Convolutional Neural Networks for Computer-Aided Detection of Musculoskeletal Abnormalities

Rohan Sikand
Ranney School

Abstract

With over 2.6 billion musculoskeletal related injuries occurring each year, the need for trained doctors is extensive–especially in rural areas where doctors aren't always available. This experiment aims to test the ability of machine learning algorithms, specifically convolutional neural networks, to accurately detect abnormalities in musculoskeletal radiographs. The Stanford MURA dataset, consisting of 14,863 musculoskeletal radiographs, was used as input data. This experiment aims not only to evaluate the use of machine learning to detect abormalities, but also to decipher which CNN architectures perform well for this task. Specifically, four different CNN architectures were tested: VGG-16, VGG-19, ResNet, and Xception. Predictions were made on a test set for each model on new radiographs. To show the results and to best compare the architectures, training and validation accuracy, training and validation loss, and a confusion matrix are shown for each architecture. From each confusion matrix, several metrics were calculated including sensitivity, specitivity, precision, classification accuracy, F_1 score, and Matthews correlation coefficient (MCC). The results are statistically significant and show that CNNs are applicable for detecting abnormalities in musculoskeletal radiographs.

1 Introduction

The musculoskeletal system is an organ system that yields the human body the ability to move. Furthermore, the system provides support, protection, and stability to the body. The system is comprised of connective tissues including bones, cartilage, ligaments, joints, tendons, and other connective tissues. Together, these elements, forming the organ system, work in unison to provide the body with important functions.

As with other organ systems, injuries occur and may result in significant abnormalities to the musculoskeletal system. Such cases may hinder the function and overall effectiveness of the system. Since the musculoskeletal system is closely intertwined and connected with various other systems and organs in the body, abnormalities can be difficult to diagnose. This can be imperative since proper diagnosis is vital for necessary treatment and overall well-being of the patient. In specific, injuries sustained to the system in the upper-body, where most organs and systems are compactly fit, diagnosis through non-invasive procedures, such as a radiograph, can become untrustworthy and inaccurate.

Preprint. Work in progress.

Applications of machine learning and deep learning to clinical medicine are increasing in popularity. The task of training an algorithm on a plethora of data points can yield high classification for certain tasks–even more accurate than human capability. In specific, the field of deep learning, the biggest subfield of machine learning, is continuing to revolutionize the field due to one particular type of algorithm: a neural network. A neural network can be used for several machine learning tasks such as classification, segmentation, and localization. Because of the powerful abilities of a neural network, large datasets can be used for training an algorithm in an efficient manner.

With image classification increasingly becoming a popular task for computer vision technologies, new algorithms and architectures are constantly being researched. Traditionally, image classification worked through feature extraction in which hand-engineered algorithms, such as Histograms of Oriented Gradients (HOG) (1) and Local Binary Patterns (LBPs) (2), choose and quantify components of the image (i.e., shape, texture). Once these features are selected, the classifier is then trained and evaluated for performance (3). However, the present-day typical approach to image classification problems is to use a variant of a neural network: a convolutional neural network (CNN) (4). CNNs utilize the convolution operation in which the features are automatically learned through filters in hidden layers–therefore bypassing the need for hand-engineering, making convolutional neural networks an efficient end-to-end model (5).

Since convolutional neural networks are extremely powerful for image data, and since musculoskeletal abnormalities can be difficult to accurately diagnose through the human eye alone, this experiment aims to test the abilitiy of convolutional neural networks for diagnosing msuculoskeletal abnormalties. With this intent, several popular and high-performing CNN architectures were tested. Each are described in greater detail in section 2.

For this experiment, we aimed not only to evaluate the use of convolutional neural networks to detect musculoskeletal abormalities, but also to decipher which CNN architectures perform well for this task. To formulate our hypothesis, we analyzed the use of CNNs applied in different medically related scenarios. In (6), a inception recurrent residual convolutional neural network was implemented to classify breast cancer through histopathological images. A convolutional neural network was used to accurately diagnose pneumonia at a radiologist level in (7). Furthermore, in (8), a deep convolutional neural network was applied to automatically detect and diagnose seizures through EEG signals. If convolutional neural networks work well experimentally in other medical image analysis tasks, then convolutional neural networks will perform well for detecting musculoskeletal abnormalities in radiographs.

2 Methods

To test the performance of the use of convolutional neural networks for detecting abnormalities in musculoskeletal radiographs, several CNN architectures (models) were implemented. Each of the models were trained, validated, and tested with the dataset described in section 2.1. These models are trained through supervised learning where the input data points are musculoskeletal radiographs labeled with their associated class (abnormal or normal). After each model was trained using the same training and validation data, the models were tested on the test data, forming predictions for completely new musculoskeletal radiographs. From this, multiple metrics were calculated and incorporated in section 3. Figure 1 provides a schematic overview of the experimental process.

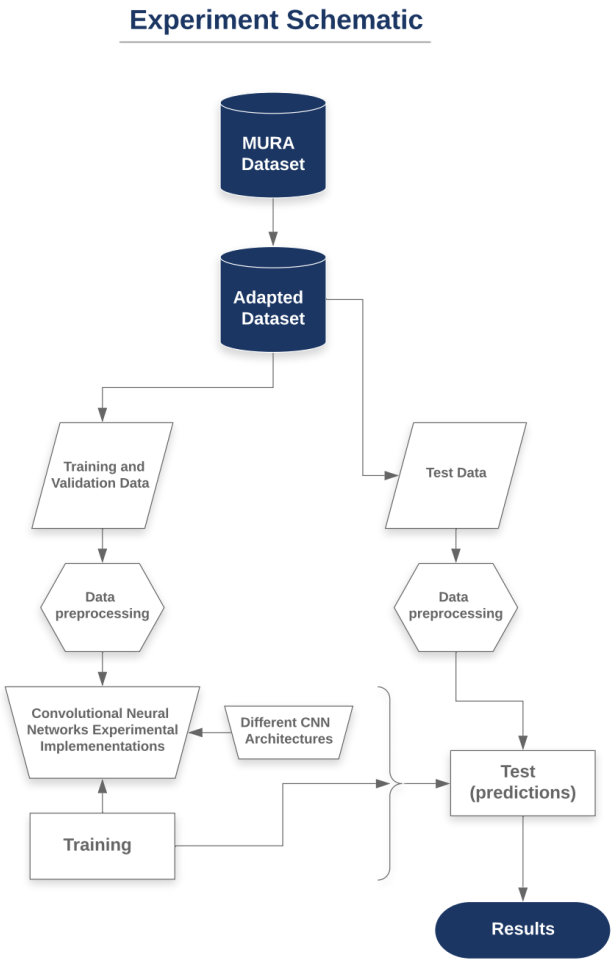

Figure 1: Complete experiment schematic, showcasing the steps and methodology taken. The original dataset was adapted to make this experiment into a binary classification problem independent of the radiograph category. The dataset was split and each radiograph was preprocessed. Several convolutional neural network architectures were experimentally implemented and trained. Each model then made test predictions which formulate the results listed in section 3.

2.1 Data and Preprocessing

The dataset that was used for this experiment was adapted from the MURA dataset which was created by the Stanford ML Group. MURA is a large dataset of musculoskeletal radiographs containing 40,561 images from 14,863 studies, where each study is manually labeled by radiologists as either normal or abnormal (9). The images were broken down into training (36,748 images), validation (3,197 images), and testing (60 images). Table 1 shows a tabular representation of the studies.

Although the original dataset contains separate directories for each of the seven categories of musculoskeletal radiographs (listed in table 1), a new directory was formed in which all radiographs were combined then separated into two classes: abnormal or normal. Thus, our experiment aims to solve a binary classification problem for all musculoskeletal radiographs independent from its category.

Table 1: The MURA dataset obtained from Stanford Machine Learning Group consists of 9,045 normal and 5,818 abnormal musculoskeletal radiographic studies of the upper extremity including the shoulder, humerus, elbow, forearm, wrist, hand, and finger. Table 1 shows the breakdown of the dataset for both training and validation of all extremities.

	Training: Normal	Training: Abnormal
Elbow	1094	660
Finger	1280	655
Hand	1497	521
Humerus	321	271
Forearm	590	287
Shoulder	1364	1457
Wrist	2134	1326

	Validation: Normal	Validation: Abnormal
Elbow	92	66
Finger	92	83
Hand	101	66
Humerus	68	67
Forearm	69	64
Shoulder	99	95
Wrist	140	97

In data science and machine learning, the act of data preprocessing is a fundamental step. It involves taking raw data and transforming it into readable and accurate data for the computer. Doing so will significantly increase accuracy. Thus, for this experiment, several data preprocessing techniques were performed.

The first technique was to resize each image to 224 x 224 pixels to meet the required input size for the neural network architectures used in this experiment–with the exception of the Xception architecture which uses an input image size of 299 x 299 pixels.

To better the image quality of each radiograph, image contrasting was applied. To contrast the images appropriately, Adaptive Histogram Equalization (AHE) is generally applied. However, in highly-concentrated areas, AHE tends to overamplify the contrast in constant or near-constant areas of an image. Thus, AHE was adjusted to Contrast Limited Adaptive Histogram Equalization (CLAHE)–a variant of AHE in which limits are applied to contrast amplification (this is especially helpful in reducing amplification in near-constant areas on image) (10). Figure 2 shows before and after contrasting of a sample image from the the dataset.

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are the primary neural network model used in most deep learning image classification problems and other computer vision tasks. Since the goal of this project is to use machine learning for differentiating between abnormal and normal musculoskeletal radiographs, convolutional neural networks were explored and implemented. CNNs are an end-to-end model, meaning it receives images (in pixel form) on one end to a class prediction at the other end.

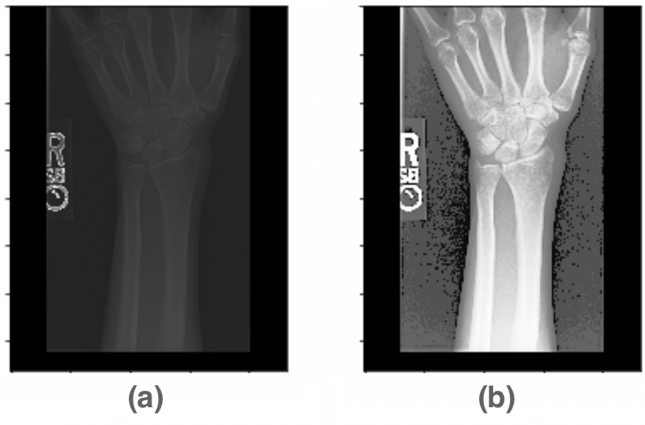

Figure 2: Before (a) and after (b) applying Contrast Limited Adaptive Histogram Equalization (CLAHE) to a musculoskeletal radiograph,

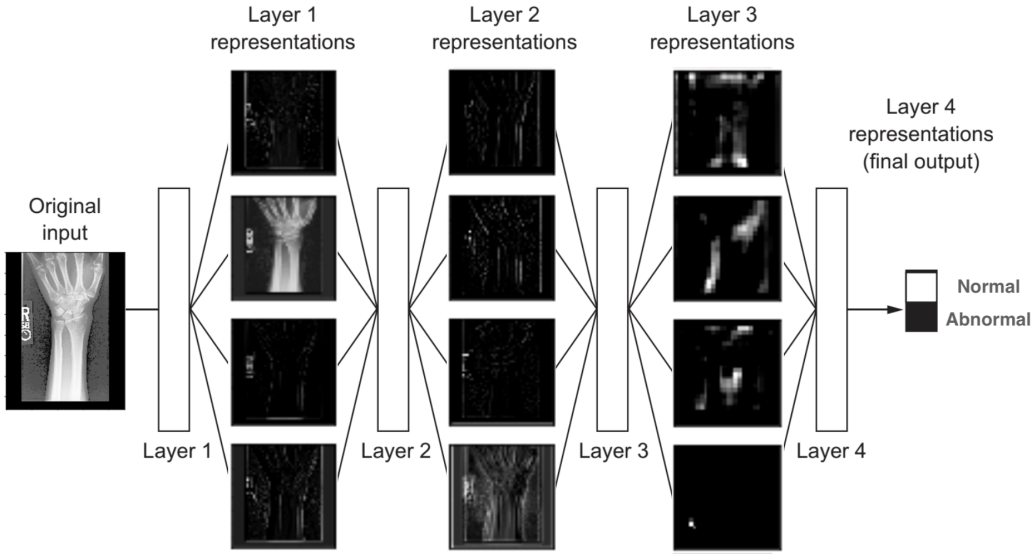

Figure 3: Convolutional Neural Networks contain additional layers used for automatic feature learning specifically for image data. This figure shows a visual representation of a CNN learning features of a musculoskeletal radiograph

Like artificial neural networks, convolutional neural networks consist of neurons that have learnable weights and biases. However, convolutional neural networks contain additional new layers (such as convolutional layers), which are engineered to learn and store representations in image data specifically. Figure 3 shows a sample representation of a CNN learning features of a musculoskeletal radiograph. Convolutional neural networks increase accuracy over traditional machine learning methods since it learns these features automatically rather than through hand-engineering.

The constituents of a CNN can be divided into three main types of layers: Convolutional Layers, Pooling Layers, and Fully-Connected Layers (which are the standard layers used in an ANN). These layers stacked together form a CNN architecture.

The convolutional layers are the base layers. These layers use a set of learnable filters (also known as kernels) of set width, height, and depth. The filters then slide, or convolve, across the image creating 2-dimensional activation maps which give the responses of each filter at every spatial position. Each layer will learn filters that activate when they see certain visual features and patterns. Each filter produces an activation map which are then stacked to produce the output of the layer (11).

Pooling layers, which generally follow convolutional layers, aim to simplify the information in the output from the convolutional layer (4). The pooling layer takes the feature maps of the convolutional layer and condenses it. There are different types of pooling, but the main one is max-pooling. In max-pooling, the pooling layer outputs the maximum activation of the filter.

Putting together these successive layers forms the base of a convolutional neural network. The final layer is a fully-connected dense layer (as used in ANNs) used to output the predicted class of the image.

2.3 CNN Architectures Experimental Implementations

Several different CNN architectures were tested (all of the layers in all of the CNN architectures listed below are visually represented in figure 4):

2.3.1 VGGNet

In (12), Simonyan et al., investigate the effect of the convolutional network depth in the large-scale image recognition setting. In the paper, they provide a detailed evaluation of two similar architectures with different depths–one with 16 layers (VGG-16) and one with 19 layers (VGG-19). Both of the models performed well in the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (13).

The main difference between the two VGG models and high performing models before it (such as AlexNet), is the amount of layers in the network and the size of the convolutional filters. Both the 16-layer VGG-16 and 19-layer VGG-19 are significantly deeper than the 8-layer AlexNet. Both models contains several blocks of convolutional layers followed by a max-pooling layer with VGG-19 containing three more convolutional layers. Another difference in VGGNet compared to AlexNet is that AlexNet uses a single 11 x 11 convolutional filter per 11 x 11 area resulting in 121 parameters ($11 * 11 = 121$), whereas both VGG models use five 3 x 3 convolutional filters per 11 x 11 area resulting in 45 parameters ($3 * 3 * 5 = 45$)–a 63% decrease (14). Thus, the effective receptive field is the same with more smaller filters and fewer parameters. Fewer parameters is important since it reduces overfitting.

2.3.2 ResNet

Adding more layers to a network's architecture, increases the difficulty of training. In some cases, when network depth increases, accuracy gets saturated. In (15), ResNet is introduced to overcome this problem. It achieves this by reformulating the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions (15). ResNet contains several variants each with different depth. For this experiment, a ResNetV2-152 was experimentally implemented.

2.3.3 Xception

The Xception architecture, introduced in (16), uses depth-wise separable convolutions to reduce overfitting. The architecture consists of 22,910,480 parameters and 126 layers. Different from the

other architectures tested in this experiment, the input size is 299 x 299 pixels. The architecture achieved a top-1 accuracy of 0.790 and top-5 accuracy of 0.945 on ImageNet data (13).

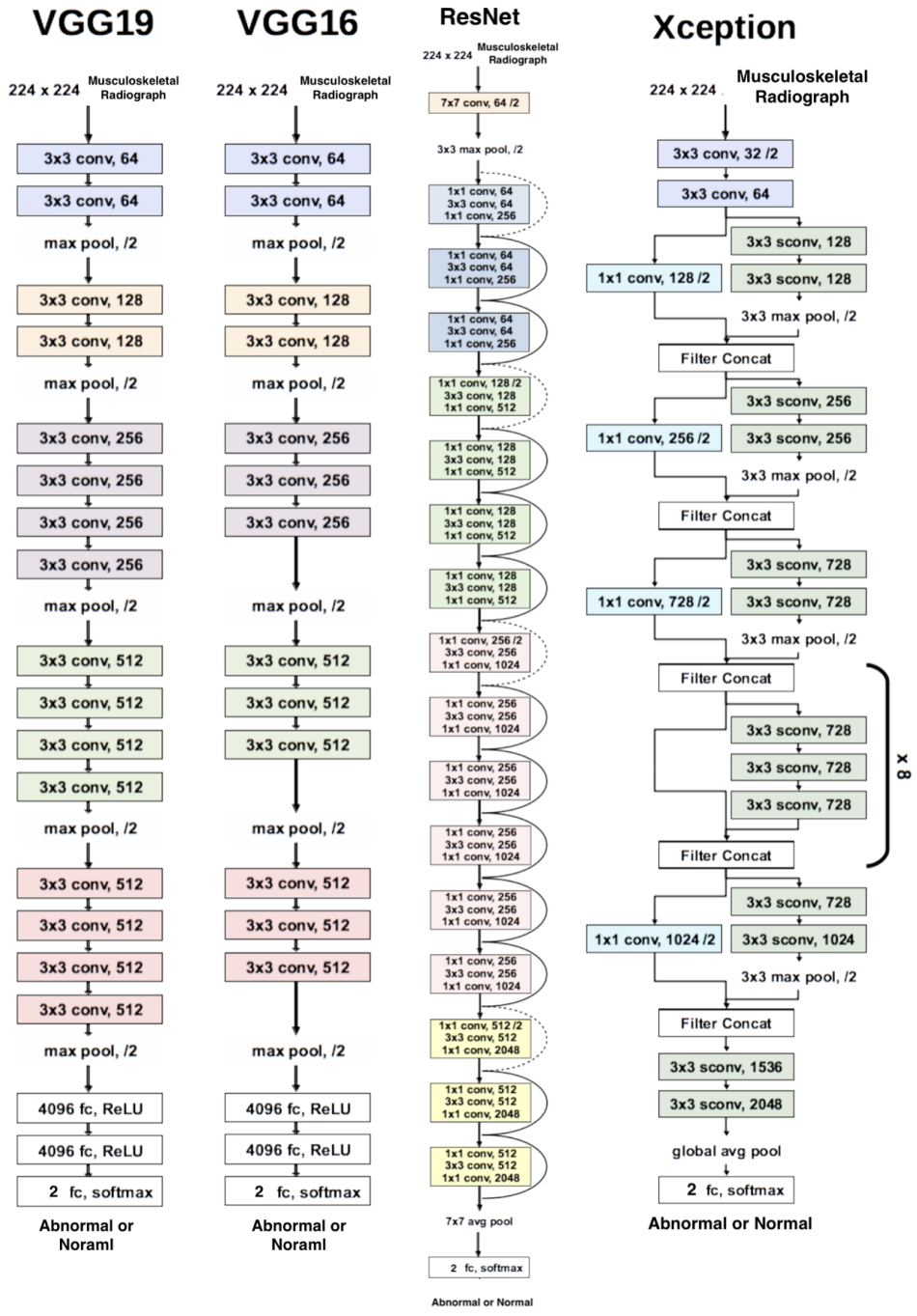

Figure 4: A visual representation of all of the CNN architectures used in this experiment. Each architecture takes in a musculoskeletal radiograph as input and return a class (abnormal or normal) as output. Images were adapted from (17).

2.4 Compiling and Training

All of the CNN architectures underwent the same compiling step and utilize the same training procedure. After prepossessing, the data needs to be compiled to fit the needs of the training procedure. In specific, an optimizer, a loss function, and a metric need to be specified. In our experiment, we used the Adam optimizer (18), the Categorical Cross-Entropy (with a softmax activation) loss function, and accuracy as a metric.

The categorical cross-entropy loss function is defined as follows:

$$loss = -\frac{1}{N}\sum_{1}^{M} T_i log(x_i) \tag{1}$$

The standard mechanism to train a neural network is to use the backpropogation algorithm (19). All of the CNN architectures in this experiment were trained using backpropogation. In specific, all of the models were trained for a total of 10 epochs (iterations of the data).

3 Results

The general purpose of this experiment is to not only evaluate the use of machine learning to detect abormalities, but also to decipher which CNN architectures perform well for this task. To test the architectures, each model listed in section 2.3, predicted the class (abnormal or normal) on 60 new musculoskeletal radiographs. The results from these predictions are shown here in a multitude of ways for proper comparison.

3.1 Training and Validation Accuracy and Loss

When a neural network is trained, stochastic gradient descent–an iterative method for optimizing an objective function–is performed. The objective function, also known as a loss function, must be defined during compiling before training. There are many different functions to choose from, so it is important to show the loss during training and validation for each epoch–a complete iteration of the data. Furthermore, it is also important to show the training and validation accuracy over each epoch since it shows how well the model works at each iteration. The graphs of training and validation accuracy and training and validation loss are shown for all machine learning models and methods performed in this experiment in figure 5 and figure 6 respectively.

3.2 Confusion Matrices

From a confusion matrix, several different metrics can be created for further evaluation. The true positive (top left), true negative (bottom right), false positive (top right), and false negative (bottom left) can be found in a confusion matrix and, in the context of this experiment, are defined as follows:

- **True Positive (TP)**: Musculoskeletal radiographs containing an abnormality and classified as abnormal.
- **True Negative (TN)**: Musculoskeletal radiographs not containing an abnormality and classified as normal.
- **False Positive (FP)**: Musculoskeletal radiographs not containing an abnormality and classified as abnormal.
- **False Negative (FN)**: Musculoskeletal radiographs containing an abnormality and classified as normal.

Figure 5: Training and validation accuracy for all models tested (VGG-16, VGG-19, ResNet, Xception)

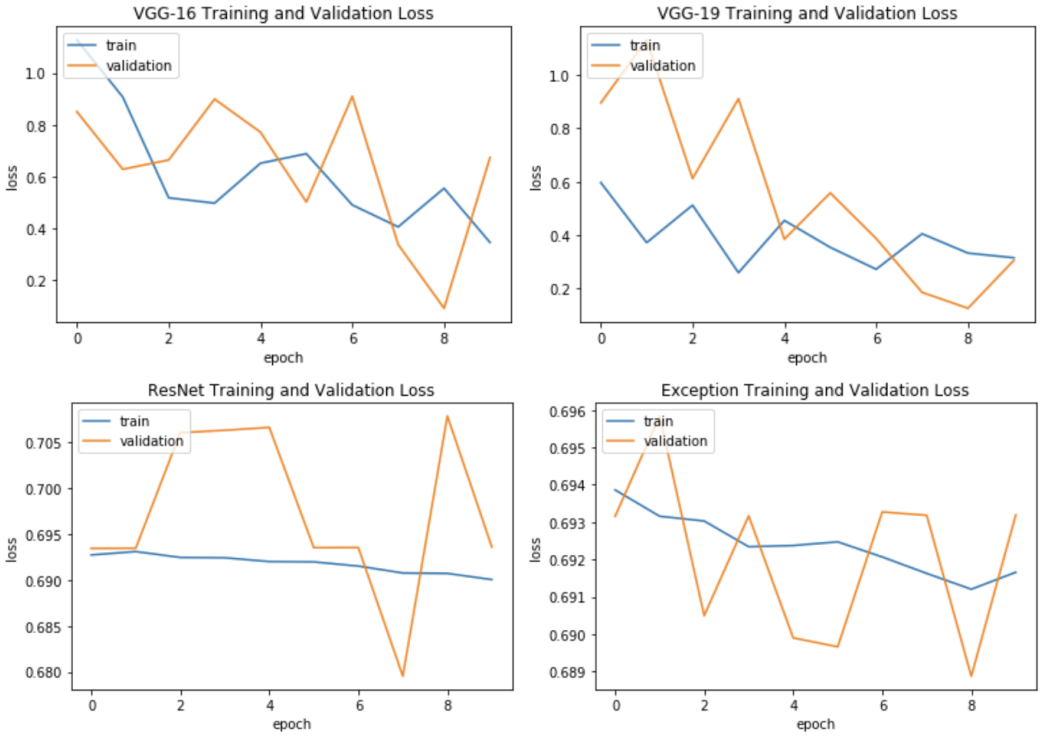

Figure 6: Training and validation loss for all models tested (VGG-16, VGG-19, ResNet, Xception)

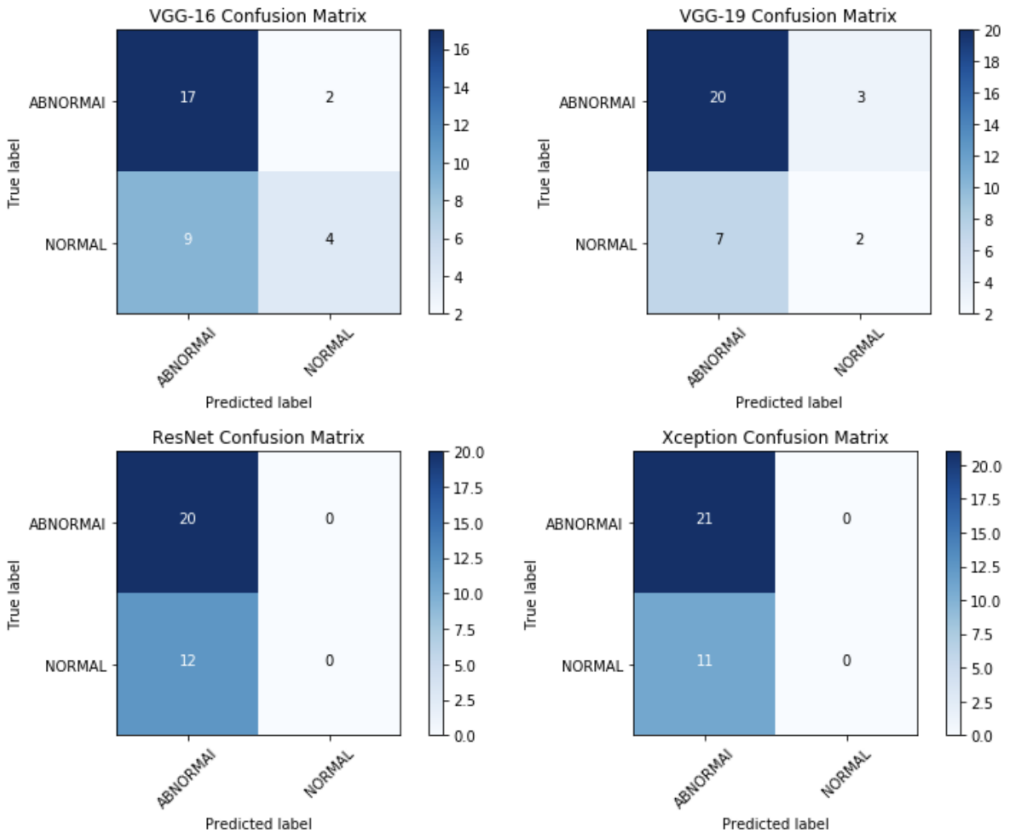

Figure 7: Confusion matrices for all models tested (VGG-16, VGG-19, ResNet, Xception).

From these numbers, more metrics can be calculated including sensitivity, specificity, precision, test accuracy, F_1 score, and Matthews correlation coefficient (MCC). Each are defined and calculated as follows:

$$Sensitivity = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

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

$$F_1 = \frac{2 * TP}{2 * TP + FP + FN} \tag{6}$$

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{7}$$

Figure 7 shows a confusion matrix for all models tested. Table 2 shows the metrics calculated, as defined above, from a confusion matrix for each model tested.

Table 2: Sensitivity, specificity, precision, test accuracy, F_1 score, and Matthews correlation coefficient (MCC) calculated from each confusion matrix for each CNN architecture.

Model	Sensitivity	Specificity	Precision	Accuracy	F_1	MCC
VGG-16	0.654	0.667	0.895	0.656	0.756	0.255
VGG-19	0.741	0.400	0.870	0.688	0.800	0.114
ResNet	0.625	NaN	1.00	0.625	0.769	NaN
Exception	0.656	NaN	1.00	0.656	0.792	NaN

4 Discussion

The results for the performance of each network are portrayed in a variety of different ways. Thus, specific comparisons can be made between each model. First, the training and validation accuracy is important to analyze for each model at each epoch. Most models gradually increased from the starting accuracy at the first epoch. However, since the graphs generate statistics for each epoch, further analysis shows that some models performed better at an earlier epoch rather than the final one. In specific, VGG-19 and Exception performed better at epoch 8 than 9 for both training and validation accuracy. Furthermore, ResNet's validation accuracy was higher at epoch 8 than 9, but the training accuracy was lower. Similar situations can be analyzed in the training and validation loss graphs. VGG-16, VGG-19, and Exception's validation loss was lower at epoch 8 than 9 whereas ResNet had its lowest loss at epoch 7. These trends are important to analyze to see, at which epoch, does each model perform the best. However, the main use of these graphs is to identify if a model is overfitting or underfitting. The shape of each graph can be used to identify potential flaws in a model which can be further identified to improve the overall performance of a model (20). Underfitting, occurring, when the model is not able to obtain a sufficiently low error value on the training set (5), can be identified through analyzing the training loss curve for each model. Due to the relatively straight curve on ResNet's training loss, it is clear that this model is underfitting. Furthermore, since the validation loss was higher for all models except for VGG-19, VGG-16, ResNet, and Exception showed signs of overfitting.

Further analysis and proper comparison can be made using the metrics (see table 2) calculated from the confusion matrices which are formed from each model's predictions on test sets–data points each model hasn't seen before. Sensitivity, also known as the true positive rate, represents the proportion of actual positives that are correctly predicted. VGG-19's sensitivity was the most desirable and the highest with a value of 0.741 whereas ResNet's sensitivity was the lowest with a value of 0.625. Specificity (also known as true negative rate), measures the proportion of actual negatives, that are predicted as negative. VGG-16 had the highest specificity at 0.667. Since the calculation for specificity resulted in division by zero for ResNet and Exception, specificity values could not be calculated. However, both of these models had the highest precision, the proportion of positives that are predicted correctly, at 1.00. The main point of comparison is the total classification accuracy for each model. VGG-19 yielded the highest accuracy at 0.688 (68.8%) and ResNet yielded the lowest at 0.625 (62.5%). The F_1 score, the harmonic mean of the precision and recall, was highest for VGG-19 and lowest for VGG-16. VGG-16 yielded the highest value, at 0.255, for Matthews Correlation coefficient–which measures the quality of a binary classification.

It is clear from these statistics, that convolutional neural networks perform better than guessing (>50%) for this task. In terms of this experiment, these results are statistically significant since it shows that CNNs are capable for such a task, but do not perform well compared to human classification (9). As stated, this experiment also aimed to specifically identify which CNN architectures perform better

than others. In terms of classification accuracy on the test set, the VGG-19 model performs the best out of the architectures tested in this experiment.

4.1 Future Work

Although the use of convolutional neural networks for detecting musculoskeletal abnormalities in radiographs is satistically signficiant as shown in this experiment, several improvements can be made. As with most machine learning tasks, especially ones involving image data, more data points for training would greatly increase accuracy and reduce overfitting. What's more, if greater computing power was able to be obtained, each model can be trained on the data for more iterations (epochs). Furthermore, the scope of this experiment is limited to the architectures listed in section 2. More CNN architectures can be explored and implemented for possible improvement. New CNN architectures can even be created from scratch. Also, other deep learning techniques can be explored such as transfer learning. Transfer learning, the act of transferring knowledge from one network to another is extremely common due to the newfound knowledge that neural networks learn general representations of data points in its earlier layers. What's more, other variants of neural networks may show improvement. Capsule networks, introduced in (21), aim to solve the shortcomings of CNNs such as the loss of pose information. This experiment only sheds light on computer-aided detection for musculoskeletal abnormalities and proves that machine learning, in specific CNNs, can be utilized for such a task and perform accurately.

4.2 Conclusion

With musculoskeletal abnormalities being one of the most common forms of injuries, efficient and accurate diagnosis is necessary. However, this isn't always possible due to a variety of shortcomings. An example of such a shortcoming is the lack of trained doctors in rural areas–especially in third world countries. AI technologies that implement neural network algorithms can greatly improve the quality of diagnosis and, most importantly, predict a diagnosis in an efficient manner. With certain patients, time can be of the essence, so it is important that any medically-related task is done in an efficient manner.

This experiment showed that CNNs are capable of such a task, but do not perform to the level of trained radiologists. However, this experiment shows that improvements are possible and can be made through further research. In conclusion, machine learning, specifically convolutional neural networks, can be a widely applicable tool to not only this task, but also to other medically-related image analysis tasks.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, June 2005.

[2] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–987, July 2002.

[3] A. Rosebrock, *Deep Learning for Computer Vision with Python*, vol. 1,Starter Bundle. PyImageSearch, 1.1.0 ed., 2017.

[4] M. A. Nielsen, "Neural networks and deep learning," 2015. http://neuralnetworksanddeeplearning.com/.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[6] M. Z. Alom, C. Yakopcic, M. S. Nasrin, T. M. Taha, and V. K. Asari, "Breast cancer classification from histopathological images with inception recurrent residual convolutional neural network," *Journal of digital imaging*, vol. 32, no. 4, pp. 605–617, 2019.

[7] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

[8] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli, "Deep convolutional neural network for the automated detection and diagnosis of seizure using eeg signals," *Computers in biology and medicine*, vol. 100, pp. 270–278, 2018.

[9] P. Rajpurkar, J. Irvin, A. Bagul, D. Ding, T. Duan, H. Mehta, B. Yang, K. Zhu, D. Laird, R. L. Ball, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng, "Mura: Large dataset for abnormality detection in musculoskeletal radiographs," 2017.

[10] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, "Adaptive histogram equalization and its variations," *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.

[11] A. Karpathy, "Convolutional neural networks (cnns / convnets)," 2018.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[14] S.-H. Tsang, "Review: Vggnet - 1st runner-up (image classification), winner (localization) in ilsvrc 2014," Sep 2018.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[16] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016.

[17] M. Leonardo, T. Carvalho, E. Rezende, R. Zucchi, and F. Faria, "Deep feature-based classifiers for fruit fly identification (diptera: Tephritidae)," pp. 41–47, 10 2018.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[20] J. Brownlee, “How to use learning curves to diagnose machine learning model performance,” Aug 2019.

[21] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic routing between capsules,” *CoRR*, vol. abs/1710.09829, 2017.