Date of publication MAR-31, 2024, date of current version JAN-25, 2024.
www.computingonline.net / computing@computingonline.net

*Print ISSN 1727-6209*
*Online ISSN 2312-5381*
*DOI 10.47839/ijc.23.1.3435*

# Computer Model for Detecting Cervical Spine Fractures based on Computed Tomography Images

**IEVGEN FEDORCHENKO[1], ANDRII OLIINYK[1], MAKSYM CHORNOBUK[1], YULIIA FEDORCHENKO[1], SERHII SHYLO[2], MYKOLA KHOKHLOV[3]**

[1]Department of Software Tools, National University «Zaporizhzhia Polytechnic», 64 Zhukovskoho str., Zaporizhzhya, Ukraine, 69063
[2]Department of electrical and electronic apparatus, National University «Zaporizhzhia Polytechnic», 64 Zhukovskoho str., Zaporizhzhya, Ukraine, 69063
[3]Department of Computer systems and networks, National University «Zaporizhzhia Polytechnic», 64 Zhukovskoho str., Zaporizhzhya, Ukraine, 69063

Corresponding author: Ievgen Fedorchenko (e-mail: evg.fedorchenko@gmail.com).

The work was carried out with the support of the state budget research project of the state budget of the National University "Zaporozhzhia Polytechnic" “Intelligent methods and tools for diagnosing and predicting the state of complex objects” (state registration number 0122U000972).

**ABSTRACT** The paper discusses methods of detecting cervical spine fractures based on computed tomography images using machine learning algorithms. Solving such a problem is important in the context of providing emergency care to patients with suspected spinal fractures, when accurate decision-making based on radiological data can be critical. In this case, such a machine learning model can speed up the work of a radiologist and reduce the importance of the human factor in making important decisions. After a review of analogs from the literature, it has been found that convolutional neural networks appear to be the most promising method. Using a publicly available dataset, a model "Fracture detection 3" based on a convolutional neural network is developed to solve the problem. The model demonstrates a classification accuracy of 98.25%, sensitivity of 99%, and specificity of 97.5%, which is ahead of the literature. For comparison with traditional methods, models based on the support vector method, decision tree, and k-nearest method are developed using a similar dataset. "Fracture detection 3" outperforms all developed models based on traditional methods in terms of classification accuracy.

**KEYWORDS** neural network; machine learning; convolutional neural network; pattern recognition.

## I. INTRODUCTION

THE unique structure and flexibility of the cervical spine make it vulnerable to injury. Injuries to this area cover a wide range, from minor muscle sprains to severe fractures and dislocations that can cause spinal cord damage and endanger the patient's life. To prevent additional neurological injury, it is critical to follow established protocols for initial assessment and treatment, including immobilization to protect the cervical spine from any unnecessary movement. Although cervical spine injuries are often obvious, relying on clinical examination alone may sometimes not be enough. In this case, radiographic imaging is necessary for an adequate response from doctors. Rapid identification and localization of any vertebral fractures is crucial to prevent neurological deterioration and paralysis following a traumatic event.

Typically, X-rays or computed tomography are used for imaging. Although computed tomography is the preferred method, X-rays may be the only option due to their lower cost [1].

In recent years, machine learning has been increasingly used in medicine. Large data sets are created based on years of scientific research, patient observation, and treatment results, which are then used to create machine learning models. Such models can be used for a wide range of tasks: predicting the patient's future condition, detecting disease based on diagnostic data, and many others. In particular, in radiology, i.e., the medical field related to the acquisition and processing of images of the human body, i.e., visualization, convolutional neural networks, machine learning models that can emulate the processes of pattern recognition in the human neural system, have become popular. Convolutional neural networks are particularly effective in this field, as they mimic the processes occurring in the human visual cortex. Convolutional neural networks, in addition to traditional fully connected layers of neurons, use the so-called convolutional operation, which removes abstract characteristics of the image [2, 3].

There are also quite successful solutions based on deep neural networks related to solving the problem of detecting cervical spine fractures based on computed tomography images. For example, [4] described the use of a convolutional neural network to solve a similar problem. The accuracy of the machine learning model was 92%, while the results of radiologists' work for the same data were 95% accurate. These data indicate the prospects of using convolutional neural networks in the context of the current task.

## II. ANALYSIS OF THE SUBJECT AREA

### A. BASIC CONCEPTS

Cervical spine injuries range from relatively mild sprains and strains to fractures and dislocations of the vertebrae, which can lead to significant spinal cord damage. The special anatomy of the ligaments and bones provides the C-spine with a large range of motion, but also makes it more susceptible to injury. Various mechanisms underlying this flexibility make the cervical spine vulnerable to injury.

Patients with a suspected cervical spine fracture are treated at the pre-hospital stage with immobilization and keeping the spinal column "in line" to prevent excessive movements.

The goal of the initial assessment of potential cervical spine fractures is to quickly recognize and identify the primary injury and to adequately protect the spine to prevent further damage. For unconscious patients, the diagnosis is mainly made using computed tomography [1].

The task of automating the detection of cervical spine fractures based on CT images can be formally described as a binary classification problem, where the input data is an array of image pixel values, and the output data is the image's belonging to one of two classes: healthy or fractured.

Machine learning methods are used to solve these problems. A distinctive feature of such methods is the ability to solve problems based on the automatic detection of patterns in large amounts of training data, rather than using predefined algorithms [5].

Popular machine learning methods used to solve the classification problem include, among others, support vector machines, artificial neural networks, k-nearest methods, and decision trees.

### B. REVIEW OF EXISTING METHODS

As mentioned above, various methods of machine learning are widely used in medicine, and in particular in radiology. The most popular machine learning methods used in radiology include the support vector method, artificial neural networks, the k-nearest method, and decision trees [6].

The Support Vector Machine (SVM) is a supervised learning algorithm used to solve classification problems. It consists in separating instances in the feature space using a hyperplane. The hyperplane is chosen in such a way as to best classify the new data. The instances in the training data that are closest to the hyperplane are called support vectors. The SVM algorithm tries to maximize the distance between hyperplanes and support vectors in each group of classes, which improves the accuracy of the model.

In radiology, SVMs are most effective when there are clear differences between the feature values of two groups, such as for predicting tumor grade or classifying different tissues based on texture [6-10].

K-Nearest Neighbor (KNN) is another supervised machine learning algorithm used to solve the classification problem. When training a model based on this method, all instances from the training sample are stored in the database. When the model runs, the attribute values of the input instance are compared to each of the values stored in the database. The initial class of an instance is determined by the class of most of the k "closest" training instances to it.

The choice of the "distance" function between the two instances is critical. Usually, Manhattan or Euclidean distances are used for quantitative attributes. For qualitative attributes, the Hamming distance can be used.

The choice of parameter k has a decisive influence on the quality of the algorithm. The algorithm can be too sensitive in the case of small k, and vice versa in the case of large values of k. One way to solve this problem is to use the so-called weighted k-nearest algorithm. In the weighted version of the method, the nearest k points are weighted using a function called the kernel function. The point is to give more weight to points that are close and less weight to points that are far away. The kernel function can be any function that decreases as the argument increases. For example, the inverse distance function can be used [6].

Decision trees are another popular supervised machine learning method used to solve classification problems. They are hierarchical tree-like structures, each node of which contains a rule according to which this object moves to one of the deeper nodes. Terminal nodes are called leaves. Each leaf corresponds to one class, which is given to the object that falls into this node during the classification process.

There are many methods for building trees based on a training sample of instances: CART, C4.5, CHAID, QUEST. All methods use a common concept: building a tree with a partition at each step of a set from some node by creating a partition rule. They differ in the principle of choosing the rule for splitting at each step, the principle of reducing the size of the resulting tree, and the condition for stopping the algorithm.

Convolutional neural networks are one of the most popular types of neural networks and methods in general used in the field of classification, and image classification in particular. Convolutional neural networks mimic structures from the visual cortex of the human brain, although they do not model their operation exactly. Unlike conventional fully connected layers of simple neural networks, convolutional neural networks use so-called convolutional layers, which effectively detect patterns in two-dimensional input data.

The convolutional layer is the most important part of the convolutional neural network. The convolutional layer includes a separate filter for each channel, which processes the previous layer piece by piece. The convolution operation uses an extremely small number of parameters, which simplifies the training process and speeds up the network [7].

Pooling layers are used to reduce the dimensionality of data. A group of pixels is compressed to a single pixel by undergoing a non-linear transformation. The max function is usually used, but other functions can be used, such as calculating the average value.

The use of neural networks, and in particular convolutional neural networks in fields related to radiology, has gained rapid popularity in recent years and continues to improve. Convolutional neural networks are used for classification, segmentation, and disease detection tasks.

### C. REVIEW OF ANALOGS FROM THE LITERATURE

The problem of detecting cervical spine fractures based on computed tomography images was discussed in [3]. A convolutional neural network developed by Aidoc was used. The results of the neural network were also compared with the average results of the assessment by radiologists of different levels. The machine learning model achieved 92% accuracy in detecting cervical spine fractures, 76% sensitivity, and 97% specificity. Radiologists' accuracy was 95%, with 93% sensitivity and 96% specificity. The fractures missed by the convolutional neural network and radiologists were similar in level and location and included fractured anterior osteophytes, transverse processes, and spinous processes, as well as lower cervical spine fractures, which are often hidden by attenuation of the CT beam.

A similar problem was solved in [6]. In this paper, the authors studied the possibility of developing a model to detect any spinal fractures from X-ray images. A convolutional neural network is used as a machine learning algorithm. According to the test results, the model achieved a level of accuracy, sensitivity, and specificity of 86%, 84.7%, and 87.3%, respectively.

The problem of detecting cervical spine fractures using a model to analyze the results of computed tomography was also solved in [7]. The model developed by the authors, based on a convolutional neural network with a layer of bidirectional long-term memory, achieved an accuracy of 77.6%, sensitivity and specificity of 77.2% and 77.6%, respectively.

A comparison of the analogs under consideration is shown in Table 1.

**Table 1. Comparison of the analyzed analogs**

| The model used | Source of information | Accuracy | Sensitivity | Specificity |
|---|---|---|---|---|
| CNN Aidoc [4] | CT | 92% | 76% | 97% |
| Expert assessment [4] | CT | 95% | 93% | 96% |
| IBM Watson Visual Recognition V3 [6] | X-ray | 86% | 84.7% | 87.3% |
| DCNN [7] | CT | 77.6% | 77.2% | 77.6% |

### D. CONCLUSIONS

Study [4] not only established a benchmark level of model quality, i.e., accuracy, sensitivity, and specificity of classification, but also provided valuable results of real radiologists' work on a dataset that is also used to evaluate the machine learning model.

These data allow us to establish that convolutional neural networks are indeed a promising technology for solving the problem. Models based on convolutional neural networks are capable of solving the task at the level of professional specialists, and therefore can be used in real medical practice as an aid.

## III. BUILDING A MACHINE LEARNING MODEL

A software implementation of a system for automatic detection of cervical spine fractures based on computed tomography images using a machine learning model based on a convolutional neural network is developed.

The problems of preventing model overtraining, developing its architecture, and selecting functions for assessing its quality are considered.

### A. PRELIMINARY PROCESSING OF INPUT DATA

All images from the dataset [8] are already resized to the same size – 224 by 224 pixels. All the data that goes to the model input has the same dimensionality. Other images that can be fed into the model should be converted to the same format.

Before processing in the model, pixel values from the image are normalized, i.e., reduced to a single small range, the values from which can be efficiently processed by the model.

Since the values of input pixels can take values from 0 to 255, the value of each pixel is converted to a range from 0 to 1.

The dataset already provides data split into training and validation sets, so there is no need for random splitting. All the models will be trained on the training set and tested on the validation set.

An example of an input image to the model is shown in Fig. 1.

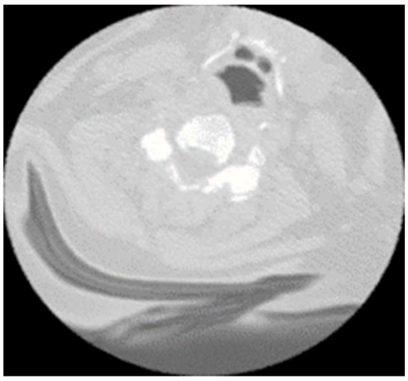


Figure 1. CT image of the cervical spine. The image was obtained from the dataset [8].

### B. SELECTION OF COST FUNCTION AND MODEL QUALITY ASSESSMENT FUNCTIONS

A loss function is used to formally quantify the model performance during training. The loss function is calculated as a numerical value from the model outputs and the actual values of the output features of the training set instances. It shows the value of the model's error, so during training the loss function is minimized to achieve the best possible model performance [9].

Since the model solves the problem of binary classification, it is natural to use a cost function called binary cross-entropy. This function can be used when the values of the output attributes of the sample instances take binary values: 0 or 1. In this case, the values predicted by the model can belong to the range from 0 to 1. The value of the binary cross-entropy increases with the distance between the actual and predicted values of the instance classes.

Metrics such as accuracy, sensitivity, and specificity will be used to evaluate the model performance during testing. These metrics are typically used to evaluate models that predict a discrete input argument, i.e., solve a classification problem. They are also used in medicine to evaluate diagnostic methods [10].

The sensitivity of a diagnostic model measures its ability to correctly identify sample with the disease condition. It is the proportion of true positives that are correctly identified by the test, given by

$$Sn = \frac{TP}{TP+FN}, \quad (1)$$

where $TP$ is true positives, $FN$ is false negatives.

The specificity of a diagnostic model measures its ability to correctly identify sample without the disease. It is the proportion of true negatives that are correctly identified by the test, given by

$$Sp = \frac{TN}{TN+}, \quad (2)$$

where $TN$ is true negatives, $FP$ is false positives.

The accuracy of a diagnostic model shows the general ability to correctly identify sample's class. It is the proportion of correctly recognized samples, given by

$$AC = \frac{Nc}{Nc+}, \quad (3)$$

where $Nc$ is correctly classified subjects, $Ni$ is incorrectly classified subjects.

Specificity and sensitivity tend to be inversely related. High-sensitivity test is more likely to detect true positives among those with a disease, while a highly specific test is better at correctly identifying true negatives in those without the condition. Consequently, it is important to evaluate both sensitivity and specificity to assess diagnostic test performance.

### *C. PREVENTING OVERFITTING*

Overfitting is a phenomenon that occurs during the training of machine learning models. It can be defined as a result that fits a particular data set too closely and therefore may not be able to process new data accurately enough. The appearance of overfitting indicates that the model does not generalize the data sufficiently, i.e., it cannot detect important patterns in the training dataset, but has enough information about this dataset to have high recognition accuracy on it.

The best way to eliminate the overfitting effect is to increase the training data set [11]. However, since we are using a publicly available dataset that has a limited size, other techniques need to be applied.

One of the well-known methods to combat overfitting in neural network-based models is the use of dropout layers [12].

The thinning applied to a layer consists in removing (setting to zero) features at the model training stage. The thinning coefficient is the proportion of features that are zeroed out. Thinning is applied only during model training. In order to compensate for the effect of reducing the number of features during thinning, the features that have not been zeroed are multiplied by 1/(1 – R), where R is the thinning coefficient [13-28].

Thinning in the model is implemented using the dropout layer from the Keras library. The value of the thinning coefficient is set to 0.1. It was chosen empirically.

### *D. BUILDING AND TESTING THE MODEL*

To solve this problem, a modified model "Fracture detection 1" is built based on a convolutional neural network. The architecture of the model is given in Table 2. The network input is a 56 by 56 pixel image.

Compared to one of the well-known models from the literature [6], the developed model has a number of differences. "Fracture detection 1" consists of a linear sequence of layers, while Visual Recognition V3 [6] has a complex branched structure and uses concat layers to combine the results of different branches. In [6], the avgpool layer is also used, which is not used in "Fracture detection 1". Softmax is used as a function for two output neurons in [6], while in "Fracture detection 1" sigmoid is used and only one output neuron is available. Overall, "Fracture detection 1" has a simpler architecture while achieving slightly better binary classification accuracy results.

**Table 2. "Fracture detection 1" architecture**

| Layer | Output shape |
|---|---|
| Convolution (4x4x1@4) | (None, 56, 56, 4) |
| MaxPooling2D (f=2;s=2) | (None, 28, 28, 4) |
| Convolution (4x4x4@8) | (None, 28, 28, 8) |
| MaxPooling2D (f=2;s=2) | (None, 14, 14, 8) |
| Convolution (4x4x6@16) | (None, 14, 14, 16) |
| MaxPooling2D (f=2;s=2) | (None, 7, 7, 16) |
| Convolution (4x4x16@32) | (None, 7, 7, 32) |
| MaxPooling2D (f=2;s=2) | (None, 3, 3, 32) |
| Dense (128) | (None, 128) |
| Dense (1) | (None, 1) |

The results of model training are shown in Fig. 2. Overall, the model achieved a classification accuracy of 85.00%.

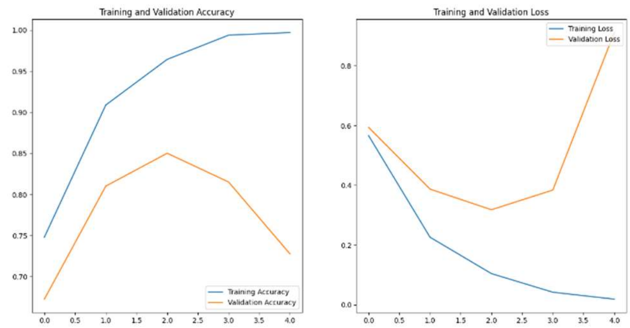



Figure 2. Graphical representation of the training of the "Fracture detection 1" model. The left part of the image contains graphs of accuracy values, the right – values of the loss function.

In order to achieve higher classification accuracy results, the "Fracture detection 2" model, which has a more complex structure, is built. Unlike "Fracture detection 1", "Fracture detection 2" has a double number of filters in each convolutional layer, a double number of neurons in the fully connected layer, and one additional fully connected layer. Such changes allowed the model to detect deeper regularities in the input data, which led to an increase in the level of classification accuracy.

The architecture of the model is given in Table. 3. The results of the model training are shown in Fig. 3. In general, the model achieved a classification accuracy of 88.75%.

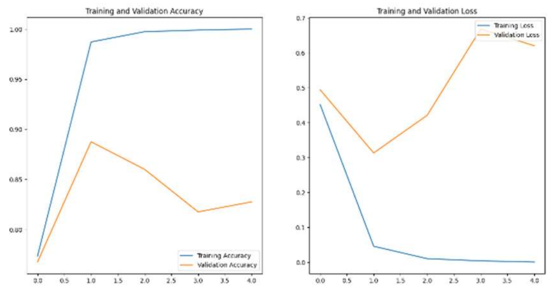



Figure 3. Graphical representation of the training of the "Fracture detection 2" model. The left part of the image contains graphs of accuracy values, the right – values of the loss function.

**Table 3. "Fracture detection 2" architecture**

| Layer | **Output shape** |
|---|---|
| Convolution (4x4x1@8) | (None, 56, 56, 8) |
| MaxPooling2D (f=2;s=2) | (None, 28, 28, 8) |
| Convolution (4x4x4@16) | (None, 28, 28, 16) |
| MaxPooling2D (f=2;s=2) | (None, 14, 14, 16) |
| Convolution (4x4x6@32) | (None, 14, 14, 32) |
| MaxPooling2D (f=2;s=2) | (None, 7, 7, 32) |
| Convolution (4x4x16@64) | (None, 7, 7, 64) |
| MaxPooling2D (f=2;s=2) | (None, 3, 3, 64) |
| Dense (256) | (None, 256) |
| Dense (256) | (None, 256) |
| Dense (1) | (None, 1) |

Obviously, there is not enough information in the compressed images, and the structure of the models is not complex enough to extract it, therefore, in order to achieve even better results of classification accuracy, a modified model "Fracture detection 3" is built based on a convolutional neural network. Unlike the "Fracture detection 2" model, "Fracture detection 3" has twice the number of filters in each convolutional layer, twice the number of neurons in the fully connected layer, one additional fully connected layer, and a dropout layer that the previous model lacked. Adding a dropout layer reduced the overtraining effect of the model and led to an increase in the maximum accuracy results obtained during training. The network accepts an image size of 224 by 224 pixels. A graphic representation of the architecture is shown in Fig. 4. The architecture of the model is given in Table. 4.

**Table 4. "Fracture detection 3" architecture**

| Layer | **Output shape** |
|---|---|
| Convolution (4x4x1@16) | (None, 224, 224, 16) |
| MaxPooling2D (f=2;s=2) | (None, 112, 112, 16) |
| Convolution (4x4x4@32) | (None, 112, 112, 32) |
| MaxPooling2D (f=2;s=2) | (None, 56, 56, 32) |
| Convolution (4x4x6@64) | (None, 56, 56, 64) |
| MaxPooling2D (f=2;s=2) | (None, 28, 28, 64) |
| Convolution (4x4x16@128) | (None, 28, 28, 128) |
| MaxPooling2D (f=2;s=2) | (None, 14, 14, 128) |
| Dropup (10%) | (None, 14, 14, 128) |
| Dense (512) | (None, 512) |
| Dense (512) | (None, 512) |
| Dense (512) | (None, 512) |
| Dense (1) | (None, 1) |

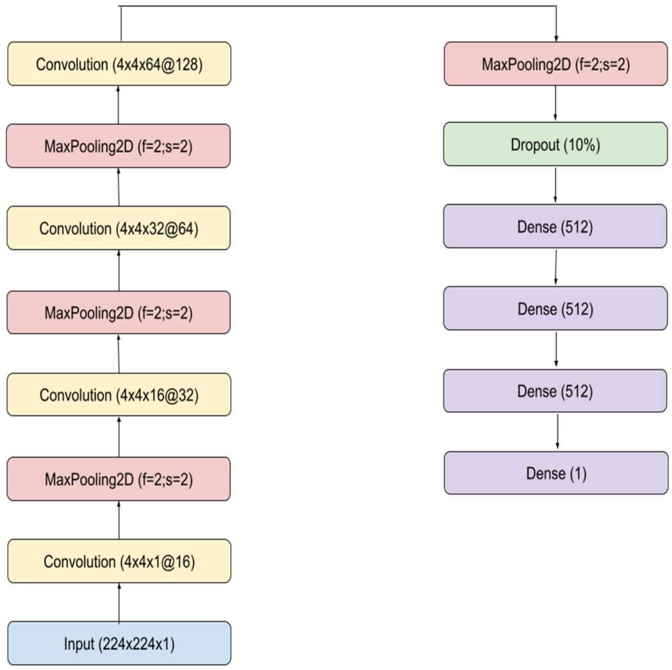



Figure 4. Architecture of "Fracture detection 3"

The results of the model training are shown in Fig. 5. At the last training epoch, the "Fracture detection 3" model achieved a classification accuracy of 98.25%. The sensitivity value was 99% and the specificity was 97.5%.

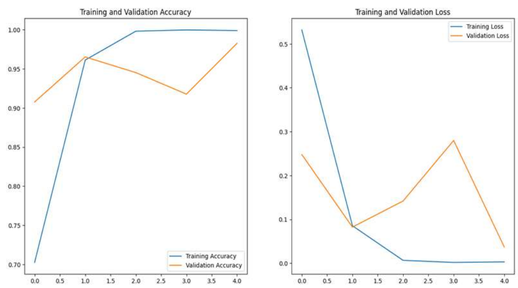



Figure 5. Graphical representation of the "Fracture detection 3" model training. The left part of the image contains graphs of accuracy values, the right – values of the loss function.

### *E. COMPARISON OF THE MODEL WITH TRADITIONAL METHODS*

To compare the "Fracture Detection 3" model with models based on traditional methods, we created models based on the support vector method, the k-nearest method, and the decision tree. The Scikit-learn library [13] was used to develop these models. A similar dataset was used to train and test each of these models [8].

Since the aforementioned traditional methods are not able to efficiently process raw image data, it is necessary to reduce the input dimensionality for these models. To solve this problem, we implemented the principal component analysis method from the Scikit-learn library. The data dimension for each of the instances was reduced from 50176 to 25 attributes.

For the model based on the support vector method, after training, the following results were obtained on the test dataset: classification accuracy – 77.25%, sensitivity – 97.5%, specificity – 57%. For the model based on the k-nearest method, after training, the following results were obtained on the test dataset: classification accuracy – 82.25%, sensitivity – 100%, specificity – 64.5%. For the model based on decision trees, after training, the following results were obtained on the test data set: classification accuracy – 58.25%, sensitivity – 68.5%, specificity – 48%.

A comparison of the results of traditional methods with the results of Fracture detection is shown in Table 5.

**Table 5. Comparison of the results of traditional methods with the results of "Fracture detection 3" model**

| The model used | Accuracy | Sensitivity | Specificity |
|---|---|---|---|
| CNN "Fracture detection 3" | 98.25% | 99% | 97.5% |
| SVM | 77.25% | 97.5% | 57% |
| K-neighbors | 82.25% | 100% | 64.5% |
| Decision tree | 58.25% | 68.5% | 48% |

### *F. COMPARISON OF THE MODEL WITH ANALOGS FROM THE LITERATURE*

A comparison of the results of the "Fracture detection 3" test with analogs from the literature is shown in Table 6. The table data indicate that the developed model is innovative and highly effective in terms of detecting cervical spine fractures. It is ahead of the known analogs in terms of accuracy, sensitivity, and specificity of classification.

The high sensitivity of the model is worth emphasizing, that is, the ability to correctly identify patients who have a spinal fracture. It is this feature of the model that will play a key role during its practical use, allowing for timely diagnosis, making a decision on a treatment strategy, including immobilizing patients with a fracture.

**Table 6. Comparison of the model with analogs from the literature**

| The model used | Source of information | Accuracy | Sensitivity | Specificity |
|---|---|---|---|---|
| CNN Aidoc [4] | CT | 92% | 76% | 97% |
| IBM Watson Visual Recognition V3 [6] | X-ray | 86% | 84.7% | 87.3% |
| DCNN [7] | CT | 77.6% | 77.2% | 77.6% |
| CNN "Fracture detection 3" | CT | 98.25% | 99% | 97.5% |

## IV. CONCLUSIONS

The problem of automatic detection of cervical spine fractures based on computed tomography images using machine learning algorithms is considered.

A review of literature analogs is carried out, based on the results of which a decision is made to develop a machine learning model based on a convolutional neural network to solve the problem.

3 models have been developed, in particular the best – "Fracture detection 3", which demonstrates 98.25% classification accuracy, 99% sensitivity, and 97.5% specificity.

We developed models based on the support vector method, decision tree, and k-nearest method using a similar dataset. "Fracture detection 3" outperforms all developed models based on traditional methods in terms of classification accuracy.

The model performance demonstrated by the test results outperforms similar systems from the literature, as well as other methods on the same dataset, which suggests that the developed model is innovative and highly efficient in terms of the task at hand.

## References


[1] M. W. Beeharry, K. Moqeem, & M. U. Rohilla, "Management of cervical spine fractures: A literature review," *Cureus*, vol. 13, issue 4, e14418, 2021. https://doi.org/10.7759/cureus.14418.

[2] G. S. Handelman, H. K. Kok, R. V. Chandra, A. H. Razavi, M. J. Lee, & H. Asadi, "eDoctor: machine learning and the future of medicine," *Journal of Internal Medicine*, vol. 284, issue 6, pp. 603-619, 2021. https://doi.org/10.1111/joim.12822.

[3] J.E. Small, P. Osler, A. B. Paul, M. Kunst, "CT cervical spine fracture detection using a convolutional neural network," *AJNR Am J Neuroradiol*, vol. 42, issue 7, pp. 1341-1347, 2021. https://doi.org/10.3174/ajnr.A7094.

[4] Y. Dong et al., "Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells," *Proceedings of the IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, 2017, pp. 101-104. https://doi.org/10.1109/BHI.2017.7897215.

[5] Oxford Advanced Learner's Dictionary, 2023, [Online]. Available at: https://www.oxfordlearnersdictionaries.com/us/definition/english/machine-learning.

[6] K. Murata, K. Endo, T. Aihara, et al., "Artificial intelligence for the detection of vertebral fractures on plain spinal radiography," *Sci Rep*, vol. 10, issue 1, article no. 20031, 2020. https://doi.org/10.1038/s41598-020-76866-w.

[7] H. Salehinejad, E. Ho, H.-M. Lin, P. Crivellaro, O. Samorodova, M. T. Arciniegas, Z. Merali, S. Suthiphosuwan, A. Bharatha, K. Yeom, M. Mamdani, J. Wilson, E. Colak, "Deep sequential learning for cervical spine fracture detection on computed tomography imaging," *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*, 2021, pp. 1911-1914. https://doi.org/10.1109/ISBI48211.2021.9434126.

[8] Spine fracture prediction from C.T., 2022, [Online]. Available at: https://www.kaggle.com/datasets/vuppalaadithyasairam/spine-fracture-prediction-from-xrays.

[9] Q. Wang, Y. Ma, K. Zhao, et al., "A comprehensive survey of loss functions in machine learning," *Ann. Data. Sci*., 2022, pp. 187-212. https://doi.org/10.1007/s40745-020-00253-5.

[10] H. B. Wong, & G. H. Lim, "Measures of diagnostic accuracy: Sensitivity, specificity, PPV and NPV," *Proceedings of Singapore Healthcare*, vol. 20, issue 4, pp. 316-318, 2011. https://doi.org/10.1177/201010581102000411.

[11] X. Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, issue 2, article no. 022022, 2019. https://doi.org/10.1088/1742-6596/1168/2/022022.

[12] Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, issue 1, pp. 1929-1958, 2014.

[13] Scikit-learn, Machine Learning in Python, [Online]. Available at: https://scikit-learn.org.

[14] A. Oliinyk, I. Fedorchenko, A. Stepanenko, A. Katschan, Y. Fedorchenko, A. Kharchenko, D., Goncharenko, "Development of genetic methods for predicting the incidence of volumes of pollutant emissions in air," *Proceedings of the 2nd International Workshop on Informatics and Data-Driven Medicine, CEUR Workshop Proceedings*, 2019, pp. 340-353.

[15] D. R. Sarvamangala, & R. V. Kulkarni, "Convolutional neural networks in medical image understanding: A survey," *Evolutionary Intelligence*, vol. 15, pp. 1–22, 2022. https://doi.org/10.1007/s12065-020-00540-3.

[16] J. Serey, M. Alfaro, G. Fuertes, M. Vargas, C., Ternero, R. Durán, R. Rivera, & J. Sabattin, "Pattern recognition and deep learning technologies, enablers of Industry 4.0, and their role in engineering research," *Symmetry*, vol. 15, issue 2, p. 535, 2023. https://doi.org/10.3390/sym15020535.

[17] C. Singh, "Machine learning in pattern recognition," *European Journal of Engineering and Technology Research*, vol. 8, issue 2, pp. 63–68, 2023. https://doi.org/10.24018/ejeng.2023.8.2.3025.

[18] J. A. Alsayaydeh, M. Nj, S. N. Syed, A. W. Yoon, W. A. Indra, V. Shkarupylo and C. Pellipus, "Homes appliances control using bluetooth," *ARPN Journal of Engineering and Applied Sciences*, vol. 14 (19), pp. 3344-3357, 2019.

[19] I. Izonin, R. Tkachenko, N. Shakhovska, N. Lotoshynska, "The additive input-doubling method based on the SVR with nonlinear kernels: Small data approach," *Symmetry*, vol. 13, issue 4, 612, 2021. https://doi.org/10.3390/sym13040612.

[20] A. Mujumdar and V. Vaidehi, "Diabetes prediction using machine learning algorithms," *Procedia Computer Science*, vol. 165, pp. 292-299, 2019. https://doi.org/10.1016/j.procs.2020.01.047.

[21] J. A. J. Alsayaydeh, W. A. Y. Khang, W. A. Indra, J. B. Pusppanathan, V. Shkarupylo, A. K. M. Zakir Hossain and S. Saravanan, "Development of vehicle door security using smart tag and fingerprint system," *ARPN Journal of Engineering and Applied Sciences*, vol. 9, issue 1, pp. 3108-3114, 2019. https://doi.org/10.35940/ijeat.E7468.109119.

[22] O. R. Rudkovskyi, G. G. Kirichek, "Interaction support system of network applications," *Proceedings of the 3rd Workshop for Young Scientists in Computer Science & Software Engineering, CS&SE@SW 2020*, CEUR-WS, 27 November 2020, pp. 11–23.

[23] A. Oliinyk, I. Fedorchenko, A. Stepanenko, M. Rud, D. Goncharenko, "Implementation of evolutionary methods of solving the travelling salesman problem in a robotic warehouse," *Lecture Notes on Data Engineering and Communications Technologies*, vol. 48, 2021, pp. 263-292. https://doi.org/10.1007/978-3-030-43070-2_13.

[24] D. Wang, C. Moore, A. Murphy, "Support vector machine (machine learning)," Reference article, Radiopaedia.org, https://doi.org/10.53347/rID-61710.

[25] Y. Y. Song, Y. Lu, "Decision tree methods: applications for classification and prediction," *Shanghai Arch Psychiatry*, vol. 27, issue 2, pp. 130-135, 2015. https://doi.org/10.11919/j.issn.1002-0829.215044.

[26] K. Taunk, S. De, S. Verma, & A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," Proceedings of the 2019 IEEE International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, 2019, pp. 1255-1260. https://doi.org/10.1109/ICCS45141.2019.9065747.

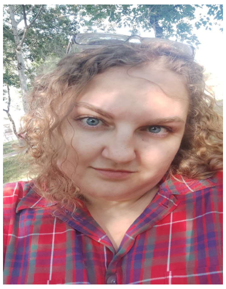
[27] I. Fedorchenko, A. Oliinyk, O. Stepanenko, T. Zaiko, A. Svyrydenko, D. Goncharenko, "Genetic method of image processing for motor vehicle recognition," *Proceedings of the 2nd International Workshop on Computer Modeling and Intelligent Systems, CMIS 2019*, CEUR Workshop Proceedings, Zaporizhzhia, Ukraine, 15-19 April 2019, vol. 2353, pp. 211-226. https://doi.org/10.32782/cmis/2353-17.

[28] O. Alshannaq, J. A. J. Alsayaydeh, M. B. A. Hammouda, M. F. Ali, M. A. R. Alkhashaab, M. Zainon and A. S. M. Jaya, "Particle swarm optimization algorithm to enhance the roughness of thin film in tin coatings," *ARPN Journal of Engineering and Applied Sciences*, vol. 17, no. 22, pp. 186–193, 2022.


***IEVGEN FEDORCHENKO, Senior Lecturer of Software Tools Department of National University "Zaporizhzhia Polytechnic". Field of scientific interests: Artificial Intelligence, Modeling, Diagnostics.***
***https://orcid.org/0000-0003-1605-8066***
***Email: evg.fedorchenko@gmail.com***

***PROF. ANDRII OLIINYK, Dr. Sc., Professor of Software Tools Department of National University "Zaporizhzhia Polytechnic". Field of scientific interests: Artificial Intelligence, Modeling, Diagnostics.***
***https://orcid.org/0000-0002-6740-6078***
***Email: olejnikaa@gmail.com***

***MAKSYM CHORNOBUK Student at the National University "Zaporizhzhia Polytechnic". Field of scientific interests: Machine Learning, Pattern Recognition.***
***https://orcid.org/0000-0003-3200-7306***
***Email: chornobuk.maksym@gmail.com***



***YULIIA FEDORCHENKO Assistant of the Software Tools Department National University of Zaporizhzhia Polytechnic. Field of scientific interests: Artificial Intelligence, Modeling, Diagnostics.***
***http://orcid.org/0000-0003-4436-3877***
***Email: fedorchenkojuliia@gmail.com***

***SERHII SHYLO Senior Lecturer of electrical and electronic apparatus Department of National University "Zaporizhzhia Polytechnic". Field of scientific interests: Artificial Intelligence, Modeling, Diagnostics.***
***http://orcid.org/0000-0002-4094-6269***
***Email: sergey.shilo@gmail.com***

***MYKOLA KHOKHLOV Senior Lecturer of Computer systems and networks Department of National University "Zaporizhzhia Polytechnic". Field of scientific interests: Artificial Intelligence, Modeling, Diagnostics.***
***https://orcid.org/0000-0001-8272-9847***
***Email: khokhlov@zp.edu.ua***