MURA Deep Learning Project

Amine Benaicha
Supervisor: Francisco Estrada

Introduction

The goal of this project was to learn and explore various deep learning concepts and to use them on a practical problem. The Musculoskeletal radiographs dataset (MURA) is a dataset that was made publicly available by the Stanford University School of Medicine and it consists of a set of 40,561 bone x-ray images [1]. The x-rays have been labelled as either normal or abnormal by 3 Stanford radiologists where a majority vote was used to decide the labels. The objective is to use deep learning to create a model that can competently preform the binary classification task using this data.

About the data:
The data is split by default into a training set (36,808 images) and a validation set (3,197 images). The data is also partitioned based on studies. A study can consist of multiple x-ray images of a single patient and a single body part, but the images can be of different angles. Having multiple views helps radiologists to come to a more informed conclusion. This means that in the dataset, all the images belonging to a study would have the same class. The categories and their break down are as listed below.

Training:

Category	Normal Images	Abnormal Images
SHOULDER	4211	4168
HUMERUS	673	599
FINGER	3138	1968
ELBOW	2925	2006
WRIST	5765	3987
FOREARM	1164	661
HAND	4059	1484
TOTAL	21935 (60%)	14873 (40%)

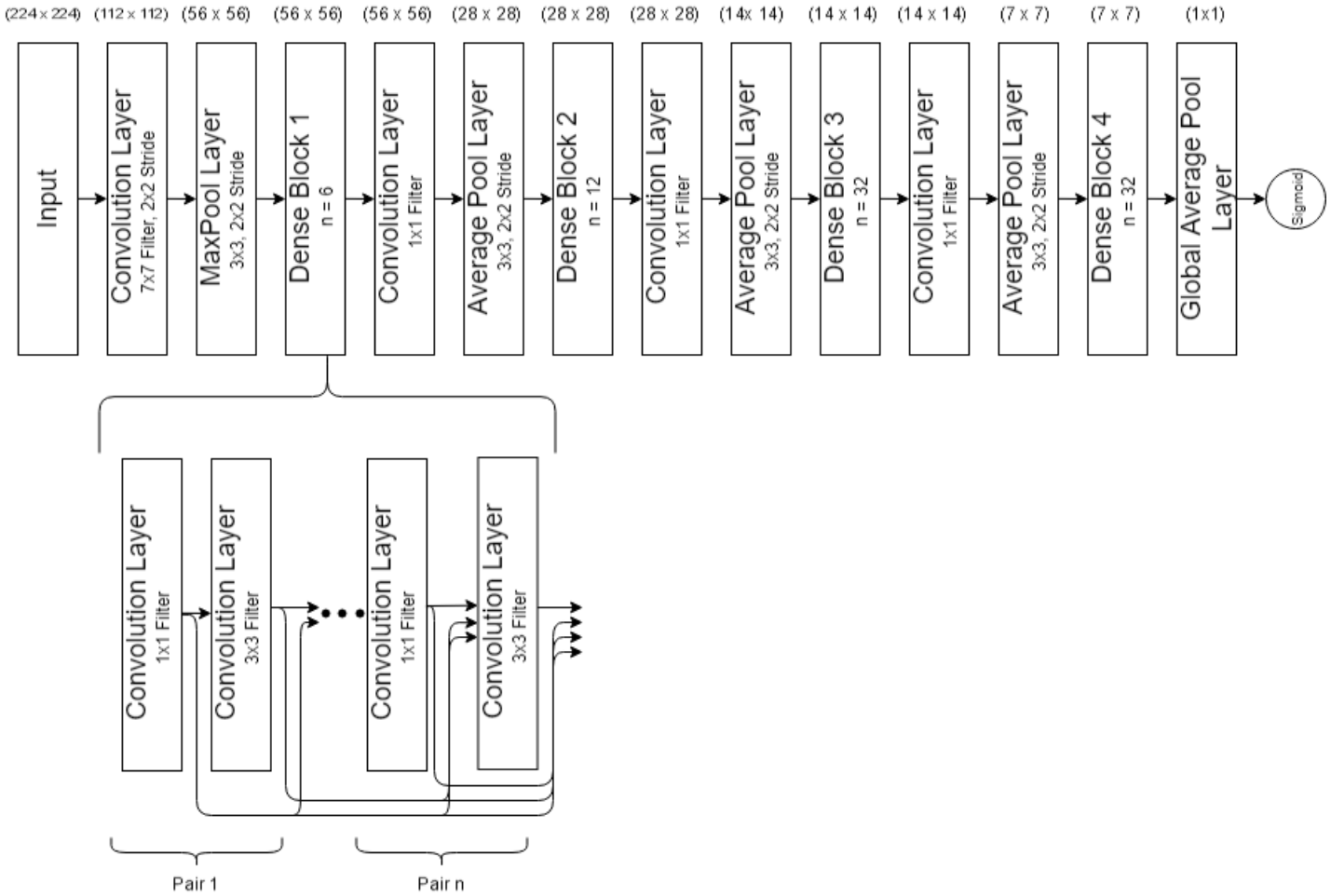
Validation:

Category	Normal Images	Abnormal Images
SHOULDER	285	278
HUMERUS	148	140
FINGER	214	247
ELBOW	235	230
WRIST	364	295
FOREARM	150	151
HAND	271	189
TOTAL	1667 (52%)	1530 (48%)

Model

After trialing multiple different network architectures, DenseNet-BC 169 [2] has shown the greatest result. DenseNet takes inspiration from both ResNet and Inception Network. As with ResNet, to help reduce information and gradients being washed out due to the numerous layers, in DenseNet a convolutional layer takes as input the feature maps produced by all the preceding convolutional layers within the dense block it belongs to. However, the input from the various layers aren't combined using summation, instead they are combined by having the feature maps concatenated similarly to Inception Network. The BC in DenseNet-BC refers to the use of bottleneck layers and Compression. The bottleneck layers help to reduce the size of the input in terms of number of channels for the densely connected convolutional layers while minimizing the amount of information that is lost. Similarly, compression is used to reduce the number of feature maps produced between Dense blocks.

The model starts with 64 filters and increases by 32 (growth rate) for every dense block. A compression rate of 0.5 is used. Drop out of 20% was used for the final fully connected layer. For optimization, Adam is used with default parameters β_1=0.9 and β_2=0.999. Overall, the model has just over 12.5M parameters.



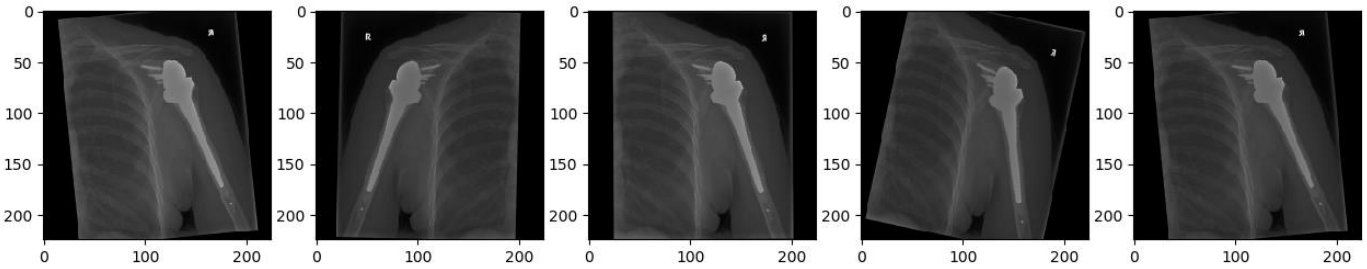
Training

Data Augmentation

When training each image is loaded in as a grayscale image with only one channel, and has the following augmentations applied to it:

- Zero padded and resized to be 224 x 224.
- Random rotation within ±30 degrees.
- Randomly flipped horizontally.

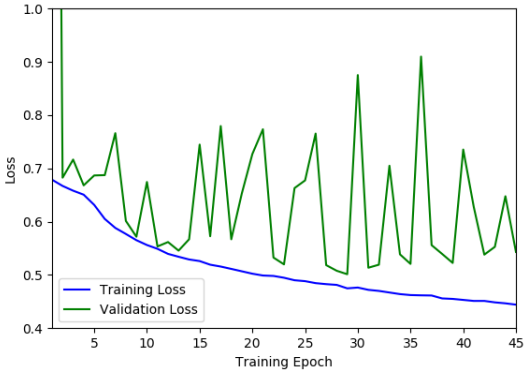
Here is an example of an image that was run through the data augmentation pipeline 5 times:



The images are also converted to be 3 channels by duplicating the single channel three times. The reason is explained in the next section.

Training

After much testing, the model is unable to learn when training using randomly initialized weights. The model would quickly learn to classify all samples as 'Normal' giving a training accuracy of 60% and a validation accuracy of 52%. Instead, the weights are initialized to be weights pretrained on ImageNet, as suggested by the authors of the MURA paper [1] [3], which allowed the model to start learning to detect meaningful features and train past 60% training accuracy. The pretrained weights were trained on colour ImageNet images meaning the model expects 3 channels on the input image. Therefore, when augmenting the images, the one channel grayscale images are duplicated and turned into 3 channel images before being feed through the network.



From this plot of the training and validation loss during training we can see that training loss is consistently being reduced, however while the validation loss is on average decreasing, it still varies wildly. It is not clear that this is an issue of overfitting since it happens all throughout training and increasing dropout influenced training loss but doesn't seem to effect validation.

Evaluation

For the best performance, an ensemble of 4 models were selected from about 15 that were trained. They were chosen based on having the lowest validation loss.

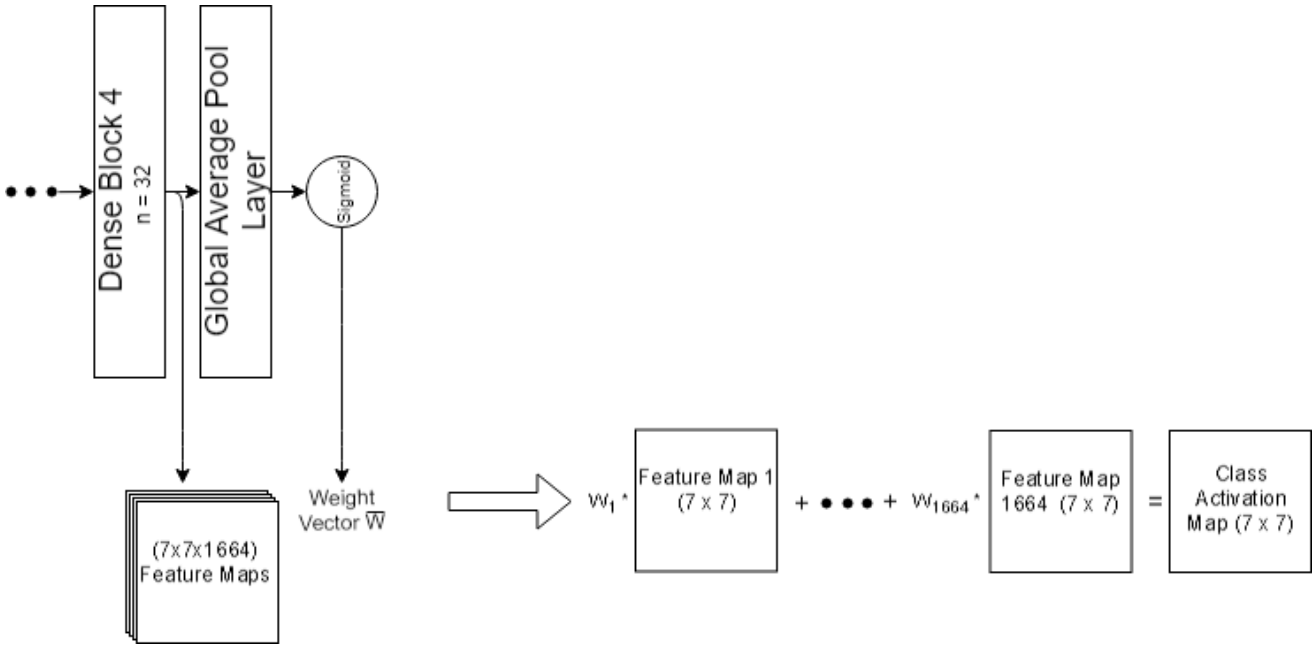
Model	Training Loss	Validation Loss	Validation Accuracy
DenseNet169_5	0.5044	0.5435	73%
DenseNet169_7	0.6012	0.6913	70%
DenseNet169_8	0.4962	0.5438	76%
DenseNet 169_9	0.5140	0.5778	72%

Evaluation was done by averaging the predictions produced by each model for a final prediction for a given image. Since each study consisted of multiple images, the predictions of all the images for a given study were averaged for a final prediction for that study. These are the final results:

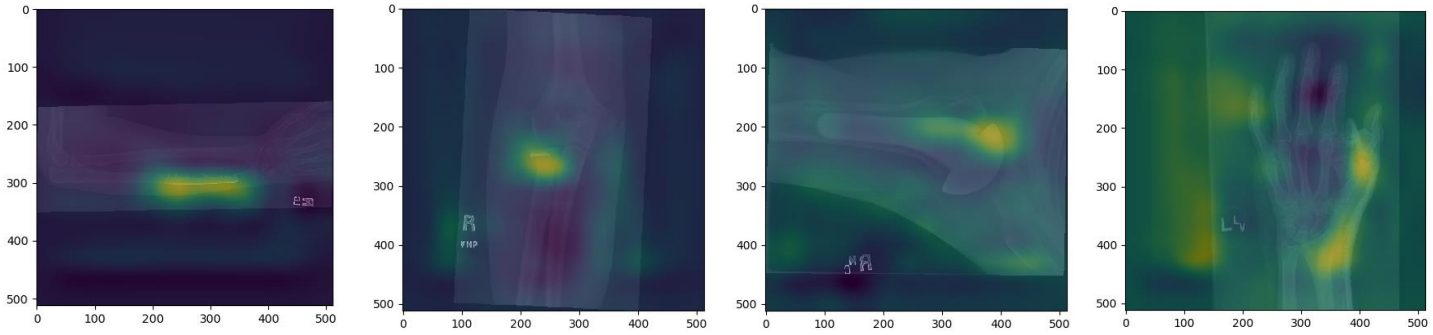
Category	Study Accuracy
SHOULDER	74.2%
HUMERUS	82.2%
FINGER	79.4%
ELBOW	83.5%
WRIST	85.2%
FOREARM	79.6%
HAND	73.6%
TOTAL	**79.8%**

Visualization

Since DenseNet makes use of a Global Average Pooling Layer before the fully connected layer, we can use Class Activation Maps(CAM) to visualize which areas of an image contributed the most when classifying [4]. A class activation map can be computed by taking the feature maps produced by the final convolutional layer and weighting each feature map with its corresponding fully connected layer weight and summing them all together.



When training the models, images of size 224x224 were used. However, at that resolution the feature maps produced at the final convolutional layer are of size 7x7 which doesn't provide a high enough resolution to identify the meaningful details. Instead when computing the class activation maps images can be scaled up to 512x512 which gives us feature maps of size 16x16. These are the class activation maps of some samples that are labelled as abnormal:



Conclusion

I was able to train an ensemble of models that together reached a reasonable validation accuracy of 79.8%, especially when considering that most of the models didn't train to have a training accuracy of more than 80%. However, there is still room for improvement as validation loss varies wildly throughout the training process (pg. 3). If that can be resolved or mitigated, I believe that the model can be finetuned and trained for longer so it can perform better. Another possible area of improvement would be to clean up the data since there are many cases of:

- Markers with letters that is influencing the network's predictions
- Multiple x-rays in a single image
- Most x-ray images are negatives but some are positives

In this project I was able to learn, implement and test the core concepts in neural networks and convolutional neural networks. Working with a challenging dataset like the MURA dataset gave me the opportunity to apply many of these concepts and it gave me an insight into how working with real data would be. Through a process of trial and error, I've become familiar with using TensorFlow, its Keras and low-level APIs. I have explored many techniques and network architectures used by researchers and practitioners. Overall I have gained a solid foundational understanding of Deep Learning.

Citations

[1] 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," *arXiv e-prints,* p. arXiv:1712.06957, 2017.

[2] G. Huang, Z. Liu, L. van der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," *arXiv e-prints,* 2016.

[3] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren and A. Y. Ng, "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning," *arXiv e-prints,* 2017.

[4] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, "Learning Deep Features for Discriminative Localization," *arXiv e-prints,* 2015.

Additional Resources

Learning Deep learning concepts: Coursera Deep Learning Specialization, Deep Learning Book
ImageNet pre-trained Weights: https://github.com/flyyufelix/cnn_finetune