Musculoskeletal Anomaly Detection

A project report submitted in fulfillment of the requirement for the

BAN 693 – Business Analytics Capstone Project

Submitted by

Raymond Balaian – sf5532

Ulysses Juan – vr6282

Soon Chye Lim – jn7736

Glee Truong – gq3622

Master of Science in Business Analytics

California State University, East Bay

Under the guidance of

Dr. Chongqi Wu

COLLEGE OF BUSINESS AND ECONOMICS

CALIFORNIA STATE UNIVERSITY, EAST BAY

Project Overview and Problem Statement: 2

MURA Dataset: 2

Figure 1: Directory Structure 3

Data Exploration and Preparation: 3

Figure 2: Binary directory structure 4

Figure 3: Categorical directory structure 4

Model Architecture: 4

Model Details: 5

Figure 5: Body Part with 7 softmax neurons output 6

Figure 6: Abnormality with 1 sigmoid neuron output 6

Figure 7: Abnormality irrespective of body part with 1 sigmoid neuron output layer 6

Figure 8: Our Model with 14 softmax neurons output 7

Figure 9: MobileNetV2 with 14 softmax neurons output 7

Classifications: 7

Figure 10: Prediction Samples 7

Figure 11: Models Results 8

Figure 12: Stanford University Results 8

Conclusions and Reflections: 9

Project Overview and Problem Statement:

In 1895 Wilhelm Conrad Roentgen discovered a new 'ray' that had the capability of passing through tissues, but not bones. Today we know this as "X-ray" or radiography. Radiography today helps physicians diagnose and understand symptoms, sometimes even before the symptoms escalate or cause additional problems. It takes years of schooling and practice to become a Radiologist. With some tuition money and Deep Learning, we can become rudimentary practitioners in identifying the abnormality of a radiograph of the musculoskeletal system.

Deep learning has the potential to help identify these issues by feeding proper data, validating to understand the results, and giving us a prediction with a certain percentage of accuracy for the possible outcome. Our attempt for this project was to use deep learning to identify the difference of skeletal structure and if there are any abnormalities in the skeletal structure. The approach of the project wasn't to replace actual physicians from the job, but to be a second layer of diagnosis to either lighten the physicians load or perhaps identify abnormalities in images that the physicians missed. The measures we are using as comparison are from the 2018 Stanford report: "*MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs*". The Stanford team had randomly selected three different radiologists to run their model up against and used Cohen's kappa to establish their overall score. We aim to get a comparable accuracy score to the Stanford model and radiologists.

MURA Dataset:

The data was provided by the Stanford Program for Artificial Intelligence in Medicine and Imaging: https://stanfordmlgroup.github.io/competitions/mura/. We have over 40,000 x-ray images that were split between training and validation folders. The training folder has 36,812 images, whereas the validation folder has 3,197 photos. In other words, the split was between 90% training and 10% validation. With such an extensive dataset, we should have more than enough data to do our research.

Figure 1 below shows the directory location of the bones, namely: elbow, finger, forearm, hand, humerus, shoulder, and wrist. From those skeletal classes the files broke into unknown patients' folders. In those folders, we can see the information on whether their bones are normal or not. A positive patient indicates the patient has an abnormality, and a negative patient

indicates the patient does not have an abnormality. Typically, we think of abnormality as a broken bone, but implants and metal hardware inside joints and bones are abnormal as well. Each image in the patient's folder varies in size, and the image is in PNG format and RGB color system. It is necessary to recognize that the pictures are RGB because the neural network will require us to define the shape of images before running through the model.

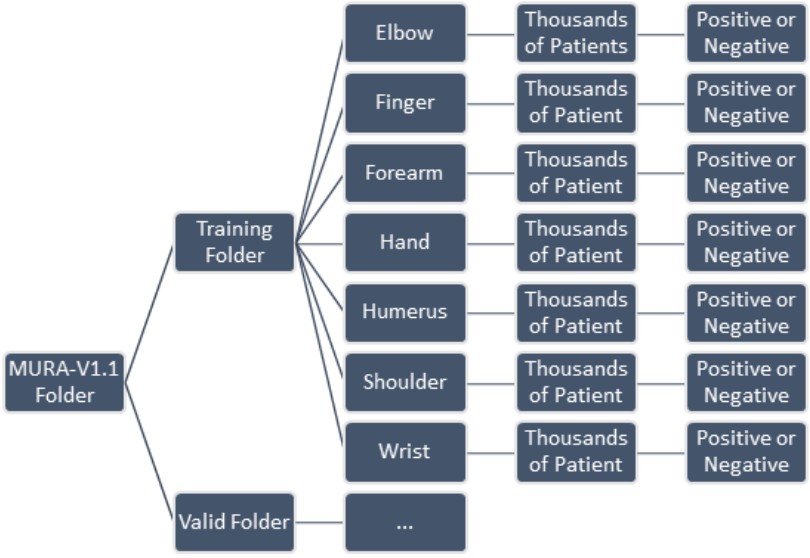


Figure 1: Directory Structure

Data Exploration and Preparation:

We are using Image Data Generator to feed each image to our model. Before we constructed our generator, we defined our input shape to 224 x 224 x 3. The shape means 224 pixels by 224 pixels by 3 RGB filters. If any images do not match the input size, we will get an error. Therefore, we converted every image to a target size of 224 by 224 by 3.

Next, we augmented our images so we can have more diverse data available for the training models. How we did this was, from the generator we were able to rescale the image size and manipulate the image by: flips (vertical and horizontal), rotations, and image shifts in multiple directions. For the validation there was no preparation needed beside rescaling to use later as a comparison.

Figure 2: Binary directory structure

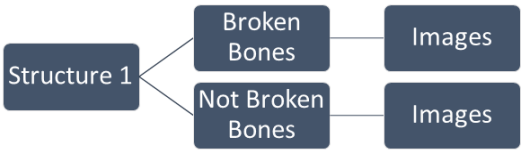
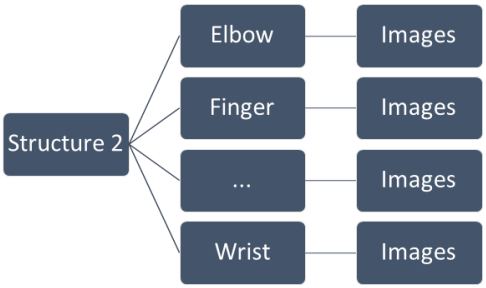


Figure 3: Categorical directory structure



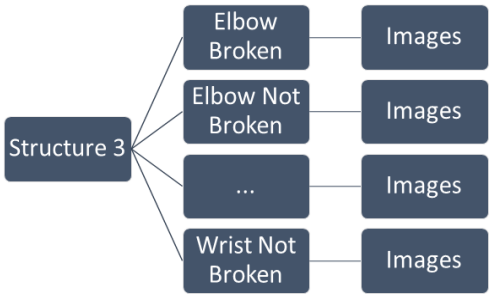


Figure 4: 14 category directory structure

Because we are using a generator, we had to change the directory structure of our files accordingly. The above diagram illustrates three different types of directory structures. If we want to determine if the bones are broken, figure 2 is desired. If we are going to discover the location of the bone, figure 3 is preferred. And finally, if we want the combination of both, figure 4 is desired. Then, we began using our generator tailored to the directory arrangement.

Model Architecture:

For our model architecture we approached it three different ways. The first is to simply classify the radiograph as normal or abnormal. This is a binary model with a single sigmoid neuron in the output layer. The second is to create 14 categories, seven body parts * normal or abnormal. This is our 14cat model and has 14 softmax neurons in the output layer. Finally, you can split the two predictions, first evaluating the body part and then determining whether the radiograph is normal or abnormal. We call this the “7+2” model and it suffers from the uncertainty of two predictions.

The Stanford competition on this topic took place in 2018 and since we have the benefit of hindsight, we can look at the top performing models there for some education. We found that most of the competitors designs were based on identification of the body part, and then a determination of normal or abnormal. This was most similar to the 7+2 design and so we started there.

Model Details:

We found a few things to be true in relation to this dataset. First off, it's LARGE. This leads to longer epoch times and longer iterations in development. Early on we dropped ~80% of the images so that we could iterate faster, but eventually we reached the point of maximum one iteration per day due to training time. After multiple runs through the dataset, we found that it's easy to over-augment images and we don't need 20 versions of the same image. A flip and a rotation seem effective without giving up a lot of epoch time.

Since we are dealing with images, all the models use 2-Dimensional Convolution layers (Conv2D). Conv2D iterates over each pixel with a kernel, and you have the option to take the kernel results as-is, or to pool them together somehow. Pooling reduces the number of pixels available to the next layer in the model. While running the MobileNet model we found Average Pooling to be more effective than Max Pooling. We also found that deeper models perform better than shallow and wide models. The reasoning behind this is that we wanted our model to be learning more than memorizing. By building a multiple layer model it cuts down on training time and is able to generalize better. Finally, and maybe the most important lesson…. it's hard to beat transfer learning. The open models we tried are more accurate than any of the models we came up with ourselves. DenseNet, VGG16/19, and MobileNet all had better accuracy than the best of our own models.

When comparing MobileNetV2 (6 layers), DenseNet (121, 169, and 201 layers), VGG(16 and 19 layers), and our teams models which had as many as 60 layers we found the MobileNetV2 implementation was the best model. Part of this can be attributed to MobileNet's lightweight design to be less computational. The benefits from this was our epoch time dropped 50% compared to the other models. While we would have liked one of our models to be the best, none of our models were able to beat the simple MobileNetV2 implementation.

In our first effort we were able to classify the body part with 92% accuracy but were only able to achieve ~69% in the normal\abnormal determination.

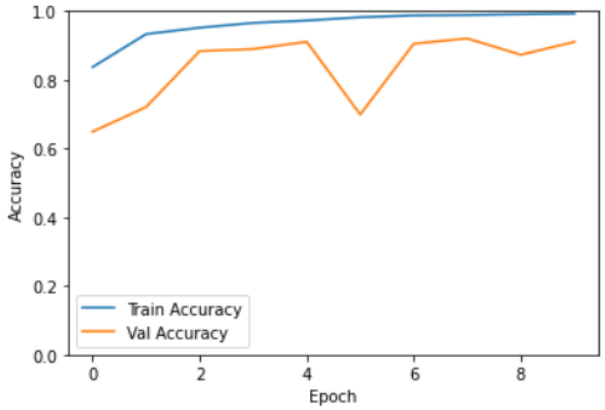


Figure 5: Body Part with 7 softmax neurons output

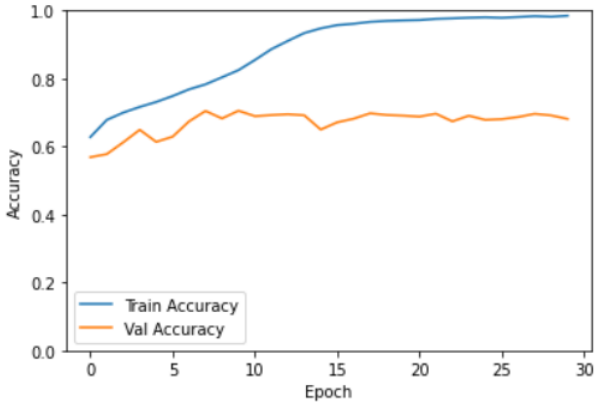


Figure 6: Abnormality with 1 sigmoid neuron output

Our second approach was binary classification, with the two classes being Normal and Abnormal, irrespective of body part. With this model we were able to attain accuracy around 72%.

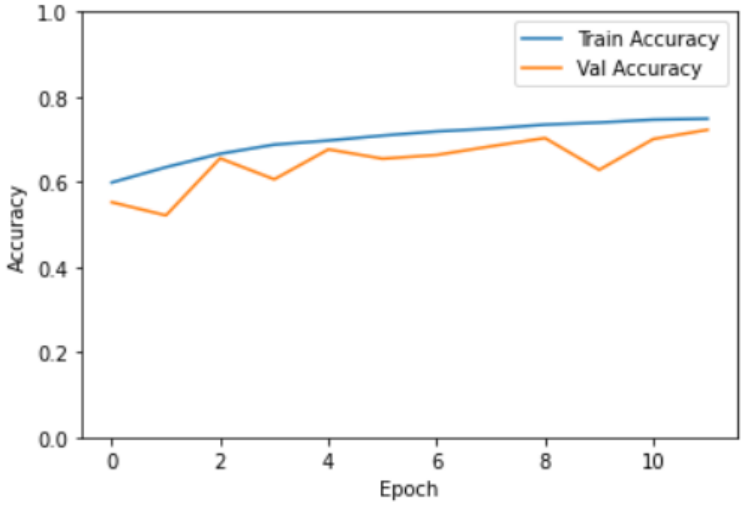


Figure 7: Abnormality irrespective of body part with 1 sigmoid neuron output layer

Our last approach was a 14-category approach (each body part, normal\abnormal). We tried our own model here with 72% accuracy, and MobileNetV2 with 74% accuracy.

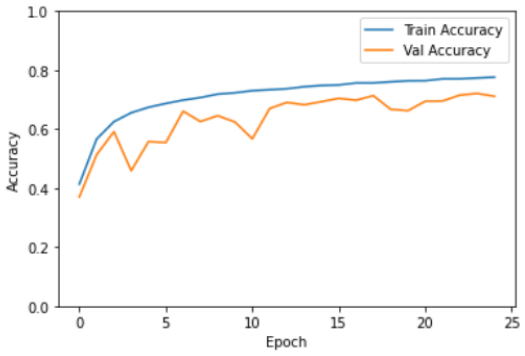


Figure 8: Our Model with 14 softmax neurons output

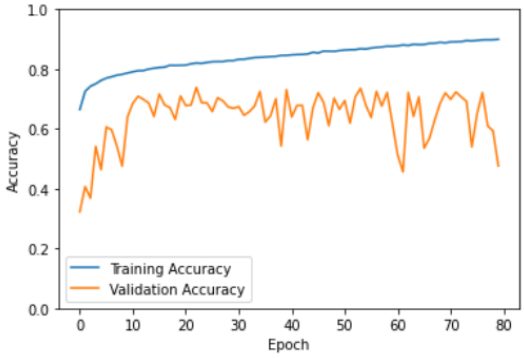


Figure 9: MobileNetV2 with 14 softmax neurons output

Classifications:

Some sample classifications are shown here. The model is pretty good, usually predicting seven out of ten, and sometimes even nine or ten out of ten with some luck. From the sample, we can see that our abnormal elbow was predicted to be a normal one. That scenario could be true if you met the person in flesh, but in our case, we could see that the person has a rod and few screws holding his elbow together. From a black box perspective, we do not know the reasoning for this classification. Our assumption could be anything from the training set just didn't have enough images of that specific injury or even an image of the medical equipment used to put the bones back in place.

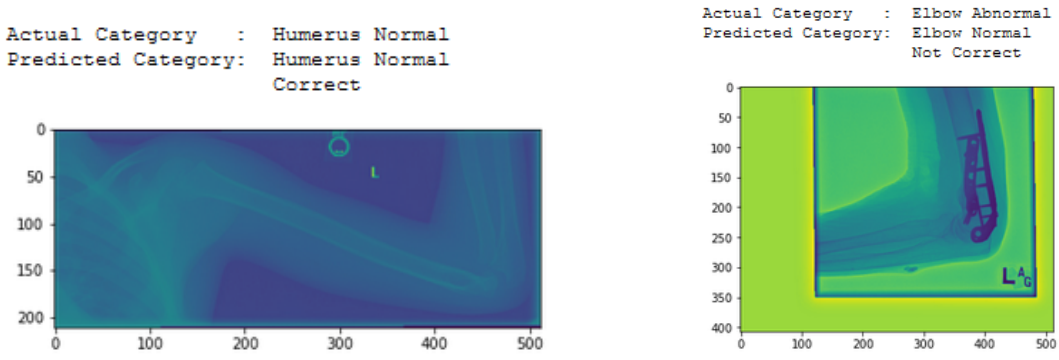


Figure 10: Prediction Samples

Validation Accuracy:

Results from our four models are as follows:

Figure 11: Models Results

Model	Conv2D	Conv2D	Conv2D	MobileNetV2
Classification Type	7 + 2	Binary	14 class	14 class
Layers	7 + 9	9	10	6
Validation Accuracy	.92 * .69 = 0.64	0.72	0.72	0.74
Naive Accuracy	.14 * .5 = 0.07	0.5	0.14	0.14

Looking at the accuracy of our models above we can see that our models are within the range of 64% to 74%. When comparing our score to the overall score from the Stanford Model and Radiologists' accuracy (below) we can see that our Conv2D and MobileNet2D accuracy is comparable. In some cases, our model scored higher compared to the overall score by a few percent.

Figure 12: Stanford University Results

	Radiologist 1	Radiologist 2	Radiologist 3	Model
Elbow	0.850 (0.830, 0.871)	0.710 (0.674, 0.745)	0.719 (0.685, 0.752)	0.710 (0.674, 0.745)
Finger	0.304 (0.249, 0.358)	0.403 (0.339, 0.467)	0.410 (0.358, 0.463)	0.389 (0.332, 0.446)
Forearm	0.796 (0.772, 0.821)	0.802 (0.779, 0.825)	0.798 (0.774, 0.822)	0.737 (0.707, 0.766)
Hand	0.661 (0.623, 0.698)	0.927 (0.917, 0.937)	0.789 (0.762, 0.815)	0.851 (0.830, 0.871)
Humerus	0.867 (0.850, 0.883)	0.733 (0.703, 0.764)	0.933 (0.925, 0.942)	0.600 (0.558, 0.642)
Shoulder	0.864 (0.847, 0.881)	0.791 (0.765, 0.816)	0.864 (0.847, 0.881)	0.729 (0.697, 0.760)
Wrist	0.791 (0.766, 0.817)	0.931 (0.922, 0.940)	0.931 (0.922, 0.940)	0.931 (0.922, 0.940)
Overall	0.731 (0.726, 0.735)	0.763 (0.759, 0.767)	0.778 (0.774, 0.782)	0.705 (0.700, 0.710)

[1]

Due note that scores above are accuracy scores, we are not able to use kappa in this case to establish a "gold standard". The reason for this is because there was no documentation of

[1] Rajpurkar, Pranav, Jeremy Irvin, Aarti Bagul, Daisy Ding, Tony Duan, Hershel Mehta, Brandon Yang et al. "Mura: Large dataset for abnormality detection in musculoskeletal radiographs.", 5

observed agreement and expected agreement. These two agreements are needed for us to establish the kappa score.

Conclusions and Reflections:

We started this project hoping to get an accuracy comparable to the Stanford team. Although we cannot fully compare our models to their kappa score, we certainly had many lessons along the way. It is easy to say, "they used DenseNet169 to get 84% accuracy", but then you implement DenseNet169 and score a 60% accuracy rate. We can easily say these factors from architectural approach, layer design, padding, shape, normalization, activation, loss function, optimizer, batch size, learning rate, pooling, and output layer can affect the accuracy between the two models. After multiple revisits and tuning, trying to find the most effective combination of them was our end goal. The fact that you can implement transfer learning using all defaults and still predict 60% accuracy into 14 classes is nothing short of amazing.

The complexity of the models is significant in all cases. All our models had at least ten layers and were generally computationally expensive. The epoch times you see in our Jupyter Notebooks are all from an Nvidia RTX3090, and we still trained for entire days sometimes. Running on basic hardware or laptop configurations is probably time-prohibitive on this dataset, even with MobileNetV2. It is worth noting that using a generator for image augmentation is much slower than being able to create tensors out of the images in advance, but the augmented images are valuable and so we sacrifice the execution time.

To our expectation, we did not expect to get a higher accuracy score compared to the Stanford overall score. Looking ahead, we think the most significant change we can make for next time is to train seven individual binary models, one for each body part. This would localize the model's learning to that body part, likely allowing greater accuracy. This was a great project for us from start to finish. It was challenging, it was hard, and in the end, we had a fantastic outcome against the naive predictions. Thanks very much for the education and the opportunity.

Bibliography:

Rajpurkar, Pranav, Jeremy Irvin, Aarti Bagul, Daisy Ding, Tony Duan, Hershel Mehta, Brandon Yang et al. "Mura: Large dataset for abnormality detection in musculoskeletal radiographs." *arXiv preprint arXiv:1712.06957* (2017).