CONVOLUTIONAL NEURAL NETWORKS TO CLASSIFY OIL, WATER AND GAS PIPE FLUID USING ACOUSTIC SIGNALS

Nafiseh Vahabi, MIEEE [*]

Department of Electronic
and Electrical Engineering
University College London
UK, London
Email: uceevah@ucl.ac.uk

David R. Selviah, MIEEE

Department of Electronic
and Electrical Engineering
University College London
UK, London
Email: d.selviah@ucl.ac.uk

ABSTRACT

The aim of the present research is to investigate the classification algorithms that identify the fluid type in oil, water and gas pipes by analysing acoustic datasets. The data is collected during 24 hours from optical laser acoustic sensors which is attached alongside the 4000 m of oil, water and gas pipes. In this research we used the sample of data from 1000 m and $20,000$ s. We implemented Artificial Neural Networks (ANN) and Conventional Neural Networks (CNN) algorithms to recognise the patterns of each fluid type by analysing its acoustic energy. Both algorithms were trained on three datasets (oil, gas and water) and tested on another dataset from different water pipe. The result of this study shows ANN and CNN algorithms classify the fluid type with the accuracy of 79.5% and 99.3% respectively when applied on the test data set.

***Index Terms*—** Convolutional Neural Networks, Artificial Neural Networks, Fluid Flow Classification, Signal Processing

1. INTRODUCTION

Downhole oil and gas pipe fluid flow classification is an increasingly important area in the oil industry and is part of the Downhole Fluid Analysis (DFA) which provides well logging and reservoir evaluation. Conventional DFA is performed by measuring one or more properties of fluid such as pressure, volume, density, Reynolds number and temperature using corresponding sensors [1]. Identifying in-well flow regime became an important component in monitoring the oil pipes and it results in low cost intervention [2], optimising and maximising the oil production [3].

It is now well established from a variety of studies, that ultrasonic techniques are robust, inexpensive and non-invasive to use as a flow meter but one of the most frequently stated problems with this technique is its requirement for pre-calibrating signals [4], [5]. The last two decades have seen a growing trend towards using machine learning algorithms in variety of industrial applications. Artificial Neural Networks (ANN) are amongst the first types of machine learning algorithm that have been employed to improve the precision of ultrasonic devices and also to automate the process of fluid flow measurement in multi-phase flow [6]. The combination of ANN and dual energy fan-beam gamma-ray attenuation technique improved the accuracy of oil, water and gas classification by approximately 5.68% using radial basis function for ANN training [5]. Identifying the pattern in gamma-ray pulse height distributions was another approach that used ANNs [7]. Scientists trained the parameters of ANNs with algorithms such as Levenberg-Marquardt [8] to describe the details of two phase flow [9] and to develop a new multiphase flow metering device for real time multiphase flow classification [8]. This new device is based on training the parameters in physical models of multiphase fluid. In another study, 199 experimental data sets fed into three-layer back-propagation neural networks and achieved 97% accuracy in its prediction of flow regime [10].

Most studies in the field of multi-phase flow classification have mainly focused on modifying the structure and parameters of of Artificial Neural Networks [11] to identify the pattern of each flow regime and have not dealt with processing the big data which is produced by their new developed sensors [8]. Data for this study were collected by using laser optical fibre Distributed Acoustic sensors [3] and shows the acoustic energy in the time-distance domain as can be seen in Fig. 1. Acoustic data gathered from oil, water and gas pipes during $20,000$ seconds and within 1000 m. For each fluid type 1000 frames generated show acoustic energy at different locations within the same period of time. Therefore, the purpose of this paper is to explore methods that can identify the

[*]The author, Nafiseh Vahabi, thanks K. Johannessen of Statoil for permitting the data recorded in their wells to be used in this paper.The authors thank Mahmoud Farhadiroushan, Tom Parker, Sergey Shatalin, Mladen Todorov of Silixa for use of their iDAS recording system. The author Nafiseh Vahabi also thanks the UK Government Research council EPSRC for her funding studentship.

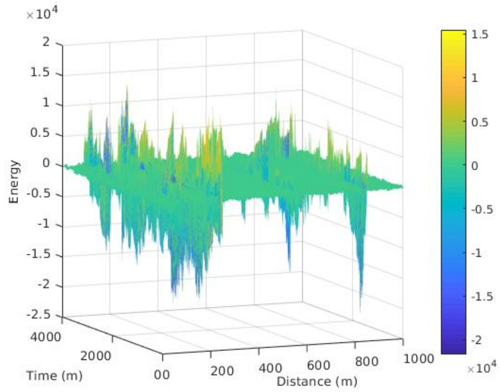

Fig. 1. Sample of acoustic raw data in time-distance domain

abstract pattern in different types of fluid flow, overcome the processing of big data and improve the accuracy of classification comparing with ANN.

2. METHODS

2.1. Articial Neural Network

The structure of our ANN contains 16 hidden layers and input data are acoustic energy of oil, water and gas dataset during $20,000$ s and within 1000 m. The data is divided into three sets; 70% for training, 15% for testing and 15% for validation. Levenberg-Marquardt (LM) is selected for training a network with a dataset containing thousands of images as is the case with our dataset. Therefore, we train ANN by the Levenberg-Marquardt (LM) [12] algorithm to update the network's weights and biases. The Levenberg-Marquardt is designed to minimise the loss functions made up of a sum of squared errors.

ANN takes three datasets and classifies them into oil, gas or water category. Fig. 2 presents the outcome of ANN on the training, testing and validation dataset. Overall, 79.5% of the predictions were correct and 20.5% were incorrect classifications. Fig. 3 illustrates the error of the ANN classification with the error of 0.020 in highest number of samples.

2.2. Convolutional Neural Network

One of the most successful types of Neural Network with a great result in a variety of computer vision and pattern recognition applications is called Convolutional Neural Network (CNN) [13]. In the network with CNN structure there are many copies of the same neuron that develop a large neural network with the smaller number of parameters. Therefore, the network does not need to learn a large number of parameters and it uses one neuron in many places. This will dramatically reduce the learning error in CNN structure.

Fig. 2. Result of classification of oil, water and gas dataset. The first three diagonal cells show the number and percentage of correct classifications by the trained network. For example, in the test confusion matrix (bottom right matrix in Fig. 2) 35 water samples are correctly classifed as water (class 1) which corresponds to 7.1% of all 493 samples. Similarly, 105 cases are correctly classified as gas and this corresponds to 21.3% of all samples and 51.1% are correctly classified as oil and this corresponds to 51.1% of all samples.

We fed Convolutional Neural Network with the pre-processed data. In the pre-processing stage, data is normalised and transformed from time-distance domain to frequency-wave number domain using two dimensional Fast Fourier Transform (2FFT) algorithm. In this implementation the Neural Network Tool box from Matlab $2017b$ version library was used. The implementation commenced with a seven layers Convolutional Neural Network because this is one of the simple CNN architecture which we started with to avoid making a complex CNN unnecessarily. Also this architecture has been used on other image datasets (MINST dataset) [13]. The layers in the network are as follow;

Layer 1: Image Input Layer In this layer the property of the input images has been specified. The size of each image in our data set is 197-by-256-by-3 which is acoustic data during $20,000$ s and within 1000 m (Fig. 1). The images are RGB type.

Layer 2: Convolutional Layer A Convolutional layer is formed by neurons that might have parallel or multilayer architecture. These neurons connect to the small regions of the input images or their previous layer. These small areas are called filters whose size needs to be defined. For each region,

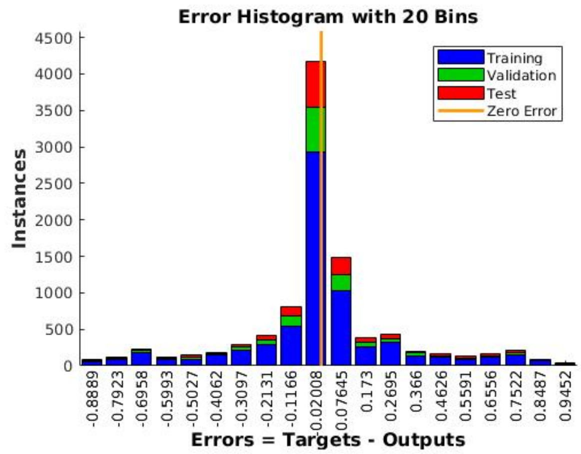

Fig. 3. Error bar for training, testing and validation dataset.

we perform a basic calculation of a neural network that is a dot product of the input and the weights, and then add a bias. An input image is convolved by scanning the filter both horizontally and vertically along the image, repeating the same calculation. In addition to the filter size, we need to specify the step size for moving the filter. In some cases the step size and filter size might result in an overlap between the local region each neuron is connected to [14]. Each filter requires the total number of $h \times w \times c$ weights, where h and w are the height and width of the filter respectively and c shows the number of colour channel in the input image. Each filter uses the same set of weights and biases to scan the whole input image. The outcome of the convolution creates a feature map. Therefore, the number of filter determines the number of feature map in the convolution layer [15], [16]. Eq. 1 computes the total number of parameters in each convolution layer.

$$P = (h \times w \times c + 1) \times (\text{Number of filters}) \tag{1}$$

The output of the convolutional layer must be an integer otherwise the filters can not fully cover the whole input image. Eq. 2 calculates the height and width of the convolutional layer;

$$\begin{aligned} &\text{Output of Convolutional Layer} \\ &= \frac{(\text{Input Size} - \text{Filter Size} + 2 \times \text{Padding})}{\text{step size}} + 1 \end{aligned} \tag{2}$$

Let's define map size as the total number of feature maps. Eq. 3 shows how we count the total number of neurons in a convulotional layer;

$$\begin{aligned} \text{Total Number of Neuron in a Convolutional layer} \\ = \text{Map size} \times \text{Number of filters} \end{aligned} \tag{3}$$

The input data set in our covolutional layer contains colour images of size 197-by-256-by-3. We defined 20 filters and a filter size is the vector $[7, 10]$ [13], the number of weights per filter is $7 \times 10 \times 3 = 210$, and the total number of parameters in the layer is $(210 + 1) \times 20 = 4220$.

Layer 3 : ReLU Layer This layer is called Rectified Linear Unit layer and is located after the convolutional layer. The activation function is defined in this layer and performs a threshold operation to each component of its input. A Rectified Linear activation function is defined in Eq 4. It should be noted that the size of the output from this layer is the same as its input [17].

$$f(x) = \begin{cases} x & 0 \leq x \\ 0 & 0 > x \end{cases} \tag{4}$$

Layer 4 : Max-Pooling Layer This layer is used to reduce the number of parameters in the network and avoid overfitting. There is no learning operation in this layer. We use max-pooling function to down sample the parameters in this layer. Max-pooling function takes an input from the activation function and outputs the maximum values of the rectangular area in its input. In the network with multiple convolutional layers, we need to have max-pooling layer between each of the two convolutional layers to reduce the number of parameters. The pool size and the step size need to be specified. If the size of the pool is more than the step size then there is some overlapping in the scanning area. Suppose that the size of the input is $n - by - n$ and the pool size is $h - by - h$, the max-pooling function reduces the output of the max-pooling layer to the size h [18]. We choose both step size and pool size to be number 4.

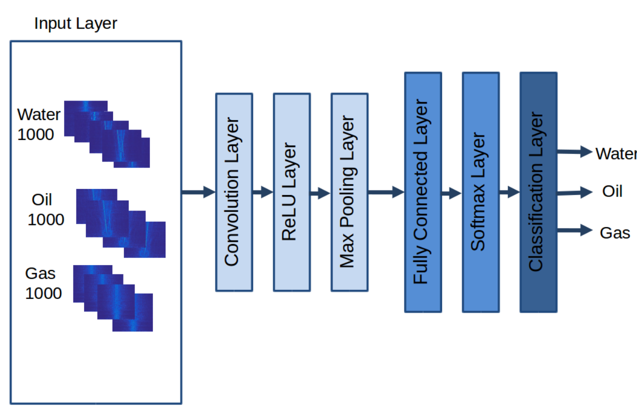

Fig. 4. Structure of Convolutional Neural Network.

Layer 5 : Fully Connected Layer All the features learned by previous layers will combine in a Fully Connected Layer to classify the input image. Therefore, the number of outputs of this layer equals the number of classes the images belongs to. In our data set we set this number to three (water, oil and gas).

Layer 6 : Softmax Layer This layer contains an activation function to perform classification in the fully connected layer. This layer is usually located after the last fully connected layer. Softmax function is defined in Eq. 5 and it is considered when we need to classify more than two classes. It is also known normalised exponential [19].

$$p(c_r|x,\theta) = \frac{p(x,\theta|c_r)p(c_r)}{\sum_{j=1}^{k} p(x,\theta|c_j)p(c_j)} = \frac{exp(a_r(x,\theta))}{\sum_{j=1}^{k} exp(a_j(x,\theta))} \tag{5}$$

where $p(c_r)$ is the class prior probability, $0 < p(c_r|x,\theta) \leq 1$, $\sum_{j=1}^{k} p(c_j|x,\theta) = 1$ and the conditional probability for class r is $p(x,\theta|c_r)$ with $a_r = ln(p(x,\theta|c_r)p(c_r))$ [19].

Layer 7 : Classification Layer Classification layer is the last layer of the network which outputs the probabilities of inputs belonging to one of the classes based on the result of the softmax function. We compute the error of multi-class classification problem with the cross entropy function (Eq. 6) [19].

$$E(\theta) = -\sum_{i=1}^{n}\sum_{j=1}^{k} t_{ij} ln y_j(x_i,\theta) \tag{6}$$

Where t_{ij} shows the ith sample has assigned to class j, the parameter vector is θ and $y_j(x_i,\theta)$ is the probability that an input i has been allocated to the class j [19].

A Gaussian distribution with a mean of zero and a standard deviation of 0.01 is commonly used for initialising the weights and biases of the convolutional neural network. The network parameters are updated by Stochastic Gradient Descent (SGD) algorithm [19] as its preferred algorithm for training thousand of images. The SGD algorithm takes the small steps towards the negative gradient direction to minimise the error function where;

$$\theta_{\ell+1} = \theta_\ell - \alpha\nabla E(\theta_\ell) \tag{7}$$

where ℓ and θ are the iteration number and the parameter vector respectively, $\alpha > 0$ determines the learning rate and the loss function is $E(\theta)$. The SGD algorithm uses the whole training set to evaluate the gradient of the loss function, $\nabla E(\theta)$ and updates the parameters in each iteration. The SGD uses a subset of the training set which is called a mini-batch during each iteration to [19] take one step toward minimising the loss function. A epoch is a period of the training algorithm over all the training data set using the mini-patches. The last term in Eq. 8 is the momentum where λ is related to the contribution of the previous gradient step to the current iteration [19].

$$\theta_{\ell+1} = \theta_\ell - \alpha\nabla E(\theta_\ell) + \gamma(\theta_\ell - \theta_{\ell-1}) \tag{8}$$

Accuracy is the ratio of the number of true labels in the test data matching the classifications from classify, to the number of images in the test data. In this case about 98.5% of the digit estimations match the true digit values in the test set.

```
  197   256    3

Training on single CPU.
Initializing image normalization.
|==========================================================================================|
|     Epoch     |   Iteration   | Time Elapsed |  Mini-batch  |  Mini-batch  | Base Learning|
|               |               |  (seconds)   |     Loss     |   Accuracy   |     Rate     |
|==========================================================================================|
|             1 |             1 |        11.94 |       1.6672 |       54.69% |     1.00e-04 |
|             3 |            50 |       615.56 |       0.0003 |      100.00% |     1.00e-04 |
|             6 |           100 |      1207.89 |       0.0000 |      100.00% |     1.00e-04 |
|             9 |           150 |      1801.37 |       0.0000 |      100.00% |     1.00e-04 |
|            12 |           200 |      2385.95 |       0.0000 |      100.00% |     1.00e-04 |
|            15 |           250 |      3059.88 |       0.0000 |      100.00% |     1.00e-04 |
|            15 |           255 |      3135.30 |       0.0000 |      100.00% |     1.00e-04 |
|==========================================================================================|
```

Fig. 5. Result of classification of oil, water and gas dataset using Convolutional Neural Network on FFT data.

Table 1. Result of Classification Algorithms

Method	Accuracy	Time s
ANN Training	80.5%	0.02
ANN Testing	79.5%	0.001
CNN Training	100%	3135.30
CNN Testing	99.3%	0.01

3. CONCLUSION

The present study was designed to investigate the algorithms for classification of the fluid types oil, water and gas datasets. Artificial Neural network and Convolutional neural Networks were implemented for this purpose. The most obvious finding to emerge from this study is that Convolutional neural Networks outperformed ANN to classify our datasets with accuracy of 99.3%. However, CNN takes longer to produce the result and this can be resolved by running the CNN on a GPU cluster.

4. REFERENCES

[1] C. Dong, M.D. O'Keefe, H. Elshahawi, M. Hashem, S.M. Williams, D. Stensland, P.S. Hegeman, R.R Vasques, T. Terabayashi, and O.C. Mullins, "New downhole-fluid-analysis tool for improved reservoir characterization," *SPE Reservoir Evaluation & Engineering*, vol. 11, no. 6, pp. 1–107, 2008.

[2] G. Roshani, S. Feghhi, A. Mahmoudi-Aznaveh, E. Nazemi, and A. Adineh-Vand, "Precise volume fraction prediction in oil–water–gas multiphase flows by means of gamma-ray attenuation and artificial neural networks using one detector," *Measurement*, vol. 51, pp. 34–41, 2014.

[3] JJ. Xiao, M. Farhadiroushan, A. Clarke, R.A. Abdalmohsen, E. Alyan, T.R. Parker, J. Shawash, and H.C. Milne, "Intelligent distributed acoustic sensing for in-well monitoring," in *SPE Saudi Arabia Section Technical Symposium and Exhibition*. Society of Petroleum Engineers, 2014.

[4] M.M.F. Figueiredo, JL Goncalves, A.M.V Nakashima, AMF Fileti, and R.D.M. Carvalho, "The use of an ultrasonic technique and neural networks for identification of the flow p. and measurement of the gas volume fraction in multiphase flows," *Experimental Thermal and Fluid Science*, vol. 70, pp. 29–50, 2016.

[5] G. Roshani, E. Nazemi, and M.M. Roshani, "Intelligent recognition of gas-oil-water three-phase flow regime and determination of volume fraction using radial basis function," *Flow Measurement and Instrumentation*, vol. 54, pp. 39–45, 2017.

[6] M. Al-Naser, M. Elshafei, and A. Al-Sarkhi, "Artificial neural network application for multiphase flow patterns detection: A new approach," *Journal of Petroleum Science and Engineering*, vol. 145, pp. 548–564, 2016.

[7] C.M. Salgado, C.M.N.A. Pereira, R.s Schirru, and L. E.B. Brandão, "Flow regime identification and volume fraction prediction in multiphase flows by means of gamma-ray attenuation and artificial neural networks," *Progress in Nuclear Energy*, vol. 52, no. 6, pp. 555–562, 2010.

[8] M. Meribout, N.Z. Al-Rawahi, A.M. Al-Naamany, A. Al-Bimani, K. Al-Busaidi, and A. Meribout, "A multisensor intelligent device for real-time multiphase flow metering in oil fields," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 6, pp. 1507–1519, 2010.

[9] E.H. Inoue, R.D.M. Carvalho, V. Estevam, A.C. Bannwart, and A.M.F. Fileti, "Development of a neural network for the identification of multiphase flow pattern," *14th International Conference on Intelligent Systems and Control*, 2013.

[10] E.A. Osman et al., "Artificial neural network models for identifying flow regimes and predicting liquid holdup in horizontal multiphase flow," *SPE production & facilities*, vol. 19, no. 1, pp. 33–40, 2004.

[11] K. Madani, "Artificial neural networks based image processing and pattern recognition: From concepts to real-world applications," in *Image Processing Theory, Tools and Applications, 2008. IPTA 2008. First Workshops on*. IEEE, 2008, pp. 1–9.

[12] J. Shawash and D.R. Selviah, "Real-time nonlinear parameter estimation using the levenberg–marquardt algorithm on field programmable gate arrays," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 1, pp. 170–176, 2013.

[13] H.C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R.M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[14] P.D. Lena, K. Nagata, and P.F. Baldi, "Deep spatio-temporal architectures and learning for protein structure prediction," in *Advances in Neural Information Processing Systems*, 2012, pp. 512–520.

[15] T.H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma, "Pcanet: A simple deep learning baseline for image classification?," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5017–5032, 2015.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[17] V. Nair and G.E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[18] J. Nagi, F. Ducatelle, G.A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L.M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 342–347.

[19] C.M. Bishop, *Pattern recognition and machine learning*, Springer-Verlag New York, 2016.