EEEE3001
Final Year Individual Project Thesis

Crystallographic orientation determination through machine learning for Nickel.

AUTHOR:	**Jiancheng SUN**
ID NUMBER:	**20216906**
SUPERVISOR:	**Dr Rikesh Patel, Dr Wenqi Li**
MODERATOR:	**Dr Kevin Webb**
DATE:	**2023/5/6**

Third *(Fourth if applicable)* year project report is submitted in part fulfilment of the requirements of the degree of Bachelor *(Master if applicable)* of Engineering.

ABSTRACT:

Artificial intelligence (AI) is burgeoning approach in signal processing area to complete tasks related to characterisation and classification. In the biochemistry area, this technique has also been applied to various application for more accurate and faster signal analysis. The Surface Acoustic Velocity (SAW) had been proofed to critical in determining the crystallographic orientation of nickel by an existing technique called Spatially Resolved Acoustic Spectroscopy (SRAS). The relation between SAW and crystallographic orientation has been obtained, and an algorithm called 'Brute Force' to correlate SAW with nickel orientation has been proposed and verified in previous research. In this project, AI techniques were applied to generate two convolutional neural networks, image-to-label classification neural network, and sequence-to-label classification neural network. These networks were used for categorizing 441 orientations of nickel based on the different SAW velocities after training them with samples and specific noise. The performance for different neural networks will be recorded and analysed, the methodology about how to properly design the network, how to design the noisy simulation data, and how to test the neural network was introduced for clarity. Finally, the conclusion about this project was related to how well the neural network fits the requirement, how to understand the testing result in an objective view, and how to improve the project to achieve better results.

Table of Contents

1 INTRODUCTION: 1

1.1 BACKGROUND REVIEW: 1
1.1.1 Microstructure: 1
1.1.2 Nickel: 1
1.1.3 Miller Index: 2
1.1.4 X-ray diffraction (XRD) Laue back-reflection (LBR) Method: 2
1.1.5 Electron Backscattered Diffraction (EBSD): 3
1.1.6 Spatially Resolved Acoustic Spectroscopy (SRAS): 3
1.1.7 Orientation Determination through Bruce Force Method 5
1.1.8 Misorientation: 5
1.2 PROJECT OVERVIEW: 5
1.2.1 Aims: 6
1.2.2 Objectives: 6
1.3 DELIVERABLES AND MILESTONES: 7
1.3.1 Deliverables with risk management: 7
1.3.2 Milestones: 7

2 LITERATURE REVIEW: 8

2.1 BRUTE FORCE METHOD: 8
2.2 ARTIFICIAL INTELLIGENCE METHOD: 8
2.2.1 Artificial Neural Network (ANN) and Deep Learning (DL): 8
2.2.2 Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM): 10
2.2.3 Convolution Neural Network (CNN): 10
2.3 TRAINING ENVIRONMENT: 11
2.3.1 Python: 11
2.3.2 MATLAB: 11
2.4 NOISE SIMULATION: 12
2.4.1 Gaussian Noise: 12
2.4.2 Gaussian White Noise 13

3 METHODOGY: 13

3.1 MAIN PROCESS: 13
3.1.1 'Image' featured processing: 13
3.1.2 'Sequence' featured processing: 14
3.2 SAW VELOCITY REPRESENTATION: 14
3.2.1 Polar Plot: 14
3.2.2 1 Dimension Numerical Sequence: 15
3.3 NOISE SIMULATION: 16
3.3.1 Gaussian Noise in MATLAB: 16
3.3.2 Gaussian White Noise in MATLAB: 16
3.3.2.1 SNR 17
3.3.2.2 Polar Plot After Noise: 17
3.3.2.3 1 Dimension Numerical Sequence After Noise: 18
3.4 IMAGE PRE-PROCESSING: 18
3.4.1 Image Cropping and Resizing: 18
3.4.2 Image to Grey: 19
3.5 CONVOLUTION NEURAL NETWORK: 20
3.5.1 2-D convolution (image classification): 20
3.5.1.1 Network Structure: 20
3.5.1.2 Training Strategy and Options: 21
3.5.1.3 Validation Strategy: 22
3.5.1.4 Verification Strategy: 23
3.5.2 1-D convolution (sequence classification): 23
3.5.2.1 Network Structure: 24
3.5.2.2 Training Strategy and Options: 25
3.5.2.3 Validation Strategy: 26
3.5.2.4 Verification Strategy: 26

4 RESULTS AND DISCUSSION: 26

4.1 Image Classification: 26

4.1.1 Training Results and analysis: 27

4.1.2 Validation Results and analysis: 29

4.1.3 Verification Results and analysis: 32

4.2 Sequence Classification: 33

4.2.1 Training Results and analysis: 33

4.2.2 Validation Results and analysis: 36

4.2.3 Verification Results and analysis: 39

5 CONCLUSION: 40

6 REFERENCE: 41

1 INTRODUCTION:

1.1 Background Review:

1.1.1 Microstructure:

In engineering applications, the structure of materials relates to its component's arrangements. Within the atomic domain, the structure is described as the atoms' organization. Generally, metals have four crystal structures: (i) face-centered cubic (fcc), (ii) body-centered cubic (bcc), (iii) hexagonal closed-packed (hcp), and (iv) tetragonal.

The next structural level is micro-dimension level, where atomic arrangements are considered as what we called microstructure. To be more detailed, the microstructure of metal material is measured by grain size, types of phases present, and crystal orientations. These factors in the micro-level can strongly influence the material physical properties such as strength, toughness, resistance, which are really important in the manufacturing industry. In one experiment, during linear friction welding (LFW), it was discovered that the orientation of single fcc-based crystal is extremely important because of changes in orientation of the primary slip [1].

1.1.2 Nickel:

Nickel is a metal in periodic table of elements with atomic number 28. The unit cell of nickel is the face-centred cubic structure, with the lattice parameter of 0.352 nm, and atomic radius of 0.124 nm [2]. Due to its plasticity and corrosion resistance, nickel and its alloys are largely applied in electrical, chemical, and aerospace engineering. For example, when nickel is combined with chromium, the hardness, ductility, fatigue resistance will be improved [3].

Understanding of the nickel or nickel alloy crystals' orientation is important engineering manipulation. For example, in the application of single crystal nickel-based turbine blade, some researchers had concluded that by optimizing the orientation of nickel, the elastic stress of the material can be reduced, thus will potentially increase the fatigue resistance of the turbine blade [4]. In this case, it is necessary to get the precise orientation of the nickel crystal using some techniques.

Figure 1. Nickel [5]

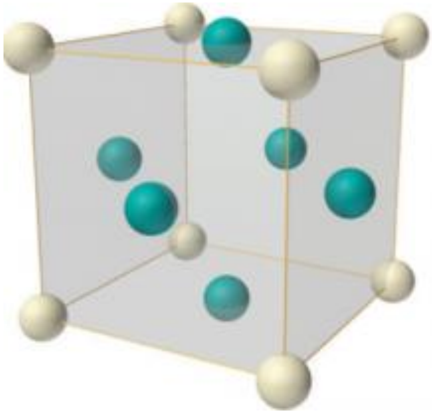


Figure 2. Face-centred cubic structure [6]

1.1.3 Miller Index:

Given the importance to determine the crystal orientation of Nickel, it is necessary to define a mathematical representation to express the orientation of crystal in the real world. Miller indices are the groups of 3 numbers indicating the direction of a plane in crystal system. A simple example about how to derive that is shown:

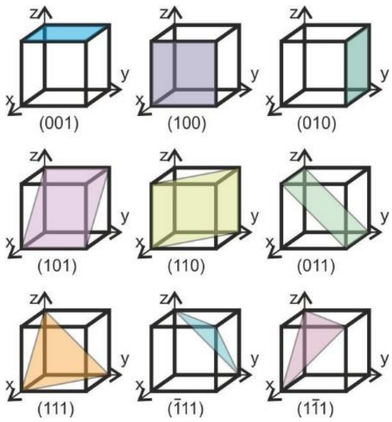


Figure 3. Miller index representations [7]

Firstly, consider the blue surface as shown in Figure 1, which is the top surface of the cubic. The intercepts of this surface to x, y, z axes are (∞, ∞, 1). In this case, derive the reciprocal of each intercept and get the result.

$$\left(\frac{1}{\infty}, \frac{1}{\infty}, \frac{1}{1}\right) = (0, 0, 1)$$

As a result, the blue plane can be represented by the miller index (0,0,1). The derivation for the rest planes is similar, and to be noted, there are a total of 6 faces equivalent to the (100) surface in the nickel crystal because of the symmetry elements.

1.1.4 X-ray diffraction (XRD) Laue back-reflection (LBR) Method:

X-rays can be considered as electromagnetic radiation waves, and the Laue back-reflection is strongly related to the Bragg diffraction. It was first proposed by William Lawrence Bragg and William Henry Bragg in 1913 responding to their discovery that Bragg peaks were produced when X-rays incite the crystals. This phenomenon can be explained by modelling the crystal as some parallel planes separated by a constant distance d using Bragg equation [7].

$$n\lambda = 2dsin\theta \qquad \textit{Eqn.1}$$

Where, d is the interplanar distance, θ is the incident angle, and λ is the light wavelength. The figure is shown below.

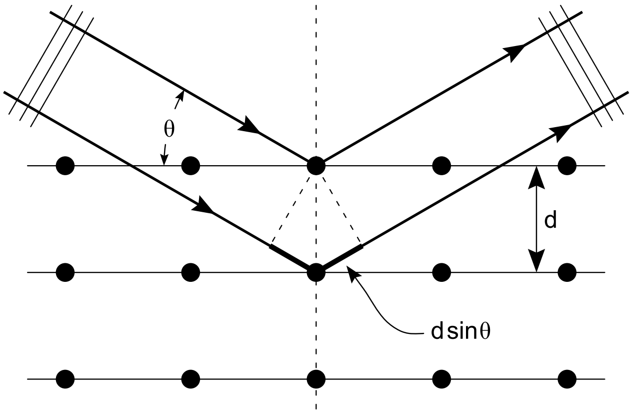


Figure 4. XRD-LBR method diagram [8] and SEM machine for EBSD [17]

Laue back-reflection (LBR) Method is achieved by a beam of white X-rays, it is incident to a stationary single crystal. In a particular orientation, d and θ are fixed, and a film is set between the light source and crystal to record the backwards diffraction. Generally, crystal orientation can be determined by the position of the spots on the film. LBR method can typically achieve a 2mm diameter system resolution with an error of 1.5 degree [9].

1.1.5 Electron Backscattered Diffraction (EBSD):

Electron Backscatter Diffraction (EBSD) is a scanning electron microscope (SEM) based technique, enabling the sample's microstructure to be observed and analysed. For EBSD, an electron beam is incident to a tilted crystal sample and the diffracted electrons will be captured with a fluorescent screen to form a EBSD pattern [10].

During the EBSD pattern formation, when the electron beam is incident to the sample, it can be scattered in all directions. These scattering forms a divergent source of electrons near the surface of sample, while some of electrons are directed to crystal planes to satisfy Bragg equations as shown in **1.1.4**. These electrons will be diffracted on the screen to form cones corresponding to each individual plane, for example the green and blue cone in Figure 5 is the (110) plane.

However, as the technique is sensitive, samples should be sufficiently polished beforehand. The major advantage of using EBSD is its resolution, typically, the spatial resolution is <10 nm [13].

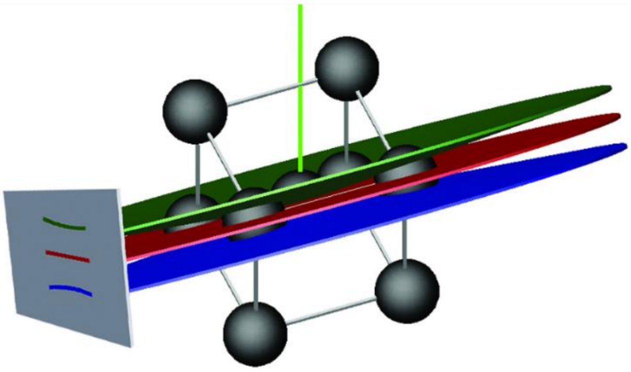


Figure 5. EBSD diagram [11]

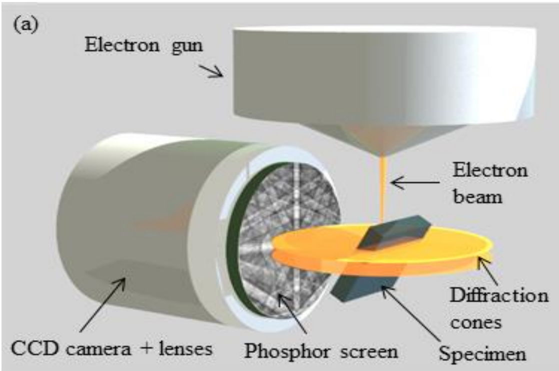


Figure 6. EBSD machine simulation [12]

1.1.6 Spatially Resolved Acoustic Spectroscopy (SRAS):

SRAS is a non-contact, non-destructive method to get the surface acoustic waves (SAW) from a material surface. This technique has been used in many applications by previous researchers to image different metals and their alloys. The realization of SRAS is completed by a pulsed laser, covered by an optical mask. The laser is incident to the sample material to be absorbed and create some surface waves. In the current research, a glass mask coated with chrome is applied to supply a fringe pattern with known and fixed spacing. To be specific, all fringes generate lots of SAWs covering a series of frequencies with the combination of source laser pulse. There is another detection laser with a knife-edged detector to select and receive SAWs whose wavelengths are equal to the fringe spacing, and those SAWs are enhanced in intensity [14]. As a result, the grating fringes spacing is equal to the SAW wavelength labelled as λ. We already know the frequency content of source laser through Fast Fourier Transformation in Figure 7, so the SAW amplitude value can be calculated by the following equation.

$$v = f\,\lambda \qquad \textit{Eqn.2}$$

Where, f is the pulse frequency and λ is the fringe spacing.

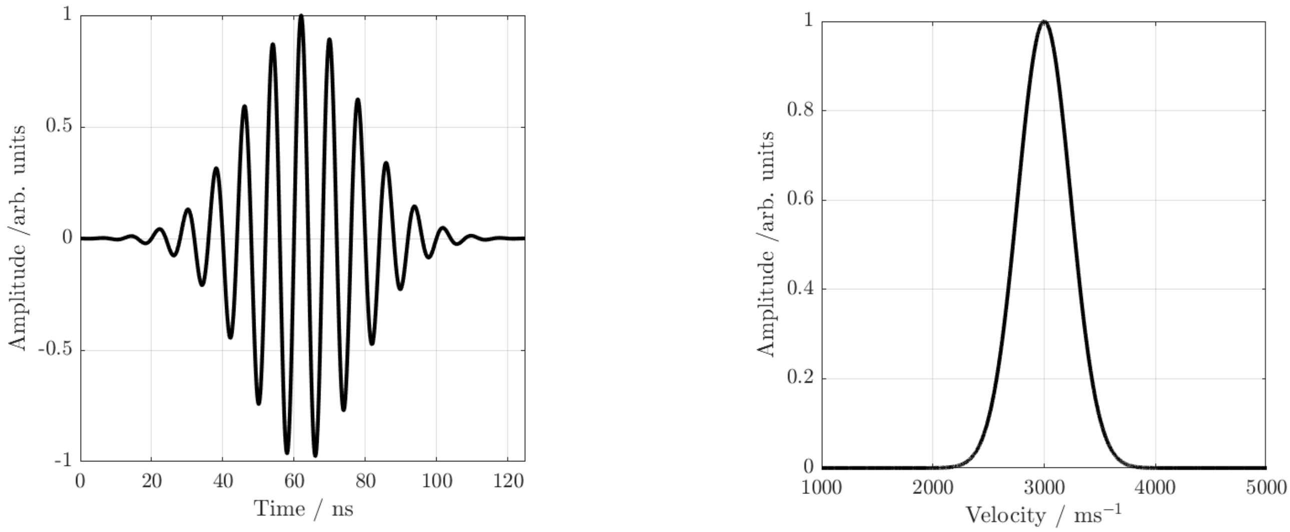


Figure 7. Fourier Transform for SAW velocity [14]

In the current system, the source laser is a Q-switched laser with pulse widths of 1-2 ns and energy of 50-150 μJ. The detection laser is a Laser Quantum wave laser with 200mW output [13]. The scanning is controlled by 2 dimensional linear moving stages to control the output image size.

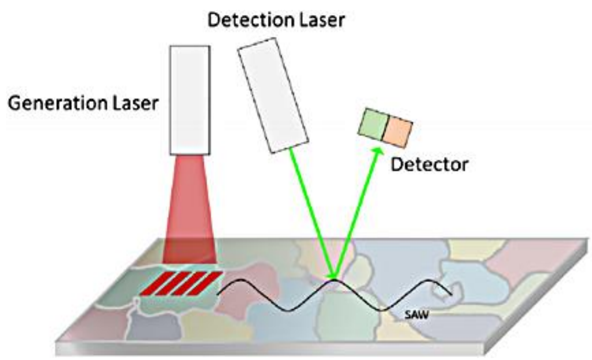


Figure 8. SRAS general construction [13]

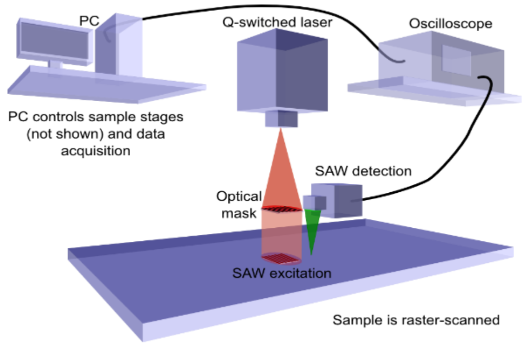


Figure 9. Q-switched SRAS construction [18]

1.1.7 Orientation Determination through Bruce Force Method

Since SAW velocity for specific plane can be obtained, to determine the orientation of crystal, it is necessary to do the SRAS scan over different planes. The scanning is taking on a random spot on the sample and make the SAWs travel in an arbitrary direction, thus different results of velocity surface can be obtained, which is a polar plot showing the SAW amplitude within 360 degrees. According to the previous research, there is a three-way relationship between material elastic constant, SAW velocity, and crystal orientation [16]. Knowing any of two allows the determination of the last one. For example, in the forward model, the SAW values can be measured through SRAS knowing the elastic constant and miller index, in this case, a database can be built to relate the miller index with SAW results. Based on the database, if a measured SAW velocity is obtained, then it will be compared with database using Brute Force method to get the most possible miller index. Typically, this process will be time-consuming and spend several days, and if there is any noise in the measurement, the accuracy will be very low, which will be verified and explained in detail in the next part.

1.1.8 Misorientation:

In this project, the difference in orientation between label and predicted plane from neural network will be assed using a value known as 'R-value', which is an industrial standard applied for over 40 years to determine the misorientation between two grains [14]. According to Wenqi [14], there are five methods to calculate R value: single-angle cos, two-angle cos, two-angle RMS, three-angle cos, and three-angle RMS. The two-angle cos method will be used in this project,

$$R - value = \arccos(\cos\varphi cos\tau) \qquad Eqn.3$$

Where φ is the minimum angle to rotate grain A to the orientation of grain B, τ is the rotation angle to superpose two grains after the first step. The range of R-value is between 0 degree and 62.8 degree, where smaller R value means two grains are closer to each other, and zero means the prediction result is the same as the label.

1.2 Project Overview:

While the normal Bruce Force Method will consume several days in determining the crystal orientations, this project will mainly focus on using the artificial intelligence method to determine the crystal orientation of nickel. Compared to the Bruce Force method, it is more timesaving as a pretrained deep learning neural network will normal take less than 10 second to recognize the test input pattern, despite of the large training time.

The first step of the project is to build two simple neural networks for the classification tasks. The first neural network is responsible for image classification, the input image is the SAW surface, shown as the polar plots image for different crystal surface, the label is the miller indices. The second neural network is responsible for sequence classification, the input data is the sequence SAW results from SRAS, the label is also the miller indices.

For the second step, specific noise with different SNRs is going to be modelled to raw data. The neural networks will be trained with noisy data, and the neural network after training will be saved for testing.

In the testing stage, the neural networks will be firstly validated with some sets of simulated noisy SAW, this method will be used for proofing the neural network can deal with the input with specific simulated noise. The desired results will be related to the comparison between different neural networks, the prediction accuracy at SNR30, and the processing time for prediction. After testing with validation data, the neural network will be verified with real data, which means there will be unknown noise in the signal, the R value results will be posted and analysed.

1.2.1 Aims:

To decrease the time consuming in the nickel crystal orientation determination process, apply artificial intelligence network for orientation classification based on SAW velocity database.

1.2.2 Objectives:

1) Gain an understanding of the background research about methods to determine crystal orientations and the project requirements, complete the project proposal.
2) Gain the basic understanding of how to program in MATLAB, including the fundamental matrix transformation and image processing algorithm.
3) Acquire the basic understanding of theory in artificial intelligence, learn how to use machine learning and deep learning method in MATLAB to design the neural network.
4) Based on deep learning algorithm, build image and sequence neural networks to determine the miller-index of nickel according to the SAW velocity on different directions.
5) Based on shape learning of polar plot, train the image neural network to determine the miller-index of nickel according to the SAW velocity on different directions.
6) Based on learning of velocity in 1-D dimension plot, train the sequence neural network to determine the miller-index of nickel according to the SAW velocity on different directions.
7) Based on the possible noise occurring in the real world, build a noise model to simulate the noise in the actual experiment, and add the noise to the raw data.
8) To consider noise in the data, update both the image and sequence neural network, and train the new neural network with noisy data with different SNR.
9) Record the different results of classification for different noise range, analyze the accuracy and try to increase it as much as possible.
10) Based on the current neural network, test the networks with real data from experiments, record the R values.

1.3 Deliverables and Milestones:

1.3.1 Deliverables with risk management:

1) The project proposal with detailed time plan on tasks, could be completed before 3rd, November (Achieved).

2) The polar plot representation in MATLAB of SAW velocity for different material (Achieved).

3) The neural network structure in MATLAB to cater for the requirements in Objectives: 5) 6) This deliverable will have some risks related to the little tutorial on building specific neural network in MATLAB. However, some possible hints from the previous methods applied in similar data (Achieved).

4) The noise model for simulating noise in the real world and applied to the raw data (Achieved).

5) Image and sequence networks are updated and pretrained with noise model, the training is adequate ignoring the actual accuracy (Achieved).

6) Artificial neural network validation using simulated noise for different SNRs, the accuracy for specific SNRs were recorded (Achieved)

7) Artificial neural network verification using real unknown noise from real experiments, the accuracy for specific planes were recorded (Achieved)

8) The completion of project thesis before the required deadline. The specific workshop and regular meetings with supervisor will be helpful in answering the questions related to the contents in the thesis. The potential risk here might be related to the loss of experimental data and MATLAB code.

Actually, the similar issues occur before the Easter break: The laptop for running code and storing data was stolen by someone from my house, because I hadn't backed up my data, so it took about the entire Easter break to redo the final year project. Fortunately, the lab book for recording is still there and I can remember almost everything in the code, this project was recovered to what it is before Easter, and the Final year project thesis is going to be completed before deadline (Achieved)

1.3.2 Milestones:

1) The polar plot representation of SAW velocity can be successfully demonstrated in the MATLAB.

2） The noise model is completed and summarized as an external function in MATLAB, ready to use in real applications.

3) The structure of neural network to deal with classification is completed, the system can be trained properly based on database, and the training result is shown to be valid for certain errors.

4) The neural network is verified with real unknown noise from experiments, the accuracy is recorded, and proper discussion will be applied to analyse

5) The final year project thesis is totally completed, ready for submission with every sections being reviewed.

2 LITERATURE REVIEW:

In this Chapter, the relevant methods and techniques applied in this project will be introduced.

2.1 Brute Force Method:

As mentioned in Chapter 1: Introduction, the brute-force method is currently applied to the SRAS system to determine the crystal orientation. Brute Force method is also called exhaustive search, which is the straight-forward method to solve a problem by trying everyone possibility. To be detailed, in order to find the problem solution, the brute-force algorithm will use a trail-and-error strategy by fitting all the possible solutions and find the most suitable one. For example, if a door can only to be unlocked by a specific key, to define the correct one, a person could try every key until finding the suitable one. On the one hand, this method is achievable if the time is sufficient. On the other hand, the accuracy for finding the correct key will be affected by other factors, like the rust on the key, means sometimes even the correct key cannot open the door because erosion affects its shape. In the current SRAS system, an overlap function defined by Li et al. is applied with brute-force method to determine the crystal orientation from the testing data [16]. However, the time consuming in Bruce-force method will increase significantly if the database is large, and this is the problem needs to be solved currently.

2.2 Artificial Intelligence Method:

2.2.1 Artificial Neural Network (ANN) and Deep Learning (DL):

Artificial neural network is an artificial intelligence or machine learning method to teach computer ‘think’ as a human brain. An ANN can be imagined as a collection of connected nodes, which is called artificial neurons. The artificial neuron is the definition inspired by the biological neurons. Usually, the ANNs are applied for modelling the real neural networks from different animals and make decisions just as the human does. In recent year, different ANNs have been applied to solve engineering problems, such as robotics, pattern recognition, computer vision, speech processing, natural language processing… The common artificial neural network structure can be summarized as input layer, hidden layer, and output layer. To be specific, the input layers are used for defining the input data dimension, which are multiplied by weights. Weights assigned to each row represent information flow. Sometimes, those weights will be added with some bias to achieve higher accuracy, and then computed by an activation function to determine the threshold of the neuron, this result will be passed to next stage, and the output will be derived at the last hidden layer [19]. Figure 10 shows a typical structure of ANN.

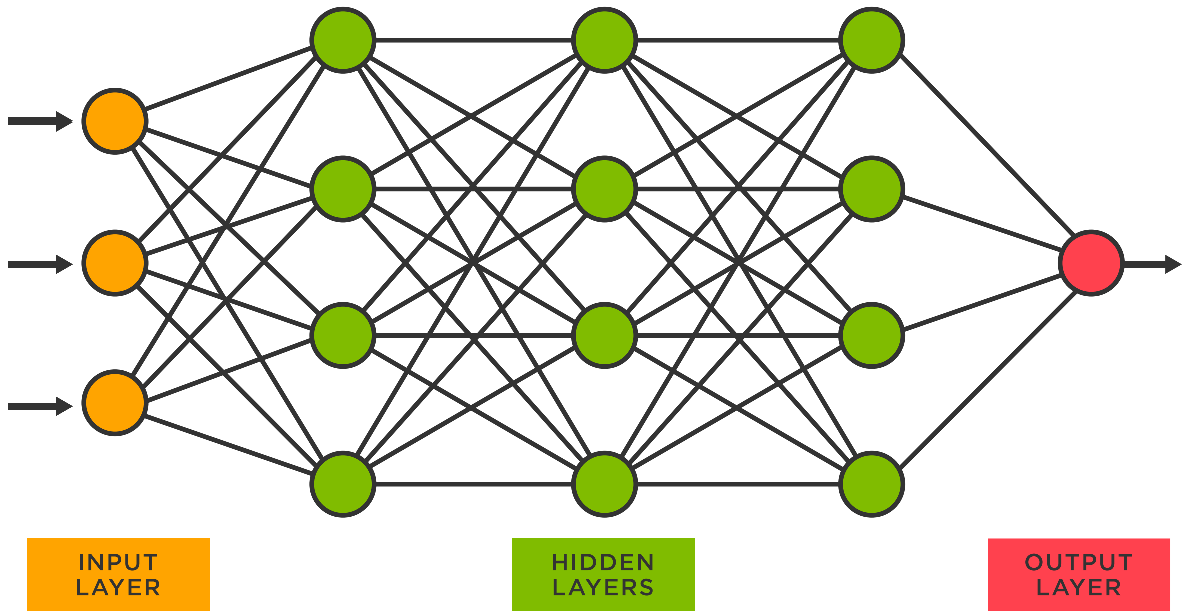


Figure 10. Typical structure of ANN [20].

When the structure of ANN is completed, it is necessary to consider how to train it. Normally, as declared before, the machine learning method is used to train the neural network. Specifically, when the neural network has more than one hidden layer, the learning process is called “deep learning”. Compared with machine learning, deep learning neural networks are more complicated and can deal with more complex data. The key advantage of deep learning is that it almost doesn’t need data pre-processing before training to do feature extraction, which can’t be avoidable in machine learning [21]. On the other hand, within the deep learning area, it consists of several neural network structures, Multilayer perceptron (MLP), Recurrent Neural Networks (RNN), Convolution Neural Networks (CNN)… Normally, MLP is a simpler model similar with CNN and specialize in image classification, so only CNN and RNN will be introduced in this thesis and applied in the project.

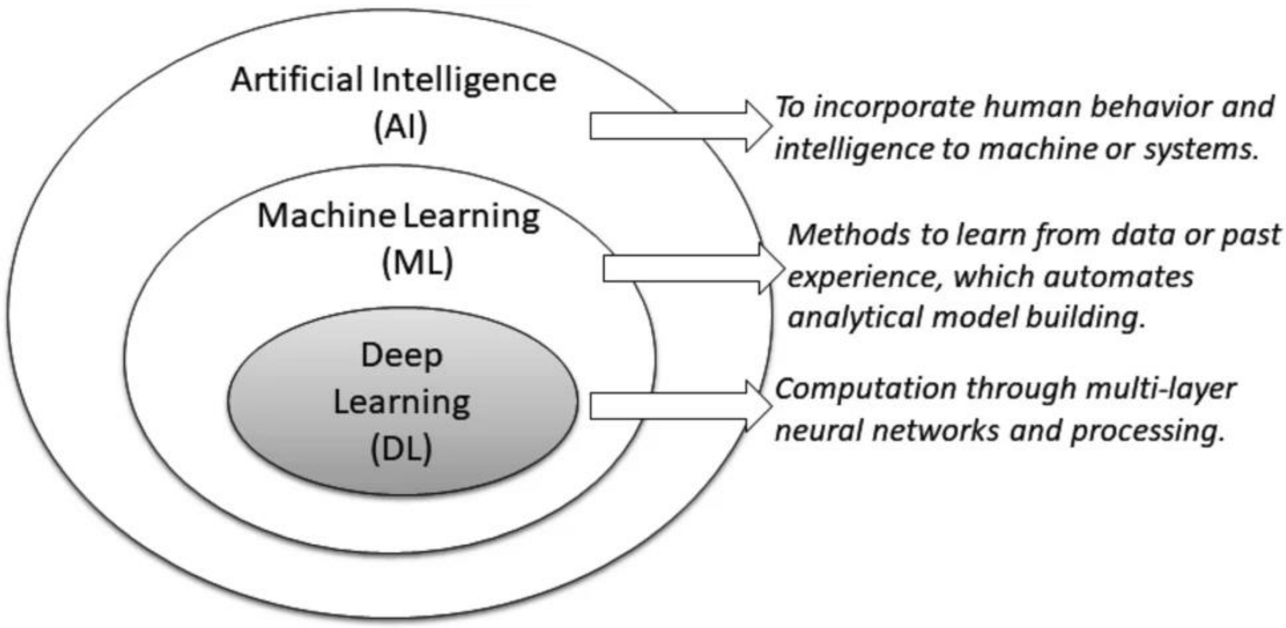


Figure 11. Relations of DL ML and AI [21].

2.2.2 Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM):

The RNN is a popular neural network applied in deep learning area. Unlike typical deep learning neural network, where the inputs and outputs are independent to each other, the connection of RNN is a circle, which means the output from one layer might affect the input of following layers as well as the former layers. To be more specific, the network will act as memorize the previous information and try to relate them with the following ones. In this case, the time-series or sequential data are especially suitable for RNN training as the network could find the time relevance included in the input data and build an algorithm according to that, thus increasing the training accuracy. However, typical RNN has the issue of vanishing gradients, and it will not be applied in this project, instead, the long sort-term memory network will be applied. As shown in Figure 12, this is a typical RNN structure, shown as a feedback neural network structure, compared with the normal feedforward structure, where the output will only be the inputs of following layers, the RNN output from hidden layers will be some of the inputs from back layers.

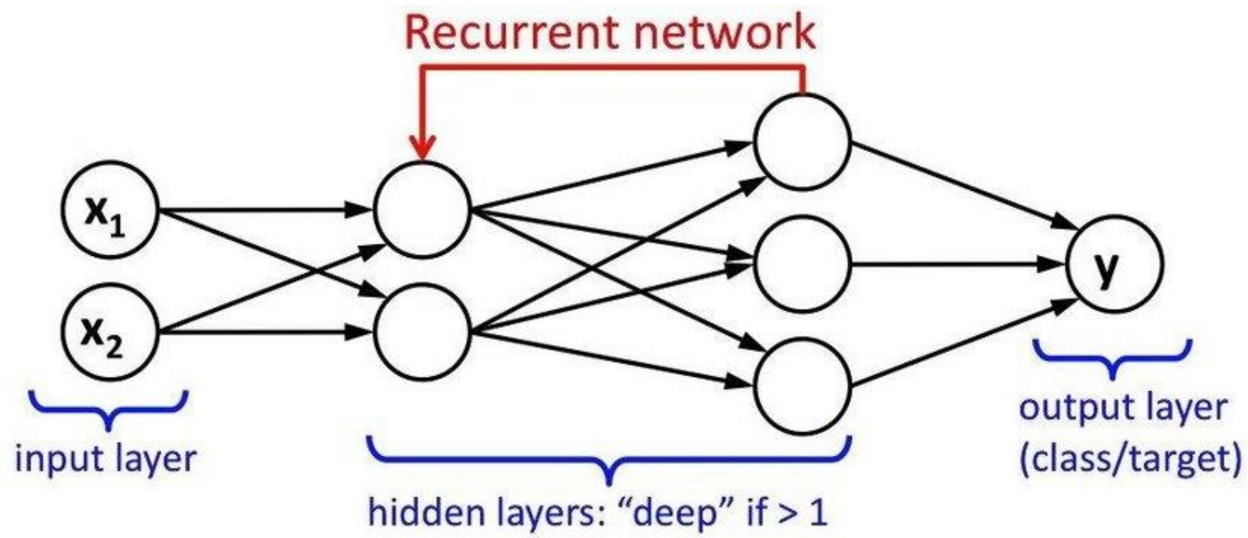


Figure 12. Typical structure of RNN [23].

LSTM is a popular kind of RNN and applies special units to deal with the vanishing gradient problem, which was introduced by Hochreiter et al [22]. If a neural network needs to store increasing information as the data size in increasing, graphically, it will explode. However, in LSTM, the unit cells with ‘Forget Gates’ are applied to ‘lose’ some irrelevant information to remain the stability of training. As LSTM solves the crucial problem in RNN, it is considered as one of the most successful RNNs.

2.2.3 Convolution Neural Network (CNN):

The Convolutional Neural Network is also a popular deep learning architecture that doesn’t need feature selection before training, thus making it to be a powerful neural network. Figure 13 shows a typical structure of CNN, containing convolution layers and pooling layers. To be noted, after each pooling, the network will create some feature maps and finally, the output will be derived through the output layer. Sometime, at specific nodes of neural network, the drop out is necessary to avoid the overfitting of training.

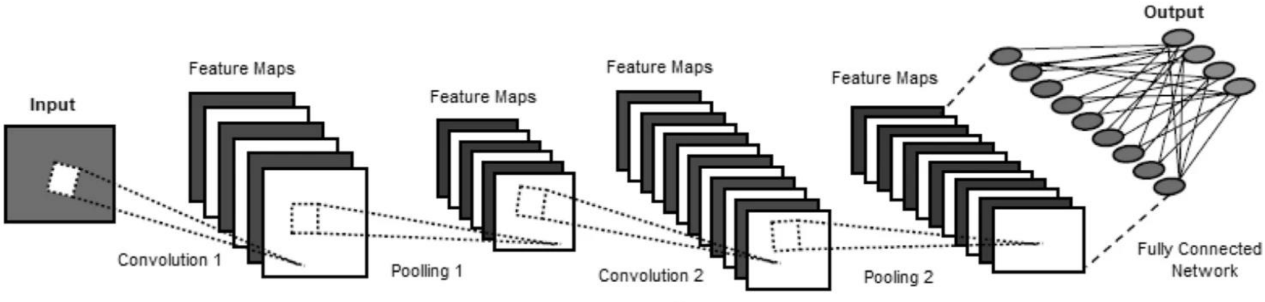


Figure 13. Typical structure of CNN [21].

Normally, CNN is designed for image, including 2D and 3D, it will consider the input 2D data as normally 3 dimensions including height, width, and depth. During the image processing, the initial image will be processed until the classification is finished. CNNs are specifically planned to deal with visual recognition, medical image analysis, natural language processing, and many more problems [24]. In this project, the CNN will be used for doing image classification as well as the sequence classification for initial data.

2.3 Training Environment:

2.3.1 Python:

There are several kinds of coding platform in the world. Some of them are born several years ago to proceed specific problems when machine learning earning techniques have not been created. With time passing, in the 1980s, Python was created by Guido van Rossum in the Netherlands as a kind of ABC programming language [25]. During several years, Python has experienced many big updates, and for now, Python is a high-level, user-friendly, general-purpose programming language.

In the 21st century, most of application in industry or in research area are based on artificial intelligence, such as machine learning or deep learning. In this case, users and developers of Python have created many useful libraries for machine learning. Some of the libraries are famous, one of the widely applied libraries is “pandas”, it uses simple and straightforward structures for implementing functions. Another useful library is “Keras”, which is commonly used for building Machine learning or Deep learning networks. This library is the best one for research and prototyping in deep learning, thus if Python will be applied in this project, “Keras” is a reasonable option.

2.3.2 MATLAB:

MATLAB is a programming language and numeric computing environment developed by MathWorks. Its main application is concentrated on matrix computation and plotting or representation of data. In the recent years, MATALB is commonly used for research purpose, especially those related to signal processing and machine learning. The user interface of MATLAB is very straightforward, along with the coding syntax, that's because unlike other programming languages, such as Python, C++, where almost everything needs coding and programming, while in the MATLAB, some of the pre-processing of signal can be achieved through specific commands and options, and as long as the data has been input, it will remain in the "workspace", allowing for any subsequent alternation.
MATLAB is one of the best tools for designing machine learning algorithms, compared with Python, machine learning in MATLAB is more original and limited defined, which means some of the neural network will be created by users themselves. When considering deep learning in MATLAB, "Deep learning Toolbox" will be applied in this project, to provide commands for creating neural network and training.

2.4 Noise simulation:

Noise is all around us in every kind of forms. In the real world, noise is commonly referred to sound. However, noise can be more accurately thought as random variations presenting in different variables, such as voltage, current, velocity. In this project, the actual representation of noise is the fluctuations in SAW velocity value.

2.4.1 Gaussian Noise:

Gaussian noise is a kind of noise, its probability density function (PDF) conforms to normal distribution. The probability density function p of a gaussian random variable can be represented as:

$$p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \qquad \textit{Eqn.4}$$

Where, μ is the mean value and σ is the standard derivation.

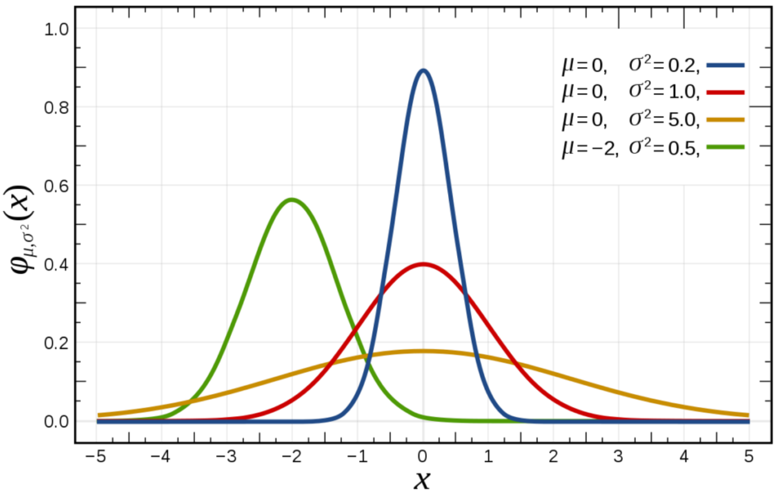


Figure 14. Gaussian Noise plot.

2.4.2 Gaussian White Noise

Gaussian is not included in Gaussian noise, instead, it includes gaussian noise. The ‘white’ means its power spectrum is flat across the entire sampling bandwidth. The ‘gaussian’ means its amplitude will be modelled with normal distribution. In MATLAB, gaussian white noise will be added through ‘AWGN’ function, which is additive white gaussian noise. This is a simple noise model to represent electron motion in the RF front end of a receiver.

Also, there are many other noises in the world. Shot noise will normally occurs when there is a potential barrier, electrons, and holes, the carriers flowing through the barrier will cause shot noise. Shot noise can be modelled as Poisson Distribution, for large numbers, the poison distribution can be considered as a normal distribution about its mean, so shot noise can be considered as Gaussian noise, thus gaussian white noise.

Thermal noise is a kind of electronic noise created by thermal agitation of electrons in a conductor, which is also called Johnson noise. In other words, current flowing through resistor will cause thermal noise. Thermal noise was first observed by John B. Johnson in 1926 and later interpreted by Harry Nyquist [26], hence thermal noise is also called Johnson noise, or Nyquist noise. Normally, thermal noise is often modelled as Gaussian white noise.

To conclude, almost all the noise mentioned above can be described as Gaussian white noise, so GWN will be the only noise source in this experiment.

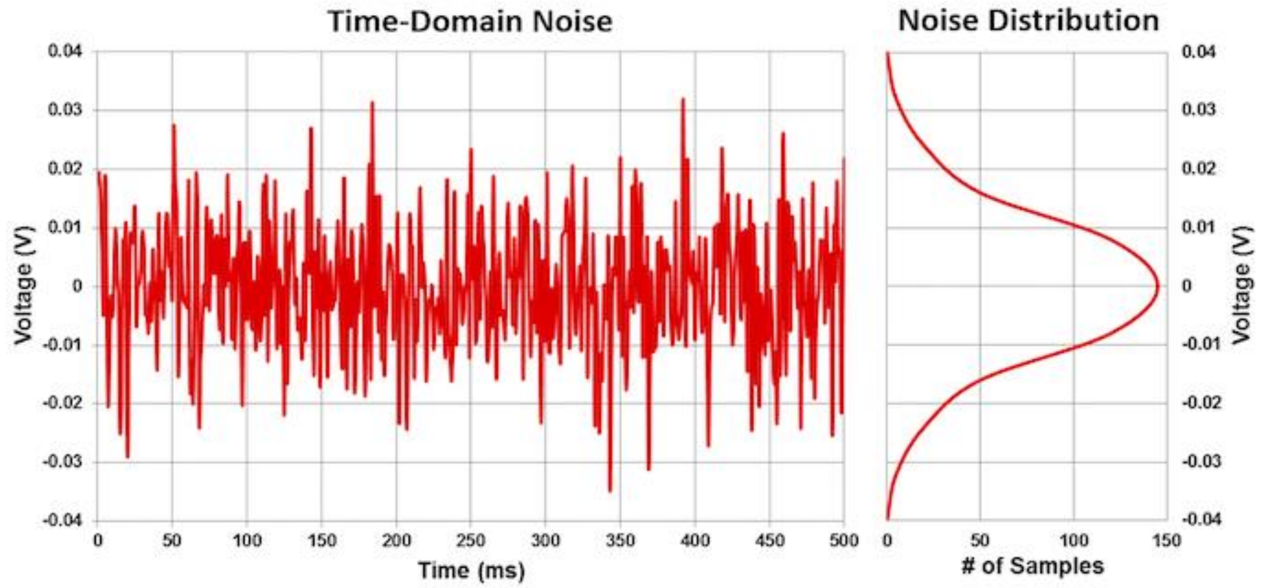


Figure 15. Gaussian white noise time domain signal and noise distribution.

3 METHODOGY:

3.1 Main Process:

3.1.1 ‘Image’ featured processing:

The first method applied to deal with SAW velocity is to represent the SAW in a polar plot, save them as images, and process images through imaged based machine learning.

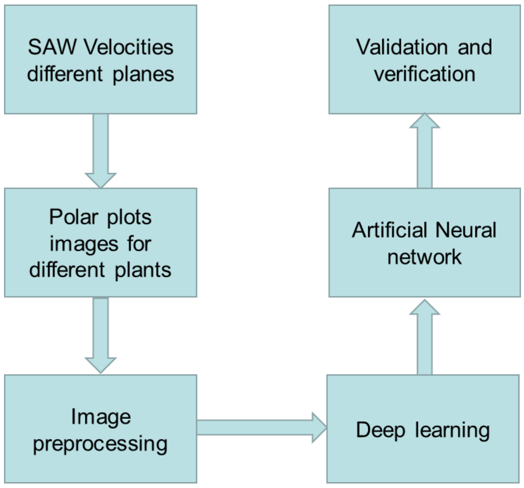


Figure 16. Image featured processing.

The main process of image-based SAW signal processing is shown as the above flow chart, the details will be explained in the following sections.

3.1.2 'Sequence' featured processing:

The second method applied to deal with the SAW velocity is to use the original value of SAW as the input of deep learning neural network and process the signal using sequence-based machine learning.

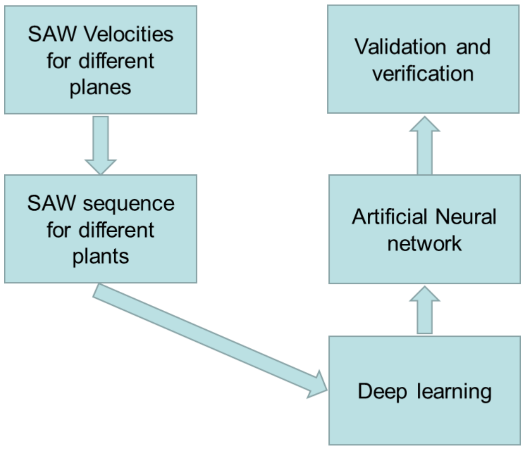


Figure 17. Sequence featured processing.

The main process of sequence-based SAW signal processing is shown as the above flow chart, the details will be explained in the following sections.

3.2 SAW Velocity Representation:

3.2.1 Polar Plot:

For the image based deep learning neural network, several image for polar plot should be generated and saved by MATLAB for future pre-processing. To be specific, given the total number of planes is 441, in this case, for each plane (surface), the SAW values were measured along different directions in a circular format. The measurement resolution for degree is 1 and range from 0 degree to 360 degree. Some examples polar plots are shown as below.

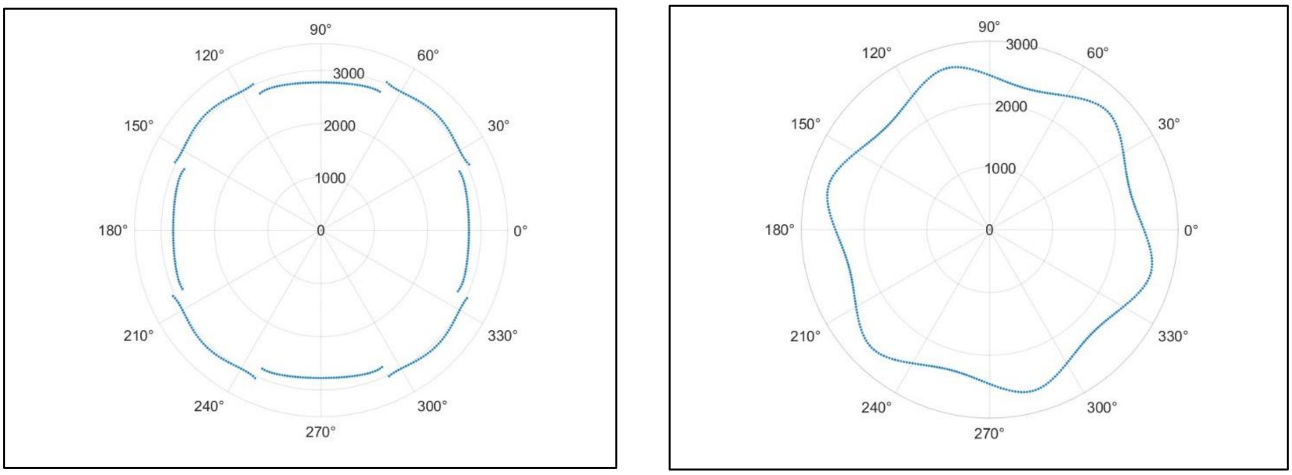


Figure 18. Example polar plots for miller index (0,0,1) and (1,1,1).

From the above Figure, it can be observed that for different miller index, which is different surfaces, the measuring results for SAW velocity are different. However, the axes on the images will affect the image classification process and make the training process to be slower. In this case, the axes should be removed, the new images are shown as follows,

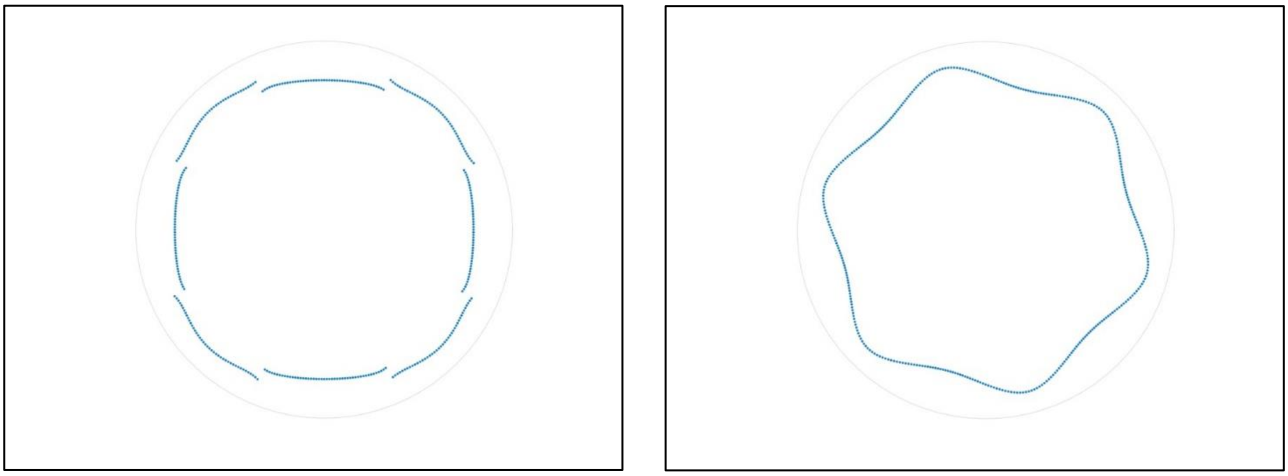


Figure 19. Example polar plots for miller index (0,0,1) and (1,1,1) without axes and labels.

By this step, the SAW polar images are ready to be pre-processed.

3.2.2 1 Dimension Numerical Sequence:

For the sequence-based deep learning, several SAW sequences for different planes should be created as the input of neural network. To be specific, given the total number of SAW surfaces is 441, in this case, for each plane (surface), SAW will be measured based on the circular route, with the resolution of 1 degree, range from 1 to 360.

Table 1. Standard SAW velocities for plane (0,0,1) with 180 directions.

Degree	**1°**	**2°**	**3°**	**4°**	**5°**	**6°**	**7°**	**8°**	**9°**	**10°**
SAW	2772.456	2774.663	2772.603	2774.253	2776.272	2777.828	2779.615	2782.755	2785.83	2788.327
Degree	**11°**	**12°**	**13°**	**14°**	**15°**	**16°**	**17°**	**18°**	**19°**	**20°**
SAW	2793.695	2795.033	2801.587	2803.936	2810.946	2813.922	2818.972	2822.944	2826.897	2831.252
…………										
Degree	**161°**	**162°**	**163°**	**164°**	**165°**	**166°**	**167°**	**168°**	**169°**	**170°**
SAW	2830.159	2830.853	2826.514	2822.122	2819.025	2812.121	2808.768	2805.411	2800.674	2798.259
Degree	**171°**	**172°**	**173°**	**174°**	**175°**	**176°**	**177°**	**178°**	**179°**	**180°**
SAW	2792.856	2788.954	2786.011	2783.721	2781.661	2777.514	2778.61	2775.934	2775.418	2772.849

The above table shows the standard SAW values for plane (0,0,1), obtained from the SAW database of nickel. It can be observed that for each degree, there is a corresponding SAW value. Given that the degrees number is 360, 441 1-D sequences with length of 360 should be corresponding to 441 planes, however, as the SAW velocity is symmetry with 180°, so only 180 values are necessary for characterization.

3.3 Noise Simulation:

3.3.1 Gaussian Noise in MATLAB:

As the Gaussian Noise will be introduced to the SAW signal while training, it is necessary to find how to add Gaussian Noise to SAW.

The function 'randn' is used for generating the normally distributed matrix. For example, if a set of Gaussian noise with mean = u, and variance = v, will be generated, the syntax should be like this:

$$noise = \sqrt{v} \times randn(180,1) + u \qquad Eqn.5$$

The sequence signal after noise should be:

$$noisy\ signal = noise + SAW\ sequence(180,1) \qquad Eqn.6$$

3.3.2 Gaussian White Noise in MATLAB:

Compared with Gaussian Noise, Gaussian White Noise is the specific one with mean = 0, so in this case, one method to model Gaussian White noise is using *Eqn.5*, so

$$noise = \sqrt{v} \times randn(180,1) + 0 \qquad Eqn.7$$

In this case, there is only one variable to modify the noise level: variance. In Gaussian White Noise, the variance of noise is equal to the power of noisy in the following format.

$$Variance = v, Power = 10log10(v)$$

Another common strategy used for adding Gaussian White Noise is using 'AWGN' function:

$$noisy\ signal = awgn(SAWsequence(180,1), SNR, 'measured', seed)\ Eqn.8$$

In this case, the Gaussian white noise with specific SNR will be added to the signal.

3.3.2.1 SNR

SNR is short for signal to noise ratio; this is a factor to reflect the noise level with respect to the signal power. The 'AWGN' function will measure the signal power based on the following equation:

$$Signal\ Power = sum\left(\frac{SAWsequence(180{,}1)\times SAWsequence(180{,}1)}{length(SAWsequence(180{,}1))}\right) \qquad Eqn.9$$

Following the similar format, the power of noise should be:

$$Noise\ Power = sum\left(\frac{(noisy\ signal - SAWsequence(180{,}1))^2}{length(noisy\ signal)}\right) \qquad Eqn.10$$

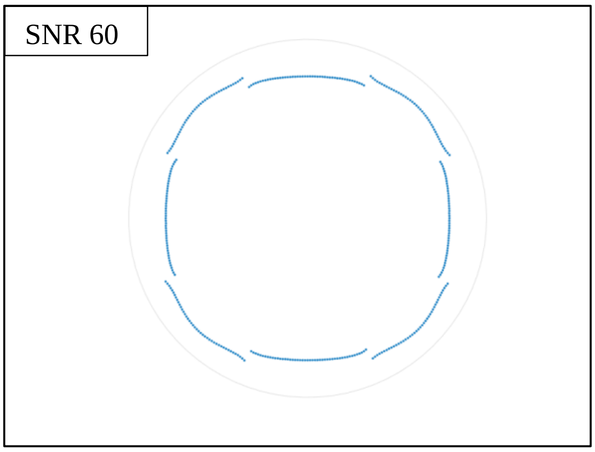
The signal to noise ration should be:

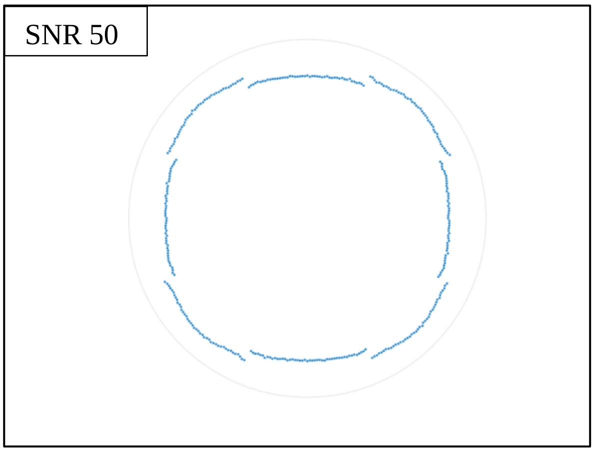
$$SNR = 10 \times \log\left(\frac{Signal\ Power}{Noise\ Power}\right)\ dB \qquad Eqn.11$$

The SNR value in this project will be set from 10 to 70, with the increasement of 10 dB, to simulate the noise in the real world. The typical SNR value in SAW measurement is 30, so some training samples with lower SNR will be necessary for increase the robustness of system, and when the SNR increases to 70, the noisy signal is equal to the original signal.

3.3.2.2 Polar Plot After Noise:

With different noise added to the signal for different SNR, the polar plot for SAW velocities are different.





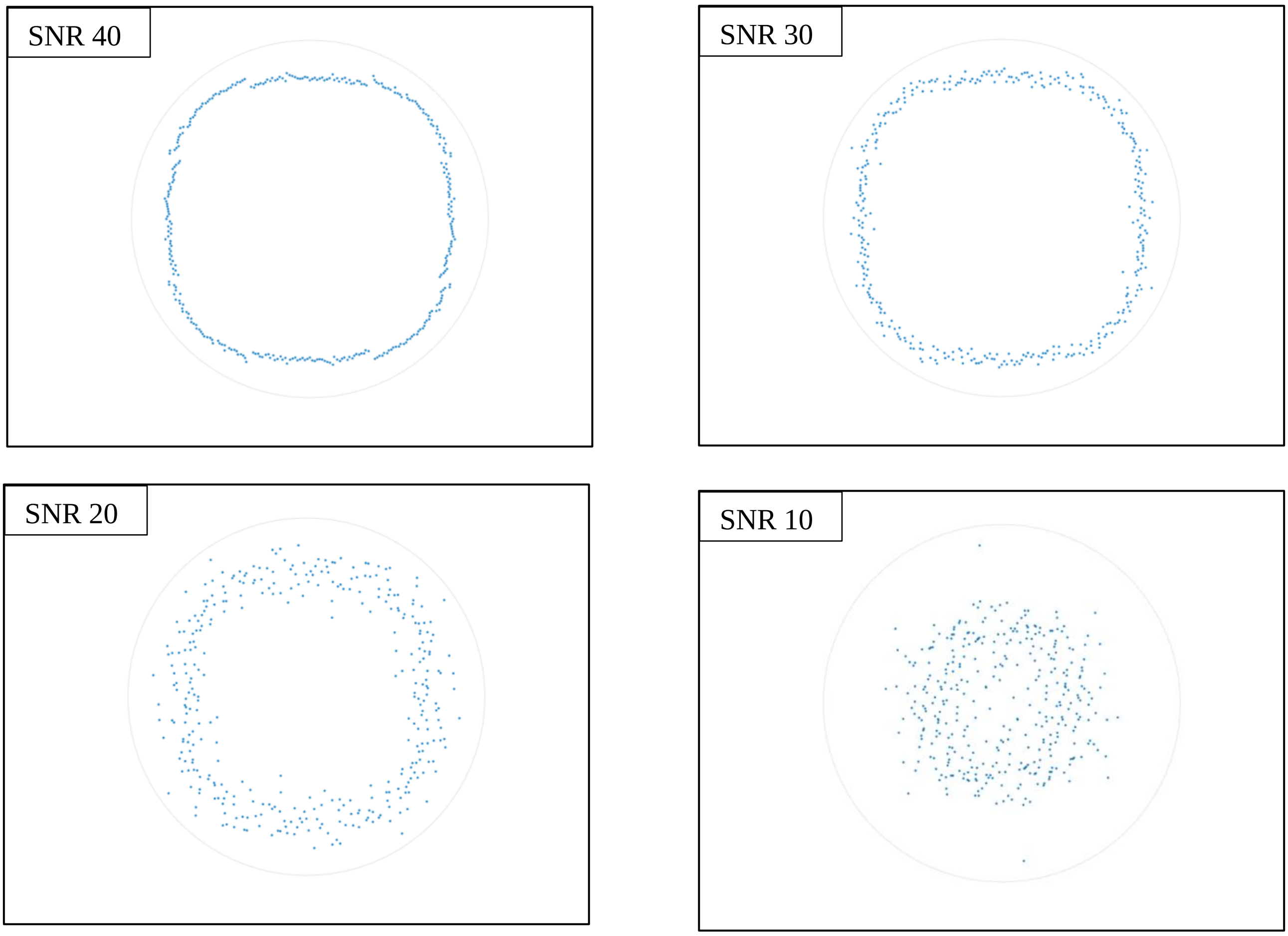


Figure 20. Noisy SAW polar plot for different SNRs for plane (0,0,1).

The above figures show the noisy SAW polar plot for different SNR, for convenience, the realization of adding noise is achieved by a self-defined external function,

$$Noisy\ image = Noise\_generation\_image(SNR, sample\ number, SAW\ sequence)$$

When the SNR is defined, the sample number means how many sets of images are produced. For example, if the sample number is 100, so there are totally 44100 number of images generated, for plane (0,0,1), there are 100 different patterns.

3.3.2.3 1 Dimension Numerical Sequence After Noise:

Similar to adding noise to SAW image, for convenience, the realization of adding noise to SAW sequence is achieved by a self-defined external function,

$$Noisy\ signal = noise\_to\_signal(SNR, sample\ number, SAW\ sequence)$$

When the SNR is defined, the sample number means how many sets of noisy sequences are produced. For example, if the sample number is 100, so there are totally 44100 number of sequences are generated, for plane (0,0,1), there are 100 different noisy sequence.

3.4 Image Pre-processing:

3.4.1 Image Cropping and Resizing:

For the image based neural network, after the original image for polar plot is obtained, to decrease the training time for the neural network, the image pre-processing is necessary.

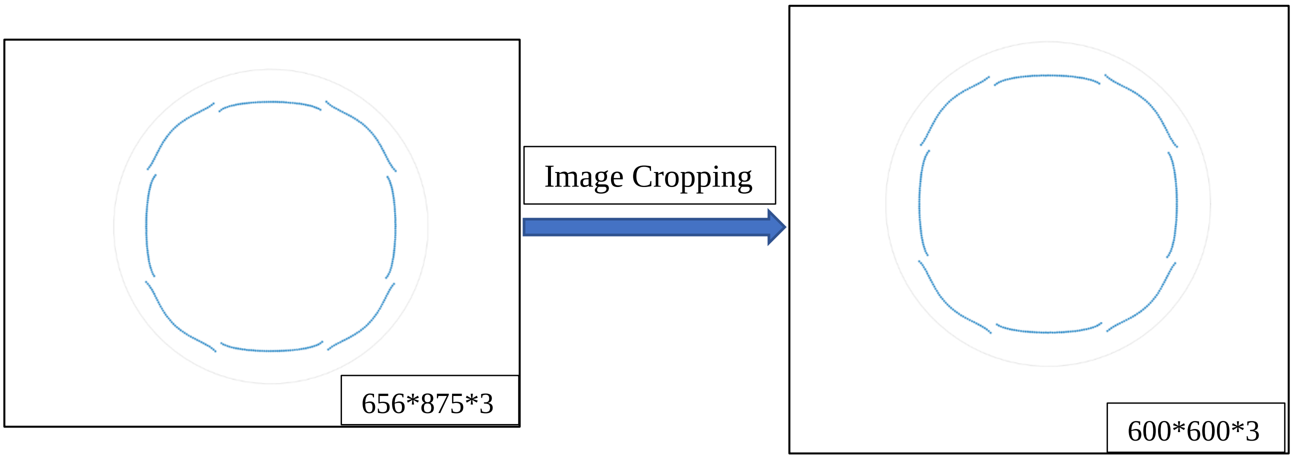


Figure 21. Image cropping process.

As indicated as the above process, the original dimension of SAW polar plot image is 656*875*3, 656 represents the height of image, and 875 represents the length of image, 3 means there are 3 channels for the RGB image. However, as can be observed, the large empty space in the figure will cause the unnecessary slowness in image classification process, so after image cropping, the new dimension is 600*600*3, remaining the essential polar plot part.

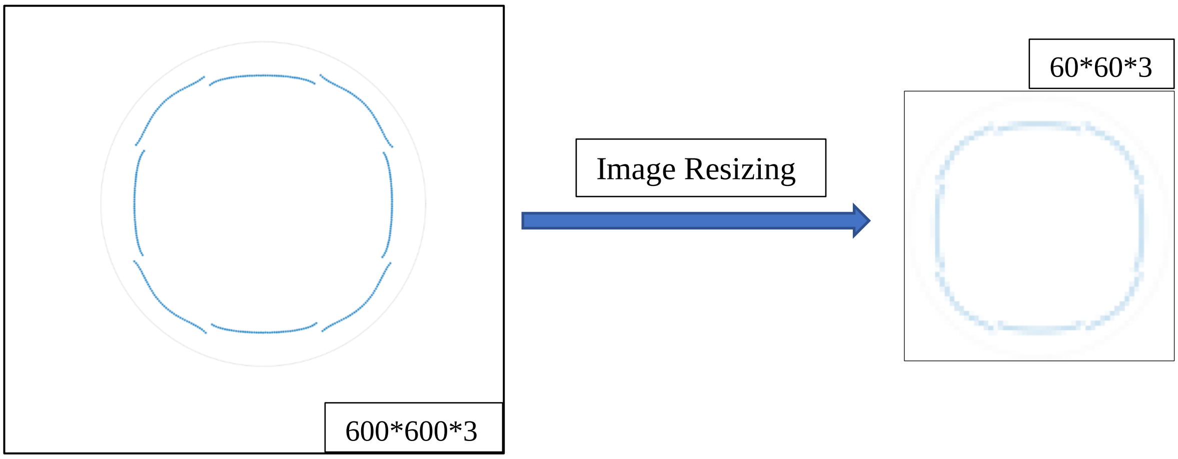


Figure 22. Image resizing process.

However, in the real practice for training, image dimension of 600*600*3 is still too large to proceed, so the image will be resized to be with dimension of 60*60*3 shown as above. This process will decrease the features for image classification but will increase the training speed in a large scale.

3.4.2 Image to Grey:

According to the observation in polar plot image, although the images are 3-channel RGB image, the useful features are only those dots located on the white background, so actually, if the neural network can recognize and categorize between background and dots, the learning can be achieved. In this case, 3 channels are not necessary, only one channel is necessary to reflect different grey values of each pixel.

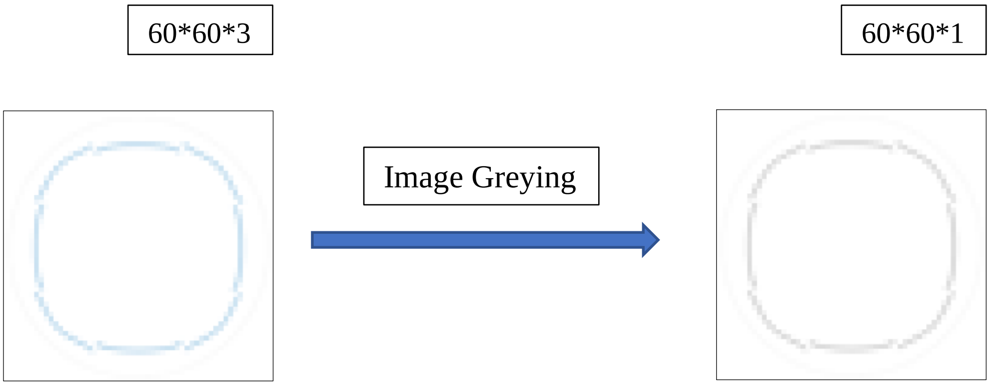


Figure 23. Image greying process.

After changing the image to grey image, the dimension of image also decreases, now, for each of the polar plot image, a matrix with only 60 columns and 60 rows can represent all the information we need.

3.5 Convolution Neural Network:

3.5.1 2-D convolution (image classification):

The neural network applied to image classification is a 6-layer convolution neural network. 6 means there are totally 6 convolution layers.

3.5.1.1 Network Structure:

Figure 24. 2-D image classification neural network structure.

The neural network structure for image classification is shown above.

Image Input Layers: The input of the neural network is 60*60*1, this layer will accept the input matrix with this dimension and apply data normalization.

Convolution2dLayer: This layer applies sliding convolutional filters to 2-D image. It will convolve the input by moving the filters along the image vertically and horizontally. The result of convolution consists of the product of weights and addition of bias [27].

Different convolution layers in the image classification neural network use difference filter size, and number of filters. For the first three layers, the filter size is always 5*5, with the number of filters increasing from 20 to 80. The resting three layers are all with filter size 3*3, with the number of filters increasing from 20 to 80. The decrease in filter size is the strategy for more careful classification when the neural network is approaching the end.

BatchNormalizationLayer: This layer is located between the convolution layer and nonlinearities, such as the activation function layers, it will normalize a mini batch of data across all observations for each channel independently [28]. The function of this layer is for speeding up the training of CNN.

ReluLayer: This layer will perform as a threshold to each input, any value less than zero will be set to zero. The equation is shown as:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad \textit{Eqn.12}$$

maxPooling2dLayer: It performs down sampling by dividing the input into rectangular polling regions and calculate the maximum of each region. For the image input, the polling will compress over the spatial dimensions. To be noted, this layer has no variables, it just applies pooling core to compress the data. The pooling size for above network is [2 2], with the stride of [2 2], this is based on a usual setting of image classification neural network for faster training.

DropoutLayer: This avoids the overfitting of neural network while training, by randomly setting half of the input to zeros.

FullyConnectedLayer: This layer is normally in the nearly end of the network, it related to how many classifications necessary in this task, we need 441 different planes, but another buffer is necessary. So, the parameter of this layer should be 442 to represent the number of the data set.

ClassificationLayer: This is the final layer for classification task of CNN, it is an essential layer for generating the loss and accuracy for each training epochs.

3.5.1.2 Training Strategy and Options:

For the image classification neural network, the input should be the pre-processed SAW images as mentioned above, the strategy is based on adding different noise to the training data to increase the robustness and noise resistivity of neural network. The constitutions of noisy data are shown in the below table 2.

Table 2. Noisy data with different SNRs, and training iteration for each SNRs

SNR	Number of samples	Training Iteration
10	5	20
20	5	20
30	5	40
40	5	20
50	5	20
60	5	20
70	5	20

As indicated in the table above, for the training of image classification neural network, different noise with different SNR was applied, for each of the SNR noise, there are 5 different samples, means there are 5 different patterns for that SNR. However, the number of 5 is far from enough for neural network training with different patterns of noise, but if the number of increases, the training will exceed the maximum RAM of the PC, so the training iteration is set to be 20, which means for each iteration the neural network is training with a new set of data and generate a new network. After all the iteration, there will be totally 20 networks, and the last one will be the best solution as it accepts all the noise patterns for training.

The options for each complete training of neural network are set as follows,

Table 3. Training options for sequence classification neural network.

Training Options	**Options choice**
Solver Name	Adam
Max Epochs	150
Mini Batch size	64
Shuffle	Every epoch
Initial Learning Rate	0.0001(default 0.001)
L2 regularization	0.01(default 0.0001)
Output Network	Best Validation Loss

3.5.1.3 Validation Strategy:

To monitor the neural network during training, also to observing whether the neural network is overfitting, the validation is necessary in the training process. Validation is realized by separating a subset of noisy data to test the neural network while training, this subset should not be included in the training data but needs to be generated following the same patterns.

To be specific, the noisy data shown in Table 2 is not all applied to training. After noise modelling, the noisy data will be randomized, and separated as training data and validation data. If the SNR ranges from 10 to 70, and each with 5 samples, that means there are totally 7*5 = 35 noisy samples, each sample has 441 noisy images, thus totally 15435 noisy images. The chosen validation images number for this project is 441, so the training images number isa 14991. The ratio between training and validation data is 34:1.

After training, it is necessary to test the network performance using new generated noisy data, the network will be tested using different SNR noise, the results will be posted in the following sections.

3.5.1.4 Verification Strategy:

Verification is different from Validation; Validation means the network is examined with some data that is similar with training data, for example, in this project, as mentioned above, validation means network will be examined with noisy data for specific SNRs generated by MATLAB, even the noise has new pattern, and the network has not seen before. However, verification means the network will be verified in the real practice, in this project, the network will be tested with real measured data of SAW and compare the result with real value.

3.5.2 1-D convolution (sequence classification):

The neural network applied to sequence classification is a 4-layer convolution neural network. 4 means there are totally 4 convolution layers.

3.5.2.1 Network Structure:

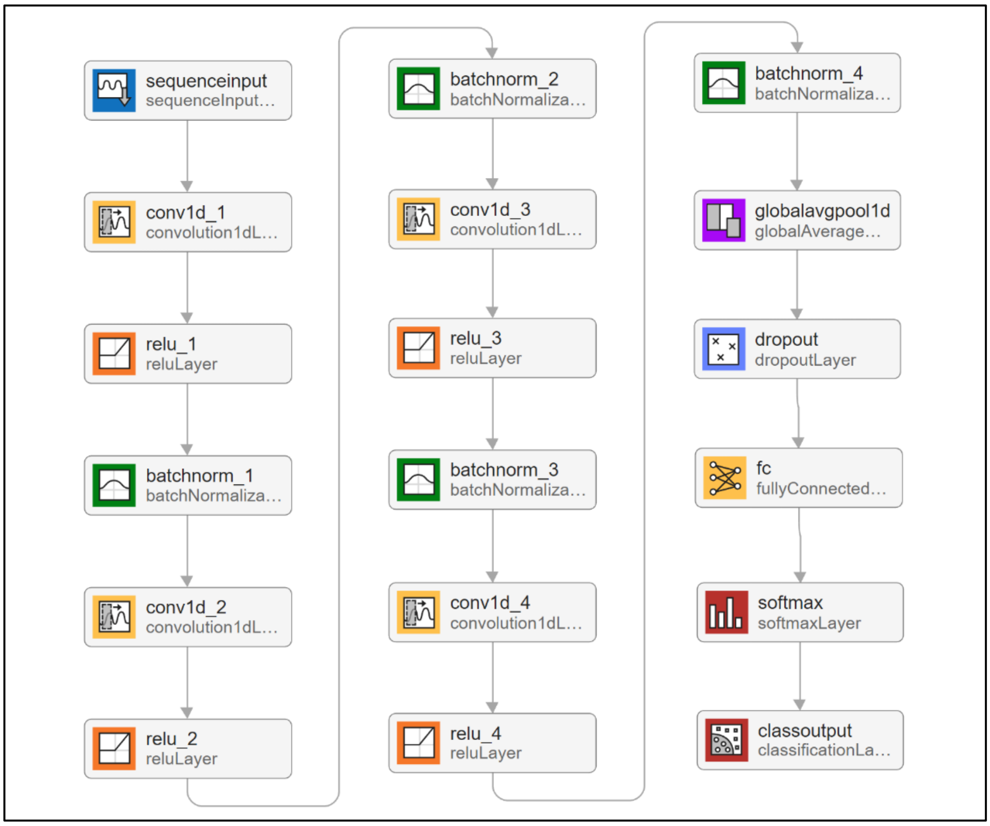


Figure 25. 1-D sequence classification neural network structure.

The figure above shows the structure of sequence classification neural network.

Sequence Input Layers: The input of the neural network is 180*1, this layer will accept the input matrix with this dimension and apply data normalization.

Convolution1dLayer: This layer applies sliding convolutional filters to 1-D sequence. It will convolve the input by moving the filters along the image vertically and horizontally. The result of convolution consists of the product of weights and addition of bias.

Different convolution layers in the sequence classification neural network use difference filter size, and number of filters. For the first three layers, the filter size is always 3, with the number of filters increasing from 32 to 128. The last layer is with filter size 6, with the number of filters equal to 256.

BatchNormalizationLayer: This layer is located between the convolution layer and nonlinearities, such as the activation function layers, it will normalize a mini batch of data across all observations for each channel independently [28]. The function of this layer is for speeding up the training of CNN.

ReluLayer: This layer will perform as a threshold to each input, any value less than zero will be set to zero. The equation is shown as:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \qquad \textit{Eqn.12}$$

globalAveragePooling1dLayer: This layer performs down sampling by outputting the average of spatial dimension of input. For this project, it pools over dimension 180 and speeding the training of neural network.

DropoutLayer: This avoids the overfitting of neural network while training, by randomly setting half of the input to zeros.

FullyConnectedLayer: This layer is normally in the nearly end of the network, it related to how many classifications necessary in this task, we need 441 different planes, but another buffer is necessary. So, the parameter of this layer should be 442 to represent the number of the data set.

ClassificationLayer: This is the final layer for classification task of CNN, it is an essential layer for generating the loss and accuracy for each training epochs.

3.5.2.2 Training Strategy and Options:

For the sequence classification neural network, the input should be the different SAW sequences, the strategy is based on adding different noise to the training data to increase the robustness and noise resistivity of neural network. The constitutions of noisy data are shown in the below table.

Table 4. Noisy data for different SNRs and training iteration for each SNRs

SNR	Number of samples	Training Iteration
10	100	20
20	100	20
30	100	20
40	100	20
50	100	20
60	100	20
70	100	20

As indicated in the table above, for the training of sequence classification neural network, different noise with different SNR was applied, for each of the SNR noise, there are 100 different samples, means there are 100 different patterns for that SNR. However, the number of 100 is far from enough network training to accept different patterns of noise, but if this number of increases, the training will exceed the maximum RAM of the PC, so the training iteration is set to be 20, which means for each iteration the neural network is training with a new set of data and generate a new network. After all the iteration, there will be totally 20 networks, and the last one will probably be the best solution as it accepts all the noise patterns for training.

The options for each complete training of neural network are set as follows,

Table 5. Training options for image classification neural network.

Training Options	**Options choice**
Solver Name	Adam

Max Epochs	150
Mini Batch size	128
Shuffle	Every epoch
Initial Learning Rate	default 0.001
L2 regularization	default 0.0001
Output Network	Best Validation Loss

3.5.2.3 Validation Strategy:

To monitor the neural network during training, also to observing whether the neural network is overfitting or underfitting, the validation is necessary in the training process. Validation is realized by separating a subset of noisy data to test the neural network while training, this subset should not be included in the training data but needs to be generated following the same patterns.

To be specific, the noisy data shown in Table 4 is not all applied to training. After noise modelling, the noisy data will be randomized, and separated as training data and validation data. If the SNR ranges from 10 to 70, and each with 100 samples, that means there are totally 7*100 = 700 noisy samples, each sample has 441 noisy SAW sequences, thus totally 308700 noisy sequences. The chosen validation images number for this project is 441, so the training sequences number is 308259. The ratio between training and validation data is 699:1.

After training, it is necessary to test the network performance using new generated noisy data, the network will be tested using different SNR noise, the results will be posted in the following sections.

3.5.2.4 Verification Strategy:

Verification is different from Validation; Validation means the network is examined with some data that is similar with training data, for example, in this project, as mentioned above, validation means network will be examined with noisy data for specific SNRs generated by MATLAB, even the noise has new pattern, and the network has not seen before. However, verification means the network will be verified in the real practice, in this project, the network will be tested with real measured data of SAW and compare the result with real value.

4 RESULTS AND DISCUSSION:

4.1 Image Classification:

In this section, the image classification neural network training result and testing result will be presented and analysed.

4.1.1 Training Results and analysis:

The figures below show the process of training image classification neural network, they are generated by MATLAB to observe the training accuracy, validation accuracy, and loss during network learning. The training is based on SAW database values of 360 directions. The training strategy and options are already declared in the section 3.5.1.2.

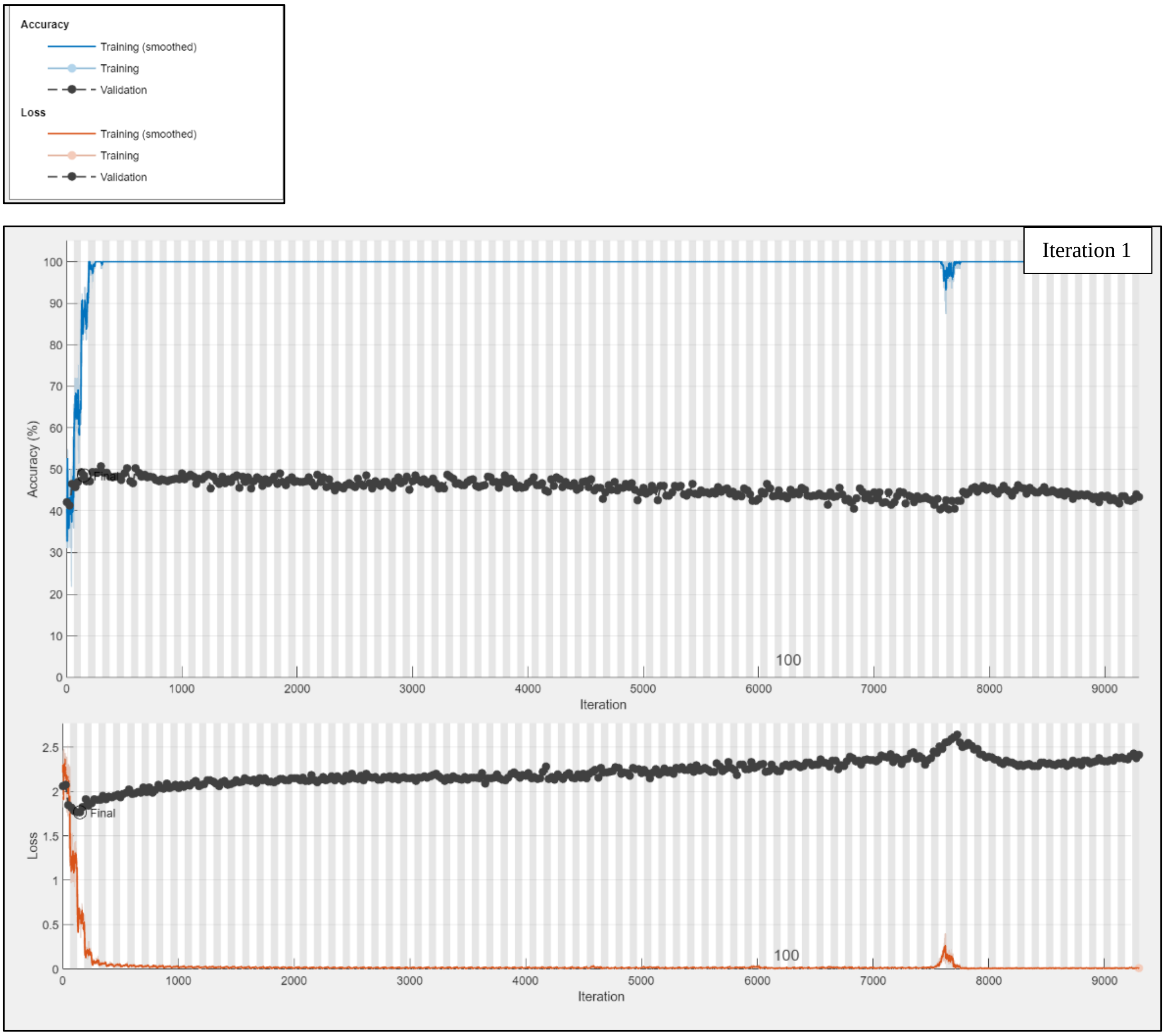


Figure 26. Training progress of iteration 1.

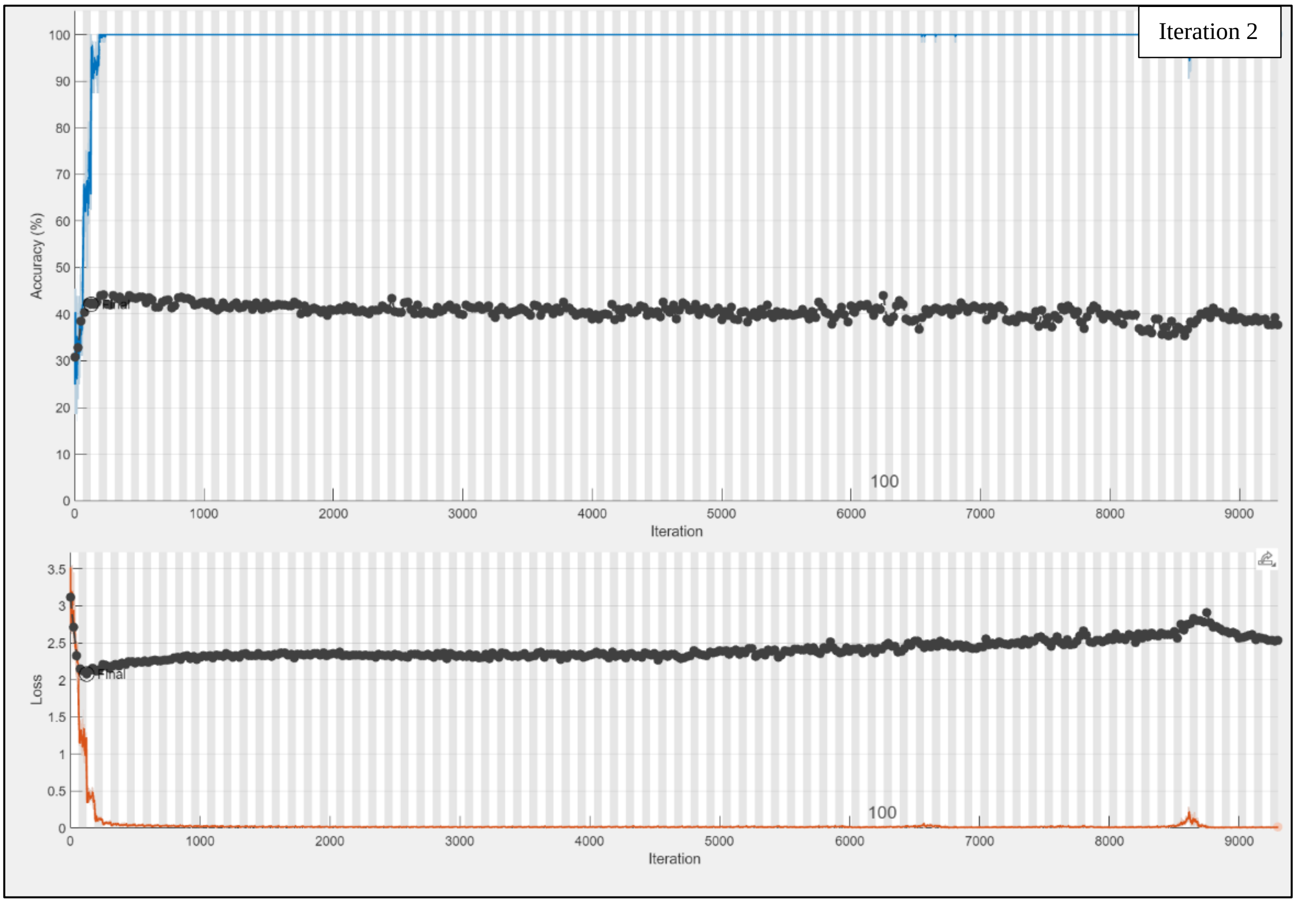


Figure 27. Training progress of iteration 2.

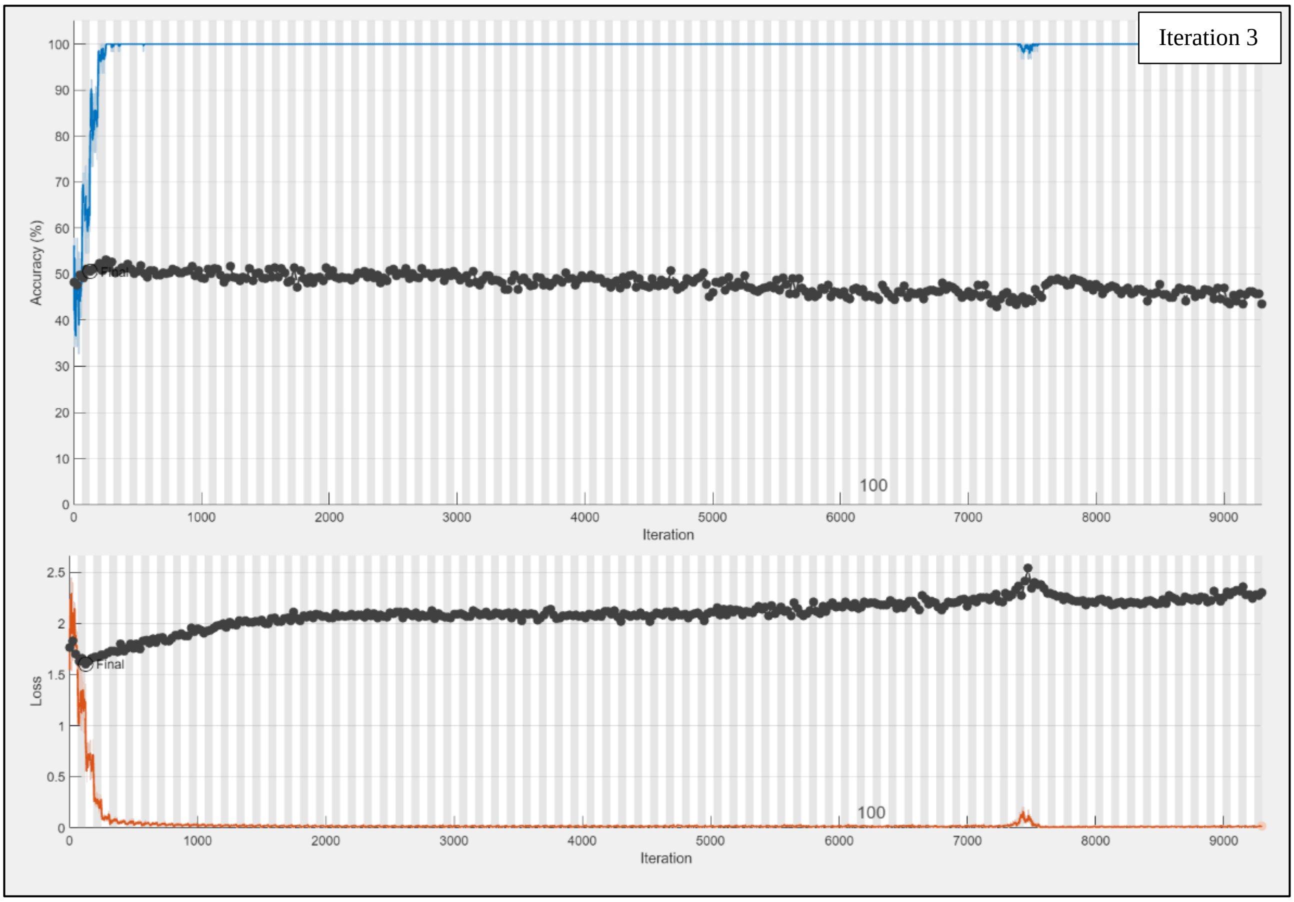


Figure 28. Training progress of iteration 3.

It can be observed that in the above figures,

1) The training result of neural network will not change if the epochs value is large enough, obviously, 150 is an enough value for each training. Probably just several iterations less than 20 are necessary to observe and analyse the training result. For example, the images above show only 3 iterations, it can be observed that, from the second iteration, even the neural network is feed with new patterns of noisy data, the validation accuracy will not have a significant increase.
2) This is an overfitting training. When the iterations increase to about 5000 in the first iteration, the validation accuracy will no longer increase, even the training is continuing. However, for the training accuracy, it will increase continuously during training, it can be observed that during the early stage, it reached almost 100% and will not change until finish. This means the neural network is memorizing the training data, instead of learning and categorizing their features as we desire.
3) When the epochs reach 150, the training will finish, and the output network will be the epoch with the minimum validation loss, at that point, the validation accuracy of network is 42.18% for 1st iteration, 48.53% for 2nd iteration, 50.57% for 3rd iteration, the network from last iteration will be used.
4) The stability and non-increasing of validation accuracy during training is probably caused by the specific structure of neural network, which means the 6-layer convolution neural network reaches its maximum to perform the image classification task, to increase the validation accuracy, more complex neural network structure is necessary.

4.1.2 Validation Results and analysis:

Validation of this neural network is achieved by testing it with new-generated noisy data from MATLAB, the validation is also based on different SNRs.

Table 6. Validation accuracy for different SNR inputs and processing time.

SNR	Accuracy (%)	Processing time (neural network prediction time)
10	0.45	0.42s
20	5.22	0.54s
30	56.10	0.54s
40	80.50	0.42s
50	96.37	0.35s
60	99.55	0.37s
70	99.55	0.41s

It can be observed from above table that for noise level with SNR = 70 or 60, the prediction accuracy for image classification neural network is almost 100%, that means when without noise or with little noise, the accuracy is very high, so in the specific circumstance, if there is no noise, the neural network must be the best approach to decide the crystal orientation as the time to proceed is real short compared with using Brute Force, which might consume several days. In this case, the neural network can be considered as a convenient database to search for answers.

When the SNR decreases gradually, the accuracy is observed to have a sharp drop at SNR = 30, with only about 56.10%, SNR = 20 with 5.22%, SNR = 10 with 0.45%, which means the neural network can deal with specific noise for SNR=30 as the accuracy in this case is just above 50%, the results can not be trusted by a 0.5 possibility.

However, these accuracies are measured by a very strict regulation, if the prediction is not the label desired, its accuracy is 0. Now, a new method to measure accuracy using R-value is applied.

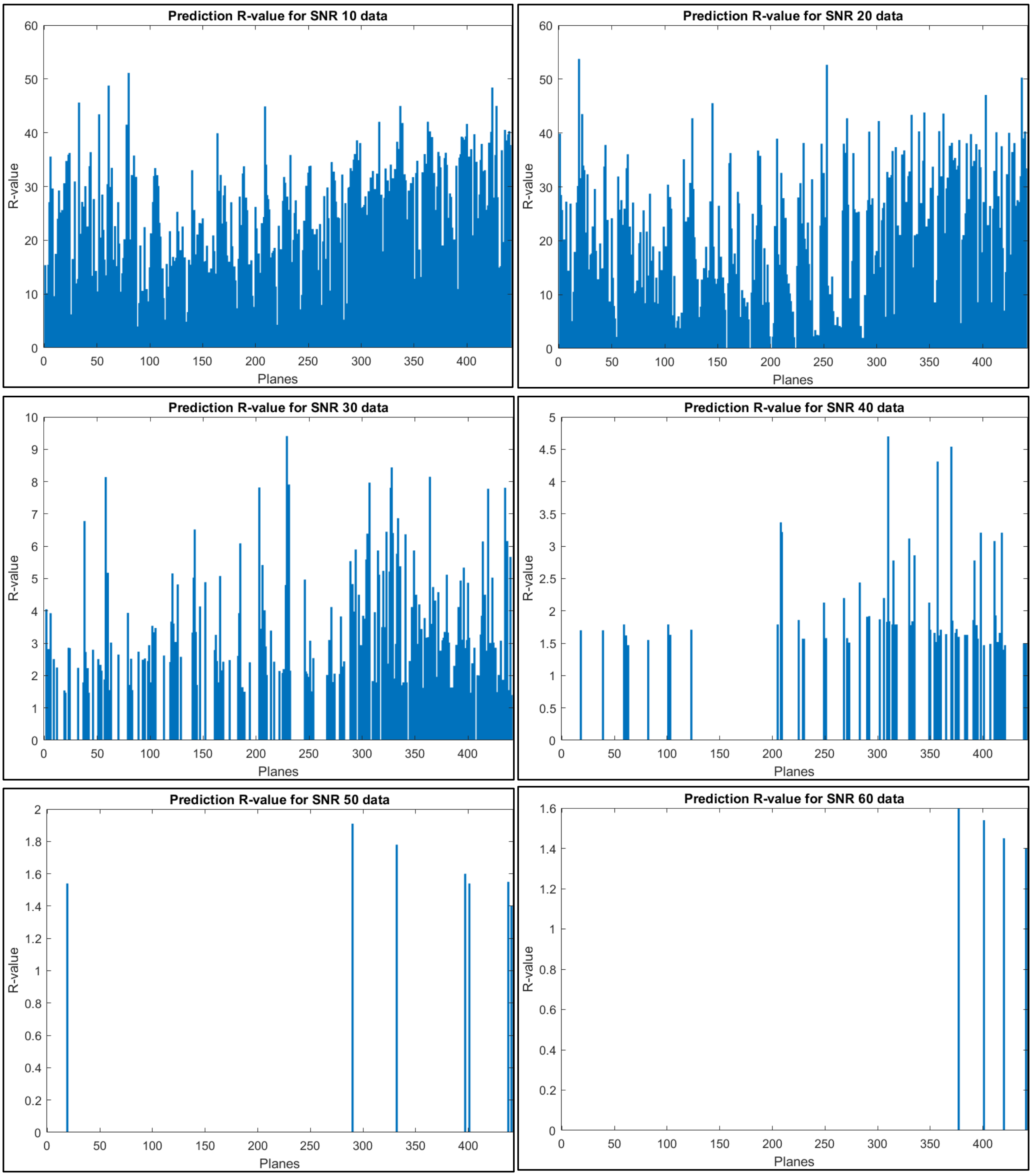


Figure 29. R-values bar chart for predictions at different SNRs.

The bar charts above show the specific R values when different SNR noise are applied to image classification neural network. It can be observed that when the SNR value decreases, for each specific plane, the prediction R-value will increase, this is because the increasing of noise will cause the wrong classification. Also, it can be observed that for SNR50 and SNR60, there are many planes with zero R-value, which means the prediction is the same as label.

When using the R value algorithm to measure the prediction accuracy of the neural network, if R value is smaller than 8°, the prediction is considered within an industrial acceptable standard. So, in table 7, the value of 8 for R value is set for threshold to predict if the misorientation is acceptable, the accuracy is calculated by how many of planes are below 8°.

Table 7. Validation accuracy for different SNRs and number of planes within acceptable standard (totally 441 planes)

SNR	Number of planes R values < 8 degree	Accuracy (%)
10	60	13.6
20	120	27.2
30	437	99.1
40	441	100
50	441	100
60	441	100
70	441	100

It can be observed that when using the new method to decide the accuracy, the accuracy for SNR = 30 to 70 is almost 100%, because almost all the predicted planes have less than 8° R-value with the labels. However, for SNR10 and SNR20, even their accuracy increases in lot compared with previous approach (to 13.6% and 27.2% alternatively), these results are still not satisfactory to some extent as they are far away from 50%. This is probably because when the SNR is too small, the shape of the polar plot changes to an unrecognized structure, far away from its original shape, thus the proportion for noise is too large for the featured signals.

Anyway, by using the R value assessment approach to evaluate the prediction accuracy, the overall accuracy with noise has increased a lot, especially for SNR=30, from previous 56.1% to current 99.1%, this means the neural network might be potentially useful in the real experiment as the typical SNR for real noise is approximate 30.

4.1.3 Verification Results and analysis:

The verification of neural network is based on the real data testing from experiments, instead of the simulated noise data in MATLAB. In the training and simulation stages, the SAW velocities are presented with 360 directions, from 0° to 360°. This means there are maximum 360 features in each plane to classify it from others. However, in the real experimental data, SAW velocities are only measured from 73 directions, also from 0° to 360° but with increasement of 5°. In this case, only 73 features are available for trained neural network to classify the planes, the results for verification are shown in the table.

Table 8. Predication result and R-value for real data with 73 directions

Plane	Prediction	R value
(0.1,0,1)	(0.1,0.05,1)	2.85
(0.35,0.2,1)	(0.05,0.4,1)	19.53
(0.2,0.2,1)	(0.05,0.2,1)	8.33
(0.3,0.15,1)	(0,0.9,1)	37.33
(0.35,0.05,1)	(0.5,0,1)	7.78
(0.1,0,1)	(0,0.05,1)	6.39
(0.3,0.2,1)	(0,0.05,1)	18.39
(0.05,0.1,1)	(0,0.5,1)	21.05
(0.2,0.15,1)	(0.3,0.9,1)	32.88
(0,0,1)	(0,0.05,1)	2.86
(0,0.2,1)	(0,0.05,1)	8.45
(0.3,0.1,1)	(0.15,0.4,1)	18.15
(0.85,0.3,1)	(0.95,0.3,1)	3.19

In Table 8, totally 13 planes are used for verification the neural network, the correct miller index is shown in the left column to compare with the prediction miller index. According to R-value assessment, there are 5 predictions with R-value less than 8°, with means the classification accuracy for real application testing is 5/13 = 38.46%. This is an unsatisfactory result compared with the accuracy of Brute Force method and cannot be trusted in the real application. However, the verification cannot reflect the real performance of neural network as the network is trained for 360 directions, 73 directions are too strict for it.

4.2 Sequence Classification:

In this section, the sequence classification neural network training result and testing result will be presented and analysed.

4.2.1 Training Results and analysis:

According to 4.1.3, if the neural network is trained with 360 directions, it will have poor performance to deal with the SAW with only 73 directions. So, in the sequence classification neural network training, two networks were trained with 360 directions and 73 directions alternatively, the training strategy and options are shown in 3.5.2.2

Similar to 4.1.1, figures below are the training progress for sequence classification neural network with 73 SAW directions.

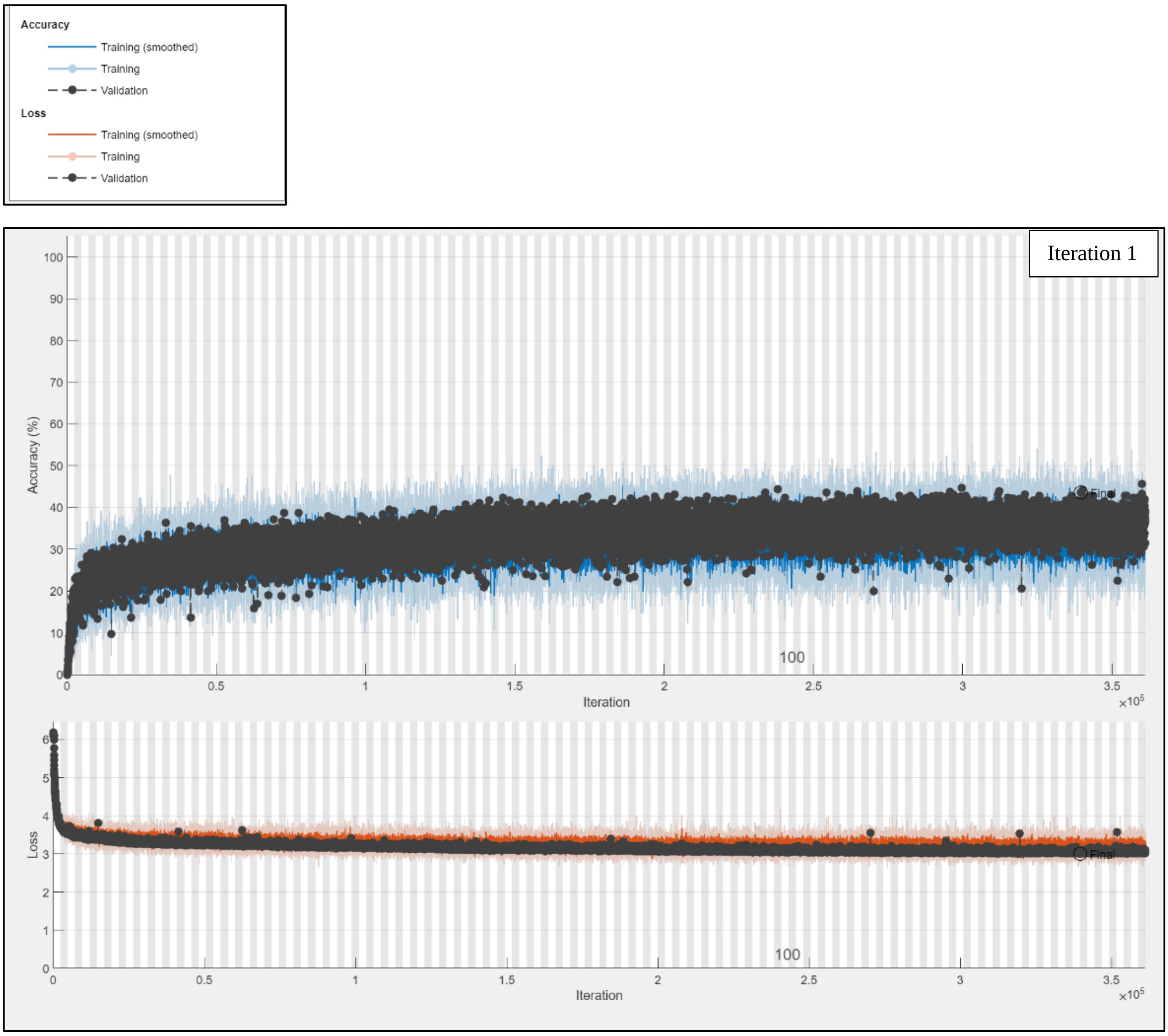


Figure 30. Training progress of iteration 1.

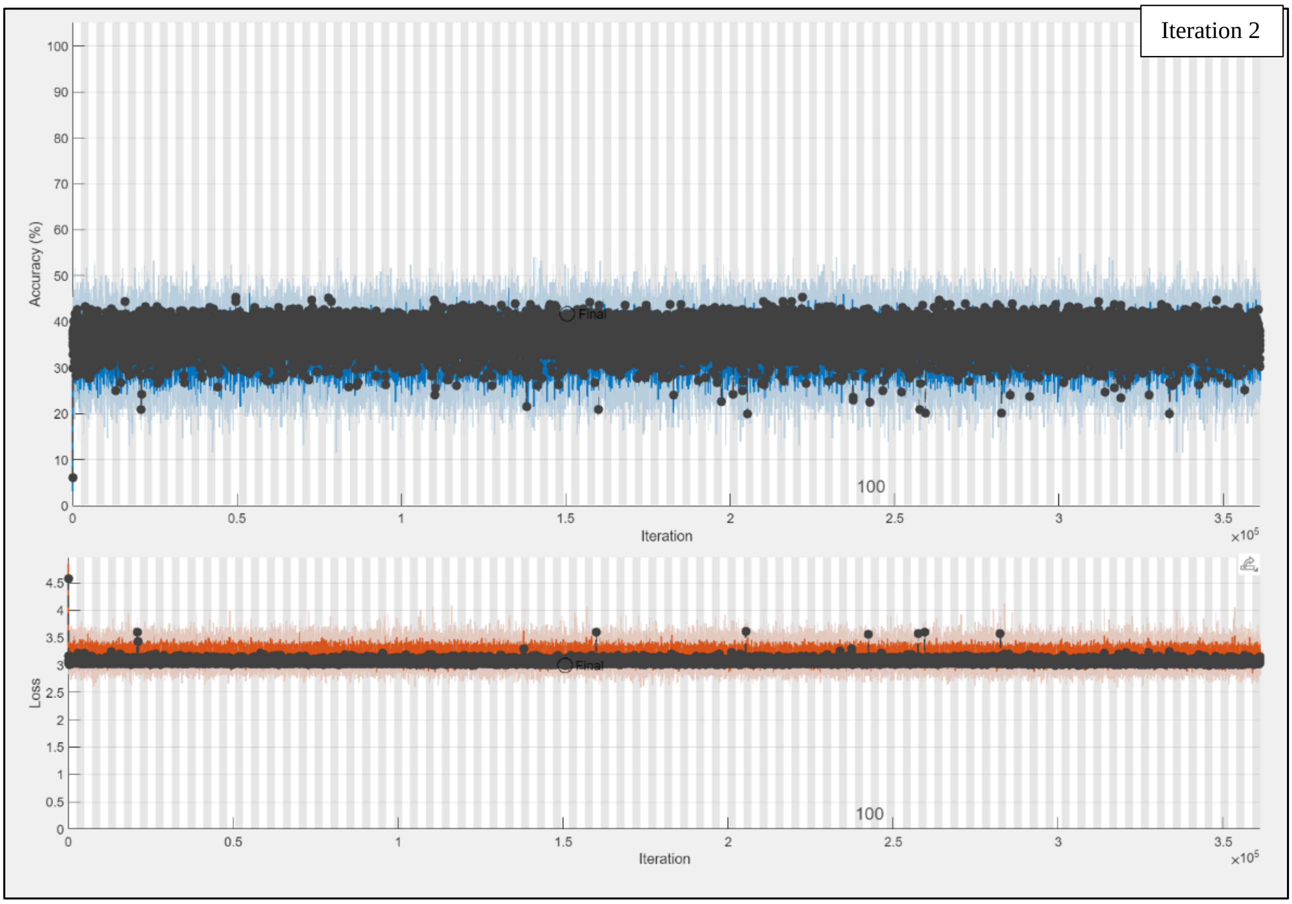


Figure 31. Training progress of iteration 2.

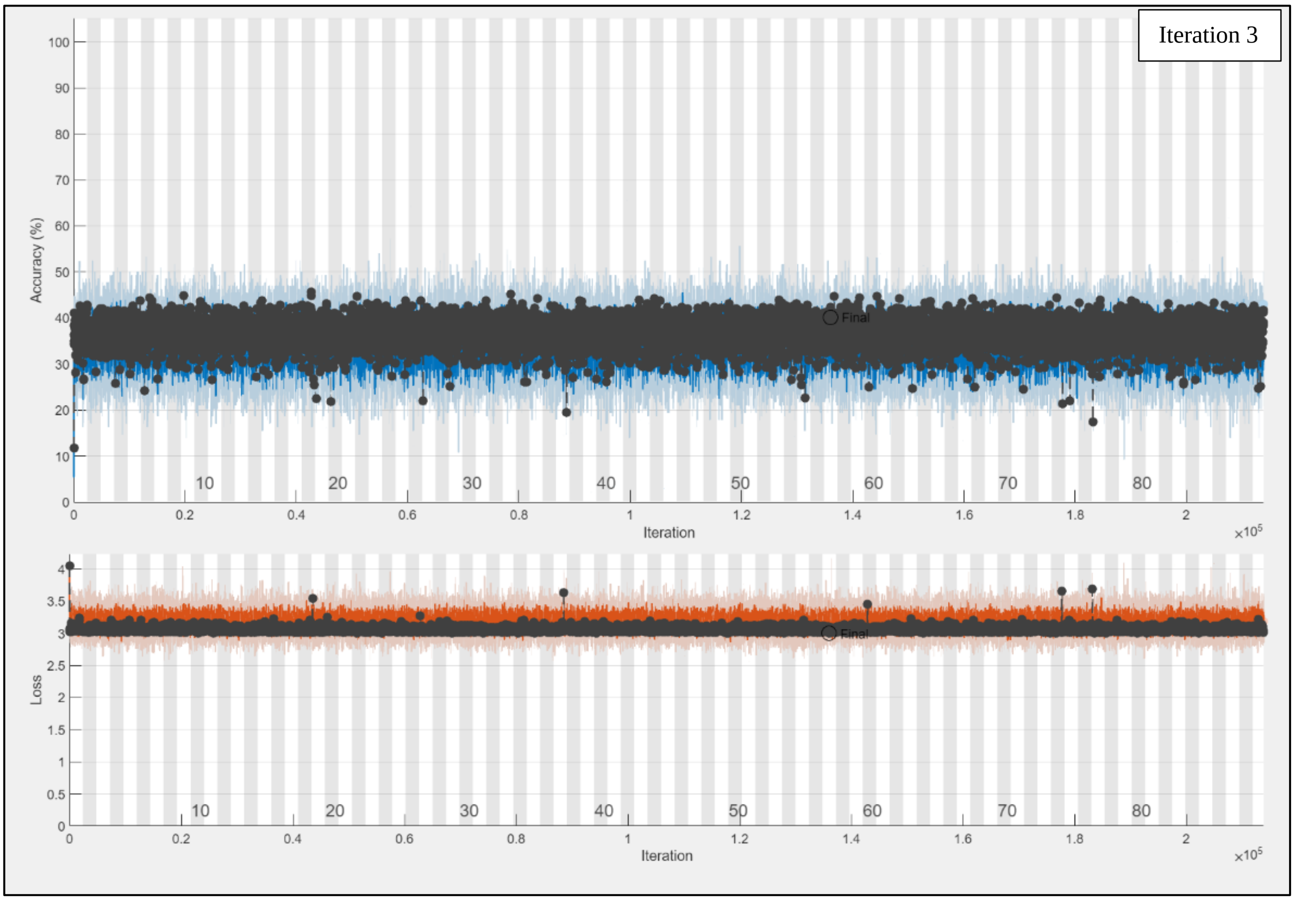


Figure 32. Training progress of iteration 3.

It can be observed that in the above figures,

1) The training result of sequence neural network will not change if the epochs value is large enough, obviously, 150 is an enough value for each training iteration as the accuracy will no longer increase after several epochs. As a result, the training plots above are all complete training with current training strategy.
2) The Figures above show only 3 iterations, from the second iteration, even the neural network is feed with new patterns of noisy data, the validation accuracy will not have a significant increase. This phenomenon happens very training iteration after the first one, so 3 iterations are sufficient for analyse the training process.
3) Considering how good the deep learning process is, the training for sequence classification neural network is neither underfitting nor overfitting, as it can be observed that the black dots for validation accuracy and blue dots for training accuracy almost overlap together during the training, but the accuracy for validation and training is also fluctuating between 30% to 40%.
4) When the epochs reach 150, the training will finish, and the output network will be the epoch with the minimum validation loss, at that point, the validation accuracy of network is 43.54% for 1st iteration, 41.72% for 2nd iteration, 40.54% for 3rd iteration, the network from 3rd iteration will be applied for future, because even its accuracy is not the highest, it has seen the most different patterns of noise.
5) The non-increasing of validation accuracy during training is probably caused by the specific structure of neural network, which means the 4-layer convolution neural network reaches its maximum to perform the sequence classification task, to increase the validation accuracy, more complex neural network structure is necessary.

4.2.2 Validation Results and analysis:

Validation of this neural network is achieved by testing it with new-generated noisy data from MATLAB, the validation is also based on different SNRs. In the validation section, to compare the result with image classification neural network, the neural network with 360 directions training is validated.

Table 9 shows the accuracy comparison for using sequence and image classification, the accuracy is not based on R-value assessment, so if the prediction is not same with the label, the accuracy for that plane is zero.

Table 9. Validation accuracy comparison for different SNR inputs.

SNR	Sequence network accuracy (%)	Image network accuracy (%)
10	2.72	0.45

20	20.63	5.22
30	61.68	56.10
40	86.39	80.50
50	97.28	96.37
60	98.64	99.55
70	98.41	99.55

As indicated in Table 9, when using the sequence classification neural network, the performance in accuracy is better than image classification neural network for smaller SNR, to be specific, for SNR = 20 and 30, there are obvious increase in the classification accuracy. This is because sequence classification neural network will recognize and learn the raw SAW values, rather than plotting the polar plot and recognizing the images, which will affected by some unimportant features. However, for the higher SNR, the sequence network accuracy is a bit smaller than image classification neural network, but that does matter as the accuracy are both satisfactory.

To be noted, for the sequence classification neural network, even the accuracy has increased a lot, its accuracy for SNR30 is still unsatisfactory, compared with higher SNRs, so the R-value assessment will also be applied in sequence classification results.

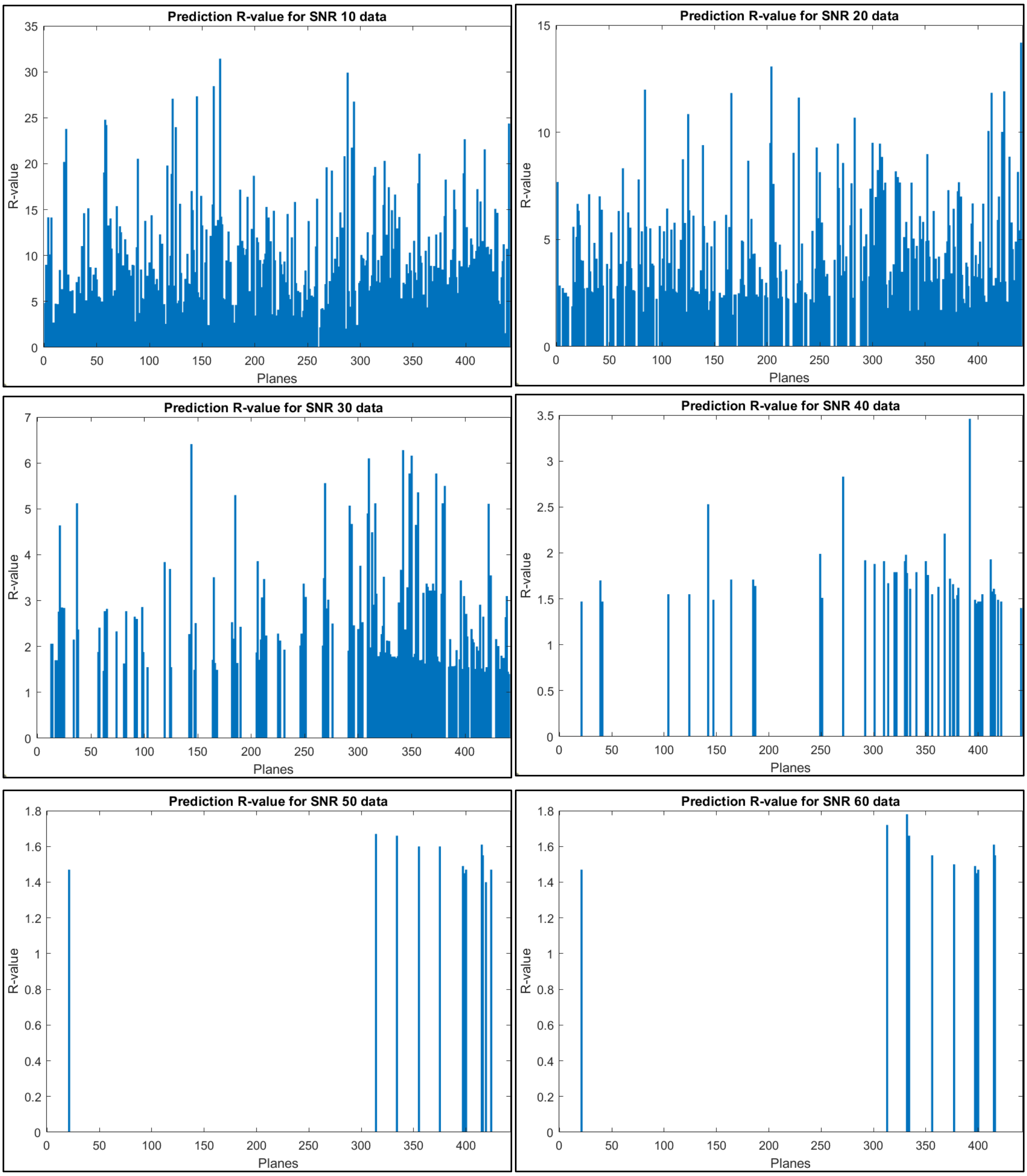


Figure 33. R-values bar charts for predictions at different SNRs.

The bar charts above show the specific R values when different SNR noise are applied to sequence classification neural network. It can be observed that when the SNR value decreases, for each specific plane, the prediction R-value will increase, this is because the increasing of noise will cause more wrong classification. Also, it can be observed that for SNR30 to 60, there are many planes with zero R-value, and the maximum R-values are smaller than 7°,which means the prediction is the same as label within the industrial standard.

Using the R-value assessment as declared in 4.1.2, the comparison between accuracies of sequence classification neural network and image classification neural network is shown in table 10.

Table 10. Validation accuracies comparison for different SNRs using R-value assessment (totally 441 planes).

SNR	Image classification number of prediction R values < 8°	Image classification accuracy (%)	Sequence classification number of prediction R values < 8°	Sequence classification accuracy (%)
10	60	13.6	250	56.7
20	120	27.2	415	94.1
30	437	99.1	441	100
40	441	100	441	100
50	441	100	441	100
60	441	100	441	100
70	441	100	441	100

It can be observed from table 10 that compared with image classification neural network, the accuracy for sequence classification increases obviously by using R-value assessment, especially for SNR20, using image classification network, the prediction accuracy is just 27.2%, using sequence classification network, the prediction can exceed 90%, this is a large breakthrough as this means the sequence classification can be useful for SNR20 to give 94.1% accurate result.

The reason for large increasing in accuracy might be related to the advantage in sequence neural network itself: the image neural network will recognized the image as a shape, so with lower SNR, it's extremely hard to track the shape, while for the sequence classification neural network, the raw SAW sequence will be processed, if the features still exist in the signal, the neural network can search and find them for classification. This is why sequence classification are preferred in this project.

However, when the SNR is 10, the accuracy of sequence classification network is 56%, which means the SNR is too small. The possible improvement might be related to increase the complexity of network structure.

4.2.3 Verification Results and analysis:

The verification of sequence neural network is based on the real data testing from experiments, instead of the simulated noise data in MATLAB. As mentioned in the 4.2.1, the sequence classification neural networks were trained with 360 features and 73 features alternatively, the results for 360 features were used for validation section, and the results for 73 features were used for this section, the verification results for sequence classification neural network applied in real data are shown in Table 11.

Table 11. Predication result and R-value for real data with 73 directions

Plane	Prediction	R value
(0.1,0,1)	(0.15,0.4,1)	21.78
(0.35,0.2,1)	(0.35,0.55,1)	16.99
(0.2,0.2,1)	(0,0.6,1)	22.65
(0.3,0.15,1)	(0.05,0.25,1)	14.89
(0.35,0.05,1)	(1,0.6,1)	32.27
(0.1,0,1)	(0,0.1,1)	8.07
(0.3,0.2,1)	(0.5,0.3,1)	10.48
(0.05,0.1,1)	(0.1,0,1)	6.38
(0.2,0.15,1)	(0,0.15,1)	11.22
(0,0,1)	(0.75,0.15,1)	39.08
(0,0.2,1)	(0,0.1,1)	5.6
(0.3,0.1,1)	(0.2,0.05,1)	6
(0.85,0.3,1)	(1,0.3,1)	4.68

In Table 11, totally 13 planes are used for verification the neural network, the correct miller index is shown in the left column to compare with the prediction miller index. According to R-value assessment, there are 3 predictions with R-value less than 8°, with means the classification accuracy for real application testing is 3/13 = 23.07%, which is less than the verification accuracy of image classification neural network in 4.1.3 (38.46%). However, only 13 planes cannot represent the neural network performance over all the 441 planes, but it can be concluded that these are unsatisfactory results compared with the accuracy of Brute Force method and cannot be trusted in the real application, even the neural network was trained with 73 directions as it is in the real data. The possible explanation for low accuracy in real data prediction for sequence network trained with 73 directions could be that the structure of CNN applied was too simple for this task.

5 CONCLUSION:

According to objectives of this project, including background knowledge review, two neural network, image-to-label classification neural network, and sequence-to-label classification neural, should be created for accepting SAW velocities with specific noise as training data, and the training results, such as the training progress, trained network, should be generated and saved. Obviously, the achievement of this project can perfectly satisfy those requirements in Objectives.

However, considering the initial aims to apply the neural network in the experiments to substitute the old Brute Force method for crystallographic orientation determination of nickel, it still has a long way to go for the neural networks applied in this project. To be specific, the experimental testing accuracy for image network is about 39%, for sequence network is about 23.07%, even far behind the 0.5 probability for neural network to 'guess the answer'.

In the other aspects, even the real data verification result is not satisfactory, the simulation results from validation of neural network present some valuable information for later researchers to refer: For the image classification neural network, by adding Gaussian white noise to simulate the real noise in the experiments, after specific training, the accuracy for SNR>30 data is 99%-110%, a specific decrease in accuracy is shown to at SNR20. For the sequence classification neural network, also after specific training, the accuracy for SNR>20 data is 99%-100%, shown to have a relatively increase in noise resistance of about 10SNR compared with image network. This indicates that if someone want to continue the work upon current, sequence-to-label CNN is a highly recommended option, besides, to overcome the deflects in this project, some possible solutions are proposed. Firstly, the noise model to simulate the noise might be too simple for the circumstances in the laboratory, because of the time reason, the noise applied in this simulation is just GWN, even this could simulate all the possible noise in the real world, such as thermal, shot, flicker noise, the cases that the average of SAW would change haven't been considered in the noise simulation. If more complex noise model will be applied in the training process of neural network, the accuracy for real data verification might increase. Secondly, even more complete noise model will be applied, the accuracy for neural networks might also be unsatisfactory because the structure of specific networks used in this projects, such as those mentioned in 3.5.1.1 and 3.5.2.1, might be too simple and shallow for this classification task with 441 different labels. Some possible indications can be obtained from the previous employment famous CNNs in the world, such as ImageNet, and GoogleNet in MATLAB.

To be noted that the MATLAB codes and all the results are upload on GitHub [29].

6 Reference:

[1] M. Karadge, M. Preuss, P.J. Withers, S. Bray, Importance of crystal orientation in linear friction joining of single crystal to polycrystalline nickel-based superalloys, Materials Science and Engineering: A, Volume 491, Issues 1–2, 2008, Pages 446-453, ISSN 0921-5093, https://doi.org/10.1016/j.msea.2008.04.064.

[2] Britannica, The Editors of Encyclopaedia. "nickel". Encyclopedia Britannica, 21 Feb. 2023, https://www.britannica.com/science/nickel-chemical-element. Accessed 7 May 2023.

[3] Ahssi MAM, Erden MA, Acarer M, Çuğ H. The Effect of Nickel on the Microstructure, Mechanical Properties and Corrosion Properties of Niobium-Vanadium Microalloyed Powder Metallurgy Steels. Materials (Basel). 2020;13(18):4021. Published 2020 Sep 10. doi:10.3390/ma13184021

[4] Savage, M. W. R. (October 27, 2011). "The Influence of Crystal Orientation on the Elastic Stresses of a Single Crystal Nickel-Based Turbine Blade." ASME. *J. Eng. Gas Turbines Power*. January 2012; 134(1): 012501. https://doi.org/10.1115/1.4004129

[5] Wikipedia. "Nickel." from https://www.britannica.com/science/nickel-chemical-element.

[6] Cambridge, U. o. "Common Cubic Structures." from https://www.gan.msm.cam.ac.uk/resources/crystalmodels/cubic.

[7] Wikipedia. "Miller index." from https://en.wikipedia.org/wiki/Miller_index.

[8] Wikipedia. "X-ray crystallography." from https://en.wikipedia.org/wiki/X-ray_crystallography.

[9] Clay, K., et al. (2010). "R value measurement issues for grain boundaries in nickel single crystal castings." Materials Science and Technology **26**(7): 781-786.

[10] Instruments, O. (2023). "What is Electron Backscatter Diffraction (EBSD)." from https://www.ebsd.com/ebsd-explained/what-is-ebsd.

[11] Instruments, O. (2023). "Electron Backscatter Diffraction (EBSD) Pattern Formation
" From https://www.ebsd.com/ebsd-explained/pattern-formation.

[12] Wikipedia. "Electron backscatter diffraction." From https://en.wikipedia.org/wiki/Electron_backscatter_diffraction.

[13] Rikesh Patel, Wenqi Li, Richard J. Smith, Steve D. Sharples, Matt Clark, Orientation imaging of macro-sized polysilicon grains on wafers using spatially resolved acoustic spectroscopy, Scripta Materialia, Volume 140, 2017, Pages 67-70, ISSN 1359-6462, https://doi.org/10.1016/j.scriptamat.2017.07.003.

[14] Li W, Sharples SD, Smith RJ, Clark M, Somekh MG. Determination of crystallographic orientation of large grain metals with surface acoustic waves. J Acoust Soc Am. 2012 Aug;132(2):738-45. doi: 10.1121/1.4731226. PMID: 22894196.

[15] Li W, Sharples SD, Smith RJ, Clark M, Somekh MG. Determination of crystallographic orientation of large grain metals with surface acoustic waves. J Acoust Soc Am. 2012 Aug;132(2):738-45. doi: 10.1121/1.4731226. PMID: 22894196.

[16] P. Dryburgh et al., "Determining the crystallographic orientation of hexagonal crystal structure materials with surface acoustic wave velocity measurements," Ultrasonics, vol. 108, p. 106171, 2020/12/01/ 2020.

[17] Nottingham, U. o. Scanning Electron Microscopy (SEM).

[18] Nottingham, O. a. P. U. o. SRAS System. S. System. **983*708 pixels**.

[19] Network and Complex Systems www.iiste.org ISSN 2224-610X (Paper) ISSN 2225-0603 (Online) Vol.3, No.1, 2013-Selected from International Conference on Recent Trends in Applied Sciences with Engineering Applications

[20] Baek, Jieun & Choi, Yosoon. (2020). Deep Neural Network for Predicting Ore Production by Truck-Haulage Systems in Open-Pit Mines. Applied Sciences. 10. 1657. 10.3390/app10051657.

[21] Sarker, I. H. (2021). "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions." SN Computer Science **2**(6): 420.

[22] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80.

[23] Mishra, Vidushi & AGARWAL, SMT & PURI, NEHA. (2018). COMPREHENSIVE AND COMPARATIVE ANALYSIS OF NEURAL NETWORK. INTERNATIONAL JOURNAL OF COMPUTER APPLICATION. 2. 10.26808/rs.ca.i8v2.15.

[24] Li B, François-Lavet V, Doan T, Pineau J. Domain adversarial reinforcement learning. arXiv preprint arXiv:2102.07097, 2021

[25] Venners, B. (2003). "The Making of Python A Conversation with Guido van Rossum, Part I." Retrieved January,1, from https://www.artima.com/articles/the-making-of-python.

[26] Walid A. Zgallai (2020) Biomedical Signal Processing and Artificial Intelligence in Healthcare. [edition unavailable]. Elsevier Science. Available at: https://www.perlego.com/book/1810183/biomedical-signal-processing-and-artificial-intelligence-in-healthcare-pdf (Accessed: 15 October 2022).

[27] MathWorks (2023). "convolution2dLayer." from https://ww2.mathworks.cn/help/deeplearning/ref/nnet.cnn.layer.convolution2dlayer.html#responsive_offcanvas.

[28] MathWorks (2023). "batchNormalizationLayer." from https://ww2.mathworks.cn/help/deeplearning/ref/nnet.cnn.layer.batchnormalizationlayer.html.

[29] Project codes release https://github.com/uwanny/Final-year-project/releases/tag/v1.0.0