Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 9, August 2021: 4044-4054

Parallel Fused Dense CNN for Identification of Production from Salt Informatics

N. Bala Vignesh[a], Dr. V. Joseph Peter[b]

[a]Research Scholar, Research Department of Computer Science, Kamaraj College, Affiliated to Manonmaniam Sundaranar University.

[b]Associate Professor, Research Department of Computer Science, Kamaraj College, Affiliated to Manonmaniam Sundaranar University.

Abstract

In this research, a novel Parallel Fused Dense Convolutional Neural Network (PFDCNN) is proposed to extract and identify production features automatically from input salt informatics. All input salt information is processed through small kernel based densely connected CNN path or phase I and large kernel based densely connected CNN path or phase II. The fused outcomes of these two phases are processed in a Fully Connected (FC) layer to perform one to one connections between input feature map and output class labels. Here, the softmax activation interprets a single vector input features into a number of salt class probabilities. The proposed output class labels are further compared with original class labels from salt dataset for performing an evaluation. Thus, this PFDCNN algorithm achieves 92% of Accuracy, 95% of Recall, 94% of F1-Score and 94% of Precision values, which is 7% of higher accuracy than the existing deep neural network and machine learning methods.

Keywords: Salt production identification, convolutional neural network (CNN), parallel fused dense CNN, deep learning, artificial intelligence, etc.

1. Introduction

Salt is the natural crystalline mineral that primarily composed with sodium chloride (NaCl) compound. It is used in various manufacturing industrial processes such as paper pulp, plastics and polyvinyl chloride. This salt enhances food taste and brings natural flavour's in foods. The consumption of excessive salt causes hypertension like cardiovascular diseases, chronic kidney disease, osteoporosis, and cancer. The highly used salt are available in twelve different forms namely, pickling salt, Himalayan block salt, smoked salt, black Hawaiian salt, red Hawaiian salt, flake salt, fleur de sel, Celtic grey sea salt, sea salt, Himalayan pink salt, kosher salt, and table salt. The salt product has been harvested through the evaporation of sea water or by salt mines. In sea salt, the sea water is filled with shallow evaporation ponds and then the water disappears by the process of evaporation leaves salt to be harvested. In underground salt, the salt miners cut or drill the rock from underground salt, and then crushed salt pieces are formed by breaking up the rock salt through machines.

An automatic identification of salt production plays a vital role to improve business profit of salt miners. In worldwide, there are five leading salt producers namely, Canada, India, China, Germany and the United States. Indian country is the tired largest producer of salt after china and United States. This research focuses an automated solution for salt production identification from salt panes available at thoothukudi district of Tamilnadu state in India. Initially, the dataset is collected based on the production information along with climate information categories like minimum temperature, maximum temperature, wind direction, wind speed, minimum pressure, maximum pressure, minimum humidity, maximum humidity, rainfall, total area, salt production for total area, salt production per a hundred acre and brain degree.

There are various machine learning [19] and neural network algorithms [20] are best suited for this salt production classification [4]. In this research, a novel Parallel Fused Dense Convolutional Neural Network (PFDCNN) is proposed to extract and identify production features automatically from input salt informatics, which is higher accuracy than the existing outperformed CNN and MLP methods. This work categorizes four subsections: The related works of salt production are detailed in Section 2; the proposed PFDCNN method is elaborated in Section 4 and Section 5 derives their conclusion.

2. Review of Related Studies

The production identification from a salt dataset is very difficult because very limited similar research works are available. There are currently available machine learning algorithms like, Naïve Bayes classifier, Random Forest algorithm, decision trees, K-Nearest Neighbor, MRF (Markov Random Field), Regression tree based methods, Conditional Random Field (CRF), and Support Vector Machine (SVM) are best suited for this salt production classification [13] [14]. These methods expect programmer interaction for identification problems [9]. In recent years, the neural network is an appropriate and effective technique to recognize patterns in real-world classification problems [6].

Sultan et al suggests ANN (Artificial Neural Network) method to extract incremental high-level features [1]. In this method, the raw data is directly classified without the intervention of specific input features in which machines act intelligent as humans to classify and recognize patterns into desired categories [21 [5] [16]. Recently, deep learning or deep neural networks algorithms are most suitable for pattern classification problems [7]. These algorithms avoid hand-crafted traditional feature extraction. Marian et al [15] and Sultan et al [17] develop Deep Learning method to learn automated features from input data. Yunzhi et al [12] developed an intelligent deep learning based CNN approach to identify salt bodies in seismic images and it captures salt features without the need of any feature extraction.

Aleksander used deep learning approach to identify salt deposits on seismic images and also classify non-salt or salt bodies in seismic images [11]. Guntur et al [10] have been developed greenhouse method based salt production, which combines prism greenhouse, threaded filter and geomembrane technology. Hero et al [8] implemented the pristine production making process by adjusting sea water concentration. These existing methods are only used for identification of salt from seismic wave images and for improving salt production. But, the salt production identification from climate information is still challenging problem. This research work developed an automated PFDCNN algorithm for detecting salt production.

3. Proposed Methodology

The proposed architecture concatenates two parallel dense CNN architectures to identify the salt production, which contains three following major phases: workstation and databases; the PFDCNN based salt production identification and performance evaluation.

3.1. Workstation and Database

The complete proposed PFDCNN method is implemented using python 3.8 working under i7-4500U GPU of Intel R-core, 16 GB Random Access Memory (RAM) and 4 GB NVIDIA GPU (Graphic Processor Unit). The salt dataset is collected from eleven salt pans in the thoothukudi district and it contains salt production details along with their climate information. The dataset composed of eight class labels: zero ton production (class 1); 1-500 tons of salt production (class 2); 501-1000 tons of salt production (class 3); 1001-1500 tons of salt production (class 4); 1501-2000 tons of salt production (class 5); 2001-2500 tons of salt productions (class 6); 2501-3000 tons of productions (class 7); 3000-3500 tons of productions (class 8).

3.2. The PFDCNN based Salt Production Identification

The main goal of PFDCNN is to extract salt production features automatically from input salt informatics. Salt production information from the dataset is processed in two different parallel phases: small kernel based densely connected CNN path and large kernel based densely connected CNN path. The outcome of these two feature maps are fused together and processed in a Fully Connected (FC) layer. Here, the softmax activation is performed over the extracted salt features from FC layer outcome to predict output classes. The overall pictorial representation of PFDCNN is visualized in **Figure. 1** and detailed methodology explained below subsections.

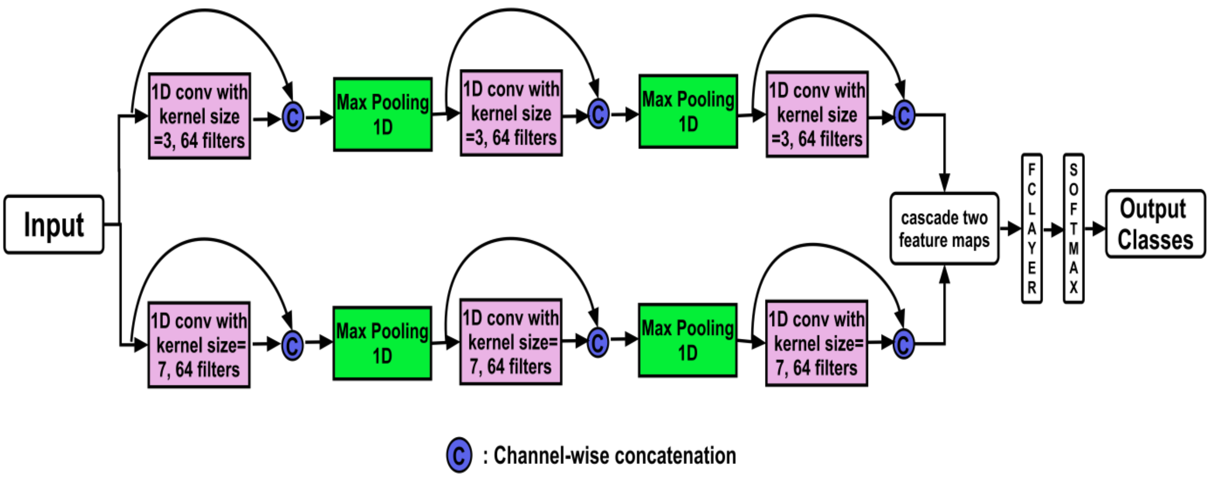

Figure.1. The architecture of PFDCNN.

3.2.1. Phase I – Dense CNN with Small Kernels

The climate and salt production information from a dataset are first processed in phase I, which is small kernel based densely connected CNN path to extract input features. In this, an input data x_i is convolved with weight w_i from small kernel size 3 and bias b_i to form feature map y_{13} is given in Eqn. (1).

$$y_{13} = f\left(\sum_{\boldsymbol{i=1}}^{n} [x_i * w_i] + b_i \right) \quad (1)$$

Where, $i=1,2,3\ldots n$. This feature map y_{13} is concatenated with input x_i to yield concatenated outcome c_{13} is defined in Eqn. (2).

$$c_{13} = y_{13} + x_i \quad (2)$$

This concatenated outcome c_{13} is downsampled using maxpooling to produce p_{13}. Further, one dimensional convolution has been performed over this downsampled data p_{13} to produce feature map y_{23} is given in Eqn. (3).

$$y_{23} = f\left(\sum_{\boldsymbol{i=1}}^{n} [p_{13i} * w_i] + b_i \right) \quad (3)$$

The feature map y_{23} is concatenated with previous max pooling outcome p_{13} to yield concatenated outcome c_{23} is mentioned in Eqn. (4).

$$c_{23} = y_{23} + p_{13} \quad (4)$$

This c_{23} is again downsampled to produce computationally less complexity outcome p_{23} and convolved using a kernel size equal to 3 to produce feature map y_{33} is defined in Eqn. (5).

$$y_{33} = f\left(\sum_{\boldsymbol{i=1}}^{n} [p_{23i} * w_i] + b_i \right) \quad (5)$$

The convolved feature map y_{33} is concatenated with previous max pooling outcome p_{23} to yield concatenated outcome c_{33} *is* mentioned in Eqn. (6).

$$c_{33} = y_{33} + p_{23} \quad (6)$$

This cascaded c_{33} is considered as phase I outcome.

3.2.2. Phase II – Dense CNN with Large Kernels

The input data from dataset is also parallelly processed in phase II or large kernel based densely connected CNN path for extracting salt features automatically. Here, an input data x_i is convolved with weight w_i having large kernel size 7 and bias b_i to form feature map y_{17} is given in Eqn. (7).

$$y_{17} = f\left(\sum_{\boldsymbol{i=1}}^{n} [x_i * w_i] + b_i \right) \quad (7)$$

Where, $i=1,2,3\ldots n$. This feature map y_{17} is concatenated with input x_i to yield concatenated outcome c_{17} is defined in Eqn. (8).

$$c_{17} = y_{17} + x_i \quad (8)$$

This concatenated outcome c_{13} is downsampled using maxpooling to produce p_{17}. Further, one dimensional convolution has been performed over this downsampled data p_{17} to produce feature map y_{27} is given in Eqn. (9).

$$y_{27} = f\,(\textstyle\sum_{i=1}^{n} [p_{17i} * w_i] + b_i\,) \tag{9}$$

The feature map y_{27} is concatenated with previous max pooling outcome p_{17} to yield concatenated outcome c_{27} is mentioned in Eqn. (10).

$$c_{27} = y_{27} + p_{17} \tag{10}$$

This c_{27} is again downsampled to produce less complexity outcome p_{27} and convolved using a large kernel size equal to 7 to produce feature map y_{37} is defined in Eqn. (11).

$$y_{37} = f\,(\textstyle\sum_{i=1}^{n} [p_{27i} * w_i] + b_i\,) \tag{11}$$

The convolved feature map y_{37} is concatenated with previous max pooling outcome p_{27} to yield concatenated outcome c_{37} *is* mentioned in Eqn. (12).

$$c_{37} = y_{37} + p_{27} \tag{12}$$

This cascaded c_{37} is considered as phase II outcome.

3.2.3. Parallel Fusion

The cascaded outcome c_{33} from phase I or small kernel based dense CNN and the cascaded outcome c_{37} from phase II or large kernel based dense CNN are fused together to produce final feature map y_{cas} is defined in Eqn. (13).

$$y_{cas} = c_{33} + c_{37} \tag{13}$$

The final feature map y_{cas} is further processed in fully connected dense layers for predicting class labels of salt production.

3.2.4. Fully Connected Layer

This layer converts final feature map into a single dimension and performs a one to one connections between input feature map and output class labels. The architecture of one to one connection from the FC layer is presented in **Figure.2.** The FC layer is used to convert the final feature map y_{cas} into a single dimension for class prediction. The final parallel fused outcome y_{cas} contains the dimension of 3x386. This dimension is converted as 1x1158 in a fully connected layer. Here, the softmax activation [18] is performed to predict the salt class probabilities.

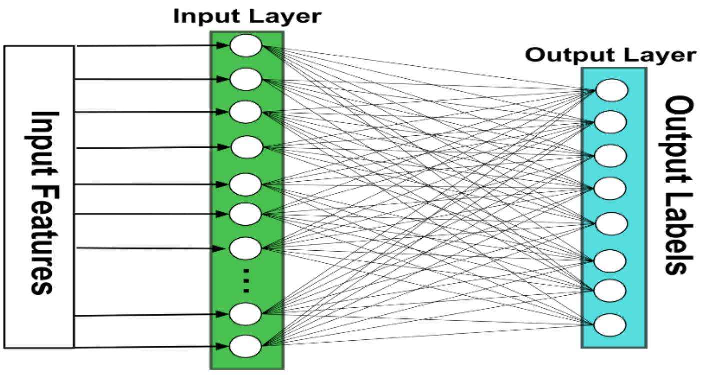

Figure. 2. Fully-connected layer architecture

3.2.5. Softmax Function

This function is mostly used in multi-class identification problems. It interprets a single vector input features into a number of salt class probabilities [3]. The predicted salt classes are compared with original salt classes from the dataset and their loss value is calculated by categorical cross entropy is given in Eqn. (14).

$$Loss = \sum_{\boldsymbol{i=1}}^{n} p(y_i) * t(y_i) \quad (14)$$

Where, *i* is the number of class labels, $p(y_i)$ is the predicted salt classes and $t(y_i)$ is the original targeted class labels. If the computed loss value is high means the same process is performed again and again until to get a desired optimal outcome. In every iterations, the network learnable weights are modified for getting high prediction performance.

3.3. Performance Evaluation

The performance of PFDCNN over eight class labels computed by Accuracy, Recall, F1-Score, and Precision are defined in Eqn. (15) - (18).

$$Accuracy = (\#TP+\#TN)/(\#TP+\#FP+\#FN+\#TN) \quad (15)$$

$$Recall = (\#TP)/(\#TP+\#FN) \quad (16)$$

$$F1\text{-}Score = (\#(2TP))/(\#(2TP)+\#FP+\#FN) \quad (17)$$

$$Precision = (\#TP)/(\#TP+\#FP) \quad (18)$$

where, # denotes the cardinality, #TP is the cardinality of correctly predicted positive salt class labels, #TN is the cardinality of correctly predicted negative salt class labels #FP is the cardinality of wrongly predicted positive salt class labels and #FN is the cardinality of wrongly predicted negative salt class labels.

4. Experimental Results and Discussion

4.1. Proposed PFDCNN Method Effectiveness

All input data from salt dataset is processed parallelly through two different phases. The proposed PFDCNN network contains two different phases. The small kernel based dense CNN layers in phase I is used to learn detailed fine-grained features from an input and large kernel based dense CNN layers from phase II is used to learn spacious features. The outcome extracted features from phase I and phase II are fused together and processed in the FC layer. Here, the softmax activation is performed to predict layers the class labels of salt production. The layer-wise parameters used in PFDCNN layers are detailed in **Table. 1.**

Table. 1. The layer-wise parameters in FCDCNN architecture.

PFDCNN							
Name of Layer	**Size (Input)**	**Outpu t (Size)**	**Parameters**	**Name of Layer**	**Size (Input)**	**Outpu t Size**	**Parameters**

Phase - I				Phase - II			
Convolution Layer-13	12x1	12x64	1D Conv, 64 filters, kernel size=3, Stride =1, ReLU [2]	Convolution Layer-I7	12x1	12x64	1D Conv, 64 filters, kernel size=7, Stride =1, ReLU [22]
Concatenate Layer-13	12x64	12x65	Concatenate input with convolution layer-I3	Concatenate Layer-17	12x64	12x65	Concatenate input with convolution layer-I7
Max Pooling Layer-13	12x65	6x65	Pool size=2	Max Pooling Layer-17	12x65	6x65	Pool size=2
Convolution Layer-23	6x65	6x64	1D Conv, 64 filters, kernel size=3, Stride =1, ReLU	Convolution Layer-27	6x65	6x64	1D Conv, 64 filters, kernel size=7, Stride =1, ReLU
Concatenate Layer-23	6x64	6x129	Concatenate Max Pooling Layer-13 with convolution layer-23	Concatenate Layer-27	6x64	6x129	Concatenate Max Pooling Layer-17 with convolution layer-27
Max Pooling Layer-23	6x129	3x129	Pool size=2	Max Pooling Layer-27	6x129	3x129	Pool size=2
Convolution Layer-33	3x129	3x64	1D Conv, 64 filters, kernel size=3, Stride =1, ReLU	Convolution Layer-37	3x129	3x64	1D Conv, 64 filters, kernel size=7, Stride =1, ReLU
Concatenate Layer-33	3x64	3x193	Concatenate Max Pooling Layer-23 with convolution layer-33	Concatenate Layer-37	3x64	3x193	Concatenate Max Pooling Layer-27 with convolution layer-37
Concatenate Layer-33 and Concatenate Layer-37 are fused together							
Name of Layer : Flatten, Size (Input) : 3x386, Size (Output) : 1x1158.							
Name of Layer : Fully connected, Size (Input) : 1x1158, Size (Output) : 1x8, Activation : Softmax, Number of classes: 8							

4.2. PFDCNN Performance

The predicted salt classes of proposed method are compared with original salt classes from the dataset to perform an evaluation. Performance measures like Accuracy, Recall, F1-score and Precision values of the proposed algorithm are calculated and their results are depicted in **Table.2.**

Table 2. Performance of PFDCNN.

Salt production data				
Method Name	**Accuracy**	**Recall**	**F1-Score**	**Precision**
PFDCNN	92	95	94	94

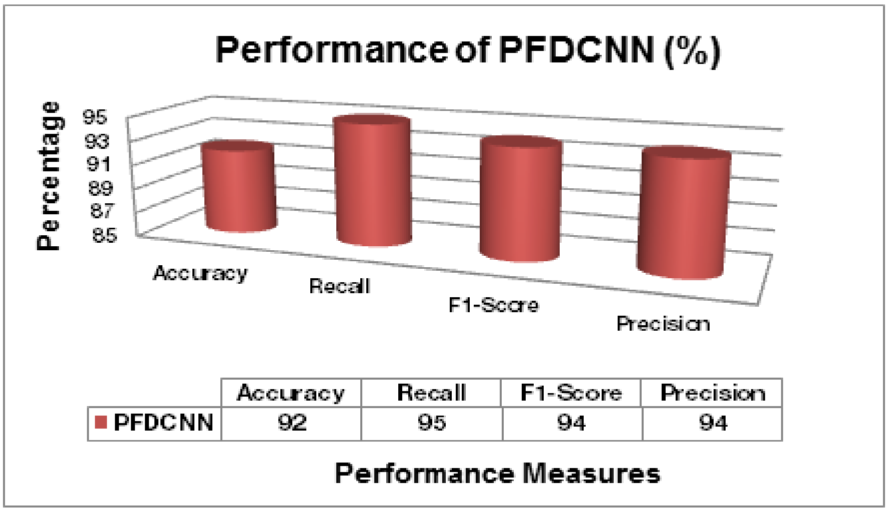

Figure.3. Performance chart of proposed Method.

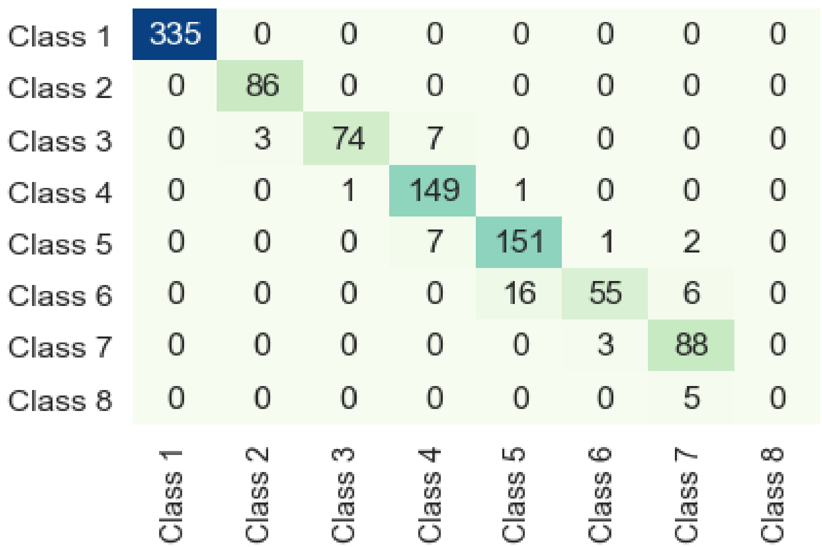

Figure. 4. Confusion matrix of proposed Method.

Thus, this proposed PFDCNN algorithm achieves 92% of Accuracy, 95% of Recall, 94% of F1-Score and 94% of Precision. Figure 3 shows the performance chart and Figure 4 shows the confusion matrix of the PFDCNN method.

4.3. Performance comparison of PFDCNN Architecture

The performance of PFDCNN algorithm is compared with existing methods. The existing Naïve Baye's, K-Nearest Neighbor (KNN), CNN and MLP methods are applied over the salt dataset and their results are detailed in **Table. 3**. Naïve Baye’s is the supervised machine learning classification method based on Baye’s theorem, which achieves accuracy value is 62%, Recall value is 62%, F1-Score value 55%, and Precision value is 64%. The KNN is one of the lazy learning algorithm, which learns data from training set very slowly. This method achieves accuracy value is 72%, Recall value is 72%, F1-Score value 70%, and Precision value is 76%.

Table.3. The performance comparison of PFDCNN method with existing CNN and MLP.

Salt production data					
S. No	**Method Name**	**Accuracy**	**Recall**	**F1-Score**	**Precision**
1	Naïve Bayes	62	62	55	64
2	KNN	72	72	70	76
3	MLP	83	85	83	87
4	CNN	86	91	91	91
5	**Proposed Method**	**92**	**95**	**94**	**94**

To overwhelm KNN and Naïve Baye’s limitations, the MLP has been applied over input salt data to learn features very fast and automatic. This MLP method earns Accuracy value is 83%, Recall value is 85%, F1-Score value 83%, and Precision value is 87%. Further, the deep learning based CNN algorithm is processed over salt input to learn automatic features for improving performance. This method earns an Accuracy value is 86%, Recall value is 91%, F1-Score value 91%, and Precision value is 91%. The results of these four methods are compared with PFDCNN results. Thus proposed PFDCNN achieves Accuracy value is 92%, Recall value is 95%, F1-Score value 94%, and Precision value is 94%. **Figure.5** shows the performance comparison of PFDCNN method with existing four methods. Thus, the proposed algorithm gained higher performance of 7% Accuracy, 4.3% Recall, 3.2% F1-score, and 3.2% Precision than the existing method.

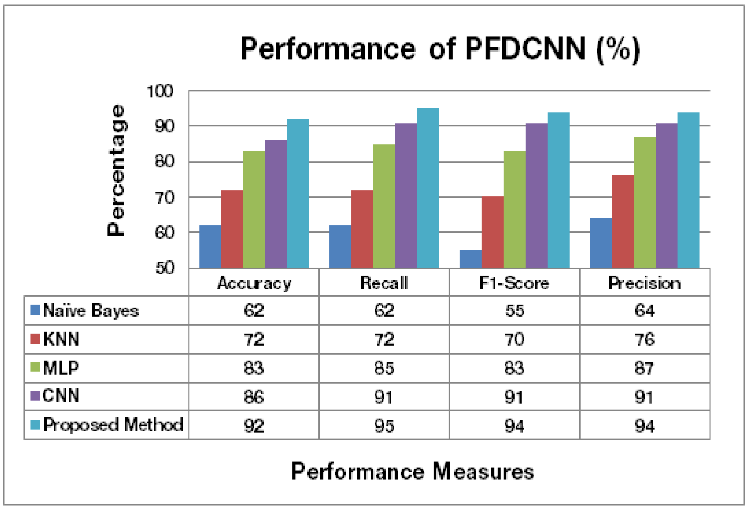

Figure. 5. The figure is center-aligned and the caption of the figure is left-aligned.

5. Conclusion

An automatic identification of salt production plays a vital role to improve business profit of salt miners. This research work developed an automated PFDCNN algorithm for detecting salt production. The predicted salt production labels from proposed algorithm are compared with original salt labels for performing evaluation. These evaluation results of PFDCNN algorithm are then compared with existing Naïve bayes, K-Nearest Neighbor (KNN), CNN and MLP methods. Thus, the proposed algorithm gained 7% higher in Accuracy, 4.3% higher in Recall, 3.2% higher in F1-score, and 3.2% higher in Precision than the existing methods.

References

[1] S. Jégou., M. Drozdzal. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In CVPRW, 117-1183.

[2] Maas, A.L., & Hannun, A.Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In Proc. ICML, 30.

[3] M. Xia, et al. (2020). Weighted Densely Connected Convolutional Networks for Reinforcement Learning. International Journal of Pattern Recognition and Artificial Intelligence, 34, 4, 2052001

[4] Abiodun, O.I et al. (2018). State-of-the-art in artificial neural network applications: A Survey. Heliyon. Elsevier, 4, e00938, doi: 10.1016/j.heliyon.2018

[5] Mounika, M., & Prabhakar, V. V. (2013). Pattern Recognition in Neural Networks. International Journal of Soft Computing and Artificial Intelligence, ISSN: 2321-404X, 1, 2, 24-26.

[6] Yegnanarayana, B. (1994). Artificial Neural Networks for Pattern Recognition. Sadhana, 19, 2, 189-238.

[7] Zhou, L., Li, Q. (2017). Image Classification using biometric pattern recognition with convolutional networks features. Computational Intelligence and Neuroscience, Hindawi, 3792805, 1-13, https://doi.org/10.1155/2017/3792805.

[8] Hero, S., Nur, R. (2015). Development of Traditional Salt Production Process for Improving Product Quantity and Quality in Jepara District, Central Java, Indonesia. *Procedia Environmental Sciences 23, 175 – 178.*

[9] Priyanka, S., & Manavjeet, K. (2013). Classification in Pattern Recognition: A Review. International Journal of Advanced Research in Computer Science and Software Engineering, 3,4, 298-306.

[10] Guntur, G., & Jaziri, A. A. (2018). Development of salt production technology using prism greenhouse method. IOP Conf. Series: Earth and Environmental Science, 106, 012082, 1-6, doi :10.1088/1755-1315/106/1/012082.

[11] Aleksandar, M. (2020). Identification of Salt Deposits on Seismic Images Using Deep Learning Method for Semantic Segmentation. ISPRS Int. J. Geo-Inf, 9, 24, 1-16, doi:10.3390/ijgi9010024.

[12] Yungi, S., & Xinming, Wu. (2018). Automatic salt-body classification using a deep convolutional neural network. SEG International Exposition and 88th Annual Meeting, 10.1190/segam2018-2997304.1, 1971-1975.

[13] Mayank. J., & Rakesh Kumar, B. (2018). A Review on Pattern Classification through Variants of Artificial Neural Network Algorithm. INTERNATIONAL JOURNAL OF ADVANCED STUDIES OF SCIENTIFIC RESEARCH (IJASSR), ISSN 2460 4010, 164-170.

[14] Jeong-Hee Lee., Jongseok, K. (2020). Pattern Detection Model Using a Deep Learning Algorithm for Power Data Analysis in Abnormal Conditions. Electronics, 9, 1140, 1-18, doi:10.3390/electronics9071140.

[15] Mariam, M. S. (2019). Classification of Pattern Recognition Techniques Used Deep Learning and Machine Learning. International Journal of Computer Science Trends and Technology (IJCST), 7, 3, ISSN: 2347-8578, 165-173.

[16] Devyani, B., Poonam, S. (2019). Review on Reliable Pattern Recognition with Machine Learning Techniques. Fuzzy Information and Engineering, 10:3, 362-377, DOI: 10.1080/16168658.2019.1611030.

[17] Sultan, A., & Simon, C. (2013). Towards Machine Learning Based Design Pattern Recognition, 244-251 978-1-4799-1568-2/13/$31.00 c 2013 IEEE.

[18] Qiuyu, Z., & Zikuang, H. (2020). Improving Classification Performance of Softmax Loss Function based on Scalable Batch Normalization. Applied Sciences, 10, 2950, 1-8.

[19] Osisanwo, F. Y. & Akinsola, J. E. T. (2017). Supervised Machine Learning Algorithms: Classification and Comparison, International Journal of Computer Trends and Technology (IJCTT), 48, 3.
[20] Xing, H. et al (2016). Technical Survey Deep Learning. International Journal of Semantic Computing, 10, 3, 417–439.
[21] Alzubaidi *et al.* (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8, 53, 1-74.
[22] Siddharth, S. & Simone, S. (2020). Activation Functions in Neural Networks, International Journal of Engineering Applied Sciences and Technology, 4, 12, 310-316, 2020.