Musculoskeletal Anomaly Detection

Capstone Project

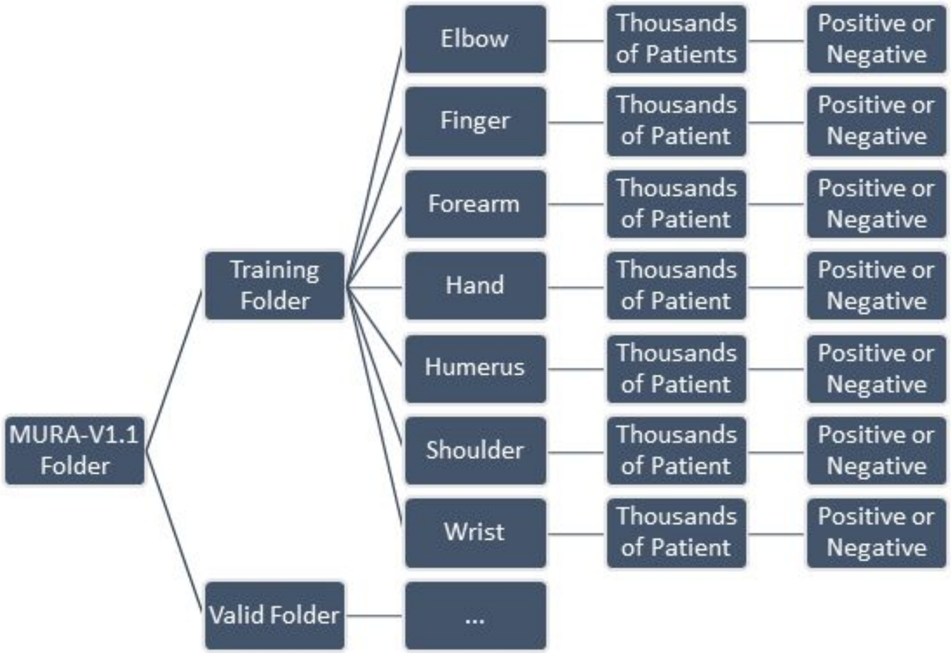
Ray Balaian
Glee Truong
Soon Chye Lim
Ulysses Juan

Overview and Problem Statement

- In this project we explore the viability of using neural networks to diagnose x-ray images.
- We aim to detect the body part in the x-ray, as well as any abnormalities like fractures or implanted medical equipments.
- The goal is to build a model that can take x-ray images as data, clean, prep and feed a neural network to produce an output which detects the body part and whether its normal or abnormal.
- We will use the accuracy of the validation set to compare our model to the naive model and other available sample models.

Data Description

- Obtained from Stanford Website
- 40,000 images with 37k training and 3k valid
- File Directory
- Images:
  - Varies in size
  - PNG
  - RGB

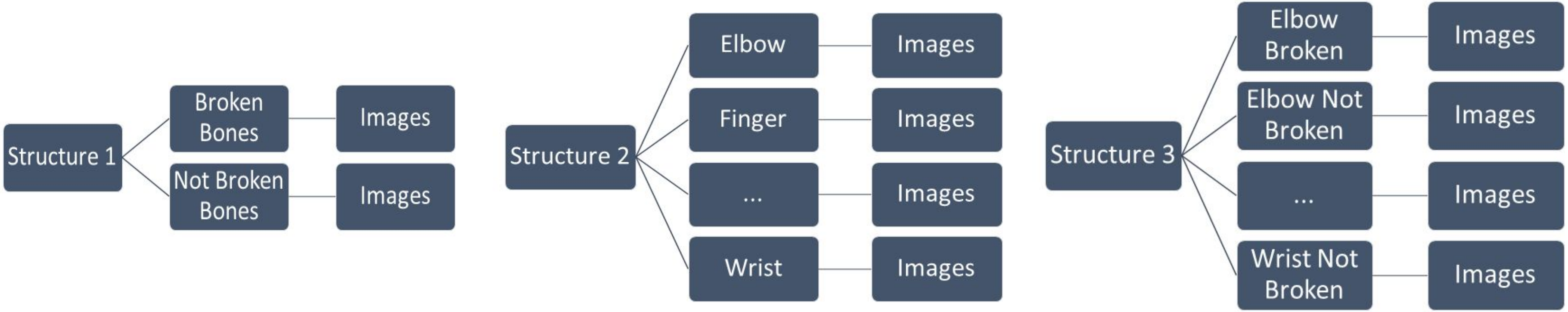


Data Cleaning

1. Target Size (224 x 224 x 3)

2. Image Data Generator
   a. Augmentation (rescale, flip, rotation, shift)

3. 3 Directory Structures for 3 Different Research Questions



Model Architecture

There are a few approaches to this problem. You can treat as such:

- Binary classification
  (normal\abnormal, ignoring body part)

- 14-class classification
  (7 body parts, normal\abnormal)

- “7+2” model: predict body part first (7 classes),
  then normal\abnormal (binary)

Models and Observations

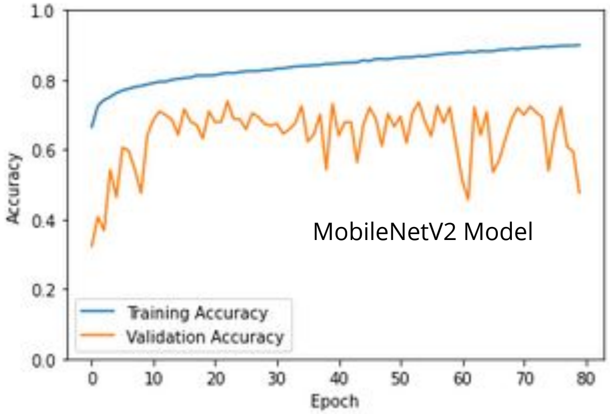
- We tried MANY models of our own design, with various:
  - Layer Types\Order, Depth, Width, Epochs, Optimizers, Batch Norm, Dropout, Regularization, Image Augmentation, etc.

Observations

- Deeper is better than wider
- Conv2D with pooling is better than without
- Better to do BatchNormalization
- It’s possible to do too much image augmentation. Some rotation and one flip seemed to work best.
- It’s easy to overfit with a deep model. Use Dropout as needed.
- Better results with ADAM vs SGD
- Accuracy varies between epochs. Use callbacks to save best weights.
- Transfer models perform the best (DenseNet, VGG, MobileNet, etc…)

Evaluation

Model	Conv2D	Conv2D	Conv2D	MobileNetV2
Classification Type	7 + 2	Binary	14 class	14 class
Layers	7 + 9	9	10	7
Validation Accuracy	.92 * .69=.64	.72	.72	.74
Naive Accuracy	.14 * .5 = .07	.5	.14	.14



MobileNetV2 Model

Stanford Study Results

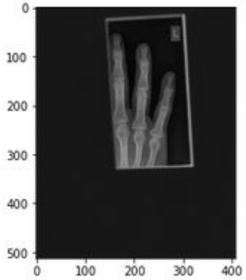
	Radiologist 1	Radiologist 2	Radiologist 3	Model
Elbow	0.850 (0.830, 0.871)	0.710 (0.674, 0.745)	0.719 (0.685, 0.752)	0.710 (0.674, 0.745)
Finger	0.304 (0.249, 0.358)	0.403 (0.339, 0.467)	0.410 (0.358, 0.463)	0.389 (0.332, 0.446)
Forearm	0.796 (0.772, 0.821)	0.802 (0.779, 0.825)	0.798 (0.774, 0.822)	0.737 (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)

Results

Actual Category : Elbow Abnormal
Predicted Category: Elbow Abnormal
Correct

Actual Category : Forearm Normal
Predicted Category: Forearm Normal
Correct

Actual Category : Finger Abnormal
Predicted Category: Finger Normal
Not Correct



Actual Category : Finger Normal
Predicted Category: Finger Normal
Correct

Correct predictions from a series of ten 100-image predictions:

Correct Predictions: 71, 74, 76, 74, 76, 72, 73, 70, 65, 71

Avg = 72.2

Conclusions and Reflections

- It is difficult and computationally expensive to build a large scale deep neural network with complex data (images).

- Model architecture and detailed hyperparameter tuning are **key** in order to excel in accuracy.

- What would we do differently?
  - Trim image counts to iterate faster.
  - More controlled testing of hyperparameters.
  - Develop seven binary models, one for each body part.