Abnormality Detection and Localization in Musculoskeletal Radiographs

CSE 598: Introduction to Deep Learning

Sunayana Gupta
1222389090
sgupt279@asu.edu

Pooja Bihani
1219789363
pbihani@asu.edu

Avinash Bawane
1220264344
abawane@asu.edu

Project Idea

We wanted to work on a project that applies deep learning to a healthcare-related issue. Since we were working with deep learning, we decided to apply some of the ideas we learnt in class about classification, convolutional neural networks, and extensions of the same to some datasets containing a lot of photographs.

We focused on the MURA dataset, a sizable dataset for the identification of anomalies in musculoskeletal radiographs, after conducting a literature study [1]. This study, conducted by the Stanford Machine Learning lab on the MURA dataset, compares the performance of their model to that of a radiologist. Therefore, in this project, we attempted to reproduce the architecture provided in the research article. Once we were successful in doing so, we next attempted to use our deep learning model to outperform the performance of the model provided in the publication.

Problem Description

As radiographic imaging is increasingly used for diagnostic and therapeutic purposes, its importance in patient care is growing significantly. This swift spread is a result of patients' demand for treatments that are faster, more accurate, less intrusive, and more economical. The use of imaging has increased as a result of technological advancements in radiologic imaging apparatus. These technological advancements make it feasible to acquire pictures with ever-increasing resolution, enabling the detection of abnormalities and more minute anatomical features.

With 30 million visits to the emergency room each year and a growing trend, musculoskeletal problems impact over 1.7 billion individuals globally and are the primary cause of severe, long-term pain and impairment. A greater average number of photos per subject is required to compensate for the higher resolution. Radiologists must interpret these pictures, which adds to their burden as the quantity of images increases. The capacity of radiologists to understand images is at jeopardy as they get increasingly complex and numerous. Machine learning may be used to automate the interpretation and diagnosis of medical images, which may lighten the workload for radiologists since modern neural networks are capable of detecting such minute irregularities in images. With this as our driving force, we began exploring for a dataset and a method. We began looking for a dataset and a method to classify abnormal and normal images from radiological images with this motivation.

Examples of similar problems include the classification of chest x-ray pictures into those that are normal or those that are influenced by COVID. [3] Sharmila Nageswaran et alcategorization .'s and prediction of lung cancer using CT scan pictures is another well-known example. [4] ANN, KNN, and RF are a few of the machine learning methods that were employed for categorization. When making predictions about lung cancer,

Literature Review

Detecting abnormalities in radiographic areas is a critical challenge in the medical domain. Radiographic tests can detect a wide range of diseases and anomalies, and they are also the most commonly used method. If such scans are accurately evaluated, doctors can rule out anomalies and eliminate the need for additional diagnoses. The musculoskeletal disorders can result in acute pain, chronic agony, and even disability. Tian published the first study on the use of Computer Automated Systems to do medical tasks in 2003, titled "Computing Neck Shaft Angle of Femur for X-Ray Fracture Detection." Feature extraction techniques were used to extract shape of femur and femur were classified based on neck-shift angle measured by it in the paper. This was one of the first significant breakthroughs in supporting the use of Automated Computer Systems to undertake medical tasks in order to obtain a second opinion [7].

All known data indicates that using deep convolutional neural network models to detect musculoskeletal abnormalities is a good choice. There is very little literature and research activity on radiographic abnormality detection. Rajpurkar and Irvin made the most significant contribution, developing the model "MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs" using DenseNet architecture and which outperformed the best radiologist in the area of wrist abnormality detection[2]. The introduction of the MURA dataset by Rajpurkar et al. [2] has greatly eased research on utilizing deep neural networks to detect an abnormality in an X-ray image of the upper extremities. It was suggested that a foundation model be built on DenseNet-169 and pre-trained on the ImageNet dataset. The gold standard was developed by a committee of three radiologists. A comparison of the DenseNet-169 model ensemble of three additional radiologists was done using Cohen's kappa score, revealing that the performance of their baseline models is still less than the worst radiologist.

A lot of research studies in this area have since undertaken additional experiments using different types of network architectures trained on the MURA dataset. D. Dias [8], for instance, investigated strategies for transfer learning for ResNet-152, VGG-19, DenseNet-121, SqueezeNet, and Inception-v3 [9]. The performance on the validation data set findings reveal that SqueezeNet achieved the maximum performance while having a far lesser number of parameters as compared to its competitors. The DenseNet-169 baseline however, has higher performance than it. N. Harini et al. [10] researched transfer learning on the dataset MURA using neural networks like Inception-v3, Xception [11], VGG-19, DenseNet169, and MobileNet on only wrist, shoulder and finger images. In a similar way, G. Chada [12] trained InceptionResNetV2, DenseNet-169, and DenseNet-201 using only finger and humerus data. In [13], an ensemble model which was composed of ResNet and VGG-19 was suggested and tested on four types of studies: wrist, humerus, elbow and finger. If the exact type of the radiograph is not known, H. El-Saadawy et al. [14] developed a classifier utilizing hierarchical structure comprising two stages of MobileNets, the first stage categorizes the bone type and the consequent stage consists of local classifiers which determines if the input is normal or abnormal.

S. Panda and M. Jangid [15] evaluated six alternative custom optimized CNN architectures using the dataset MURA and according to the performance metric determined that Adam [16] was the best optimizer based on their trials. The models trained on the MURA dataset are usually intended to be classification models, but B. Guan et al. [17] tagged fractures in the arm in the dataset with bounding boxes so that they are able to localize fractures in the image using their object detection method. Because the test set is not publicly available, efforts on the MURA dataset in the literature are often evaluated on the validation set. On the test set, the DenseNet-169 ensemble baseline [2] was reported with an overall Kappa score of 0.705 for all categories of research. However, the performance of future tasks was reported using different metrics, or the validation set performance was compared to the DenseNet-169 baseline test set performance. Another study that achieved great performance was [18], which used a capsule network with two convolutional capsule layers to reach an average Kappa score of 0.801. The average Kappa of 0.836 was also reached by the other study, which suggested a multi-scale convolutional neural network with completely linked graph convolutional network [19].

Dataset Description

MURA (Musculoskeletal Radiographs) is a sizable radiograph dataset that includes 40,000 pictures of the musculoskeletal system in the upper extremity. [2] Radiologists manually classify each picture as normal or abnormal. Our database, MURA, comprises 9,045 normal and 5,818 pathological musculoskeletal radiography examinations of the upper extremity, encompassing the shoulder, humerus, elbow, forearm, wrist, hand, and finger (each study has pictures from several perspectives). The resolution and aspect ratio of the clinical photographs varies. The dataset is divided into three categories: test (783 patients, 1,199 studies, 3,197 photos), validation (783 patients, 13,457 studies, and 36,808 images), and training (206 patients, 207 studies, 556 images). None of the sets have any patients in common.

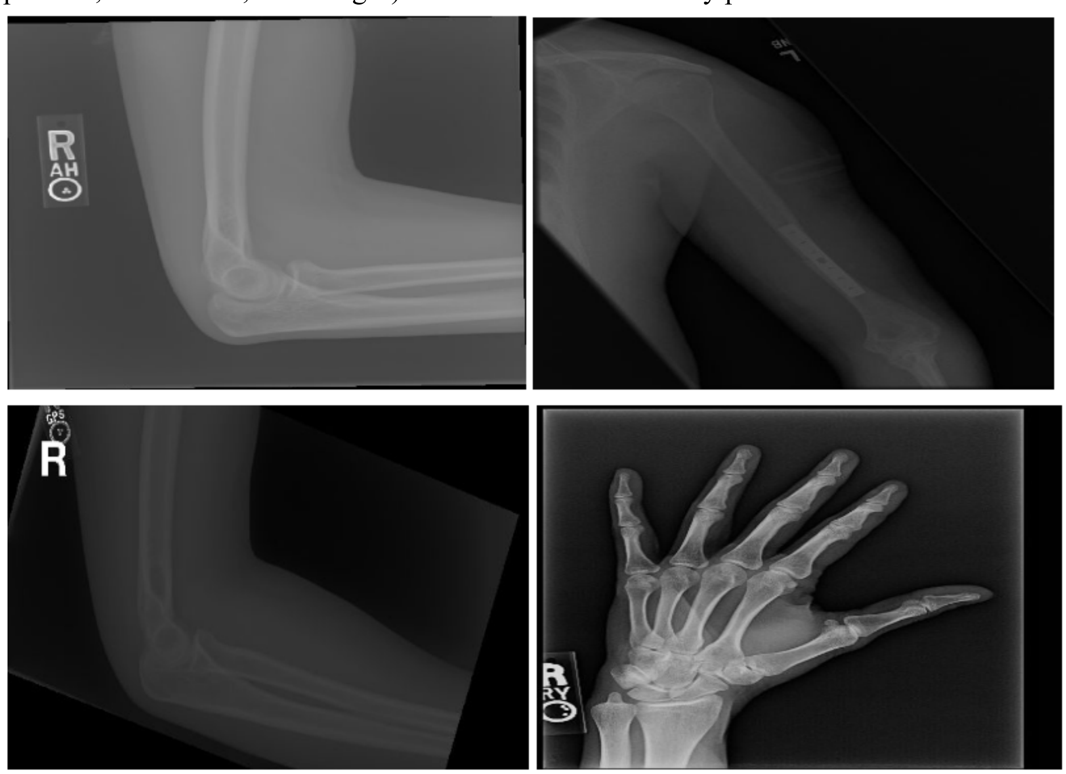

Figure 1 Sample images of MURA dataset

The distribution of perfect and imperfect radiographs investigations is shown in Table 1 as a whole.

Study	Train		Validation		Total
	Normal	Abnormal	Normal	Abnormal	
Elbow	1094	660	92	66	1912
Finger	1280	655	92	83	2110
Hand	1497	521	101	66	2185
Humerus	321	271	68	67	727
Forearm	590	287	69	64	1010
Shoulder	1364	1457	99	95	3015
Wrist	2134	1326	140	97	3697
Total No.of Studies	8280	5177	661	538	14656

Table 1: Table showing the distribution of studies in the MURA dataset

Baseline Evaluation and Metrics

VGG16 has been selected as the foundational model for our analysis. We choose VGG16 because it is straightforward to comprehend and apply and has a track record of producing successful results on the imagenet dataset. Still effective in establishing a solid enough foundation to achieve a score of almost 80% on traditional tasks like classifying cats and dogs. Additionally, VGG16 offers us a "sweet spot" where it is not too deep to encounter issues like vanishing gradients but is yet sufficiently deep to provide respectable results.

The following findings, which we will use as our baseline results, were obtained after running the VGG16 model on the dataset's humerus portion:

```
              precision    recall  f1-score   support

           0       0.53      0.55      0.54        76
           1       0.75      0.73      0.74       140

    accuracy                           0.67       216
   macro avg       0.64      0.64      0.64       216
weighted avg       0.67      0.67      0.67       216
```

Based on precision, recall, f1-score, and accuracy, we will compare the various models that we develop.

Experiments and Results

Data augmentation

To further enhance our outcomes, we invested a significant amount of effort on data processing the photographs. Convolutional neural networks can classify items even when they are arranged in multiple orientations thanks to invariance. Data augmentation is a technique for producing fresh, atypical "data." This has two advantages: first, it enables the generation of "additional data" from sparse data, and second, it inhibits overfitting. Our model's loss is low when the parameters are tweaked properly, and this sweet spot is where we want to be during optimization. To attain good performance when there are many parameters, we would need to present the machine learning model to a proportionate number of samples. Additionally, the complexity of the task the model must do affects the number of parameters required. A CNN can be resistant to translation, viewpoint, size, and illumination, for example (Or a combination of the above). Essentially, this is how data augmentation works. In the actual world, we might have a dataset of photos shot under a specific set of circumstances. But there are many distinct situations in which our target application might be present, including variations in scale, brightness, orientation, and position. By providing our neural network with additional synthetically modified data, or data after augmentation, we can account for these situations.

Here, we are performing following data augmentation techniques:

(i) Resizing of images -

The images which are input to the machine learning model should be of the same size and this resizing of images is a very much vital task to be carried out.

(ii) Horizontal Flip -

An image flip means reversing the rows or columns of pixels in the case of a vertical or horizontal flip respectively.

(iii) Crop from Center -

Crops the image from the center discarding the irrelevant features of the image and selecting only the relevant features.

(iv) Random rotation of images -

A rotation augmentation randomly rotates the image clockwise by a given number of degrees from 0 to 360.

The rotation will likely rotate pixels out of the image frame and leave areas of the frame with no pixel data that must be filled in.

(v) Sharpening of images -

Sharpening of images highlights the boundary/lines creating more contrast for the area of interest and background.

(vi) Gaussian Blurring of images -

Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. It is used to reduce image noise.

Data Normalization

Histogram equalization is a technique for image processing that alters the histogram's intensity distribution to change the contrast of an image. Giving the final image a linear cumulative distribution function is the goal of this technique. When the applied data of the image is represented by close contrast values, histogram equalization is highly effective in boosting the local contrast of the item in the image. By making this change, the histogram's intensity can be more evenly spread, allowing places with low local contrast to increase their contrast without harming the overall contrast.

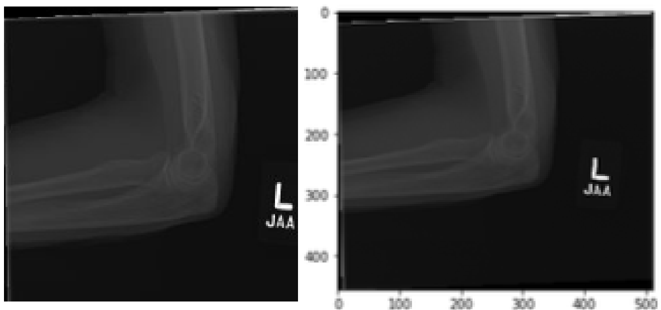

Figure 2 Images before and after processing

Methodology

We have replicated the architecture of the research paper where they used Denesenet 169. [2] We ran different models on individual studies of all the body parts listed in the MURA dataset. We worked on Resnet as it takes less memory, has faster inference time, and allows deeper networks to be trained easily.

We also tried to work on other models like MobileNet and Inception and NasNet. This allowed us to do a comprehensive study, let us do a quantitative and qualitative analysis of the models as well as the data, and in the end, report the variations observed. We kept the image size constant to 224, 224, 3 throughout the experiments and also all the preprocessing on all the images. Also we kept a constant batch size of 24 for all our studies.

Experiment - 1 VGG16

After segregating the data into different directories according to the body parts mentioned in the reference paper and preprocessing the images, we trained each of the body parts using the pretrained VGG16 model on imagenet data taken from the Keras library. We worked on pretrained models here because the size of our dataset is not that high to work on models made from scratch. We had limited resources to train our models and with those only pretrained models give better results on less data and less processing time.

After taking the pretrained model of VGG16 we removed the last layer of the model and added a batch normalization layer and a global average pooling layer as it was needed to normalize the contribution of each layer for every mini batch. The pooling layer was needed as we were going to add further layers to the model so we needed to reduce the cluster size so we do not increase the complete model size to something untrainable. After this we added some dropout layers to let the model not get overtrained as dropout layers help in regularization of the model. Post this we added leaky Relu activation and finally the last layer for classification. We trained the model with callbacks on the validation accuracy so that the model does not get overtrained at any point and it stops training with a patience of 10. We ran this model with all the required body parts, and below are the results we got from this experimentation.

Body part	Precision	Recall	F1 Score
Wrist	0.72	0.73	0.70
Forearm	0.64	0.65	0.58
Finger	0.62	0.65	0.60
Hand	0.63	0.54	0.49
Shoulder	0.67	0.69	0.62
Elbow	0.73	0.77	0.73
Humerus	0.64	0.64	0.67

Table 2: Table showing the results of VGG16 model

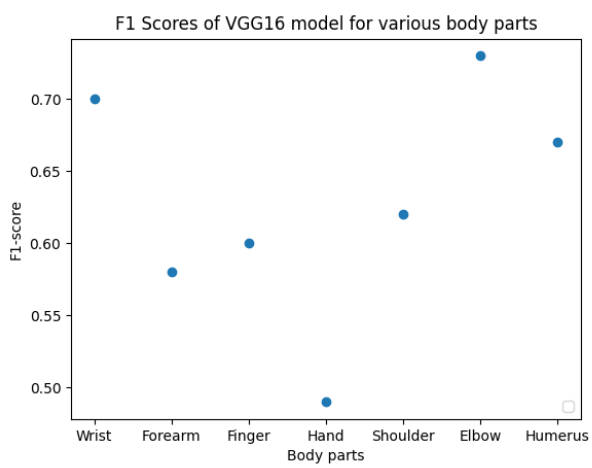

Figure 3 Performance of VGG16 model with different body datasets

As we can see in Fig 3, we got best results for the elbow part and worst results for the hand. This is because of the intricacies in the way our bones are structured. The elbow part of the arm somehow gets very clearly seen in the X-rays and hence the abnormality in it is also very clearly distinguished from the normal X-Rays. Same goes for the wrist and humerus part. The VGG16 model seems to give bad results for the hand, finger and shoulder. This is because the structure itself of these parts is very complicated and the abnormality sometimes is very difficult to identify in the X-ray of these parts of the arm.

Experiment - 2 ResNet50

We discovered that when the deeper networks began to converge, a degradation issue emerged: as network depth increases, accuracy becomes saturated before quickly degrading. In order to tackle this problem, we thought of using Resnet which has a residual function that was developed using the stacked non-linear layers, $F(x) = H(x) - x$, where $F(x)$ and x represent the identity function (input=output), respectively. Optimizing the original, unreferenced mapping H is more difficult than optimizing the residual mapping function $F(x)$. [5]

A residual learning framework in deep learning networks aids in maintaining positive outcomes throughout a network with several layers. Utilizing residual blocks, which rely on residual mapping to maintain inputs, the deep residual network addresses vanishing gradients and saturating accuracy. With these things in mind we tried ResNet50 on our data and got below results

Body part	Precision	Recall	F1 Score
Wrist	0.72	0.73	0.70
Forearm	0.60	0.61	0.54
Finger	0.65	0.70	0.65
Hand	0.62	0.63	0.65
Shoulder	0.62	0.63	0.65
Elbow	0.62	0.60	0.49
Humerus	0.59	0.59	0.62

Table 3: Table showing the results of ResNet50 model

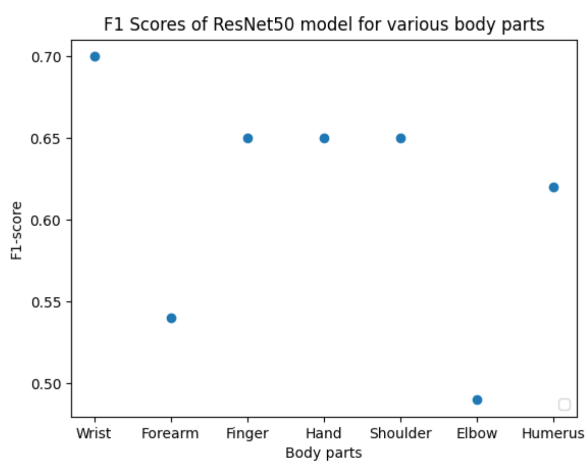

Figure 4 Performance of ResNet50 model with different body datasets

As we can see in Fig 4, here, the results we got from ResNet50 are not very impressive and they are barely classifying the images into normal and abnormal. So this made us realize that we need a more dense network as ResNet. It is also known that ResNets have many layers which are not contributing much to the overall network and can be dropped. So for our case the contributing layers are so small that the classification could not be done properly.

Experiment -3 DenseNet169

Densenet 169 is the model from our reference research paper. This model has much more layers as compared to the VGG16 model that we took as our baseline in this project. In CNN, the distance between the input layer and the output layer (and the gradient in the opposite direction) increases to the point where information can disappear entirely before getting to the other side. Maximum information (and gradient) flow is ensured by DenseNet. It only connects every layer directly to one another to do this. DenseNets take advantage of the network's potential through feature reuse rather than obtaining representational power from extremely deep or wide architectures. [6] Contrary to popular belief, by connecting in this manner DenseNets require less parameters than a comparable classic CNN because redundant feature maps are not required to be learned and it also solves the problem of vanishing gradients. Below are the results we got when we ran the densenet169 model on our Mura dataset.

Body part	Precision	Recall	F1 Score
Wrist	0.77	0.78	0.75
Forearm	0.70	0.74	0.69
Finger	0.69	0.75	0.67
Hand	0.72	0.64	0.57
Shoulder	0.73	0.76	0.75
Elbow	0.74	0.78	0.72
Humerus	0.79	0.81	0.81

Table 4: Table showing the results of DenseNet169 model

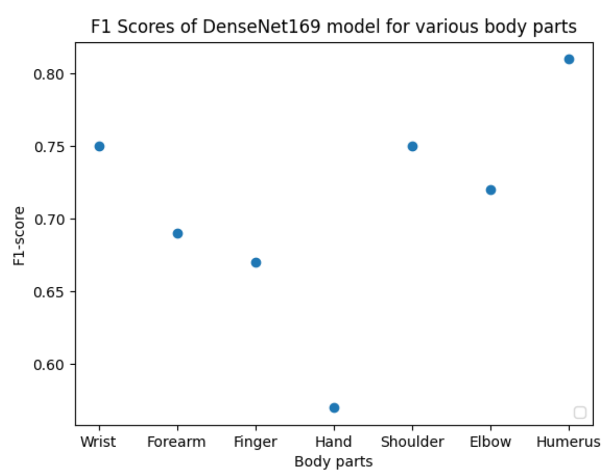

Figure 5 Performance of DenseNet169 model with different body datasets

As you can see from Fig 5, Except hand all the results are significantly improved with the densenet results. This is because each layer of the densenet is connected to every other layer. For each layer the feature map is used as an input to the next layer. For this the DenseNet169 has multiple small dense blocks within the network which have the same feature size. This makes the concatenation possible and this way a very deep network is generated. This network is denser and deeper than the VGG16 model and hence gave better results.

Experiment - 4 InceptionV3 model

We further worked on the Keras InceptionV3 model in hope for better results compared to the Densenet169 model. It is a popular image recognition model and has been demonstrated to achieve more than 78.1% accuracy on the ImageNet dataset. It was created based on the GoogleNet architecture used in ILSVRC. Although Densenet169 gave better results compared to the VGG16 model, because of its too dense layers, it took a lot of time to get trained. To reduce this training time we worked on Inception Net. Below are the results we got using the InceptionV3 model.

Body part	Precision	Recall	F1 Score
Wrist	0.76	0.78	0.75
Forearm	0.72	0.76	0.70
Finger	0.65	0.70	0.67
Hand	0.70	0.73	0.70
Shoulder	0.73	0.76	0.75
Elbow	0.74	0.77	0.72
Humerus	0.83	0.82	0.82

Table 5: Table showing the results of InceptionV3 model

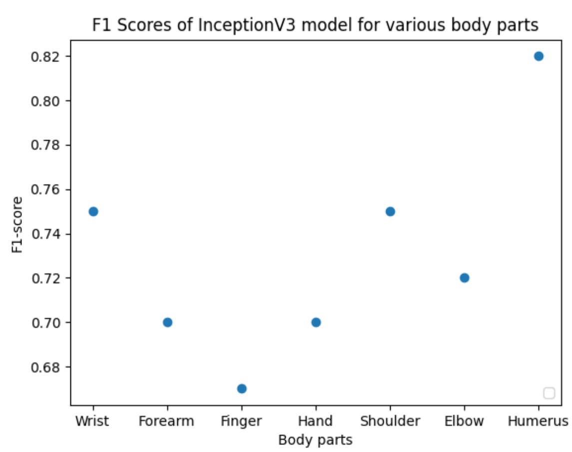

Figure 6 Performance of InceptionV3 model with different body datasets

The InceptionV3 model only has 48 layers as compared to the 169 layers in Densenet169 [Fig 6]. But surprisingly it gave almost at par results to densenet169, if not better. The humerus results are actually better in the InceptionV3 network. This is because of the clever design of the InceptionV3 model. One of the key characteristics of the Inception architecture is their adaptation of Lin et al.'s "Network in Network" approach, which increased the neural networks' capacity for representation. The Inception Net was designed to reduce the computation time and also simultaneously keep up highly efficient classification with high accuracy. With the increase in size and depth the designers of Inception model wanted to keep sparsity in the model which accounted for the high speed training of the network.
InceptionV3 uses 3 techniques to achieve this, factorizing the convolutions, auxiliary classifiers and efficient grid size reduction. Convolutions are factorized with the intention of lowering the number of connections and parameters without lowering network performance. It basically replaces the bigger size of 5* 5 to two convolutions of 3*3 which inturn decreases the required computation. They also did asymmetric convolutional factorization by replacing 3*3 to 3*1 and 1*3 convolutions. With factorization, the total number of parameters is decreased for the network as a whole, overfitting is less likely, and as a result, the network can grow deeper. Auxiliary classifiers are added to act as regularisers so that less overfitting takes place. The grid size reduction process is different in InceptionV3 net as compared to other models. In InceptionV3 instead of simply using max pooling layer it uses convolutional layers with 2*2 stride in parallel with the pooling layers and concats them so that less information loss takes place.

All these subtle changes make a huge difference in performance even with a much smaller network compared to the densenet.

Experiment - 5 NASNet trained from scratch (failed experiment)

NASNet is designed to find the best neural network structure by reinforcement learning. In essence, the goal is to find the optimal combination of the search space's characteristics, which include filter sizes, output channels, strides, the number of layers, etc. The accuracy for the searched architecture on the provided dataset was the reward after each search activity in this Reinforcement Learning context. NASNet helps us get away from designing the network and thinking how many layers of neurons per layer are to be included in the model.

We tried with multiple configurations to create NASNet on our Agave clusters provided by ASU and ran them with a maximum 2000 neurons for every layer and with 100 layers. But due to limited resources we could only run 10 iterations of the reinforcement cycle of training and reward. We ran the reinforcement learning model on the humerus data only and got an accuracy of 62% only. Training this model took a very long time on the smallest dataset(humerus) that we had and gave an accuracy of only 62% so we decided to drop this model and instead tried the pretrained NASNet model.

But NASNet in itself is a very powerful technique to generate custom deep learning models for any kind of dataset but works only when we have a lot of resources at our disposal.

Experiment-5 NASNet pretrained model (failed experiment)

As we could not train the NASNet that we were training from scratch beyond a certain number of iterations it didn't help getting a better accuracy than the Inception model. So we decided to work on the pretrained model of NASNet again provided by Keras. This model is designed using reinforcement learning while working on training on imagenet dataset. Below are the results we got when we did our pre-training and testing using Keras's NASNet.

Body part	Precision	Recall	F1 Score
Wrist	0.74	0.75	0.71
Forearm	0.70	0.73	0.67
Finger	0.65	0.69	0.63
Hand	0.74	0.64	0.57
Shoulder	0.67	0.69	0.70
Elbow	0.71	0.74	0.68
Humerus	0.73	0.74	0.75

Table 6: Table showing the results of NasNet pretrained model

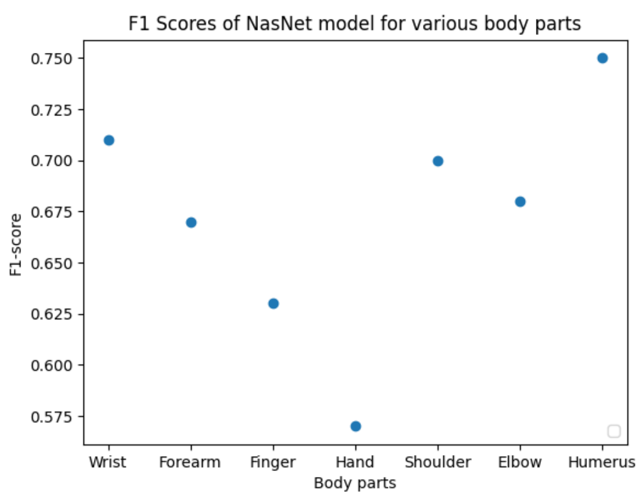

Figure 7 Performance of NASNet model with different body datasets

As we can see from Fig 7, this model gave poor results as compared to InceptionV3 and densenet models. This is because this model is custom designed to work on the imagenet and its layers are modified to get better validation accuracy for imagenet and as we know imagenet dataset is very different as compared to Mura dataset. The major difference between NASNet pretrained model and other pretrained model is that NASNet is designed with reward and reinforcement using imagenet while other models architecture is designed to work on any dataset but they are just pretrained on imagenet. Therefore, the NASNet pretrained model gives state of the art accuracy for imagenet but fails miserably when worked upon on any different dataset.

Localization of abnormality

Along with classification, we have also done localization of abnormality in the abnormal Mura images. This was done using the Class Activation Map(CAM) as done in our reference paper. This basically signifies the part of the radiograph which contributes most to the model's prediction of abnormality. The discriminative image areas that the CNN utilized to identify a given category are shown on a class activation map for that category. The idea to identify the discriminative area is to perform a global average pooling layer on the convolutional feature map just before the final output layer. In CAM, we project the weights of the output layer back on the convolutional feature map to identify the importance of certain image regions. The spatial average of each unit's feature map is produced by global average pooling at the final convolutional layer. To create the final output, these values are added up in a weighted manner.

We had to create a bounding box and the related item category in order to accomplish localization. We segmented the heatmap using a straightforward thresholding method in order to get a bounding box from the CAMs. We started by segmenting the areas where the value is greater than 20% of the maximum value of the CAM. Next, we took the segmentation map's biggest linked component's bounding box. Below is the resulting image that we got after performing the CAM.

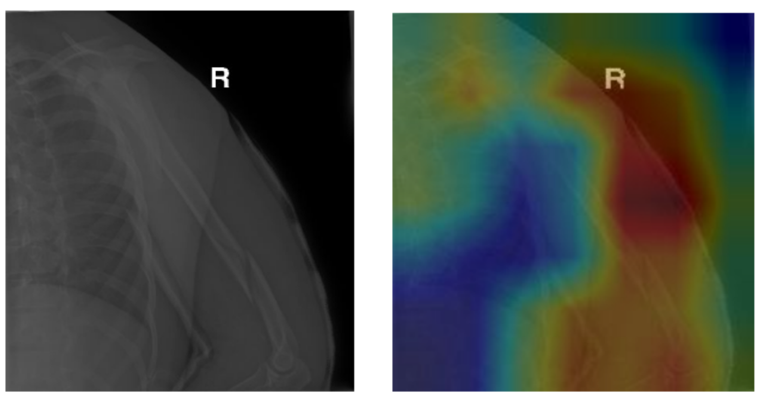

Figure 8 Localisation of abnormality (right) in the original image (left)

Conclusion

In this work we have used different algorithms for classification of the X-ray images into normal and abnormal, and by far we got the best results using InceptionV3. Firstly, we used VGG16 but we saw that as the number of epochs were increasing the accuracy started getting saturated and afterwards degraded rapidly so we needed a shallow network for this and residual net fitted completely this criteria as it adds a residual part of the previous layer instead of direct mapping.

After training the ResNet model we found that the number of parameters in ResNet was very high hence it needed a large time to train many of these layers were barely contributing hence there was introduction to DenseNet whose layers are very narrow and add a small set of new feature maps. Further, we tried Inception V3 whose parameter size was even reduced due to the factorizing convolutions and accuracy was even better as shown in figure 9. The last method we tried is NasNet which improves reinforcement learning to get an optimized model but as we had limited resources the number of iterations taken were maximum 10 hence the results obtained NasNet can still be better.

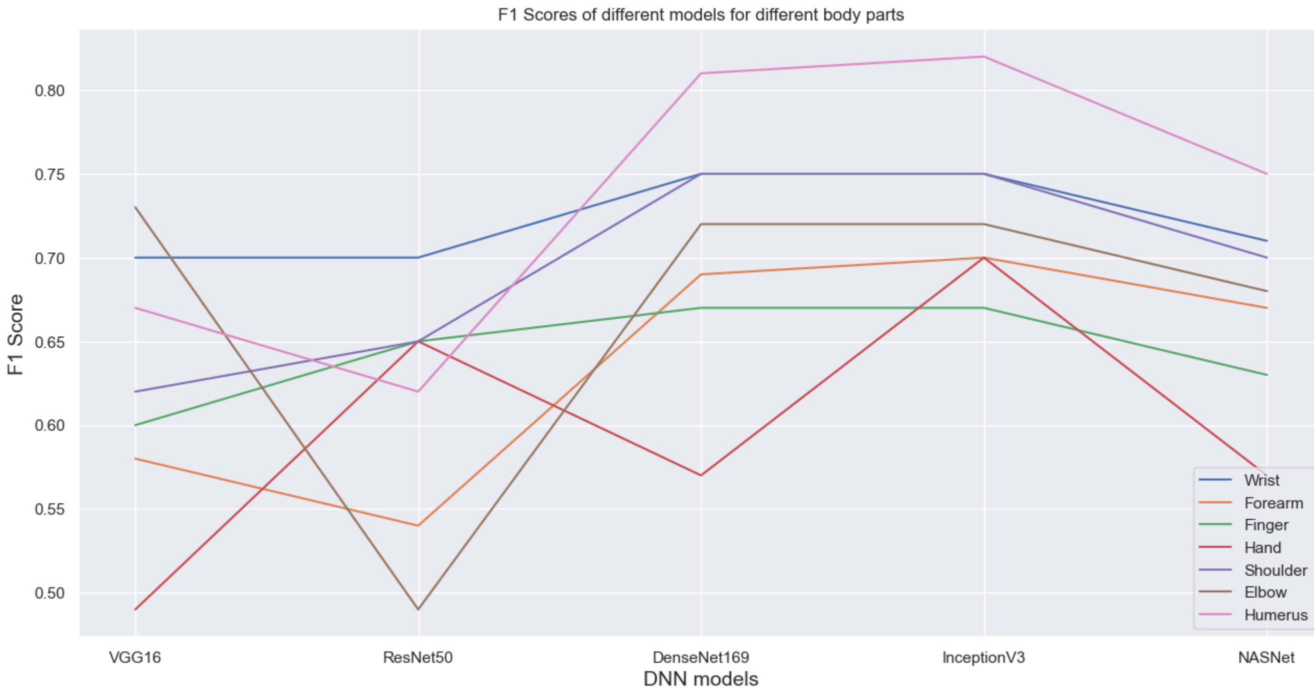

Figure 9 Performance comparison of five models against different body dataset

Reference

[1] https://stanfordmlgroup.github.io/competitions/mura/
[2] Rajpurkar, P., Irvin, J., Bagul, A., Ding, D., Duan, T., Mehta, H., ... & Langlotz, C. (2017). Mura: A large dataset for abnormality detection in musculoskeletal radiographs. *arXiv preprint arXiv:1712.06957.*
[3] https://eurradiolexp.springeropen.com/articles/10.1186/s41747-020-00203-z
[4] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9424001/
[5] https://arxiv.org/pdf/1512.03385.pdf
[6]https://towardsdatascience.com/architecture-comparison-of-alexnet-vggnet-resnet-inception-densenet-beb8b116866d
[7] Tian T P, Chen Y, Leow W K, Hsu W, Howe T S and Png M A (2003) Computing Neck-Shaft Angle of Femur for X-Ray Fracture Detection
[8] D. D. A. Dias, "Musculoskeletal Abnormality Detection on X-Ray Using Transfer Learning," Master's thesis, Universitat Pompeu Fabra, 2019
[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger (2017) "Densely Connected Convolutional Networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
[10] N. Harini, B. Ramji, S. Sriram, V. Sowmya, and K. Soman (2020) "Musculoskeletal radiographs classification using deep learning," in Deep Learning for Data Analytics.

[11] F. Chollet, (2017) “Xception: Deep Learning with Depthwise Separable Convolutions,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
[12] G. Chada, (2019) “Machine Learning Models for Abnormality Detection in Musculoskeletal Radiographs”.
[13] T. C. Mondol, H. Iqbal, and M. Hashem, (2019) “Deep CNN-Based Ensemble CADx Model for Musculoskeletal Abnormality Detection from Radiographs,” in 2019 5th International Conference on Advances in Electrical Engineering (ICAEE).
[14] H. El-Saadawy, M. Tantawi, H. A. Shedeed, and M. F. Tolba, (2020) “A TwoStage Method for Bone X-Rays Abnormality Detection Using MobileNet Network,” in Advances in Intelligent Systems and Computing. Springer, 2020.
[15] S. Panda and M. Jangid, (2020) “Improving the Model Performance of Deep Convolutional Neural Network in MURA Dataset,” in Smart Systems and IoT: Innovations in Computing. Smart Innovation, Systems and Technologies.
[16] D. P. Kingma and J. Ba, (2015) “Adam: A Method for Stochastic Optimization,” in the 3rd International Conference for Learning Representations, San Diego.
[17] B. Guan, G. Zhang, J. Yao, X. Wang, and M. Wang, (2020) “Arm fracture detection in X-rays based on improved deep convolutional neural network”.
[18] A. F. Saif, C. Shahnaz, W. P. Zhu, and M. O. Ahmad, (2019) “Abnormality Detection in Musculoskeletal Radiographs Using Capsule Network”.
[19] S. Liang and Y. Gu, (2020) “Towards Robust and Accurate Detection of Abnormalities in Musculoskeletal Radiographs with a Multi-Network Model”.