Abnormality Bone Detection in X-Ray Images Using Convolutional Neural Network

Hiep Xuan Huynh[1(✉)], Hang Bich Thi Nguyen[3], Cang Anh Phan[2], and Hai Thanh Nguyen[1]

[1] College of Information and Communication Technology, Can Tho University, Can Tho, Vietnam
hxhiep@ctu.edu.vn, nthai@cit.ctu.edu.vn

[2] Faculty of Information Technology, Vinh Long University of Technology Education, Vinh Long, Vietnam
cangpa@vlute.edu.vn

[3] Department of Multimedia, Vinhlong Radio and Television Station, Vinh Long, Vietnam
nguyentbichhang41@gmail.com

Abstract. Medical imaging plays a role as a crucial source of data for disease detection and diagnosis. Recent advancements in machine learning and deep learning have become an efficient tool for medical image analysis. Medical image research laboratories are rapidly creating machine learning systems to achieve the professional performance of humans. However, both machine learning and deep learning methods are complex and require a lot of expertise, resources, knowledge, and time to train. Those create a significant barrier for researchers. In this study, we propose a convolutional neural network architecture to detect abnormalities in bone images. The proposed method provides insight into medical images and explains in detail how the model supports the diagnosis.

Keywords: Abnormality detection · Musculoskeletal radiographs · X-ray images · Convolutional neural network

1 Introduction

In recent years, abundant and expandable data sources in the field of health and the rapid development of technologies have contributed to improving the effectiveness of diagnosis and treatment. Health data is very various, the most common of which is medical images. Medical images include Computerized tomography(CT), MRI, and radiography images that perform the internal organs visually.

With features such as faster, cheaper, readily available, and easy to use, bone X-rays are one of the initial clinical measures used for doctors to make a

P. C. Vinh and A. Rakib (Eds.): ICCASA 2020/ICTCC 2020, LNICST 343, pp. 31–43, 2021.
https://doi.org/10.1007/978-3-030-67101-3_3

diagnosis or detect other abnormalities for the different parts of the body. The musculoskeletal conditions affect more than 1.7 billion people worldwide [1] and are the second most common cause of serious disabilities, and have the 4th greatest impact on the overall health of the world population when considering both death and disability. The treatment for osteoarthritis is long and costly. The causes of musculoskeletal problems may be due to external influences such as trauma, accidents, or playing sports, but also due to conditions such as genetics, knee arthritis, osteoporosis, and cancer. Proper diagnosis and abnormal findings are very important for treatment. But an increasing number of musculoskeletal patients is a major challenge in diagnosis. Automatic anomaly detection base on the computer can become very useful for diagnosis as well as saving time. Various machine learning processes have played an important role in the classification of medical imaging. Decision Forests [2], Support vector machine [3] or K-Means clustering [4], Integrating spatial fuzzy clustering [5] showed significant results in medical imaging classification. In addition, in deep learning, Convolutional NeuralNetwork (CNN [6] or ConvNet [7]) has been widely used in classifying images and segmentation problems. The improvements in deep learning in medical images have brought many promising results. For instance, analysis of electronic health records [8], bone tumor diagnosis [9], skin cancer detection [10] or most recently discovered COVID-19 [11,12]. However, this process is not only based on medical professional knowledge, medical industry standard, medical system but also requires knowledge in machine learning.

In this study, we present a novel method based on a convolutional neural network to detect abnormalities in musculoskeletal radiographs images. The method is expected to a good tool for disease diagnosis based on medical images.

The remaining of this work is presented as follows. Section 2 covers works related to detecting problems base on bones including bone disease and abnormalities. Section 3 introduces the architecture of our convolutional neural network for detecting abnormalities in bone images. We present details of the used dataset and experimental Scenarios in Sect. 4. Finally, experimental results are discussed in Sect. 5 respectively.

2 Related Work

Musculoskeletal disorders include many types of abnormalities in the bones, soft tissues, and joints. Finding an anatomical X-ray abnormality is not an easy task, so it is also an attractive topic in medical image classification. There have been many studies on bone images based on machine learning methods: diagnosing bone tumors by Naïve Bayesian model [9], detecting knee bone tumors by fractional method Seg-Unet [13] uses multi-tasking deep learning architecture, or estimates the probability for primary malignant bone tumors [14], detect abnormality in lower extremity radiographs [15] uses convolution neural networks (CNNs) and evaluates bone age [16].

MURA [17] is a large dataset containing images of musculoskeletal x-rays. In the MURA dataset, a 169 layer neural network (densenet) is used to predict anomalies. MURA provides better performance in comparison to the best

radiologist performance in detecting abnormalities on finger and wrist studies. This model performs lower than best radiologist performance in case of detecting the abnormality in elbow, forearm, hand, humerus, and shoulder studies. To improve model performance, various network structures are proposed. [18] uses Deep Convolutional Neural Network, [19] uses Multi-Network Model to Detect Abnormalities in Musculoskeletal Radiographs.

3 Deep Learning Architecture for Detecting Abnormalities in Bone Images

3.1 Data Preprocessing and Normalizing

Because the radiographs are of different sizes, resizing is needed to have an equal image size. First, they are resized to 64×64 pixels. Then, they were resized to 128×128 pixels and finally to 224×224 pixels. In the first case, when the image size is 64×64 pixels, the image is of poor quality, and most features are lost. In the second case, with an image size of 128×128 pixels, the image quality is improved compared to the first case. In the last case, when the image size is 224×224 pixels, the feature maps are lost less, thus improving accuracy and reducing losses, but training time is higher. Considering the image quality and training time in the second and third cases, the final 150×150 pixels image was chosen for model training. Hence Fig. 1 is a presentation of the image after resizing.

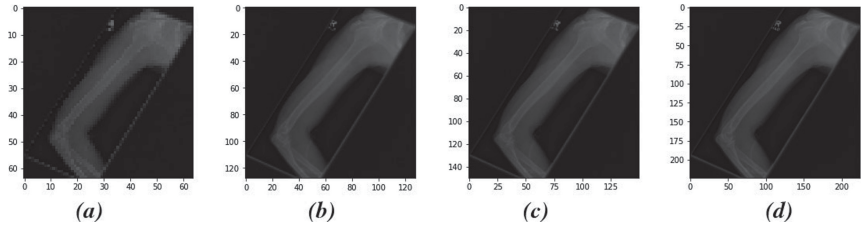

Fig. 1. When the image size is increased, more features are included: (a) image size 64×64 (b) image size 128×128 (c) image size 150×150 (d) image size 224×224.

3.2 The Proposed Convolutional Neural Network Architecture

We chose on humerus studies in the MURA dataset for our Convolutional Neural Networks model. And then we compare the results obtained with the results of training humerus studies by a 169-layer DenseNet baseline model [17]. Our Convolutional Neural Networks with structures are illustrated in Fig. 2. The CNN contains two Convolutional layers, followed by a MaxPooling layer and a Fully Connected layer.

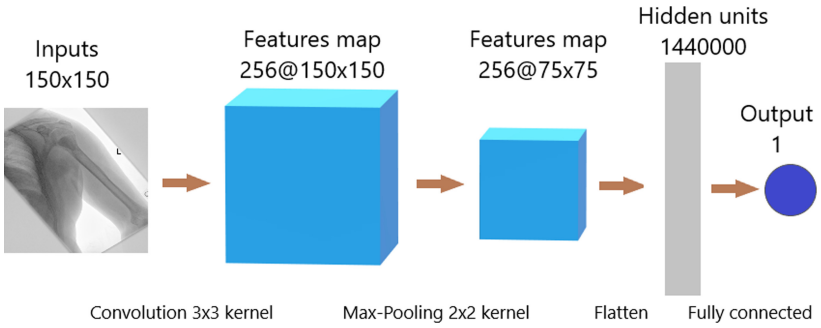

Fig. 2. A shallow convolutional neural network architecture.

For more details, in Fig. 3, CNN receives input data that is 150×150 pixels in size. The first convolution layer contains 64 filters or kernels, the filter itself being a 3×3 integer matrix. The second convolution layer contains 256 filters or kernels. The Convolutional Layer is a set of feature maps and each of these feature maps is a scanned copy of the original input but extracted into specific features/properties. How to scan depends on the Convolution Filter or the kernel. Here, we use a 3×3 size kernel, scans the input data matrix, from left to right, top to bottom, and multiplies each value of the input matrix with the kernel matrix and then adds up. Via the activation function "relu", we get the feature map. The Max-Pooling layer is used to reduce the dimensions of the generated feature maps. We used the Max-Pooling layer with a size of 2×2. The output from the Max-Pooling layer will be flattened to convert tensor in the multidimensional form to tensor 1D.

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_5 (Conv2D)            (None, 150, 150, 64)      640
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 150, 150, 256)     147712
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 75, 75, 256)       0
_________________________________________________________________
flatten_3 (Flatten)          (None, 1440000)           0
_________________________________________________________________
dense_3 (Dense)              (None, 2)                 2880002
=================================================================
Total params: 3,028,354
Trainable params: 3,028,354
Non-trainable params: 0
_________________________________________________________________
```

Fig. 3. A convolutional neural network implementation.

Moreover, CNN is implemented with Adam optimizer [20] with a batch of size 32. To avoid overfitting, we use Early stopping to stop the algorithm before the loss function reaches a value too small. The loss function is built to compare the difference between the predicted output and the actual output. Cross-entropy is a loss function, and its value can be minimized. This helps neural networks evaluate the probability of predicting a data sample corresponding to a class. For binary classification tasks, we computed binary cross-entropy loss during training by the formula 1.

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y_i}) + (1 - y_i)\log(1 - \hat{y_i}) \tag{1}$$

3.3 Tools

All experiments were implemented in Keras, trained and tested on a 64-bit Windows system equipped 8 GB of memory. Keras is an open source for neural networks written in python language. It is a high-level API that can be used together with famous deep learning libraries such as Tensorflow, CNTK, and Theano. Keras has some advantages such as: easy to use, fast model building, can run on both CPU and GPU, support to build CNN, RNN, and can combine both 2. Compared to other libraries, Keras is simple, user-friendly, yet very powerful. So, we choose Keras for our study.

4 Experiments

4.1 Dataset Description

MURA [17] is a large dataset of bone X-rays, collected by the Stanford ML group. Algorithms are tasked with determining whether an X-ray study is normal or abnormal. MURA dataset was published with 40,561 images from 14,863 studies including seven body tissues (elbow, finger, hand, humerus, forearm, shoulder, and wrist). Each study was labeled as either normal or abnormal by radiologists. The data set was separated into two parts, including the training set and validation set. MURA is one of the largest public radiographic image datasets (Fig. 4).

4.2 Evaluation Metrics

We have calculated the Training and Validation Accuracy, Training and Validation Loss, Cohen-kappa Score, Area Under the Receiver Operating Characteristic Curve (ROC-AUC) [22] for a general assessment of classifications. While the accuracy metric directly reflects the performance of the model, the Cohen-kappa score [21] is a metric that measures inter-rater agreement for qualitative or categorical items. In the case of musculoskeletal research, kappa statistics provide

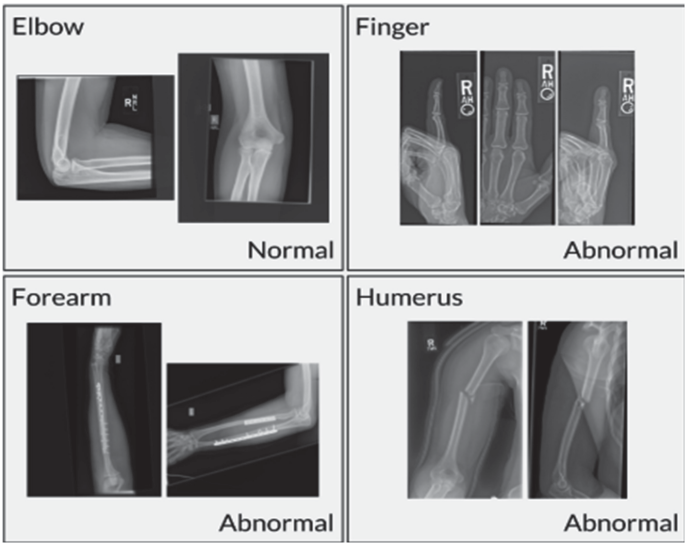

Fig. 4. MURA dataset contains 14,863 musculoskeletal studies of the upper extremity including the shoulder, humerus, elbow, forearm, wrist, hand, and finger. These examples show a normal elbow study (top left), an abnormal finger study (top right), an abnormal forearm study (bottom left), and an abnormal humerus study with a fracture (bottom right). The dataset is freely available at https://stanfordmlgroup.github.io/competitions/mura/

much more valuable information because k often considers random consensus. The formula to calculate Cohen's kappa for two raters is:

$$K = \frac{P_0 - P_e}{1 - P_e} \tag{2}$$

where P_0 represents consensus values observed among the evaluation variables and P_e represents probability assumptions ability consensus.

ROC is a graph of one axis is Sensitivity (or true positive rate), the other is Specificity (true negative rate) for a binary classification system. ROC is a probability curve and AUC represents a level or measure of separation. With TP represents true positive, TN represents true negative, FP represents false positive, and FN represents false negative; True positive rate (TPR) and true negative rate (TNR) are presented by the following formula:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4}$$

Besides, we use the confusion matrix to visualize model performance. Each row of matrices represents instances in a predictive class while each column represents

instances in an actual class (or vice versa). The confusion matrix makes it easy to see if the system is confusing two layers (usually mislabeled as another layer). Here, we label the two layers normal (0) and abnormal (1) as Table 1.

Table 1. Confusion matrix.

	Predicted class		
Actual class		Normal	Abnormal
	Normal	TN	FP
	Abnormal	FN	TP

Accuracy is defined as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

4.3 Experimental Scenarios

The learning rate is an important metric for the model. It controls how quickly the pattern is adjusted to the problem. A learning rate that is too large can cause the model to converge too quickly to a non-optimal solution, while a learning rate that is too small can make the process difficult. Therefore, we choose four different model scenarios to compare accuracy by changing the learning rate hyperparameter of the optimizer that implements the Adam algorithm. First, we train the model with a default learning rate of 0.001. After that, we in turn reduced the learning rate to 0.0001; 0.00001 and finally 0.000001. In addition, we also train the model with epochs of 12, 50, and 100. For each learning rate, the epoch with higher accuracy will be chosen. We compared accuracy in four learning rate scenarios in Table 2 to achieve good performance on our problem.

Table 2. Comparison of model accuracy in 4 scenarios.

Scenarios	Learning rate	Epoch		
		12	50	100
Scenario 1	0.001	0.635	0.514	0.597
Scenario 2	0.0001	0.576	0.674	0.684
Scenario 3	0.00001	0.604	0.625	0.615
Scenario 4	0.000001	0.563	0.608	0.604

In scenario 1, we train our model with the default learning rate hyperparameter of optimizer Adam is 0.001. We got a model accuracy of 0.635 at epoch 12. This accuracy is higher than that of the MURA model (0.600) on humerus

study. Continue to train the model with epoch 50 and 100, the model's accuracy is 0.514 and 0.597, respectively, lower than MURA model's accuracy.

In scenario 2, the learning rate hyperparameter is reduced to 0.0001. We achieve higher accuracy than scenario 1 at epoch 50 (0.674) and epoch 100 (0.684). However, the model has low accuracy at epoch 12 (0.576) compared to scenario 1 (0.635).

Continue to reduce the learning rate hyperparameter to 0.00001 in scenario 3 and 0.000001 in scenario 4, we get the highest accuracy for each scenario of 0.625 and 0.608 respectively. Both of these precision is higher than the MURA model but lower than the accuracy in scenario 1 and 2. The successive precision drops in scenarios 3 and 4, so we stop reducing the learning rate. Comparing all 4 scenarios (Fig. 5), we choose the one with the highest accuracy to use for the training model.

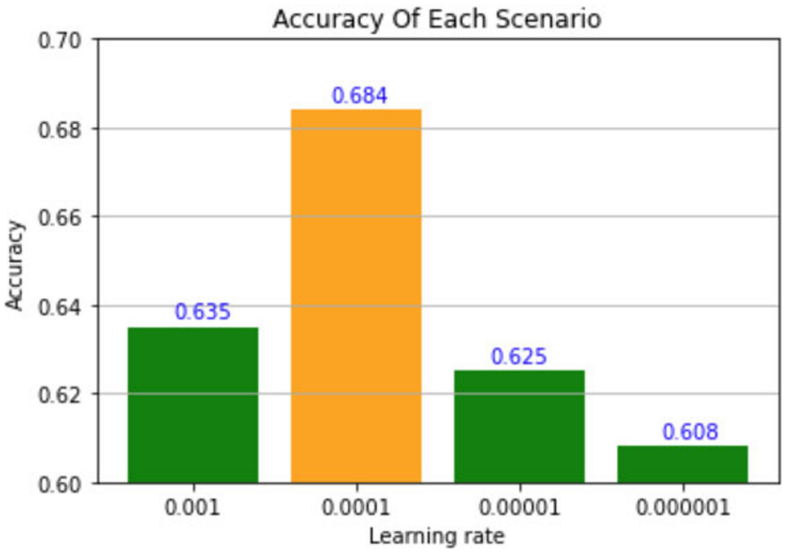

Fig. 5. Using the best accuracy of each scenario for comparison, scenario 2 with learning rate 0.0001, accuracy 0.684 is chosen to train the model.

5 Conclusion

5.1 Results

Our CNN model reached the overall 67.33% of training and 68.4% validation accuracy. The humerus image was correctly classified 64.28% for the abnormal class and 72.29% for the normal class. The training and validation accuracy is visualized in Fig. 6.

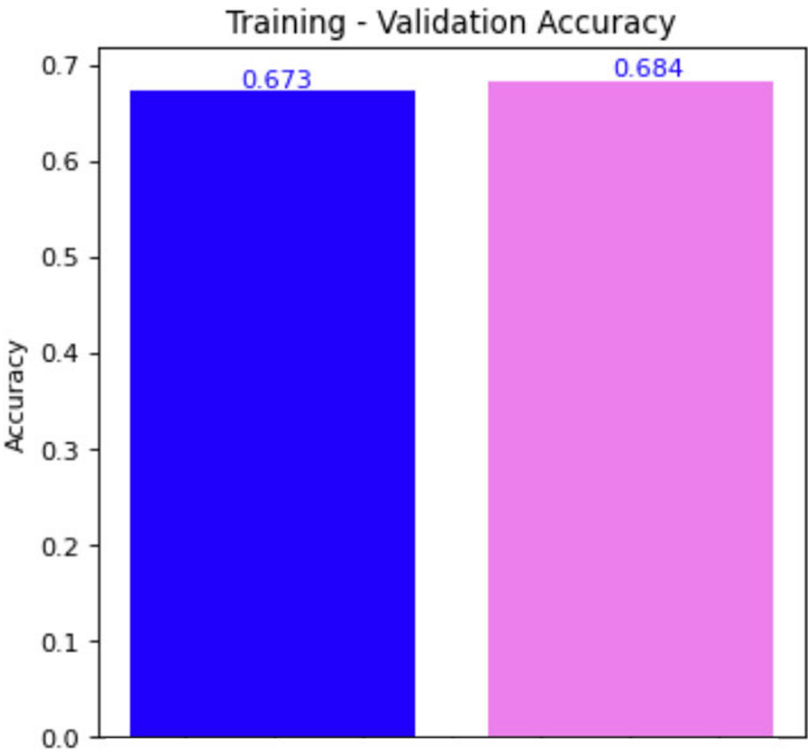

Fig. 6. The visualization of accuracy during training.

In the case of an unbalanced data set, the ROCAUC appears to be a more efficient measure than accuracy to evaluate performance. Our model achieved 0.659 and 0.655 for AUC training- validation respectively. The AUC is illustrated in Fig.7.

With 1,272 images in the training data, 288 images in valuation data, and 727 studies cases of humerus; We have shown the training results using a confusion matrix (Fig. 8). The humerus is correctly classified with 197/288 images. 90/140 images correctly predicted to be abnormal (TP); 50/140 abnormal images that are incorrectly predicted as normal (FN). 107/148 correctly predicted images are normal (TN) images; 41/148 normal images are incorrectly predicted to be abnormal (FP).

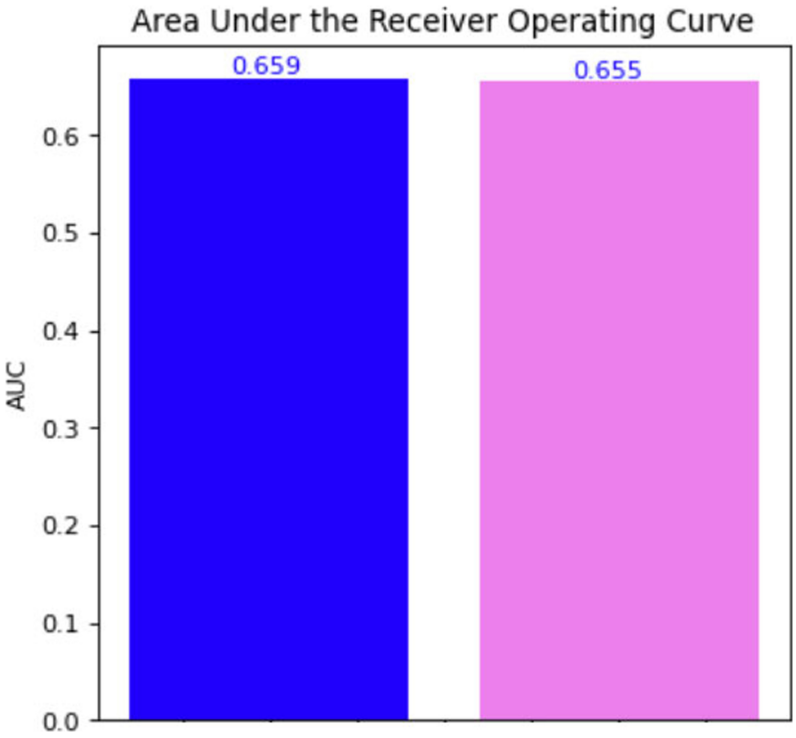

Fig. 7. The visualization of AUC during training.

The result of our model is better than the result of training on humerus studies by a 169-layer DenseNet baseline model (60.0%)[17]. In Table 3, MURA's model compares performance between 3 radiologists and model on the Cohen's kappa statistic. We highlight the result of MURA's model (red) and our model (green) on humerus studies to express performances.

Table 3. Comparison between MURA's model and our CNN model.

	Radiologist 1	Radiologist 2	Radiologist 3	MURA's Model	our model
Humerus	0.867	0.733	0.933	0.600	0.684

5.2 Discussion

Detecting abnormalities in musculoskeletal x-rays has important clinical applications. An abnormality detection model can be used to support the radiologist for faster review and approval. Besides, the detection of normal musculoskeletal

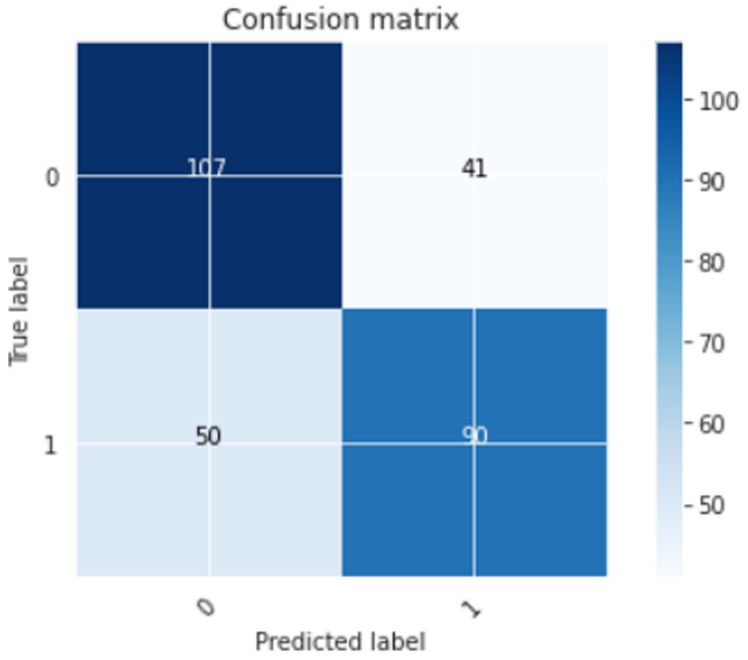

Fig. 8. 197/288 images of humerus are accurately classified at 68.4%.

helps to save time, treatment costs for patients. Therefore a computer-based automatic anomaly detection model can lead to disease interpretation through diagnostic imaging, removal of reducing agents, and standardization of quality.

We presented an approach to use Convolutional Neural Network to detect abnormality in musculoskeletal radiographs.

Due to a limited number of computational resources, we only implemented a shallow CNN architecture for image classification tasks on humerus in large MURA data sets. We hope our framework can help to classified musculoskeletal x-ray images to be better and contribute to the development of more diagnostic imaging methods. There are still many other machine learning methods and more in-depth research is needed on larger sets of images in the future.

References

1. Weinstein, S.L., Yelin, E.H., King, S.: The Burden of Musculoskeletal Diseases in the United States (BMUS), 3rd edn., p. 12 (2016)
2. Nedjar, I., El Habib Daho, M., Settouti, N., Saïd, M., Chikh, M.: Random forest based classification of medical x-ray images using a genetic algorithm for feature selection. J. Mech. Med. Biol. **15**(02), 1540025 (2015). https://doi.org/10.1142/S0219519415400254

3. Çamlica, Z., Tizhoosh, H.R., Khalvati, F.: Medical image classification via SVM using LBP features from saliency-based folded data. In: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), Miami, FL, pp. 128–132 (2015). https://doi.org/10.1109/ICMLA.2015.131
4. Shrivastava, K., Gupta, N., Sharma, N.: Medical image segmentation using modified K means clustering. Int. J. Comput. Appl. **103**, 12–16 (2014). https://doi.org/10.5120/18157-9341
5. Li, B.N., Chui, C.-K., Chang, S., Ong, S.: Integrating spatial fuzzy clustering with level set methods for automated medical image segmentation. Comput. Biol. Med. **41**, 1–10 (2010). https://doi.org/10.1016/j.compbiomed.2010.10.007
6. Nguyen, H.T., Huynh, H.T., Tran, T.B., Huynh, H.X.: Explanation of the convolutional neural network classifying chest X-ray images supporting pneumonia diagnosis. EAI Endorsed Trans. Context-Aware Syst. Appl., 1–7 (2020). https://doi.org/10.4108/eai.13-7-2018.165349
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In Proceedings of the IEEE CVPR, pp. 770–778 (2016)
8. Shickel, B., Tighe, P.J., Bihorac, A., Rashidi, P.: Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. IEEE J. Biomed. Health Inform. **22**(5), 1589–1604 (2018). https://doi.org/10.1109/JBHI.2017.2767063
9. Do, B.H., Langlotz, C., Beaulieu, C.F.: Bone tumor diagnosis using a Naïve Bayesian model of demographic and radiographic features. J. Digit. Imaging **30**(5), 640–647 (2017). https://doi.org/10.1007/s10278-017-0001-7
10. Ascalu, A., David, E.O.: Skin cancer detection by deep learning and sound analysis algorithms. A prospective clinical study of an elementary dermoscope. EBioMedicine **43**, 107–113 (2019). https://doi.org/10.1016/j.ebiom.2019.04.055
11. Toğaçara, M., Ergenb, B., Cömert, Z.: COVID-19 detection using deep learning models to exploit Social Mimic Optimization and structured chest X-ray images using fuzzy color and stacking approaches. Comput. Biol. Med. **121**, 103805 (2020). https://doi.org/10.1016/j.compbiomed.2020.103805
12. Wang, S., et al.: A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis. Eur. Respir. J., 1399–3003 (2020). https://doi.org/10.1183/13993003.00775-2020
13. Do Nhu, T., Joo, S.-D., Yang, H.-J., Jung, S., Kim, S.H.: Knee bone tumor segmentation from radiographs using Seg-Unet with dice loss. In: 25th International Workshop on Frontiers of Computer Vision (IW-FCV2019), Gangneung, South Korea, pp. 1–3 (2019)
14. Benndorf, M., Neubauer, J., Langer, M., Kotter, E., et al.: Bayesian pretest probability estimation for primary malignant bone tumors based on the Surveillance, Epidemiology and End Results Program (SEER) database. Int. J. Comput. Assist. Radiol. Surg. **12**, 485–491 (2017). https://doi.org/10.1007/s11548-016-1491-3
15. Varma, M., Lu, M., Gardner, R., et al.: Automated abnormality detection in lower extremity radiographs using deep learning. Nat. Mach. Intell. **1**, 578–583 (2019). https://doi.org/10.1038/s42256-019-0126-0
16. Larson, D.B., Chen, M.C., Lungren, M.P., Halabi, S.S., Stence, N.V., Langlotz, C.P.: Performance of a deep-learning neural network model in assessing skeletal maturity on pediatric hand radiographs. Radiology **287**(1), 313–322 (2017). https://doi.org/10.1148/radiol.2017170236. RSNA
17. Rajpurkar, P., et al.: MURA: large dataset for abnormality detection in musculoskeletal radiographs. In: 1st Conference on Medical Imaging with Deep Learning (MIDL 2018), pp. 1–10. arXiv:1712.06957 [physics.med-ph]

18. Panda, S., Jangid, M.: Improving the model performance of deep convolutional neural network in MURA dataset. In: Somani, A.K., Shekhawat, R.S., Mundra, A., Srivastava, S., Verma, V.K. (eds.) Smart Systems and IoT: Innovations in Computing. SIST, vol. 141, pp. 531–541. Springer, Singapore (2020). https://doi.org/10.1007/978-981-13-8406-6_51
19. Liang, S., Gu, Y.: Towards robust and accurate detection of abnormalities in musculoskeletal radiographs with a multi-network model. Sensors (Basel, Switzerland) **20**(11), 31–53. https://doi.org/10.3390/s20113153
20. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization (2014). arXiv:1412.6980v9
21. McHugh, M.L.: Interrater reliability: the kappa statistic. Biochem. Med. (Zagreb) **22**(3), 276–282 (2012)
22. Park, S.H., Goo, J.M., Jo, C.-H.: Receiver operating characteristic (ROC) curve: practical review for radiologists. Korean J. Radiol. **5**(1), 11–18 (2004)