Abnormality Detection in Musculoskeletal Radiographs

Ravi Raj Choudhary, Somesh Choudhary and Gaurav Meena
Department of Computer Science, Central University of Rajasthan, Rajasthan, India
E-mail: raviraj@curaj.ac.in, gaurav.meena@curaj.ac.in

Abstract. A radiographic study is a typical procedure utilized to find different kinds of abnormalities, where the identification of Musculoskeletal Abnormalities from the norm stands out as a crucial task. In this paper we suggest some deep learning methods to identify such Musculoskeletal Abnormalities using the MURA dataset which is one of the largest collections of upper extremity radiographs. We make use of DenseNet and VGG deep learning models to try to draw some unprecedented findings and results. In addition, we try to visualize all our results and findings using comparative graph study. We compare our model with models proposed during the Stanford ML Group MURA Competition using Cohen Kappa Statistic. The obtained results show that deep convolutional neural networks can achieve results which are very close and even better compared to current state-of-the-art models. We achieve comparable model performance to state-of-the-art performances in three of seven study types.

1. Introduction
Musculoskeletal abnormality detection in radiographic studies is a very important task in the medical field. A variety of diseases or abnormalities can be detected using radiographic studies and this is also the most common way to do it. If such studies are interpreted correctly then the clinicians can rule out abnormalities and rule out the need of further diagnostics. Also, it can help the clinicians to avoid errors while examining a study. In addition, most of the long-term pain, severe pain and disabilities are caused my musculoskeletal abnormalities, which makes the correct identification of such abnormalities a crucial task. Also, some medical centers also have by standing clinicians which work in only emergency scenarios, such clinicians do not have adequate experience and expertise for correct identification of abnormalities on radiographs. In case of busy medical centers, experienced radiologists may not be available. In such a case, clinicians can take hours of valuable time to assess radiographic images and chance of incorrect interpretation of musculoskeletal abnormality is also there which affects the patient care.

There are a number of research articles which discusses and investigates bone abnormality but most of those are limited to bone fracture only. Musculoskeletal abnormalities not only cover bone fracture, but it also covers a wide range of diseases like bone deformities, osteoporosis, congenital diseases, and different kinds of arthritis. There is not much research done on such abnormalities. MURA dataset, which contains thousands of radiographic studies for musculoskeletal abnormality detection, is crafted by Rajpurkar towards this research field and, as far as my knowledge goes, there is no other work for radiographic abnormality detection. The objective of our model is to make use of deep learning methods to find out musculoskeletal abnormalities in radiographic images provided in MURA dataset. We make use of DenseNet169...
and VGG16 which are frequently used models for visual tasks. The models are trained and evaluated using MURA dataset crafted by Rajpurkar. The results are compared with Stanford ML Group MURA dataset competition leaderboard (Link) as well as results obtained by Rajpurkar himself. In addition, we also try to visualize the results obtained by our models.

This paper is divided in 5 sections. Section 1 discusses about the basic overview of the problem of Musculoskeletal Abnormality Detection. Available literature describes in section II, section III and IV shows methodology and results of proposed work.

2. Literature Review
Detection of abnormality in a radiographic study is a very important task in the medical field: A variety of diseases or abnormalities can be detected using radiographic studies and this is also the most common way to do it. If such studies are interpreted correctly then the clinicians can rule out abnormalities and rule out the need of further diagnostics. Musculoskeletal conditions affect more than 1.7 billion people every year. These conditions can cause of severe pain, long-term pain and even disability. First publication regarding the use of Computer Automated Systems to perform medical task was made by Tian in 2003 in the paper entitled “Computing Neck Shaft Angle of Femur for X-Ray Fracture Detection”. In this paper, he made use of feature extraction technique to extract femur contour and classified femur based on neck-shift angle measured by it. This was the first major breakthrough for backing the use of Computer Automated Systems to perform medical task to get a second opinion [1].

Researchers from National University of Singapore made some advancements and proposed a new computer aided system in the paper entitled “Detection of Femur and Radius Fractures in X-ray Images”. The paper proposes an approach which uses multiple detection method for detecting fractures in the femur and radius. Different kinds of features are extracted using these methods. For the detection of fracture in femur, neck-shaft angle is extracted. The paper discusses extraction of other general features which may prove vital for detecting fractures. It makes use of SVM and Bayesian classifier for the classification task. [2]. Later in 2005, Lum, Leow and Chen proposed the idea of using a combination of classifiers to deploy a computer aided system to find bone fractures in x-ray images in the paper entitled “Combining Classifiers for Bone Fracture Detection in X-ray Images”. They managed to achieve up to 98% accuracy on test set [3]. Machine learning techniques were very popular for the task of image recognition and classification. But in recent times, CNNs gained vast popularity in computer vision related task due to them being more accurate. Deep convolutional neural networks perform great in image classification task due to their operability on large datasets, deeper models and better training algorithms.

In year 2009, Princeton University introduced a huge dataset of around 14 million images divided into around 21,000 categories. They invited people around the world to participate in the image classification task based on this dataset. They also proposed a classification model named ImageNet in the paper entitled “ImageNet: A Large-Scale Hierarchical Image Database” [4].

Over the course of 8 years between 2012-2017 this competition (ILSVRC: ImageNet Large Scale Visual Recognition Challenge) witnessed the introduction of well-known models such as AlexNet, VGG, GoogLeNet, ResNet etc. These models prove to be state-of-the-art in computer vision tasks. First use of CNNs for image classification task on ImageNet dataset was illustrated in 2012 in a paper entitled “ImageNet Classification with Deep Convolutional Neural Networks” [5]. This CNN model outperformed existing state-of-the-art which made CNNs the most favourite technique for image classification task. This model is popularly known as ConvNet. After the success of ConvNet, CNNs became first choice for image classification task. During the ILSVRC-2014, a new model named GoogLeNet was introduced in a paper entitled “Going Deeper with Convolutions” which is a 22 layered CNN[6]. The model was implemented such that it could obtain an architecture with increased depth and width while keeping computational cost at the
lower side. In year 2015, Visual Geometry Group of University of Oxford proposed a model named VGG in the paper entitled “Very Deep Convolutional Networks for Large-Scale Image Recognition”. This VGG model made use of CNNs which were up to 19 layers deep. VGG model achieved state-of-the-art standards in ILSVRC-2015[7].

The use of deep convolutional neural networks proved to be very successful in case of computer vision tasks. None of the other artificial intelligence techniques could achieve the benchmark set by CNN techniques. As the popularity of CNNs increased, more and more models of CNNs appeared on the surface. People started to use very deep convolutional neural networks which were of the depth two hundred and even more. Further researches in deep CNNs led to the craft of Densely Connected Neural Networks. Dense Convolutional Network or DenseNet was introduced in a paper entitled “Densely Connected Convolutional Networks” in year 2018 by Gao Huang of Cornell University and three other authors. DenseNet has up to 201 layered architecture. They proposed the use of Dense Block in the architecture. These DenseNet models outperformed existing state-of-the-art in nearly all the aspects. Although, these convolutional neural networks models used ImageNet dataset, which consists of non-medical images, for training, they could still work well for medical images also. Many medical researchers made use of these popular models to conduct researches on medical images also. U-Net is a convolutional neural network model which was introduced in 2015 in a paper entitled “U-Net: Convolutional Networks for Biological Image Segmentation”. They used a sliding window convolutional network method for their model and won first prize at ISBI cell tracking challenge 2015[8]. In year 2017, CheXNet model was introduced in a paper entitled “Radiologist-Level Pneumonia Detection on Chest X-ray using Deep Learning”[9] . They developed a convolutional neural network with 121 layers. The model was trained on CheX-ray14 dataset which contained 100,000 images with 14 diseases. The results of model were compared against four radiologists. This model achieved state-of-the-art in case of each 14 diseases. In the same year, researchers from The University of Adelaide published a paper entitled “Detecting hip fractures with radiologist-level performance using deep neural networks”. They proposed a 172-layered DenseNet model. The model achieved an accuracy of 0.99 on validation set[10].

All available researcher show that choosing a deep convolution neural network model for musculoskeletal abnormality detection task might prove to be a great selection. In the context of radiographic abnormality detection, only a little literature and research work is available. The major contribution is made by Rajpurkar and Irvin who together developed the model entitled “MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs” in which they used DenseNet architecture and their model outperformed best radiologist performance in wrist abnormality detection[11]. Several other participants deployed their models in the same Stanford ML Group MURA Dataset competition and managed to outperform Rajpurkar’s model. However, most of the participants opted for ensemble of models rather than single model. The leaderboard consists of top 70 participants and uses Cohen Kappa statistic to evaluate models. The performance of the models of these 70 participants lies between 0.843 to 0.518[12].

3. Methodology
In this section we explains the theoretical concepts behind the implementation of the model and it also covers the scope of the paper. We define some theoretical background which is essential to understand the theme. Different learning models used in the implementation and evaluation metrics used for performance analysis are described.

3.1. Artificial Neural Networks
Artificial Neural Network (ANN) falls under the category of supervised learning algorithms. It is also called Multi-layer Perceptron. It is designed to replicate the way the human brain works
to process information. An ANN comprises multiple layers. It has one input layer, one output layer, and at least one hidden layer. Perceptrons are the fundamental building blocks of the ANN, and each layer might contain several perceptrons. The following figure is an example of ANN.

![ANN with one hidden layer](image1)

**Figure 1.** ANN with one hidden layer

**3.2. Convolutional Neural Networks**

Convolutional Neural Networks belongs to the class of deep neural networks. So basically, it is a deep learning algorithm. It can learn weights and bias for a given input and can assign importance to different objects in an image. Also, it can differentiate various objects of the image. It consists of a Convolutional layer which reduces the number of parameters in ANN which gives CNN an upper hand over ANN. A convolution operation can extract high-level features like edges. For this task, it makes use of a filter or kernel which moves over the complete input image with a stride value. A convolution operation produces feature maps. A CNN also has pooling layers which are used to reduce the dimensionality of feature maps. The following figure shows an example of CNN. CNN are a quite broad subject where there are many configurations of a single network architecture and we cannot say with full confidence that on

![Convolutional Neural Network Architecture](image2)

**Figure 2.** Convolutional Neural Network Architecture
which configuration the architecture performs the best for a given task. But there are some models which are well-known for their great performance in image classification task that we are currently dealing with. In general, these models are trained and evaluated on ImageNet dataset. In this section, we are going to describe two models which are used in this paper.

3.2.1. VGG  VGG model was introduced by Oxford Visual Geometry Group researchers. VGG is short for Visual Geometry Group model. The salient feature of this model is to add multiple convolutional layers of kernels of size 3*3. This model shows good overall performance on ImageNet dataset. This model generalizes very well for image classification task on other dataset also which makes it a potential solution for our problem. There are many variations of this model based on number of layers used in the architecture. Most popular variants are VGG16 and VGG19. VGG16 has 16 layers whereas VGG19 is a 19 layered architecture. In this paper, we are using VGG16 architecture. Following figure shows overall architecture of VGG. VGG16 is a simpler model than other deep learning models. It uses kernels of size 3x3 for convolution. It uses max pooling to reduce the volume further. There are two fully-connected layers with 4,096 nodes each. And it makes use of softmax activation function to squeeze the output between 0 and 1. The input to VGG16 is images of size 224*224. The size of images is reducing as they pass through the network. Following figure shows the architecture of VGG16.

![Figure 3. Architecture of VGG](image)

3.2.2. DenseNet DenseNet architecture was introduced by Gao Huang of Cornell University and three other authors in a research paper entitled “Densely Connected Convolutional Networks”. DenseNet is built on research that CNNs are more efficient and accurate for training task if the links between layers close to inputs and layers close to outputs are shorter. To implement this research, authors introduced a Dense Block, where every layer treats the feature maps produced by its previous layers as an input for itself. And, the feature maps produced by each layer are used as inputs for all the subsequent layers. This reduces the training time because all the layers have direct access to gradient of loss function. The following figure shows the example of Dense Blocks. The traditional CNNs have only one link between each layer and it’s following layer that means L number of links for L number of layers. Whereas, DenseNet has $\frac{L(L+1)}{2}$ direct links, L being number of layers. These Dense Blocks are linked together using transition layers. Transition layers perform convolution and pooling operations. The following figure shows overall architecture of DenseNet. Just like VGG, DenseNet also has
Figure 4. Architecture of VGG16[13]

Figure 5. Example of five-layer dense block

Figure 6. Architecture of three-layer deep DenseNet
different models depending upon the layers used in the model architecture. Some of those are DenseNet121, DenseNet169, DenseNet201 and DenseNet264.

4. Proposed Method
In this paper, we implement two deep learning models, VGG16 and DenseNet169. Our proposed models are somewhat different from existing models. Following section provides details about both models.

4.1. VGG16
The model is forged using 5 convolution blocks. After each convolutional block, pooling operation is performed. The type of pooling performed is Max-pooling with stride value 2. First convolutional block consists of 64 filters which is the depth of the block. Each filter is of size 3x3. Number of convolutional layers in the block is 2.

(i) Activation function used at the end of block is ReLu activation. Second convolutional block consists of 128 filters which is the depth of the block. Each filter is of size 3x3. Number of convolutional layers in the block is 2. Activation function used at the end of block is ReLu activation.

(ii) Third convolutional block consists of 256 filters which is the depth of the block. Each filter is of size 3x3. Number of convolutional layers in the block is 3. Activation function used at the end of block is ReLu activation.

(iii) Fourth convolutional block consists of 512 filters which is the depth of the block. Each filter is of size 3x3. Number of convolutional layers in the block is 3. Activation function used at the end of block is ReLu activation.

(iv) Last convolutional block consists of 512 filters which is the depth of the block. Each filter is of size 3x3. Number of convolutional layers in the block is 3. Activation function used at the end of block is ReLu activation.

At the output layers, sigmoid activation function is used which gives an output value between 0 and 1.

4.2. DenseNet169
A total of 5 dense blocks are used in the architecture. A 1x1 convolution is applied at each transition layer. Sigmoid function is used at the output layer to obtain a output value between 0 and 1. The following table shows the architecture of used DenseNet169 model.

4.3. Experimental Setup
We trained our VGG16 and DenseNet169 models on MURA dataset. The MURA dataset needs not to be partitioned in training and validation set since it is already partitioned. Training set about 90% of all the images and validation set has remaining 10% images. The model has to be trained on study basis and not on per image basis because every study has multiple images of same extremity of same patient taken from different view angles. A single image of a study does not guarantee the detection of abnormality. A different view angle might show an abnormality which is not visible in one view as per Figure 8. The following figure shows basic pipeline used in this paper. We resize the variable size images to a fixed size of 224*224 pixels according to ImageNet standards and then we augment the images as mentioned in section 5.1.1. For the evaluation of our models, we used the quantitative evaluations metrics described in section 5.2, which then are compared with our baseline model given by Rajpurkar et al. initial work.
| Layer               | Output Size | Layer Description                  |
|---------------------|-------------|------------------------------------|
| Convolution         | 112 × 112   | 7 × 7 conv, stride 2               |
| Pooling             | 56 × 56     | 3x3 max pool, stride 2             |
| Dense Block(1)      | 56 × 56     | 1 × 1 Conv                         |
| Transition Layer(1) | 56 × 56     | 2x2 avg pool, stride 2             |
| Dense Block (2)     | 28 × 28     | 1 × 1 Conv                         |
| Transition Layer(2) | 28 × 28     | 2x2 avg pool, stride 2             |
| Dense Block (3)     | 14 × 14     | 1 × 1 conv                         |
| Transition Layer(3) | 14 × 14     | 2x2 avg pool, stride 2             |
| Dense Block (4)     | 7 × 7       | 1 × 1 conv                         |
| Classification Layer| 14 × 14     | 7 × 7 global avg pool, 1000D fully-connected, sigmoid |

Table 1. Architecture of DenseNet169

Figure 7. Block Diagram of Model

4.4. Dependencies
- Keras 2.1.6 is used with TensorFlow as backend
- TensorFlow 1.15.0 GPU version is used
- NVIDIA 940MX GPU is used and in order to run TensorFlow on this GPU, following applications are prerequisite:
  - Compute Unified Device Architecture (CUDA) 10.1
  - NVIDIA CUDA Deep Neural Network Library (cuDNN) 7.6.5
- Python version 3.6 is used
- Scipy 1.1.0
- Numpy
- Pandas
- OpenCV-python
- Keras-vis
- Matplotlib
- Tqdm
It is better to setup a separate virtual environment for this task so that these particular versions of dependencies do not interfere with existing configurations of the machine. We used Anaconda (64-bit version) to setup a separate virtual environment for deploying our models. Further we require internet connectivity while running the models so it can download ImageNet weights. If the machine is not GPU-enabled, then the user can change “tensorflow-gpu==1.15” to “tensorflow==1.15” written in ‘prerequisite.txt’ file to make use of CPU only. These dependencies can be installed by simply running the command ‘pip install -r requirements.txt’ and ‘pip install -r prerequisite.txt’ in the command prompt of corresponding software installed on the machine.

5. Results
5.1. Data Set
We are using MURA dataset in this paper which is a huge collection of over 40 thousand radiographic images. The objective here is introducing a classification model to classify whether the study has any musculoskeletal abnormality or not. By musculoskeletal abnormality we mean fractures, degenerative bone diseases, inserted hardware and other kind of abnormalities. The dataset contains only upper extremity radiographic studies where each study contains one or more images showing different radiographic views. The input from the dataset is a study, which is basically a set of images, and the output by the model is a binary prediction \( y \in \{0, 1\} \), where 0 indicates the absence of abnormality and 1 indicates presence of abnormality. Each study belongs to one of the seven standard upper extremity radiographic study types: Elbow, forearm, finger, hand, humerus, shoulder and wrist. Each study was labeled as abnormal or normal manually by board certified radiologists of Stanford Hospital. Table 2 summarizes the distribution of abnormal and normal studies. The dataset is already splitted in training and validation sets. Dataset contains a total of 14,863 studies conducted on a total of 12,173 patients. Total number of images in the dataset is 40,561.

5.1.1. Data Augmentation
As we are dealing with upper extremity studies, it is a possibility that the dataset contains unbalanced studies on training set because it might take only one side of the extremity in account. For example, it might have studies of only right hand. So,
augmentation of dataset images is crucial so that high-level features are learned by our models. These features are invariant to transformations like horizontal flips and small degree rotations. Such transformations can occur in real-life scenarios also at the time of capturing radiographs. We make two such transformations which are as follows:

(i) Horizontal Flip: We randomly make horizontal flips on studies to make our data symmetric so that the model can learn both sides of the extremity.

(ii) Small Rotations: Random rotations up to 30 degrees are applied on studies. It enforces the models to deal with small variations that might occur in a normal radiographic study.

5.1.2. Loss Function

The process of learning from data or to find the solution to the problem, ideally, we have labels making learning Supervised Learning; in this process training data is used to generate a new predictor function. This function attempts to map input data to labels. We expect this function to generalize for unseen data also. For this, we produce some output label, desired output, for every data point in our dataset. Obviously, the predicted labels are not identical to actual labels, so we try to calculate error value by comparing predicted labels with actual labels. To move close to a good solution, we always try to minimize this error value. The set of operations that we propose to find the error value is known as Loss function. Loss functions might be different for different models depending on the data. So basically, loss functions are generally generated depending upon properties of any given dataset.

5.1.3. Binary Cross Entropy

Cross-entropy loss is a metric to evaluate the performance of a classification model having probability value as output. As the predicted label moves away from actual label, cross entropy loss increases. So, lower loss value indicates a better model. For a perfect model loss value is 0. Cross entropy for a binary classification model is given by:

\[-y \log(P) + (1 - y) \log(1 - P)\]  

5.2. Quantitative Results

We compare results obtained by VGG16 and DenseNet169 models with the baseline proposed by Rajpurkar. We also compare overall performance of each of our models with the leader board of Stanford ML Group MURA Dataset Competition.

5.2.1. Baseline

The baseline model which we are using to compare our results against is the model proposed by Rajpurkar. This model obtained a Kappa value of 0.705, sensitivity of 0.815 and specificity of 0.887. The table below shows the results obtained by baseline model:

| Study     | Elbow | Finger | Forearm | Hand | Humerus | Shoulder | Wrist |
|-----------|-------|--------|---------|------|---------|----------|-------|
| Kappa     | 0.710 | 0.389  | 0.737   | 0.851| 0.600   | 0.729    | 0.931 |

Table 2. Performance of Baseline model

5.2.2. VGG16

We trained VGG16 model over 70 epochs with a learning rate of 0.00001 and default parameters. The results obtained are tabulated below: VGG16 achieved overall accuracy of 0.767, sensitivity of 0.701, specificity of 0.827 and a Kappa score of 0.532. Whereas, the baseline model achieved considerably higher scores of sensitivity, specificity and Kappa compared to our model. The following table shows the comparison between results obtained by VGG16 and baseline model. Table 4 clearly shows that baseline model clearly outperforms VGG16 model in all the metrics. Nevertheless, in case of finger study, baseline model obtains a Kappa of 0.389.
Table 3. Result analysis of VGG16

| Study   | Accuracy | Sensitivity | Specificity | Precision | Kappa  |
|---------|----------|-------------|-------------|-----------|--------|
| Elbow   | 0.800    | 0.760       | 0.838       | 0.821     | 0.599  |
| Finger  | 0.750    | 0.732       | 0.771       | 0.786     | 0.501  |
| Forearm | 0.750    | 0.668       | 0.833       | 0.801     | 0.501  |
| Hand    | 0.747    | 0.539       | 0.892       | 0.778     | 0.453  |
| Humerus | 0.753    | 0.771       | 0.736       | 0.734     | 0.507  |
| Shoulder| 0.714    | 0.669       | 0.757       | 0.729     | 0.427  |
| Wrist   | 0.830    | 0.749       | 0.895       | 0.853     | 0.652  |
| Overall | 0.767    | 0.701       | 0.827       | 0.789     | 0.532  |

Table 4. Comparison of results between VGG16 and Baseline

| Model   | Accuracy | Sensitivity | Specificity | Precision | Kappa  |
|---------|----------|-------------|-------------|-----------|--------|
| Baseline| -        | 0.815       | 0.887       | -         | 0.705  |
| VGG16   | 0.767    | 0.701       | 0.827       | 0.789     | 0.532  |

whereas VGG16 obtains a Kappa of 0.501. So VGG16 was able to outperform baseline model in only one case. Still VGG16 showed a respectable performance. We visualised the performance of VGG16 graphically using number of epochs versus metrics graph. The following figure shows that graph. We also visualised the loss functions of training and validation set over the epochs.

Figure 9. Visual analysis of VGG16 performance

Following Figure 10 shows that graph. In conclusion, we can say that VGG16 could not perform excellently on medical radiographic studies, but it still showed respectable performance.

5.2.3. DenseNet-169 We trained DenseNet model over 40 epochs with initial learning rate of 0.0001, ImageNet weights and default parameters. Every time the validation loss hits a plateau, learning rate is decayed by a factor of 10 in subsequent epoch. The results obtained by DenseNet169 are tabulated below. DenseNet169 obtained an accuracy score of 0.809, sensitivity of 0.721, specificity of 0.890, precision of 0.858 and a Kappa score of 0.616. It achieved specificity more than the baseline model and a very considerable accuracy. The following table shows the comparison between the performance of baseline model and DenseNet169. Our DenseNet169
Table 5. Result analysis of DenseNet169

| Study  | Kappa (DenseNet169) | Kappa (Baseline) |
|--------|---------------------|------------------|
| Elbow  | 0.711               | 0.710            |
| Finger | 0.558               | 0.389            |
| Forearm| 0.674               | 0.737            |
| Hand   | 0.523               | 0.851            |
| Humerus| 0.666               | 0.600            |
| Shoulder| 0.512              | 0.729            |
| Wrist  | 0.680               | 0.930            |
| Overall| 0.616               | 0.705            |

Table 6. Comparison of results between DenseNet169 and baseline

model outperforms baseline model in the study of elbow, finger and humerus. It also outperforms the best radiologist’s performance in finger study. So, DenseNet169 shows a good overall performance. We visualised the performance of DenseNet169 graphically using number of epochs versus metrics graph. The following figure shows that graph. We also visualised the loss functions of training and validation set over the epochs. Following Figure 12 shows that graph. As we mentioned previously, learning rate changes whenever the loss function of validation set hits a plateau. Following figure shows the visualization of decay of learning rate overtime.
Figure 11. Visual analysis of DenseNet169

Figure 12. Loss function of DenseNet169

Figure 13. Visualization of learning rate decay
6. Conclusion

In this paper, we looked at the effect of using well-known models which are trained on ImageNet dataset, a dataset consisting non-medical image, to detect musculoskeletal abnormalities using the MURA dataset. We used VGG16 and DenseNet169 models for the cause. Although, our models outperformed baseline models in some aspects, but none of these could beat the overall performance of baseline model. But, on the bright side, our models achieved a very good accuracy score. DenseNet169 performed better than our baseline in many aspects. It even gained higher specificity score than baseline model. Our models secured respectable positions on Stanford University ML Group MURA Dataset competition leaderboard. In conclusion, although these models and even current state-of-the-art models have not achieved the performance level of clinicians, but they may prove to be of vital help for providing second opinion and even for prioritising work in emergency scenarios.

References

[1] Tian T P, Chen Y, Leow W K, Hsu W, Howe T S and Png M A 2003 Computer Analysis of Images and Patterns ed Petkov N and Westenberg M A (Berlin, Heidelberg: Springer Berlin Heidelberg) pp 82–89 ISBN 978-3-540-45179-2
[2] Lim S E, Xing Y, Chen Y, Leow W K, Howe T S and Png M A 2004
[3] Vineta Lai Fun Lum, Wee Kheng Leow, Ying Chen, Tet Sen Howe and Meng Ai Png 2005 IEEE International Conference on Image Processing 2005 vol 1 pp I–1149
[4] Dong J, Dong W, Socher R, Li L, Kai Li and Li Fei-Fei 2009 2009 IEEE Conference on Computer Vision and Pattern Recognition pp 248–255
[5] Krizhevsky A, Sutskever I and Hinton G E 2012 Advances in Neural Information Processing Systems 25 ed Pereira F, Burges C J C, Bottou L and Weinberger K Q (Curran Associates, Inc.) pp 1097–1105 URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep.pdf
[6] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Proceedings of the IEEE conference on computer vision and pattern recognition pp 1–9
[7] Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition (Preprint 1409.1556)
[8] Ronneberger O, Fischer P and Brox T 2015 Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 ed Navab N, Hornegger J, Wells W M and Frangi A F (Cham: Springer International Publishing) pp 234–241 ISBN 978-3-319-24574-4
[9] Chandra T B and Verma K 2020 Proceedings of 3rd International Conference on Computer Vision and Image Processing (Springer) pp 21–33
[10] Gale W, Oudkerk-Rayner L, Carneiro G, Bradley A P and Palmer L J 2017 CoRR abs/1711.06504 (Preprint 1711.06504) URL http://arxiv.org/abs/1711.06504
[11] Rajpurkar P, Irvin J, Bagul A, Ding D, Duan T, Mehta H, Yang B, Zhu K, Laird D, Ball R L, Langlotz C, Shpanskaya K, Lungren M P and Ng A Y 2017 Mura: Large dataset for abnormality detection in musculoskeletal radiographs (Preprint 1712.06957)
[12] https://stanfordmlgroup.github.io/competitions/mura/
[13] 2009 https://mc.ai/the-architecture-behind-cnn-and-vggnet-16/