolution of the signal is improved to approximately 4 minutes.

CHAPTER 4
Materials and Methods

In this chapter, we first introduce the Leaf wetness sensor which the Leaf wetness sensor signal using for estimating Leaf wetness duration. Then, in section 4.2, we present a residual building block that will be used as a sub-module in our proposed network. We explain the proposed deep neural network in-depth with mathematical explanations in the last section 4.3.

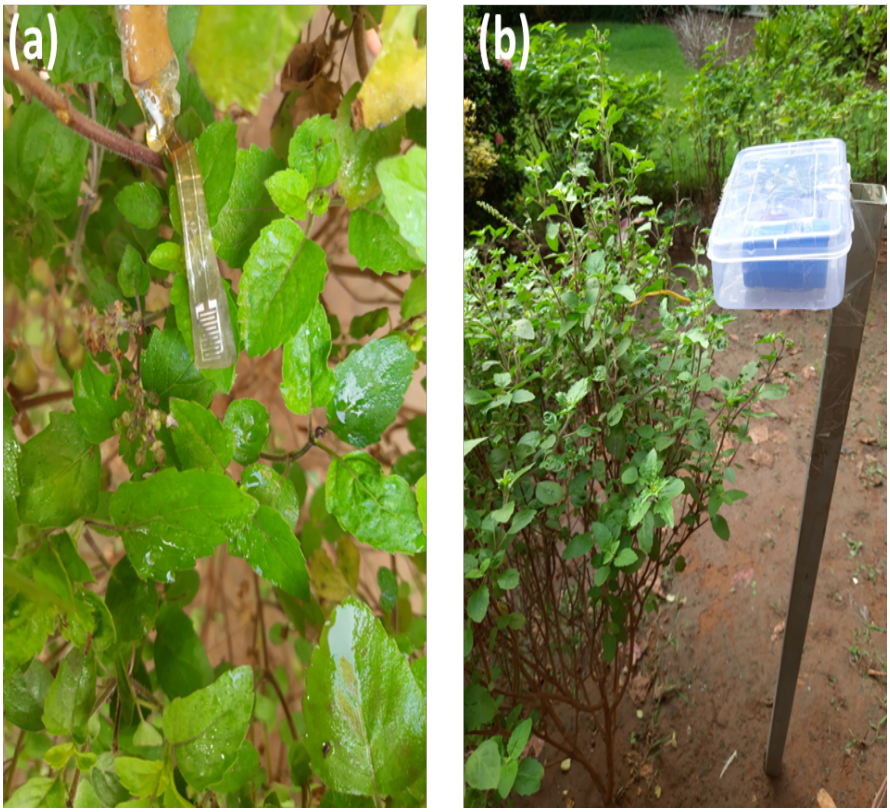


Figure 4.1: (a) Deployment of the LWS sensor on the leaves, (b) in-house developed sensor system to record the sensor data

4.1 Leaf wetness sensor

In our work, we have used the leaf wetness sensor, which is fabricated on the flexible substrate and reported in our earlier work [5]. The fabricated LWS produces a change in the sensor frequency when exposed to the wetness of the plant leaves. This frequency information is converted to the LWD information using the total variation denoising (TVD) and baseline correction methods [5]. The data displayed in the following sections were self-collected from sensors put in different fields. For the field measurements, the sensors have been deployed at a 45-degree angle as suggested in [5]. Further, we developed the in-house IoT-enabled systems, which reads the sensor data periodically every 30 minutes and upload the data to the Thing Speak TM server. Fig. 4.2 shows the LWS and the sensor system that has been used to gather LWS data from the field.

4.2 Residual Network

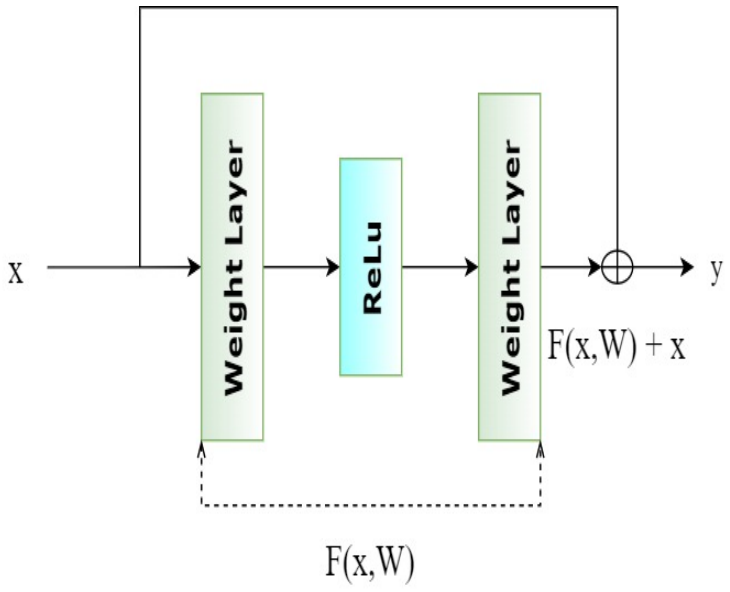


Figure 4.2: Residual building block architecture

The residual network shown in Fig. 4.2 is one of the blocks used extensively in deep learning models for obtaining a high-resolution sensor signal from a low-resolution signal. Unlike interpolation, this model is popular for feature extraction and recognizing the patterns depicted by the input for accurate prediction. Several layers of these blocks are linked to form a deep learning architecture, where each of these blocks learns certain features of the input data during the

training process. Because of its effective pattern recognition capabilities, a residual network is utilized to increase the quality of the sensor signal in terms of better time-resolution. It takes advantage of a shortcut connection to speed up the training process as shown in Fig. 4.2. The internal layers of the residual building block are weight layers that are used in the learning process. The block is designed in such a way that the network can learn both complex and basic functions. The equation that governs the working of the residual block is as follows

$$y = F\left(x, W_i^{[L]}\right) + x \tag{4.1}$$

where x and y are the n dimensional input and output vector of the residual network, respectively. The function $F(.)$ represents the residual mapping function for individual residual blocks, while $W_i^{[L]}$ represents the weight parameter allocated to each layer in the residual block. Here, L and i stand for level and layer, respectively. It is required that x and $F(.)$ have the same dimensions. If $F(.)$ only has an only one level with a single layer then Eq. (4.1) is similar to the linear layer as follows:

$$y = W_1^{[1]}x + x \tag{4.2}$$

Intuitively, the additional layers should enhance the accuracy of the predictions, but this is not always the case. A deeper network may sometimes encounter performance degradation problems [18]. The proposed architecture discussed next uses this residual network as a module.

4.3 Proposed Architecture

The proposed architecture for time-resolution of sensor data is shown in Fig. 4.3. The network takes a low-resolution signal without an up-sampled version as input and progressively predicts a high-resolution signal of level L defined as $\log_2 S$ where S is the scaling factor. The upper branch of the network is used for the feature extraction process while the lower branch is used for the reconstruction of a high-resolution signal at every level.

As discussed earlier in Section 4.2, the fundamental goal of the residual block is to extract features of the input signal. It is used in the upper branch of our proposed network for the feature extraction process at multiple scales S. The residual block consists of convolution, batch normalization, [31] and activation ($relu$) layers as shown in Fig. 4.3(b). The output of module 1 present in the upper branch at level L = 1 is connected to the next module at L = 2 and so on. The

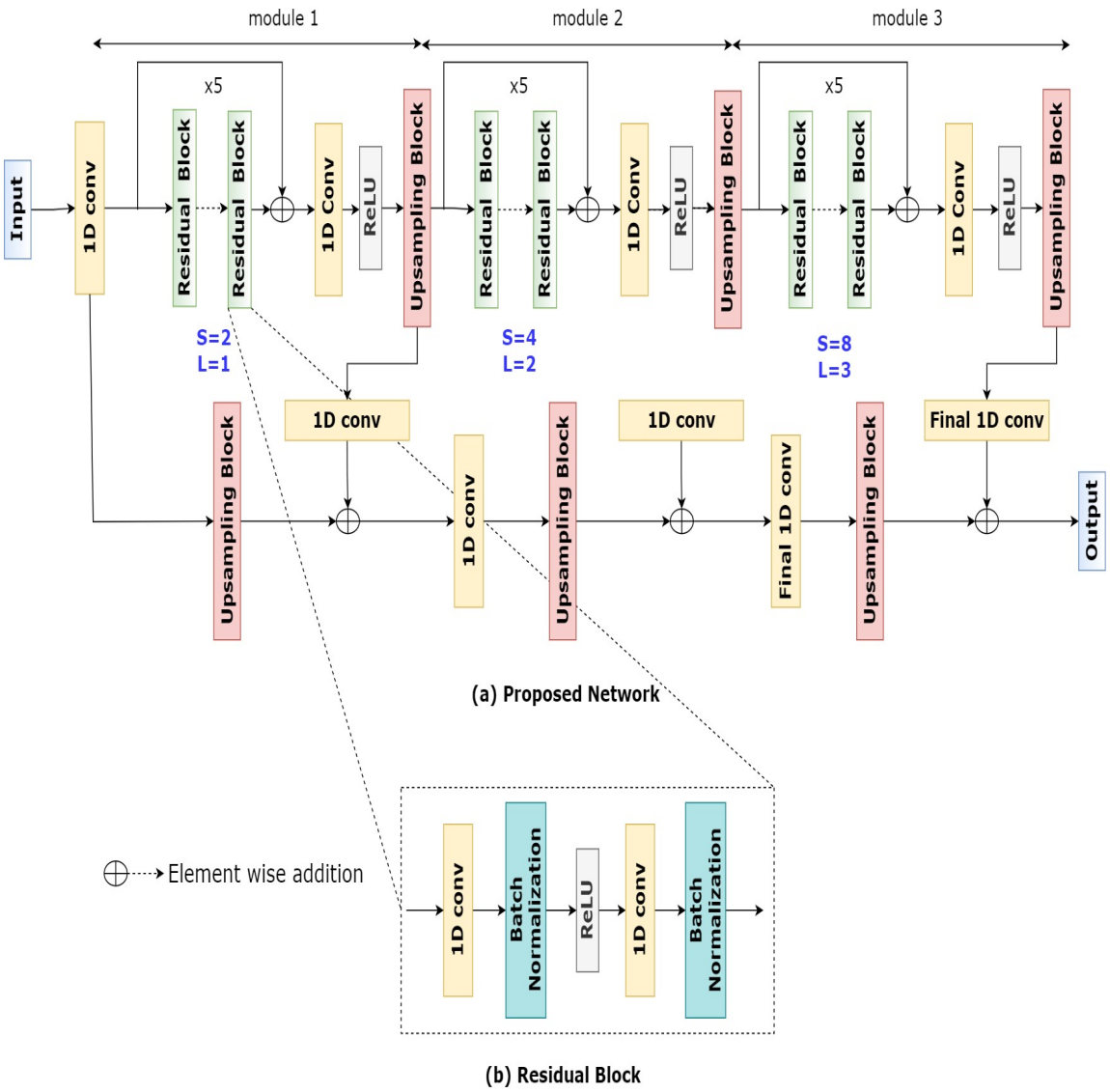


Figure 4.3: Proposed architecture which use residual block in every modules for better pattern recognition of sensor signals that help for achieving increasing time-resolution of the sensor signals for scale of 8

lower branch is responsible for the reconstruction of a high-resolution signal. In this branch, the main signal is fed as an input. Next, the output of module 1 in the upper branch is element-wise added to its corresponding *upsampling* block in the lower branch. This process repeats at all levels. For instance, the module at level 1 up-samples the 48-sampled input signal by a scale of 2, at level 2 by a scale of 4, and at level 3 by a scale of 8, as illustrated in Fig. 4.3(a). In the first module, the residual blocks in the upper branch extract feature from the 48-sampled input signal, which is then up-sampled by a scale of 2. The input signal up-sampled in the lower branch is added to the up-sampled features representations which come from an upper feature extraction branch. As a result, we obtain a scale 2 reconstructed time-resolution output for the original signal at the end of module 1. This process continues for all the other modules.

4.3.1 A mathematical explanation of proposed deep neural network

Let the input to the architecture is denoted by x. The output of the convolution layer at any level L is denoted as,

$$c(x) = \mathrm{W}_i^{[L]} \cdot x + b \tag{4.3}$$

where b is the bias vector, and $\mathrm{W}_i^{[L]}$ is the weight vector for the i^{th} layer. Consider $X_u^{[L]}$ and $X_l^{[L]}$ as the inputs to the upper and lower parallel branch, respectively, for each level L. Using Eq. (4.3), the signals for both branches are given as follows

$$X_u^{[1]} = X_l^{[1]} = \mathrm{W}_1^{[1]} \cdot x + b. \tag{4.4}$$

The residual network output is denoted by $R^{[L]}$. In the upper feature extraction branch, the residual network output using eq:1 for any level L is expressed as follows

$$R^{[L]} = F\left(X_u^{[L]}, W_i^{[L]}\right) + X_u^{[L]} \tag{4.5}$$

For any input x, $U(x)$ is defined as an up-sampling [15] function. Let $\mathcal{R}^{[L]}$ be the result of processing $R^{[L]}$ using the convolution layer and activation function [32]. Then the convolved up-sampled features represented as $c(U(\mathcal{R}^{[L]}))$ are added to the up-sampled signal from the lower parallel branch to construct the L^{th} level output given as follows

$$y^{[L]} = U(X_l^{[L]}) + c(U(\mathcal{R}^{[L]})) \tag{4.6}$$

Here, $y^{[L]}$ is the output at the L^{th} level. This process is repeated for all the other levels. At end of the level $L = 3$, we obtain a time-resolution signal $y^{[3]}$ of scale 8. Now our predicted signal $y^{[3]}$ will compare with the ground truth signal of the training dataset by calculating error using Mean square error(MSE). Then the deep neural network will use the Adam [33] optimizer to update the weight of layers of the proposed architecture.

CHAPTER 5
Results And Discussion

5.1 Data Collection and Training Details

The proposed architecture is applied to the leaf wetness sensor (LWS) signal for increasing its time-resolution. A dataset of input-target signal pairs is created in which 384 sampled signals are used as target signals from the LWS signal. Next, down-sampling these signals by a scale of 8 is used to generate 48 sampled signals as low-resolution input signals. Here, we consider the dataset of 33 days to build input-target signal pairs to train our architecture.

To evaluate the performance of our model, we have compared the results of our model with bi-cubic interpolation [15], residual network with pre-upsampling [18], and other state of art methods SRPCNN [29],AudioUNetGAN[26] and MU-GAN[23]. In the implementation of our proposed network, convolution layers consist of 16 filters having a kernel size of 3×3 except for the final convolution which consists of 1 filter of kernel size 3×3. In the proposed architecture, five blocks of the residual network are used in each module for better feature extraction. Here, the Adam optimizer [33] is used with a default learning rate of 0.001. The values of momentum and epsilon are set to 0.99 and 0.001 respectively for batch normalization [31].

5.1.1 Loss Function

The loss function used for training the model is taken as mean squared error (MSE) defined as follows

$$MSE = \frac{1}{n}\sum_{i=0}^{n}\left(y_i - \hat{y}_i^{[3]}\right)^2 \tag{5.1}$$

where y_i is the actual signal, $\hat{y}_i^{[3]}$ is the predicted signal from the architecture, and n is the total number of training signals. Fig.5.1(a) shows the comparison of MSE

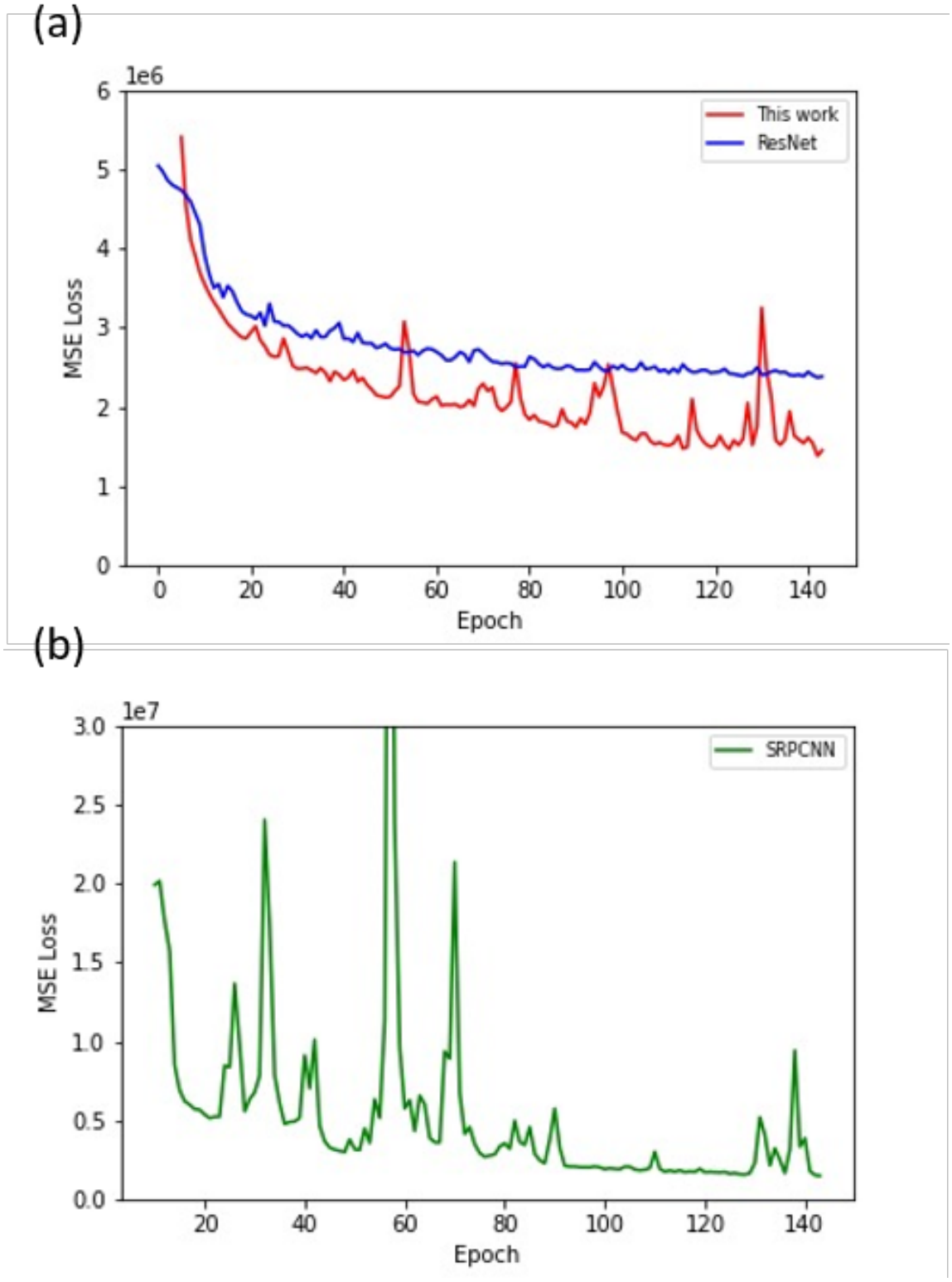


Figure 5.1: Comparison of training loss that indicate Mean square error loss for (a) ResNet and proposed work; (b) SRPCNN

loss for the proposed architecture with the ResNet, while Fig. 5.1(b) represents the loss for SRPCNN. The loss obtained while training the SRPCNN network is fluctuating heavily and is starting to increase again at the end of 140 epochs. On the other hand, the loss obtained while training ResNet, as seen in 5.1(a), is void of fluctuating values but it converges at a loss values higher than the proposed work. Out of all the three methods, the proposed network produces the best results. From Fig. 5.1(a) it can be seen clearly that in the case of our architecture the loss converges to a minimum value faster compared to ResNet.

5.2 Time-resolution analysis

5.2.1 Stem plot analysis

In this section, a comparative analysis of the quality of the time-resolution signal generated from all the methods along with the proposed architecture is discussed. For doing this analysis, three main signals acquired from the LWS are taken and are shown in Figs. 5.2(a)-5.4(a). A specific segment of the main signal which is a 384-sampled signal is taken for comparative analysis. These segments are down-sampled by a scale of 8 and is provided as inputs to the deep learning models. Figs. 5.2b(i)-5.4b(i) represent the down-sampled versions of the segments.

To improve the time-resolution of the signal, the estimated samples generated using various methods are placed in between the actual samples. The blue-colored data points are the estimated samples generated using bi-cubic interpolation, ResNet, SRPCNN, MU-GAN, AdioUNetGAN and the proposed architecture. The estimated samples obtained using various methods are shown in Figs.5.2b(ii)-(vii), 5.3b(ii)-(vii), and 5.4b(ii)-(vii) along with the original sensor data marked in red. The most apparent observation from Figs5.2b(ii)-(vii) is that there are multiple samples (shown in blue) inserted in between the consecutive input samples (shown in red) thereby resulting in an improved time-resolution of the original signal. The signal length has increased from 48 to 384 as a result of the improved time-resolution. Here the scale is chosen as 8 which signifies that between two successive samples in the original signal, seven values have been inserted. A similar interpretation is carried out using signals acquired from additional LWS at various locations and are shown in Figs. 5.3(b) and 5.4(b) respectively.

For the time-resolved signal to be accurate, the magnitude of interpolated values should vary smoothly between the samples. It can be observed in Fig. 5.2b(iv) that there is an abrupt variation in the magnitude of the interpolated values us-

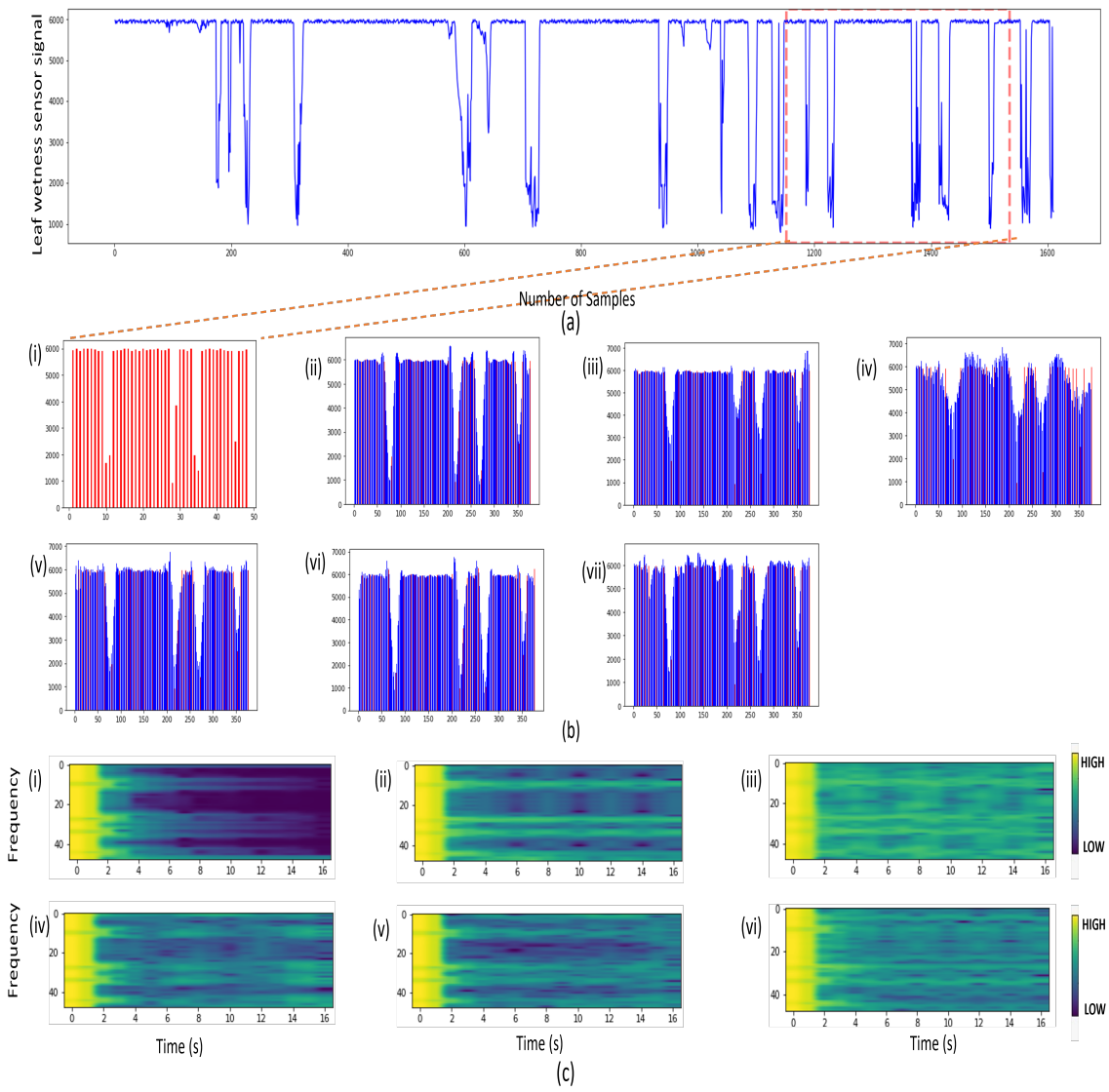


Figure 5.2: (a) **Leaf Wetness Sensor** *signal* 1; (b) **Stem plot** depicts the variation in density of the signal before and after applying time resolution. The process of time resolution increases the samples of the time series signal therefore the density of the signal increases afterward. **Stem plot** of (i) highlighted 48-point signal; **interpolated signals** using (ii) Bi-cubic (iii) ResNet (iv) SRPCNN (v) MU-GAN (vi) AudioUNetGAN (vii) This work; (c) **The spectrogram** is a visual representation of the spectrum of frequencies of a signal as it varies with time.**Spectrogram of the interpolated signals using** (i) Bi-cubic (ii) ResNet (iii) SRPCNN (iv) MU-GAN (v) AudioUNetGAN (vi) This work

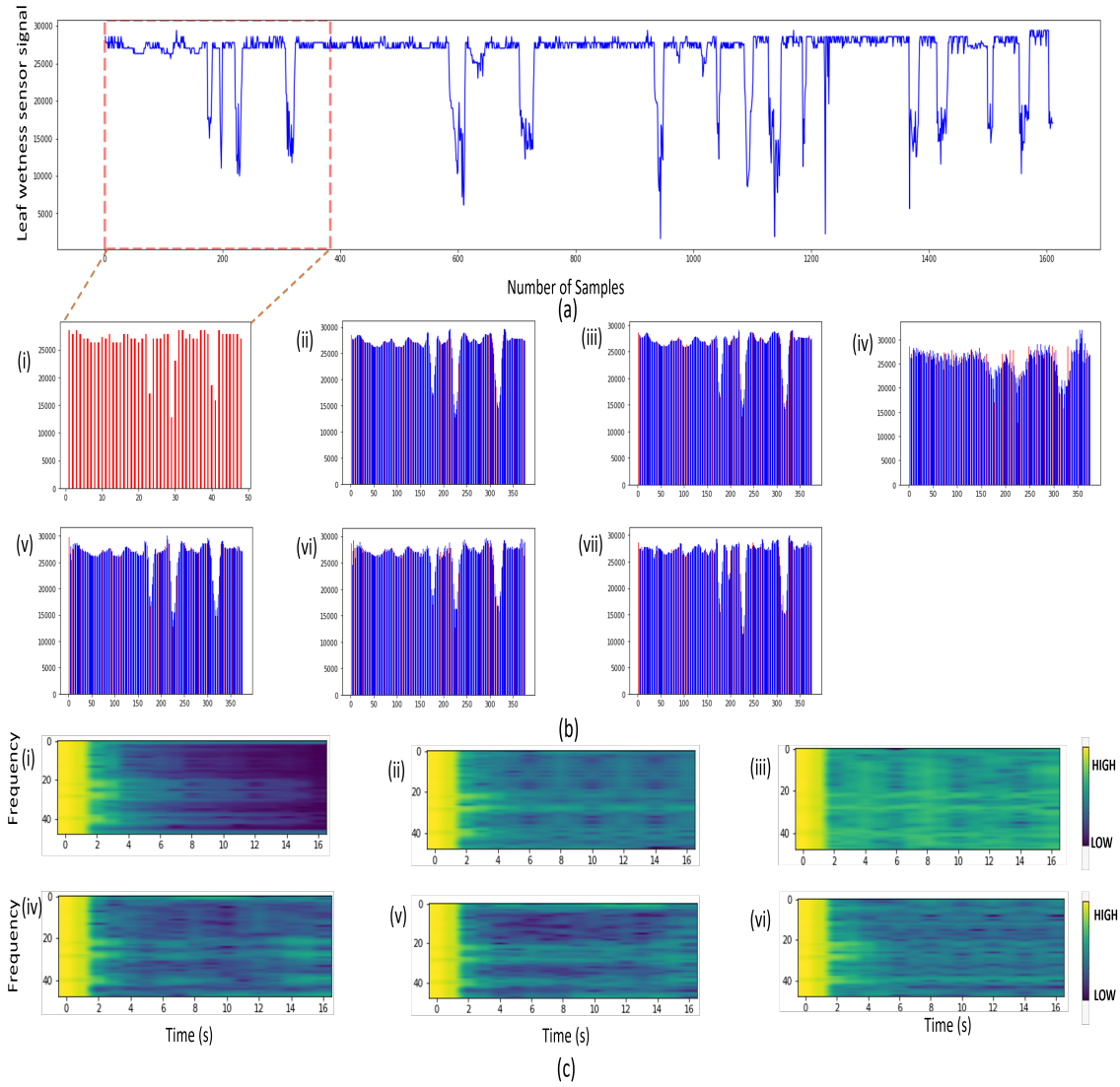


Figure 5.3: (a) **Leaf Wetness Sensor** *signal* 1; (b) **Stem plot** depicts the variation in density of the signal before and after applying time resolution. The process of time resolution increases the samples of the time series signal therefore the density of the signal increases afterward. **Stem plot** of (i) highlighted 48-point signal; **interpolated signals** using (ii) Bi-cubic (iii) ResNet (iv) SRPCNN (v) MU-GAN (vi) AudioUNetGAN (vii) This work; (c) **The spectrogram** is a visual representation of the spectrum of frequencies of a signal as it varies with time.**Spectrogram of the interpolated signals using** (i) Bi-cubic (ii) ResNet (iii) SRPCNN (iv) MU-GAN (v) AudioUNetGAN (vi) This work

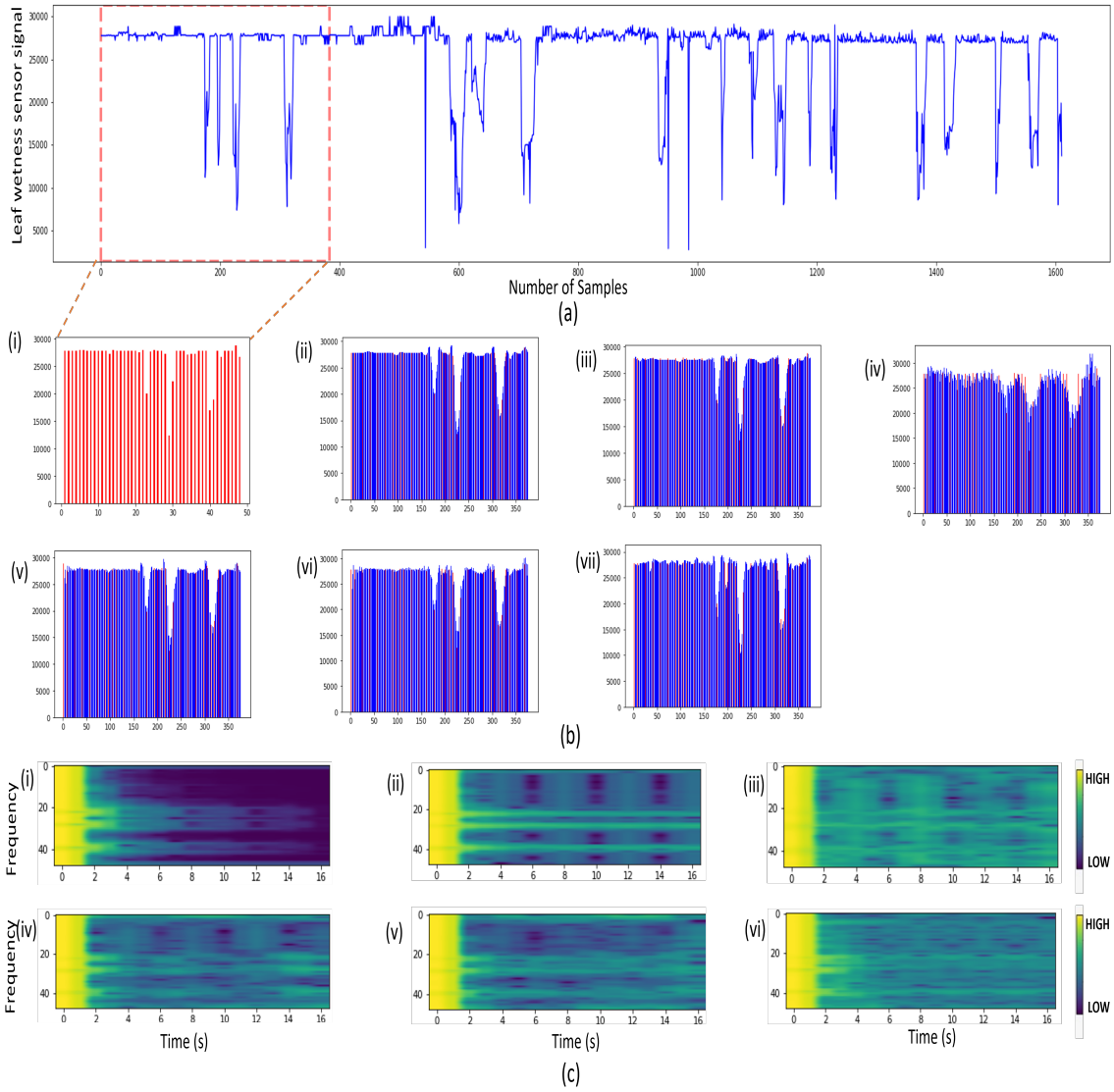


Figure 5.4: (a) **Leaf Wetness Sensor** *signal* 1; (b) **Stem plot** depicts the variation in density of the signal before and after applying time resolution. The process of time resolution increases the samples of the time series signal therefore the density of the signal increases afterward. **Stem plot** of (i) highlighted 48-point signal; **interpolated signals** using (ii) Bi-cubic (iii) ResNet (iv) SRPCNN (v) MU-GAN (vi) AudioUNetGAN (vii) This work; (c) **The spectrogram** is a visual representation of the spectrum of frequencies of a signal as it varies with time.**Spectrogram of the interpolated signals using** (i) Bi-cubic (ii) ResNet (iii) SRPCNN (iv) MU-GAN (v) AudioUNetGAN (vi) This work

ing the SRPCNN method. Similarly, the time-resolved signal using ResNet also shows abrupt variation in the magnitude of the interpolated samples. This can be observed in Fig. 5.2b(iii), where few interpolated data samples have higher estimated values even when the original signal is declining. It can be observed from Fig. 5.2b(vii) that our proposed architecture estimates the interpolated samples properly and does not suffer from any issues discussed for other methods. Similar observations can be made for other signals shown in Figs. 5.3(b)-5.4(b) respectively.

5.2.2 Spectrogram analysis

In order to quantify the issues related to the quality of interpolated values discussed in the aforementioned paragraph, we have used the spectrogram of the signal for qualitative analysis. The nature of interpolated values affects the spectrogram. If the interpolated signal values vary abruptly in magnitude and do not conform to the trends observed in the original signal, high frequency variations are recorded in the spectrogram. On the other hand, a smooth signal where the magnitude varies gradually will produce low frequency components in the spectrogram. The quality of the spectrogram is evaluated using the log-spectral distance (LSD) [34] metric defined as

$$LSD = \frac{1}{M}\sum_{m=1}^{M}\sqrt{\frac{1}{K}\sum_{k=1}^{K}(P(l,k)-\widehat{P}(l,k))^2} \tag{5.2}$$

where P and $\widehat{P}$ are the log-spectral power magnitudes of actual signal y and predicted signal $\hat{y}^{[3]}$, respectively. These are defined as $P = \log|T|^2$, where , where T stands for the signal's short-time Fourier transform (STFT). For the time and frequencies axis, we utilize the m and k indexes, respectively. Here, the lower LSD value indicates matching frequency content. It can be observed from Figs. 5.2(c)-5.4(c) that the spectrogram of the interpolated signal generated using the proposed architecture is better compared to other methods. This can be quantified using the LSD metric which is evaluated for spectrogram of interpolated signals and is given in Table 5.1. It can be observed from the table that the spectrogram for bi-cubic interpolated values is found to be the least accurate as the LSD values are much higher than the rest of the methods. Moreover, the ResNet, SRPCNN, AudioUNetGAN and MU-GAN have comparable performances but their LSD values are more than the proposed model. The spectrogram generated for the interpolated signal using the proposed model indicates that it is able to

capture the signal's frequency components better than the rest of the methods. This can be clearly seen in Table 5.1 where the lowest LSD values is obtained for the proposed architecture compared to the other methods.

Table 5.1: LSD values comparison for various methods

Methods	*signal* 1	*signal* 2	*signal* 3
Bi-cubic[15]	3.72	3.63	3.90
ResNet[18]	1.34	1.42	1.40
SRPCNN[29]	1.38	1.073	1.38
AudioUNetGAN[26]	1.34	1.09	1.27
MU-GAN[23]	1.38	1.14	1.37
This work	**1.30**	**1.069**	**1.09**

5.2.3 Signal to noise ratio analysis

Next, the quantitative comparison of the proposed architecture with the existing methods is evaluated in terms of the signal to noise ratio (SNR). In signal processing literature, the SNR is a commonly used metric for quantifying distortion present in the signal and is defined as

$$SNR\left(\hat{y}^{[3]}, y\right) = 20\log_{10}\frac{\|y\|_2^2}{\|\hat{y}^{[3]} - y\|_2^2} \tag{5.3}$$

where y is the target signal and $\hat{y}^{[3]}$ is the predicted signal. A higher SNR value indicates better signal quality. The SNR value calculated for all the methods is shown in Table 5.2. Here, the signals used for comparison are the LWS signals shown in Figs. 5.2-5.4. It can be observed from the Table 5.2 that the SNR values obtained for all three signals using the proposed architecture are greater than the other methods.

Table 5.2: SNR values comparison for various methods

Methods	*signal* 1	*signal* 2	*signal* 3
Bi-cubic[15]	14.58	22.27	21.23
ResNet[18]	14.93	22.99	21.81
SRPCNN[29]	13.48	18.52	18.08
AudioUNetGAN[26]	16.31	22.48	21.47
MU-GAN[23]	16.19	21.99	21.27
This work	**17.07**	**25.53**	**23.89**

5.3 Estimation of Leaf Wetness Duration (LWD)

In this section, the advantage of using a time-resolution signal in estimating the LWD values is discussed. The signal analyzed in this work is obtained from a leaf wetness sensor (LWS). The point where the signal value falls below the threshold level is considered to be the starting point of the event and when the signal value rises above the threshold level, the point is marked as the ending point of the wetness event. The time duration between the subsequent starting and stopping points corresponds to the LWD value [5]. The leaf wetness duration (LWD) values have been extracted from both the time-resolved signal as well as the original signal from the sensor.

To capture the LWD values accurately, it is necessary to apply data pre-processing techniques and clean the data. A technique called total variation denoising (TVD) has been used for this purpose. The method has been employed because it maintains the sharp rise and fall of the signal. This contributes positively in the sense that the baseline values which correspond to starting and ending points of the events will not be disturbed. The algorithm of pre-processing and feature extraction has been discussed in detail in one of our earlier works [5].

5.3.1 Representation of LWD values of Leaf wetness sensor signal

The bar graphs depicted in Fig. 5.5 are the representative of the LWD values extracted from the signals before and after applying the time-resolution methods. The values of LWD extracted from the fabricated sensors have been compared with the commercial sensors (*pythos* 31) for the purpose of bench-marking the sensors. Fig. 5.5(a) shows the LWD values extracted from the original signal using one of our earlier works[5]. It can be observed from Fig. 5.5(a) that the minimum absolute difference of the LWD values between the values from fabricated sensors and commercial sensors is ±0.5 hours. This is because the sensor output is recorded after every 30 minute interval. The main advantage of using time-resolution methods is that the minimum absolute difference between the fabricated and the commercial sensor gets reduced as seen in Figs. 5.5(b)-(g). However, there will be be fluctuations in the differences. Specifically, for Fig. 5.5(g) which plots the LWD values extracted from the time-resolved signal using proposed architecture. It can be noticed that most of the LWD values extracted from *sensors* 1, 2 and 3 are close to that from a commercial sensor with absolute differ-

ence of ±0.1. For the rest of the methods, namely, bi-cubic interpolation, ResNet, SRPCNN, AudioUNetGAN, and MU-GAN show a significant fluctuation in the LWD values ranging in between ±(0.2 to 0.4). This is evident from the bar graphs shown in Fig. 5.5(b)-(f) which are of varying heights signifying the error in the estimation of LWD values using these methods. The LWD values from the processed signals of lab-fabricated LWS are compared with those from the commercially available LWS. The RMSE from earlier work [5] is for signals that have not been processed. These signals have a time resolution of 30 minutes, hence having the largest room for error to occur in the estimation of LWD values. After time-resolution, the time between subsequent samples reduces to 4 minutes, thereby reducing the minimum absolute error. Bi-cubic interpolation and residual network have comparable performances. SRPCNN gives poor results owing to the disturbance caused in the baseline frequencies after using this method. Both the GAN methods also show less than average results. Overall, from Table III, it is evident that the progressive network proposed in this work displays a superior performance.

5.3.2 Root mean square error analysis for LWD of the sensor signal

To quantify the above discussion, root means square error (RMSE) is used as an evaluation metric which is defined as follows

$$RMSE = \sqrt{MSE} \tag{5.4}$$

where MSE is given in Eq. (5.1). The LWD values obtained from lab-fabricated sensors have been compared against those from commercial sensors using this metric. The values of the RMSE recorded for various methods have been tabulated in Table 5.3. It can be observed that the proposed architecture gives minimum RMSE and hence performs better compared to the other existing methods.

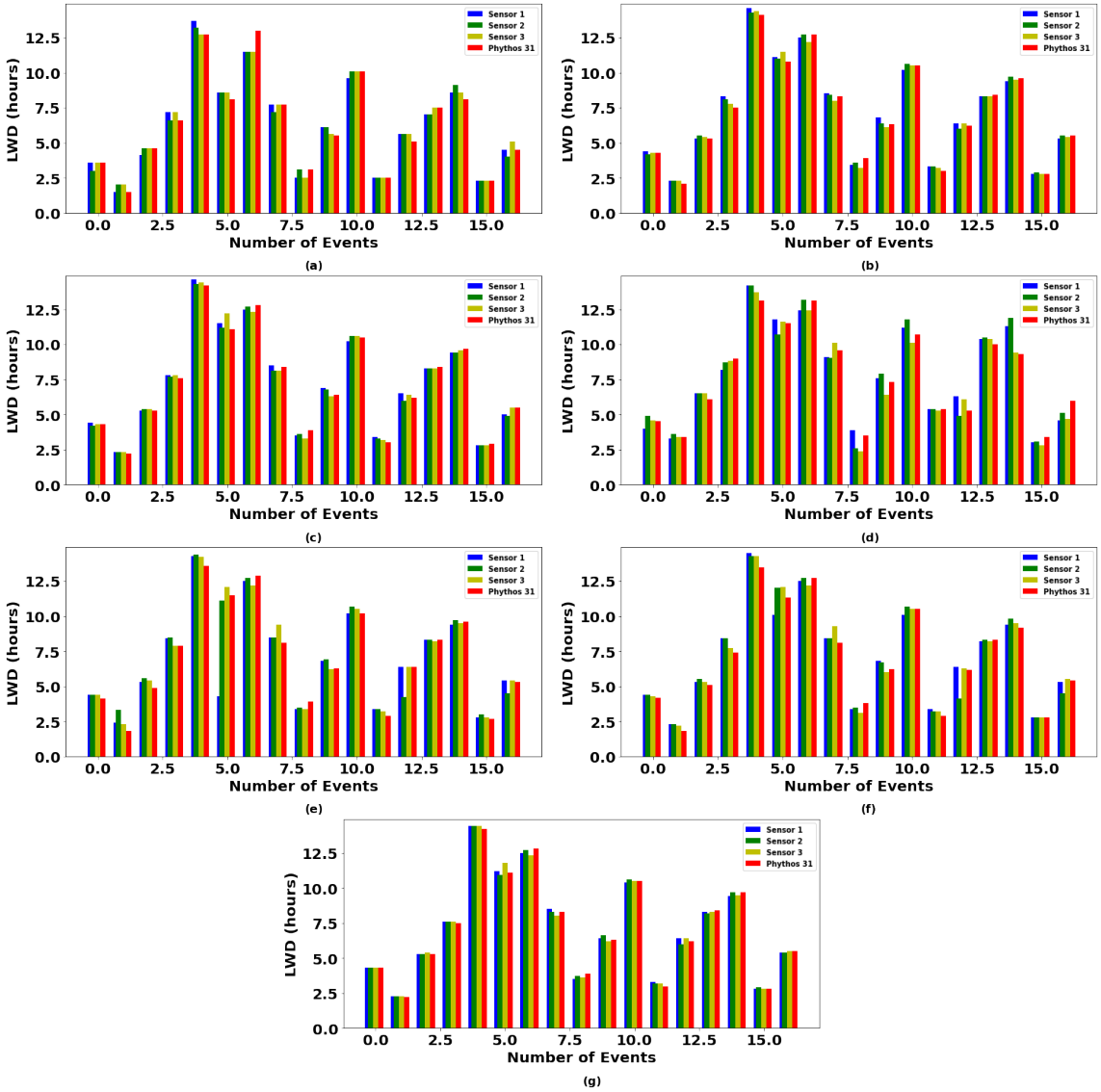


Figure 5.5: LWD values (a) without time-resolution (earlier work) (b) Bi-cubic interpolation (c) ResNet (d) SRPCNN (e) AudioUNetGAN (f) MU-GAN (g) This work

Table 5.3: Comparison of RMSE values for various methods

Methods	*sensor* 1	*sensor* 2	*sensor* 3
Earlier work [5]	0.5856	0.5800	0.5052
Bi-cubic[15]	0.3360	0.2482	0.3162
ResNet[18]	0.3058	0.2249	0.3547
SRPCNN[29]	0.8000	0.8762	0.6216
AudioUNetGAN[26]	0.5397	0.7076	0.4864
MU-GAN[23]	0.7875	0.7886	0.4832
This work	**0.1909**	**0.1514**	**0.2578**

CHAPTER 6
Conclusions

Leaf wetness duration is an essential parameter that determines plant growth and spread of diseases. Accurate estimation of LWD is essential for successful prediction of diseases in plants. In this thesis, a deep learning model for improving the time-resolution of the LWS signals is proposed. The model is used for reconstructing a high-resolution LWS signal from its low-resolution counterpart. This helps in improving the estimation of LWD parameter. The model can successfully learn prior information from historical data of LWS. By training from the LWS signal's dataset, the model can successfully predict high-resolution signals from the low-resolution signals using an efficient progressive feature extraction process. The qualitative and quantitative results of the proposed architecture are superior compared to the existing methods when evaluated in terms of root means square error.While the performance of the proposed network surpassed the other methods used for comparison, the time for training the model was the highest. The model complexity for higher levels of time-resolution could be much higher. In future, we would like to address this issue by proposing alternate deep learning architecture.

References

[1] A. Sajeena, J. John, B. Sudha, A. Meera, and S. Karthika, "Significance of botanicals for the management of plant diseases," in *Plant Health Under Biotic Stress*, pp. 231–243, Springer, 2019.

[2] M. G. Lawrence, "The relationship between relative humidity and the dewpoint temperature in moist air: A simple conversion and applications," *Bulletin of the American Meteorological Society*, vol. 86, no. 2, pp. 225–234, 2005.

[3] L. Granke and M. Hausbeck, "Effects of temperature, humidity, and wounding on development of phytophthora rot of cucumber fruit," *Plant Disease*, vol. 94, no. 12, pp. 1417–1424, 2010.

[4] M. Kumar, A. Kumar, and V. S. Palaparthy, "Soil sensors-based prediction system for plant diseases using exploratory data analysis and machine learning," *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17455–17468, 2020.

[5] K. S. Patle, R. Saini, A. Kumar, S. G. Surya, V. S. Palaparthy, and K. N. Salama, "Iot enabled, leaf wetness sensor on the flexible substrates for in-situ plant disease management," *IEEE Sensors Journal*, vol. 21, no. 17, pp. 19481–19491, 2021.

[6] L. Huber and T. Gillespie, "Modeling leaf wetness in relation to plant disease epidemiology," *Annual review of phytopathology*, vol. 30, no. 1, pp. 553–577, 1992.

[7] T. Gillespie and G. Kidd, "Sensing duration of leaf moisture retention using electrical impedance grids," *Canadian Journal of Plant Science*, vol. 58, no. 1, pp. 179–187, 1978.

[8] H. Häckel, "New developments of an electrical method for direct measurement of the wetness-duration on plants," *Agricultural Meteorology (Netherlands)*, 1980.

[9] R. Getz, "Report on the measurement of leaf wetness," *WMO (Commission for Instruments and Methods of Observation), Rome*, 1991.

[10] A. Madeira, K. Kim, S. Taylor, and M. Gleason, "A simple cloud-based energy balance model to estimate dew," *Agricultural and Forest Meteorology*, vol. 111, no. 1, pp. 55–63, 2002.

[11] R. A. Miranda, T. D. Davies, and S. E. Cornell, "A laboratory assessment of wetness sensors for leaf, fruit and trunk surfaces," *Agricultural and forest meteorology*, vol. 102, no. 4, pp. 263–274, 2000.

[12] P. C. Sentelhas, T. J. Gillespie, M. L. Gleason, J. E. B. Monteiro, and S. T. Helland, "Operational exposure of leaf wetness sensors," *Agricultural and Forest Meteorology*, vol. 126, no. 1-2, pp. 59–72, 2004.

[13] V. Kuleshov, S. Z. Enam, and S. Ermon, "Audio super resolution using neural networks," *arXiv preprint arXiv:1708.00853*, 2017.

[14] O. Rukundo and H. Cao, "Nearest neighbor value interpolation," *arXiv preprint arXiv:1211.1768*, 2012.

[15] D. Khaledyan, A. Amirany, K. Jafari, M. H. Moaiyeri, A. Z. Khuzani, and N. Mashhadi, "Low-cost implementation of bilinear and bicubic image interpolation for real-time image super-resolution," in *2020 IEEE Global Humanitarian Technology Conference (GHTC)*, pp. 1–5, IEEE, 2020.

[16] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, *et al.*, "Scipy 1.0: fundamental algorithms for scientific computing in python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[17] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, "Deep convolution neural network for image recognition," *Ecological Informatics*, vol. 48, pp. 257–268, 2018.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[19] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[20] Z. Lu, X. Jiang, and A. Kot, "Deep coupled resnet for low-resolution face recognition," *IEEE Signal Processing Letters*, vol. 25, no. 4, pp. 526–530, 2018.

[21] H. Li, X.-j. Wu, and T. S. Durrani, "Infrared and visible image fusion with resnet and zero-phase component analysis," *Infrared Physics & Technology*, vol. 102, p. 103039, 2019.

[22] B. Li and D. Lima, "Facial expression recognition via resnet-50," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 57–64, 2021.

[23] S. Kim and V. Sathe, "Adversarial audio super-resolution with unsupervised feature losses," 2018.

[24] X. Li, V. Chebiyyam, K. Kirchhoff, and A. Amazon, "Speech audio super-resolution for speech recognition.," in *INTERSPEECH*, pp. 3416–3420, 2019.

[25] Y. Liu, "Recovery of lossy compressed music based on cnn super-resolution and gan," in *2021 IEEE 3rd International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, pp. 623–629, IEEE, 2021.

[26] J. King, R. V. Torné, A. Campbell, and P. Liò, "An investigation of pre-upsampling generative modelling and generative adversarial networks in audio super resolution," *arXiv preprint arXiv:2109.14994*, 2021.

[27] J. Gu, H. Chen, G. Liu, G. Liang, X. Wang, and J. Zhao, "Super-resolution perception for industrial sensor data," *arXiv preprint arXiv:1809.06687*, 2018.

[28] G. Liang, G. Liu, J. Zhao, Y. Liu, J. Gu, G. Sun, and Z. Dong, "Super resolution perception for improving data completeness in smart grid state estimation," *Engineering*, vol. 6, no. 7, pp. 789–800, 2020.

[29] G. Liu, J. Gu, J. Zhao, F. Wen, and G. Liang, "Super resolution perception for smart meter data," *Information Sciences*, vol. 526, pp. 263–273, 2020.

[30] I. de Paz-Centeno, M. T. García-Ordás, O. García-Olalla, J. Arenas, and H. Alaiz-Moretón, "M-srpcnn: A fully convolutional neural network approach for handling super resolution reconstruction on monthly energy consumption environments," *Energies*, vol. 14, no. 16, p. 4765, 2021.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, PMLR, 2015.

[32] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] V. Kuleshov, S. Z. Enam, and S. Ermon, "Audio super-resolution using neural nets," in *ICLR (Workshop Track)*, 2017.