

Geochemical exploration numerical modeling using convolutional neural network (Case study: Gonabad region)

M. Tahmooresi¹, B. Babaei¹, S. Dehghan^{1*}

1- Dept. of Mining , Mahallat Branch, Islamic Azad University, Mahallat, Iran.

* Corresponding Author: *dr.saeed_dehghan@yahoo.com*
(Received: January 2022, Accepted: June 2022)

Keywords

Numerical modeling
Geochemical exploration
Convolutional neural network
Khorasan Razavi
Iran

Abstract

Modeling of mineral potentials to identify promising districts in large exploration regions for detailed exploration operations is one of the main stages of exploration. In this research, a new approach based on a convolutional neural network is proposed for geochemical numerical modeling and mineral potential exploration. In the first step, in order to create the intelligent geochemical exploration modeling, the codes of the convolutional neural network algorithm and its evaluation indicators are programmed in MATLAB environment. After preprocessing of stream sediment geochemical data, including identification of outliers, estimation of censored data, and data normalization and standardization, factor analysis is performed in order to reduce the dimension of the study space, identify the main variables that control the concentration of deposit elements, and define factors. The variables used in modeling are the result of factor analysis of stream sediment data. The average accuracy of the mentioned modeling is obtained as 96%. In the second step, using the geostatistical method (universal kriging), the average accuracy of estimation points via ArcGIS software is calculated to be 75%. At the end of this study, the performances of numerical modeling using convolutional neural network and universal kriging as well as the support vector machine and its integration with the continuous genetic algorithm, which was studied in the previous article, are compared. The evaluation results show that machine learning algorithms are more accurate in identifying promising mineral districts compared to traditional methods. It is important to note that the results of this study are in good agreement with the results of field studies and mineralized sampling.

1. INTRODUCTION

Deep learning has entered dramatically into many applications of artificial intelligence (AI), including object and speech recognition, noise

reduction of speech signals, machine translation, and emotion analysis, and has

provided the solution to many complex AI problems. Table 1 shows the history of the development of deep learning and its various applications [1].

In recent years, in-depth learning in the fields of mining engineering and earth sciences has also entered, the following are some of the studies conducted in these fields:

Table 1. Important milestones in the development of deep learning and its various applications [1].

Year	Contributor	Contribution
300 BC	Aristotle	Introduced associationism, started the history of human's attempt to understand the brain.
1873	Alexander Bain	Introduced neural groupings as the earliest neural network models, inspired Hebbian learning rule.
1943	McCulloch & Pitts	Introduced the MCP model, which is considered the ancestor of the artificial neural model.
1949	Donald Hebb	Considered the father of neural networks, introduced the Hebbian learning rule, which lays the foundation of modern neural networks.
1958	Frank Rosenblatt	Introduced the first perceptron, which highly resembles the modern perceptron.
1974	Paul Werbos	Introduced backpropagation
1980	Teuvo Kohonen	Introduced self-organizing map
1980	Kunihiko Fukushima	Introduced neocogitron, which inspired convolutional neural network
1982	John Hopfield	Introduced Hopfield network
1985	Hilton & Sejnowski	Introduced Boltzmann machine
1986	Paul Smolensky	Introduced harmonium, which is later known as the restricted Boltzmann machine
1986	Michael I. Jordan	Defined and introduced recurrent neural network
1990	Yann LeCun	Introduced LeNet, showed the possibility of deep neural networks in practice
1997	Schuster & Paliwal	Introduced bidirectional recurrent neural network
1997	Hochreiter & Schmidhuber	Introduced LSTM, solved the problem of vanishing gradient in recurrent neural networks
2006	Geoffrey Hinton	Introduced deep belief networks, also introduced layer-wise pretraining technique, opened current deep learning era.
2009	Salakhutdinov & Hinton	Introduced deep Boltzmann machines
2012	Geoffrey Hinton	Introduced dropout, an efficient way of training neural networks

Palafox et al. (2017) stated the large volume of high-resolution images that have been obtained by the Mars Reconnaissance Orbiter, has opened a new frontier for the automatic detection of earth shapes on the surface of Mars. In this study, convolutional neural networks (CNNs) have been used to identify a wide range of earth shapes on the surface of Mars [2]. Pires de Lima et al. (2019) emphasized that machine learning methods, including CNN, are not a substitute for the expertise of geologists, and specialists must provide the labels required to train the algorithms, in other words, machine learning methods are a means of applying the expertise of skilled geologists to large volumes of data [3]. Baraboshkin et al. (2019) used CNNs to reduce the time required in order to accurately describe rocks. They used color distribution analysis and feature extraction. In the mentioned study, the famous neural network architectures (Alexnet, VGG, Googlenet, Resnet) were used. The accuracy of Googlenet architecture was assessed at about 95% of validation data, which examined 50 meters of drilling core in terms of rock type per minute [4]. Obianuju Lynda (2019) used CNN to classify satellite images. They started using this method, and the speed and accuracy of satellite image classification and subsequently production of geological or topographic maps increase [5].

Maitre et al. (2019) applied CNN to automatically identify mineral samples among suspended particles of natural sand. They stated that the results are very promising for the samples used in the study area, but for other areas, this identification has not been done correctly, which can be due to differences in mineralogy or taking pictures of the samples from different light sources. They concluded that it is essential to study more samples in order to extract the features [6]. Hassan et al. (2019) stated land use mapping by applying remote sensing data, due to similar spectra of different objects, is a challenge. For this purpose, CNN was employed to classify hyperspectral data [7]. Liu et al. (2020) used image segmentations to identify the ore particle size distribution. They expressed, however, that the adhesion of ore particles and dark areas in the explosion and conveyor belt images usually leads to low accuracy [8]. Fan et al. (2020) noted that the detection and classification of rocks are important issues in the prospecting phase; detection based on the study of thin sections, in addition to time and cost consumption, cannot guarantee accuracy. Also, it cannot be useful in field studies. In the mentioned study, a smart mobile phone's camera to take pictures of rocks and the linked software based on CNN were used to identify various lithologies with high accuracy and speed in field

studies [9]. Si et al. (2020) applied CNN for intelligent separation between coal and bedrock on the coal face. In the mentioned study, in order to prevent overfitting, three methods of “dropout”, “weight adjustment” and “batch normalization” have been used. Also “data augmentation” function has been applied to enhance visual data to improve neural network performance [10]. Razak and Jafarpour (2020) used CNN to aggregate subsurface flow data, identify geological scenarios and reduce uncertainty [11]. Madhuanand et al. (2021) used deep CNN to determine the surface of coal mines on Sentinel-2 images. They noted, that coal mines are important sources of methane emissions, the second and most important greenhouse gas, and monitoring the emission of methane from coal mining requires determining the exact location of coal mines. The purpose of the aforesaid paper is to determine the surface of coal mines from satellite images using deep learning methods. For this purpose, different pre-trained architectures of CNN (VGG, Resonant, and Dancenet) have been applied also to learn CNNs, a list of well-known coal mines from different countries, and a set of 13-band Sentinel-2 image segmentations entitled “Coal Mines” and “Non-Coal Mines” were prepared [12].

As can be seen in all the above studies, the input data into CNN are segmentations of images. However, in the studies conducted by Pryshliak et al. (2018) and Sharma et al. (2019), the data used were numeric data, but they applied the processes of normalization and conversion of numerical data into image data in order to prepare data to enter into CNN [13, 14].

In this study, in order to compare the accuracy of conventional and novel methods in classifying samples and identifying mineral potential zones, CNN (from machine learning methods) and geostatistics (from traditional methods) are used. CNN codes are programmed in a MATLAB environment. ArcMap software is used to accomplish the geostatistical methods. According to the data processing in the previous study (create stream maps, perform factor analysis, and establish fuzzy factor score maps), in this study, by mentioning a summary of the previous article, only its results are used.

2. METHODOLOGY AND APPROACHES

2.1. Methodology

CNN architecture includes convolution, pooling, fully connected, softmax, and

classification layers. Also, in order to improve training, techniques such as using Rectified Linear Unit (ReLU) activation function, batch normalization, and dropout are considered. In convolution operations, important features of an image or a signal (for example edges) are automatically extracted according to the kernel used. Due to the use of a large number of filters in each convolution layer and the high computational volume, the pooling layer is applied to reduce the dimensions of the feature map produced by the convolution layer [15]. Up to this stage, feature extraction is performed. According to the type of data employed in this research, the data are entered into a fully connected layer. A fully connected layer identifies and classifies objects in the image [16]. To prevent overfitting, relevant functions are used in this layer. The final layer of fully connected in the CNN architecture has output neurons equal to the number of classes [15]. The ReLU nonlinear activation function is applied in between fully connected layers, the derivative of the ReLU function is 0 (If the value is negative) and 1 (if the value is positive). The output of ReLU does not have a maximum value (It is not saturated) and this helps gradient descent. The function returns 0 if the input is negative, but for any positive input, it returns that value back. After the fully connected layer is placed softmax layer, the output is probability values ranging from 0 to 1 and equal to the number of classes. In the classification layer, the highest probability of output of the Softmax layer is set to one and about other categories to zero [17].

The training process in the neural network is step-by-step. In this method, the data are entered into the network and after passing through different layers of the network by applying weights and nonlinear functions to the data, the result that is obtained at the end of the network may differ from the desired result, therefore, the weights in this algorithm should be corrected by the error back propagation method. One of the common algorithms employed to train deep networks is stochastic gradient descent (SGD). The slope of a function can be considered as the gradient. The gradient is used to gauge the change in the weights with respect to the inputs. The higher the gradient, the faster the model can learn from the inputs, and when the gradient eventually reaches 0, the learning process gradually stops.

Sometimes, during back propagation, the weight change may be so small that the model does not learn anything new or learns at a negligible rate. This is a vanishing gradient problem. Another problem is the exploding

gradient where the weight assigned is very large as the derivative with respect to the input is huge.

Thus, the network may remain as one of the minimum local points at the error level, or the network training speed may be too slow and the result will not be accurate enough, hence many network training employ techniques to improve the training process as follows:

One of the techniques is batch normalization, in this method, normalization is performed in the middle layers of the network. In other words, the output of the layers is normalized again. Another method is to add dropout layers, in this way a number of neurons are randomly removed from the training process at each repetition or training step, so that, neurons are trained independently, as a result of which the network can discover the powerful features [18].

To evaluate the performance of machine learning algorithms, indicators such as accuracy, precision, recall, F1_Score, and AUC (the area under the receiver operating characteristic (ROC) curve) are generally used. Obviously, increasing the values of the mentioned indicators indicates the ability and accuracy of the designed model. The following parameters are used to define the indicators:

TP (True Positives): The number of data in Class (1) that is correctly identified.

FP (False Positives): The number of data that the classifier incorrectly places in class (1).

FN (False Negatives): The number of data that the classifier incorrectly places in class (2).

TN (True Negatives): The number of data in Class (2) that has been correctly identified.

Accuracy is the ratio between the correctly classified samples to the total number of samples as follows:

$$Accuracy = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision is the same classification accuracy that is defined in a particular class.

$$Precision(Class1) = \frac{TP}{TP + FP} \quad (2)$$

$$Precision(Class2) = \frac{TN}{TN + FN} \quad (3)$$

Another factor is the recall or sensitivity, which is the ratio of the number of data correctly classified in class (1) to the total data actually in class (1) and in the case of class (2), this factor is called specificity.

$$Recall(Class1) = \frac{TP}{TP + FN} \quad (4)$$

$$Recall(Class2) = \frac{TN}{TN + FN} \quad (5)$$

Another factor used for evaluation is F1_Score, which is obtained from the following equation:

$$F_1\text{-Score} = \frac{2 \times \text{precision} \times \text{Recall}}{(\text{precision} + \text{Recall})} \quad (6)$$

The classifier is used to perform the prediction operation. One of the predict function outputs is label_predict which shows the class of each instance and is used for the confusion matrix plot. (a confusion matrix, that is known as an error matrix, each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa) and the second output is the score_label matrix that the probability of placement each instance in the corresponding class shows [19, 20]. To evaluate the CNN classifier performance (i.e. to determine the classifier's ability to classify positive images as positive and negative images as negative), the area under the ROC curve is used (AUC) and its value is between zero and one. In general, the rule of thumb for interpreting the AUC value is as follows [21, 22]:

$$AUC = \frac{1}{2} \left(\frac{TP}{TN + FN} + \frac{TN}{TN + FP} \right) \quad (7)$$

$$AUC=0.5 \text{ No discrimination} \quad (8)$$

$$0.6 \geq AUC > 0.5 \text{ Poor discrimination} \quad (9)$$

$$0.7 \geq AUC > 0.6 \text{ Acceptable discrimination} \quad (10)$$

$$0.8 \geq AUC > 0.7 \text{ Excellent discrimination} \quad (11)$$

$$AUC > 0.9 \text{ Outstanding discrimination} \quad (12)$$

In this study, in addition to using CNN, the geostatistical method (kriging) is used to predict values in any geographical location. The kriging method fits a mathematical function to a certain number of points, or all points in a given radius, to determine the output value for each location. The kriging method is a multi-step process that includes data statistical analysis, variogram modeling, and creating a kriging surface. Kriging surfaces are the sum of an estimate for the trend surface $\mu(s)$ plus the kriging prediction for the residual process $\delta(s)$ [23]. This function is more appropriate when there is a spatial correlation or directional bias in the data. Experimental

semivariogram $\gamma(h)$ is computed as half the average squared difference among the components of data pairs, according to the following equation (Isaacs, E. and Srivastava, M.R, 1989):

$$\gamma(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} [Z(x) - Z(x+h)]^2 \quad (13)$$

where $N(h)$ is the number of pairs of data locations a vector h apart and $Z(x)$ is the measurement at point x . Before using a semivariogram in estimation, it is necessary to fit a suitable mathematical model. In this study, the spherical model has the best performance and is defined as follows:

$$\gamma(h) = C_0 + C \left(\frac{3h}{2a} - \frac{1}{2} \frac{h^3}{a^3} \right), \quad 0 < h \leq a \quad (14)$$

$$\gamma(h) = C_0 + C, \quad h > a \quad (15)$$

$$\gamma(h0) = 0, \quad h = 0 \quad (16)$$

where C_0 is the nugget effect, a is the radius of influence (or range), and C is the difference between the nugget effect and sill.

The result of geostatistical interpolation for estimating a variable at a non-sampled location X_p is $Z(x_p)$, which is defined as a linear combination of observational values in the neighborhood as follows:

$$Z^*(x_p) = \sum_{i=1}^n \lambda_i Z(x_i), \quad \sum_{i=1}^n \lambda_{i=1} \quad (17)$$

In the above equation, λ_i are kriging weights for observation points [24]. Different kriging methods are used to interpolate point data. Kriging is a random technique similar to inverse distance weighted in which a linear combination of weights at known points is used to predict unknown points. The kriging system is expressed in terms of covariances, which are generally obtained by estimating and modeling a semivariogram [25].

There are three fundamental types of kriging with the assumption of the global mean (μ_z) of the underlying Gaussian process $Z(x)$. They are simple, ordinary, and universal kriging. Simple kriging considers a known and constant mean, ordinary kriging assumes a global mean that is constant but unknown. Universal kriging supposes a variable global mean [26].

Universal kriging is considered as the following model:

$$Z(s) = \mu(s) + \varepsilon(s) \quad (18)$$

As can be observed in Fig. 1, the observed data are considered as solid circles and defined trend $\mu(s)$ as a second-order polynomial. From the polynomial difference with the original data, error $\varepsilon(s)$ is obtained. The mean of the errors will be zero. In addition, a linear process or any number of other functions can be thought of as $\mu(s)$ or universal kriging.

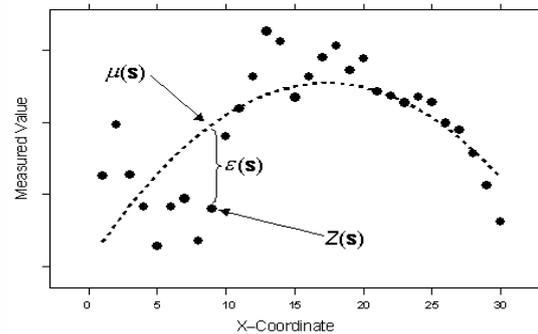


Figure 1. Universal kriging model [27].

As can be seen, one of the drawbacks of the geostatistical method is the definition of the spatial relationship between samples with a mathematical model called variogram. However, the mentioned method cannot always be used due to some assumptions, nature of real data, and the multiplicity of influential variables, hence the methods should be used that extract data features and learn them to identify the relationship between data. For this purpose, intelligent algorithms have been also used in this research.

2. 2. Study Area

Due to the possibility of collecting data, both numerical and visual data, the Gonabad region in this study has been selected. This region with an area of 995 square kilometers includes two zones Helali (in the northwest) and Kalateh Ahani (in the southeast) in Khorasan Razavi province (Fig. 2). The age of the rock units in the study area is related to Paleozoic to Cenozoic periods. Major rock units in the Paleozoic include limestone and quartzite sandstone. In the Mesozoic period, the Shemshak formation has the most widespread in the region. The main members of this formation include volcanic units with intermediate layers, sandstone, slate, and black shale. In terms of structural geology, folds, faults, and dikes are abundant in the mentioned formation, which in some areas has caused alteration and

mineralization. During the Cenozoic, the main rock units include basalt, andesite, rhyolite, and dacite, which are the result of magma penetration

with acidic, intermediate, and mafic compositions and are sometimes covered by Quaternary layers including clay, marl, silt, and gypsum.

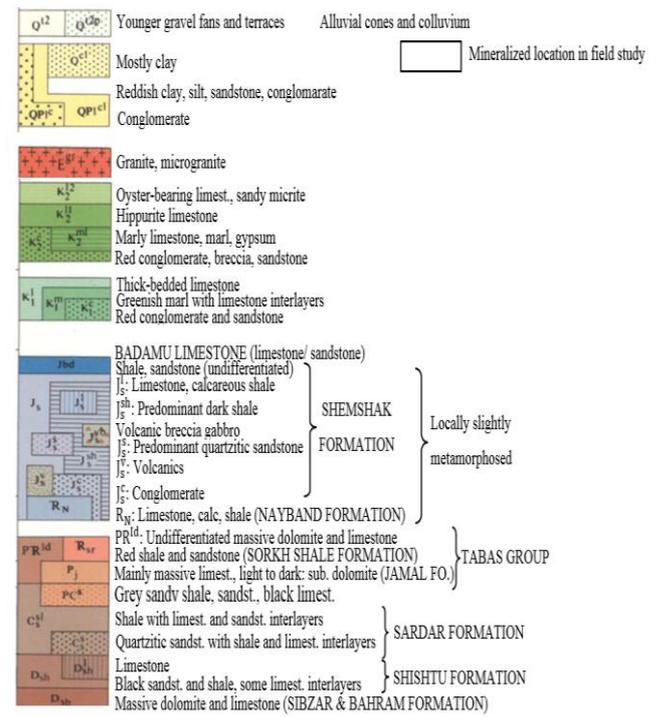
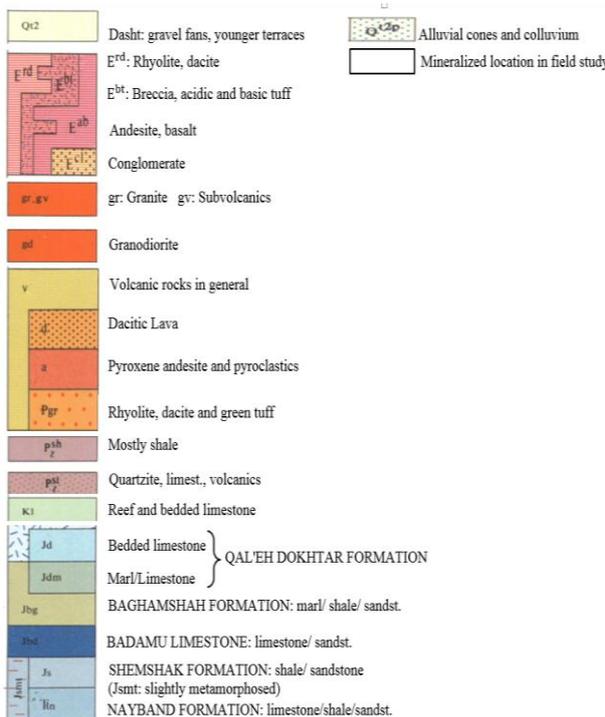
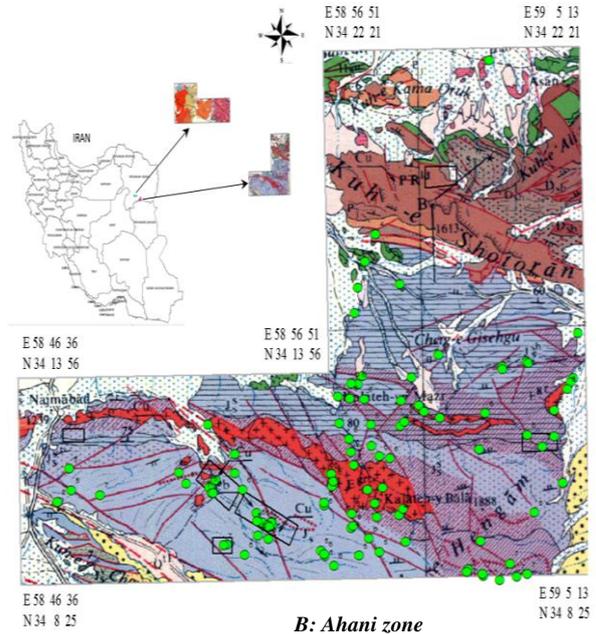
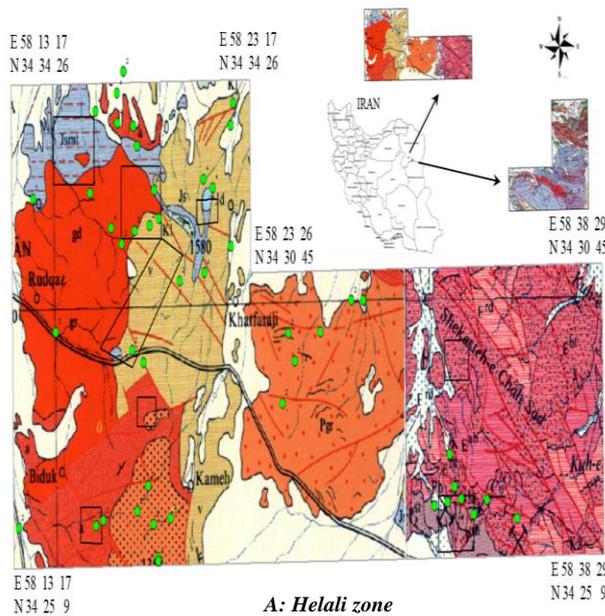


Figure 2. General schematic of anomalous samples via factor analysis on the geological map (1:250000) in the Gonabad area (A: Helali zone, B: Ahani Zone).

2.3. Implementing Method

In this study, 580 samples of stream sediments have been studied, of which 340 samples are related to the Helali zone and the rest is related to the Ahani zone. ICP_OES analysis for determination of 35 elements content in various samples is applied. In the preprocessing phase, the following steps are performed, identification of outliers using the Tukey box, estimation of censored data using Cohen's method, data normalization, and standardization by calculating of enrichment index. Then, factor analysis is used to identify the main variables that control the concentration of deposit elements and reduce the dimensionality of a set of data. In performing the mentioned analysis, all variables (elements) enter into the analysis process, and eventually, the number of factors reduces to seven factors with special values of more than one. According to the changes in factor scores at the sampling locations, the fuzzy factor score [28] of each sample is calculated so that data values fall between zero and one. Using the fuzzy weight of the samples, the probability maps of the mineral potential obtained from the factor analysis are drawn and anomalous areas are determined. Fig. 2 shows the fuzzy factor scores maps related to the distribution of anomalous samples of each factor that has been superposed on the geological map.

In order to perform intelligent exploration modeling in classifying samples and determining geochemical anomaly areas, the codes of the CNN algorithm are written in the MATLAB environment which the program code is present in Appendix.

Definition of the input data file is the most vital part. The number of rows and columns is equal to the number of samples and attributes (respectively) and the last column includes the targets. The features used, are a number of factors, and one feature is related to the lack of mineralization. The data applied are fuzzy factor scores and the target is identified anomalies related to each factor. In setting the input file, the number of anomalous and non-anomalous samples are selected in equal numbers. Also, 75%, 10%, and 15% of data are training, validation, and test data respectively. Data are selected with random permutations using the randperm function. In this research, CNN architecture includes fully connected, softmax, and classification layers and the ReLU activation function. CNN training is also performed by the SGD method. In order to improve network training in MATLAB, the functions of normalizing

symmetrically and dropout are used. According to the extraction of seven factors in factor analysis and one feature related to the lack of mineralization, eight categories are defined. Determining the proper number of hidden neurons is based on the trial rule that is defined with minimal error and highest accuracy, also size of the input and output layers should be considered. To evaluate the ability and accuracy of modeling using CNN, indicators such as accuracy, precision, recall, F1_Score, and AUC are employed. The relevant codes are programmed in the MATLAB environment.

Also, to compare the accuracy of intelligent exploration modeling with the geostatistical method, the universal kriging method is applied. The first step in using the kriging method is to check out the normality of the data. For this reason, the logarithm of the data is considered. The kriging outputs are predicted values and error estimation values for each sample which are applied to evaluate the performance of the method.

3. RESULTS AND CONCLUSIONS

3.1. Results And Discussion

Due to the large exploration region and high sampling density, it is not possible to determine the promising zones using traditional methods in terms of time and cost consumption, hence in this research, intelligent exploration modeling is employed. The evaluation results of geochemical numerical modeling using the CNN algorithm via MATLAB software are indicated in Table 2.

Table 2. Results of numerical modeling by CNN algorithm (MATLAB output)

```
*****
Results For Simple Create: Test Data
Accuracy: 96.1039
Precision: [100 90 100 91.6667 100 100 100 95.4545]
Recall: [100 100 100 91.6667 100 100 66.6667 95.4545]
F1_Score: [100 94.7368 100 91.6667 100 100 80 95.4545]
AUC: [1 1 1 0.9987 1 1 1 0.9983]
*****
```

Table 3 shows the confusion matrix obtained from numerical modeling of CNN on geochemical data in the study area. As can be seen, out of the total samples, 77 samples have been selected for test data, of which 74 samples have been correctly identified by the algorithm in the relevant classes.

Table 3. Confusion matrix obtained from numerical modeling of CNN on geochemical data.

		CM for Simple Create: Test Data							
Output Class		1	2	3	4	5	6	7	8
		Target Class	1	2	3	4	5	6	7
1		4 5.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
2		0 0.0%	9 11.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 1.3%
3		0 0.0%	0 0.0%	4 5.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
4		0 0.0%	0 0.0%	0 0.0%	11 14.3%	0 0.0%	0 0.0%	1 1.3%	0 0.0%
5		0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 9.1%	0 0.0%	0 0.0%	0 0.0%
6		0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	16 20.8%	0 0.0%	0 0.0%
7		0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 2.6%	0 0.0%
8		0 0.0%	0 0.0%	0 0.0%	1 1.3%	0 0.0%	0 0.0%	0 0.0%	21 27.3%
		100% 0.0%	100% 0.0%	100% 0.0%	91.7% 8.3%	100% 0.0%	100% 33.3%	66.7% 4.5%	95.5% 3.9%

Fig. 3 indicates the ROC curve achieved from CNN modeling. The curve is located above the bisector line of the diagram. The average AUC is 99%, which indicates outstanding discrimination of the modeling algorithm.

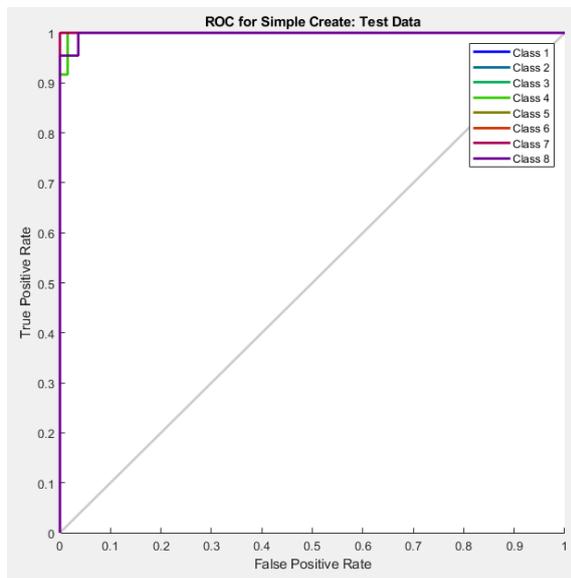
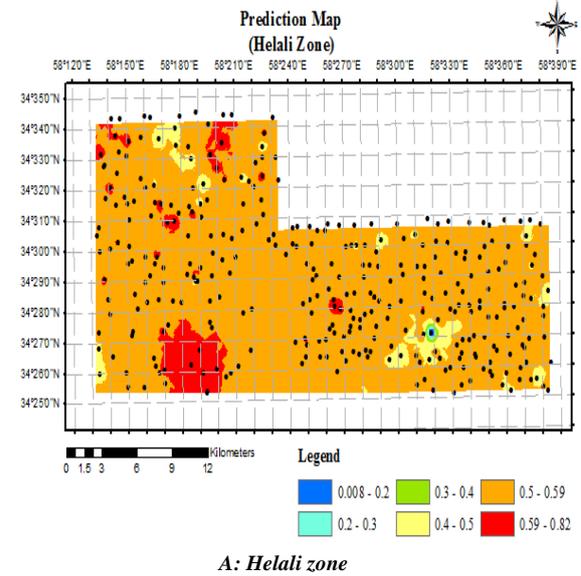


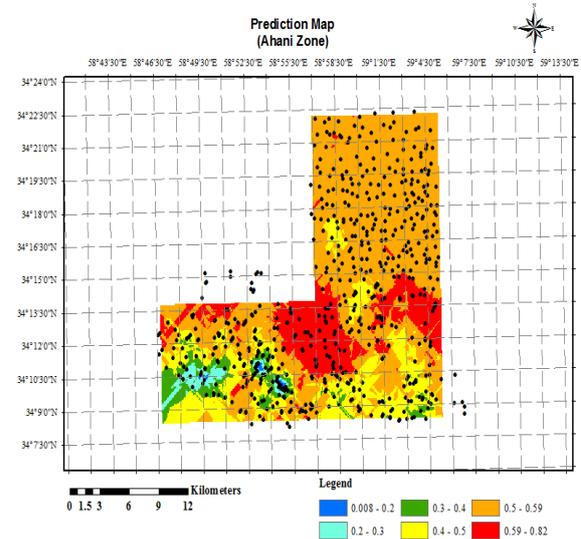
Figure 3. ROC curve achieved from numerical modeling of CNN on geochemical data.

Exploration studies based on geostatistics are also used to compare the accuracy of the methods. According to be variable global mean, universal kriging is utilized. Fig. 4 shows fuzzy factor scores prediction maps using the universal kriging method. The aim of the applying universal kriging method is to predict $Z(x)$ at an unsampled area. Using ArcMap software, prediction errors can

easily be accessed for the layer created by the kriging method for each of the samples. The mean error can be generalized to identify all samples, both anomalous and non-anomalous samples. Thus, the average accuracy of this method in identifying the class of samples can be approximated.



A: Helali zone

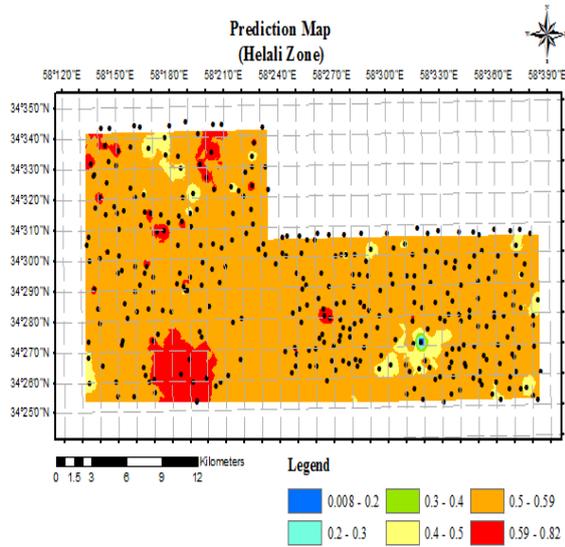


B: Ahani zone

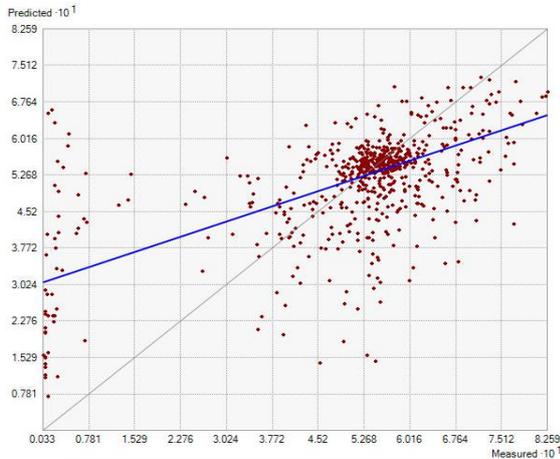
Figure 4. Prediction map by kriging (A: Helali zone, B: Ahani Zone).

A scatter diagram is the easiest way to graphically represent the relationship between two quantitative variables which is as an output of the kriging method implementation. The highest accuracy of estimation is when the fitted line to data has an angle of 45° or near to it. After

determining the correlation among the variables, the behavior of unknown variables (measured or predicted) can be easily predicted by other variables. Fig. 5 shows the scatter diagrams of the predicted vs. measured values. Prediction means the model response. As can be seen, the y variable (predicted) tends to increase as the x variable (measured) increases. Hence, there is a positive correlation among the variables.



A: Helali zone



B: Ahani zone

Figure 5. Scatter diagram of the predicted vs. measured values. (A: Helali zone, B: Ahani Zone).

As can be seen in Table 4, the average accuracy of numerical modeling via the CNN algorithm in identifying specimens class on test data is calculated as 96%. According to previous studies, the average accuracy of the support vector machine algorithm is 98% [29] and the average accuracy of the geostatistical method is 75%.

Table 4. Comparison accuracy of the geostatistical method, support vector machine algorithm, and its integration with genetic algorithm and convolution neural network.

Average accuracy of numerical modeling via CNN in identifying samples class (Test data)	Average accuracy of support vector machine and its integration with genetic algorithm in identifying samples class (Test data)	Average accuracy of geostatistical method in identifying samples class
96%	98%	75%

3. 2. Conclusions

By superimposing the geological map and the factor analysis map (Figs. 2), it can be seen that anomalies are located in the Helali zone, mainly in bodies of igneous rocks (andesite, basalt, rhyolite, dacite, green tuff, granite, granodiorite, and volcanic rocks), in Ahani zone in granite and microgranite and in Shemshak formation in shale, sandstone, limestone, black shale, and quartzite sandstone, which can be due to numerous dykes and faults in the region and intrusion of magmatic fluids into the fractures. In terms of mineralization, there is a possibility of significant reserves in the black shales that are geochemical dams and are caused by the accumulation of minerals and play a key role in the prospecting and exploration of mineral deposits.

It is noteworthy that the location of anomalous samples from factor analysis has a good overlap with the location of mineralized sampling in field studies (Fig. 2).

This article is the results section of the research related to Miss Mandana Tahmooreesi's dissertation that has been done in the Mahallat Branch, Islamic Azad University [30].

REFERENCES

[1] Haohan Wang, and Bhiksha Raj, "On the Origin of Deep Learning," ArXiv: 1702.07800v4 [cs.LG] 3 Mar 2017.

[2] Leon F. Palafox, Christopher W. Hamilton, Stephen P. Scheidt, and Alexander M. Alvarez, "Automated detection of geological landforms on Mars using Convolutional Neural Networks," Computers & Geosciences, vol. 101, pp. 48-56, 2017.

[3] Rafael Pires de Lima, Alicia Bonar, David Duarte Coronado, Kurt Marfurt, and Charles Nicholson, "Deep convolutional neural networks

as a geological image classification tool," *The Sedimentary Record*, pp. 4-9, 2019.

[4] E.E. Baraboshkin, L.S. Ismailova, D.M. Orlov, E.A. Zhukovskaya, G.A. Kalmykov, O.V. Khotylev, E.Yu. Baraboshkin, and D.A. Koroteev, "Deep Convolutions for In-Depth Automated Rock Typing," arXiv:1909.10227v3 [cs.CV] 27 Sep 2019 .

[5] Nzurumike Obianuju Lynda, "Systematic survey of convolutional neural network in satellite image classification for geological mapping," IEEE, DOI: 10.1109/ICE_CCO48375.2019.9043261 [2019 15th International Conference on Electronics, Computer and Computation (ICECCO)], 2019.

[6] Julien Maitre, Kévin Bouchard, and L. Paul Bédard, "Mineral grains recognition using computer vision and machine learning," *Computers & Geosciences*, vol.130, pp. 84-93, 2019.

[7] Hayder Hasan, Helmi Z.M.Shafri, and Mohammed Habshi, "A comparison between support vector machine (SVM) and convolutional neural network (CNN) models for hyperspectral image classification," *Earth and Environmental Science*, vol.357, pp. 1-10, 2019.

[8] Xiaobo Liu, Yuwei Zhang, Hongdi Jing, Liancheng Wang and Sheng Zhao, "Ore image segmentation method using U-Net and Res_Unet convolutional networks," *Royal society of chemistry*, vol.10, pp. 9396-9406, 2020.

[9] Guangpeng Fan, Feixiang Chen, Danyu Chen, Yan Li, and Yanqi Dong, "A deep learning model for quick and accurate rock," *Hindawi Recognition with Smartphones*, vol.2020, pp.1-14, 2020.

[10] Lei Si, Xiangxiang Xiong, Zhongbin Wang, and Chao Tan, "A deep convolutional neural network model for intelligent discrimination between coal and rocks in coal mining face," *Hindawi Mathematical Problems in Engineering*, Vol. 2020, pp.1-12, 2020 .

[11] Syamil Mohd Razak, and Behnam Jafarpour, "Convolutional neural networks (CNN) for feature-based model calibration under uncertain geologic scenarios," *Computational Geosciences*, vol.24, pp.1625-1649, 2020.

[12] L. Madhuanand, P. Sadavarte, A.J.H. Visschedijk, H.A.C. Denier Van Der Gon, I. Aben and

F.B. Osei, "Deep convolutional neural networks for surface coal mines determination from sentinel-2 images," *European Journal of Remote Sensing*, vol. 54, pp. 296-309, 2021.

[13] M. Pryshliak, S. Subbotin, and A. Oliinyk, "Constructing a method for the conversion of numerical data in order to train the deep neural networks," *Journal of Enterprise Technologies*, vol.95, pp.1-7, 2018 .

[14] Alok Sharma, Edwin Vans, Daichi Shigemizu, Keith A. Boroevich and Tatsuhiko Tsunoda, "DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture," *Scientific Reports*, vol. 9, pp. 1-7, 2019.

[15] Timea Bezdan, Nebojsa Bacanin, "Convolutional neural network layers and architectures," *Data science and digital broadcasting systems, international scientific conference on information technology and data related research*, DOI: 10.15308/Sinteza-2019-445-451, 2019.

[16] S H Shabbeer Basha, Shiv Ram Dubey, Viswanath Pulabaigari, and Snehasis Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image Classification," *Indian Institute of Information Technology Sri City, Andhra Pradesh- 517646, India*. arXiv:1902.02771v3 [cs.CV] 19 Nov 2019.

[17] Jiayao Chen, Tongjun Yang, Dongming Zhang, Hongwei Huang, and Yu Tian, "Deep learning based classification of rock structure of tunnel face," *Geoscience Frontiers*, vol. 12, pp. 395-404, 2021 .

[18] Abhijit Roy, "An introduction to gradient descent and backpropagation," <https://towardsdatascience.com/an-introduction-to-gradient-descent-and-backpropagation-81648bdb19b2>.

[19] <https://www.mathworks.com>

[20] Umut Ozkaya, Saban Ozturk, and Mucahid Barstugan, "Coronavirus (COVID-19) Classification using Deep Features Fusion and Ranking Technique," <https://arxiv.org/pdf/2004.03698>.

[21] Shengping Yang, and Gilbert Berdine, "The receiver operating characteristic (ROC) curve," *The Southwest Respiratory and Critical Care Chronicles*, vol. 5, pp. 34-36, 2017.

[22] Ibrahim Kandel, Mauro Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," The Korean Institute of Communications and Information Sciences, pp. 1-4, 2020 .

[23] Olaf Berke, "Methodology Exploratory disease mapping: kriging the spatial risk function from regional count data," International Journal of Health Geographics, vol.3, pp.1-11, 2004.

[24] Farhad Misaghi, Shadi Dayyanidardashti, Kourosh Mohammadi, and M.R. Ehsani, "Application of Artificial Neural Network and Geostatistical Methods in Analyzing Strawberry Yield Data," 2004 ASAE/CSAE Annual International Meeting Sponsored by ASAE/CSAE Fairmont Chateau Laurier, The Westin, Government Centre Ottawa, Ontario, Canada 1 - 4 August 2004.

[25] Marco Bezzi, and Alfonso Vitti, "A comparison of some kriging interpolation methods for the production of solar radiation maps," Geomatics Workbooks, pp.1-5, 2005.

[26] Ibrahim L. Olokodana, Saraju P. Mohanty, and Elias Kougiianos, "Krig-Detect: Exploring Alternative Kriging Methods for Real-Time Seizure Detection from EEG Signals," DOI: 10.1109/WF-IoT48130.2020.9221260, 2020.

[27] Kemal Sulhi Gundogdu, and Ibrahim Guney, "Spatial analysis of groundwater levels using universal kriging," Journal of Earth System Science, vol.116, pp. 49-55, 2007.

[28] Feridon Ghadimi, Mohammad Ghomi, and Mojtaba Aref Sedigh, "Identification of Ti-Anomaly in Stream Sediment Geochemistry using Stepwise Factor Analysis and Multifractal Model in Delijan District, Iran," IJMGE Int. J. Min. & Geo-Eng, Vol. 50, No.1, pp77-95, 2016.

[29] Mandana Tahmooresi, Behnam Babaei, and Saeed Dehghan, "Intelligent geochemical exploration modeling using multiclass support vector machine and integration it with continuous genetic algorithm in Gonabad region, Khorasan Razavi, Iran," Arabian Journal of Geosciences, 141012 (2021). <https://doi.org/10.1007/s12517-021-07306-w>

[30] Mandana Tahmooresi, Data mining and intelligent optimization of support vector machine and convolutional neural network using genetic algorithm in order to modeling for mineral potential exploration (Case study: Gonabad

arena), Ph.D. Dissertation, Mahallat Branch, Islamic Azad University, Mahallat, IRAN. 2022 (Note: Final defend) [In Persian].

APPENDIX: NUMERICAL MODELING OF CONVOLUTIONAL NEURAL NETWORK IN MATLAB

```

Clc;
Clear;
close all;
%% Classification Usin Deep learning CNN
%% Load and Divide Data
Gonabad=xlsread('E:\Mahallat\Mahallat\Resale_
mahallat\word\My
paper\Yazd\Classification\DataSets\dataentry.xls');
save('E:\Mahallat\Mahallat\Resale_mahallat\wor
d\My
paper\Yazd\Classification\DataSets\Gonabad.mat');
ListFolder =
{'Gonabad.mat','fisheriris.mat','d166_9598_clean2_
2.mat'};
NameDataSet = ListFolder{1};
[dataTrain,dataValid,dataTest] =
LoadDivideData(NameDataSet);
%% Set Layers and Train Options
numFeatures = size(dataTrain.Inputs,2);
numClasses = numel(unique(dataTrain.Targets));
layers = CreateLayers(numFeatures,numClasses);
%% Set Options
lr = 1e-2;
MaxE = 500;
MinB = 12;
options =
SetTrainOptions(lr,MaxE,MinB,dataValid);
%% Train NetWorks
[netTrain,info] =
trainNetwork(dataTrain.Inputs,dataTrain.Targets,
layers,options);
%% Prediction and Evaluation
% Train Data
Labels = dataTrain.Targets;
ResultsTrain = EvaluatePlot(netTrain,...
dataTrain,Labels,'Simple Create: Train');
% Test Data
Labels = dataTest.Targets;
ResultsTest = EvaluatePlot(netTrain,...
dataTest,Labels,'Simple Create: Test');
*****
function [TrainData,ValidData,TestData] =
LoadDivideData(name)
name = ['DataSets/',name];
data = importdata(name);
if isstruct(data)
data = data.data;
end
VaNum = round(NSamples * VaPercent / 100);
R = randperm(NSamples);
% load RFH

```

```

trIndex = R(1 : TrNum);
tvIndex = R(TrNum + 1:TrNum + VaNum);
tsIndex = R(1 + TrNum + VaNum : end);
TrainData.Inputs = Inputs(trIndex,:);
TrainData.Targets = Targets(trIndex);
ValidData.Inputs = Inputs(tvIndex,:);
ValidData.Targets = Targets(tvIndex);
TestData.Inputs = Inputs(tsIndex,:);
TestData.Targets = Targets(tsIndex);
end
*****
function          layers          =
CreateLayers(numFeatures,numClasses)
% convolution2dLayer(2,2,'Name','conv')
numHiddenNeuron = 20;
layers = [
featureInputLayer(numFeatures,'Normalization','r
escale-symmetric')
fullyConnectedLayer(numHiddenNeuron)
reluLayer('Name','sig')
fullyConnectedLayer(numClasses)
softmaxLayer
classificationLayer('Name','classification')];
end
%      reluLayer('Name','relu')
%      leakyReluLayer
*****
function          Results          =
EvaluatePlot(netTrain,data,Labels,Name)
% Prediction
[YPred,scores] = classify(netTrain,data.Inputs);
Lau = unique(Labels);
groups = zeros(size(YPred));
groupshat = zeros(size(YPred));
for i = 1:numel(Lau)
    ID = ismember(YPred,Lau(i));
    groupshat(ID) = i;
    ID = ismember(Labels,Lau(i));
    groups(ID) = i;
end
confmat = confusionmat(groups,groupshat);
Results = PrecisionRecall(confmat);
[Targets,Groups]
H_MultiClassCMROC(groups',groupshat');
figure,plotconfusion(Targets,Groups)
title(['CM for ',Name,' Data'])
for i = 1:size(confmat,1)
[~,~,AUC(i)]
=
perfcurve(Targets(i,:),scores(:,i),1);
end
Results.AUC = AUC;
disp(['Results For ',Name,' Data'])
disp(Results)
disp('*****')
figure,plotroc(Targets,scores')
title(['ROC for ',Name,' Data'])
end
*****
function          [Targets,Groups]          =
H_MultiClassCMROC(Targets,Groups)
Tu = unique(Targets);
Temp1 = zeros(size(Targets));
Temp2 = Temp1;
for i = 1:numel(Tu)
    Ind1 = find(Targets==Tu(i));
    Temp1(Ind1) = i; %ok
    Ind2 = find(Groups==Tu(i));
    if isempty(Ind2)
        Ind2=Ind1(1);
    else
        Temp2(Ind2) = i; %ok
    end
end
Targets = full(ind2vec(Temp1));
Groups = full(ind2vec(Temp2));
end
function Results = PrecisionRecall(confmat)
nC = size(confmat,1);
for i = 1:nC
    Precision(i) = confmat(i,i)/sum(confmat(:,i))*100;
    Recall(i) = confmat(i,i)/sum(confmat(i,:))*100;
    F1_Score(i)
    =
    2*Precision(i)*Recall(i)/(Precision(i)+Recall(i));
end
Accuracy = sum(diag(confmat))/sum(confmat(:));
Results.Accuracy = 100*Accuracy;
Results.Precision = Precision;
Results.Recall = Recall;
Results.F1_Score = F1_Score;
% disp(Results)
End
*****
function          options          =
SetTrainOptions(Ir,MaxE,MinB,dataValid)
Validation = {dataValid.Inputs,dataValid.Targets};
options = trainingOptions('sgdm', ...
'MiniBatchSize',MinB, ...
'MaxEpochs',MaxE, ...
'InitialLearnRate',Ir,...
'LearnRateSchedule','piecewise',...
'LearnRateDropFactor',0.7, ...
'LearnRateDropPeriod',50, ...
'ValidationData',Validation, ...
'ValidationFrequency',50, ...
'ValidationPatience',Inf, ...
'Shuffle','every-epoch',...
'Verbose',0, ...
'Plots','training-progress',...
'ExecutionEnvironment','cpu');%none,'training-
progress'
% 'ValidationData',augimdsValidation, ...
% 'ValidationFrequency',3, ...
End

```