



Automated Detection of Breaks and Fractures in X-Ray Bone Images

Maithreyi Gopalakrishnan, Jaymee Sheng, Maurizio Valesani
{mgopalak, jzs, valesani} @ stanford.edu

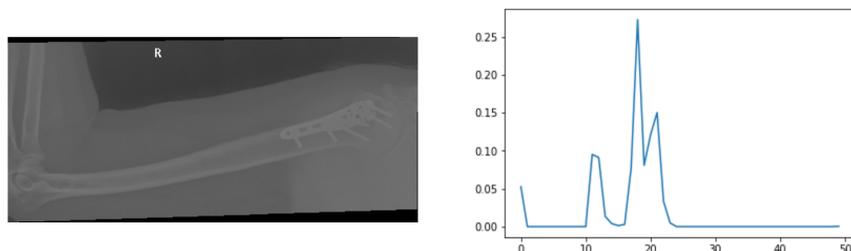
Introduction

Today when a radiologist diagnoses a break or fracture in a bone, he or she simply does so by the eye, which could be prone to error. The motivation of this project is to utilize machine learning techniques to automatically detect breaks and fractures in bone X-ray images. This can serve as an initial flagging system or as a secondary check to verify that the radiologist has identified all anomalies. Eventually, such an algorithm can be extended to identify breaks on a 3-dimensional scale, i.e. with CT scan data.

We compare two different approaches to identifying anomalies in X-ray images: first, we input normalized histograms of images to perform logistic regression and k-nearest neighbors; second, we input images directly into a convolutional neural network.

Data and Features

- We use MURA (musculoskeletal radiographs), a large public dataset of bone X-rays, and focus on the humerus for the scope of this project
- The resolutions of these images varied, with one dimension of the images being around 512 pixels and the other ranging from 200-600 pixels
- For supervised learning, we convert each image into a normalized histogram so it can be directly compared with histograms of other images
- For the neural network, we convert all images to the same size by padding with pixels of the color of the mode of the image



Example of a positive (anomalous) training image and its normalized histogram with bin=50.

Models & Results

| Model | Test Accuracy |
|--|---------------|
| Logistic regression | 0.61 |
| Logistic regression with L2 regularization ($\lambda = 0.001$) | 0.64 |
| k-nearest neighbors ($k = 10$) | 0.67 |
| Convolutional neural network | 0.60 |

k-NN:

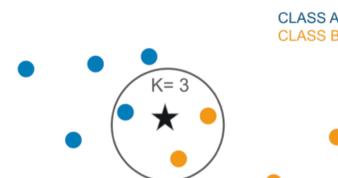
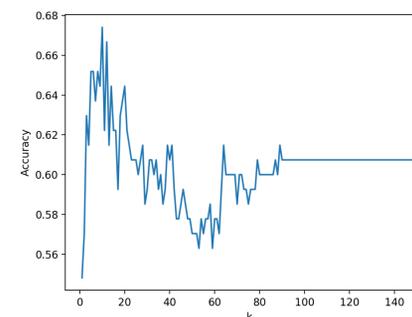


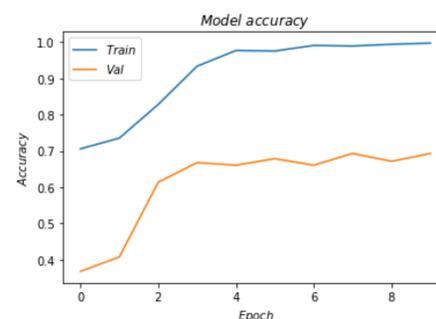
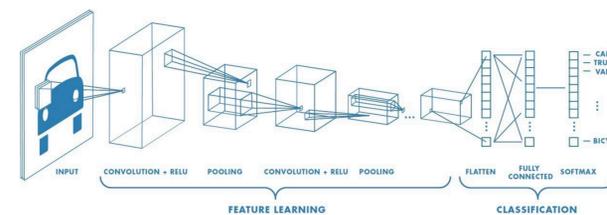
Illustration of k-NN algorithm. [1]
If $k=3$, the unknown point will be assigned to class B.



The optimal k based on performance on test data is $k=10$.

CNN:

Example structure of a CNN for multi-class classification. [2]



The final structure of the CNN consists of :

- an input convolutional layer
- 3 additional convolutional layers (with pooling and dropout layers in between)
- a flattening layer
- 3 dense layers (with dropout layer in between)
- output dense layer

The ReLU function is used as the activation function for all convolutional and dense layers except the final output layer, which uses the sigmoid function.

Discussion

- Surprisingly, the “laziest” learning algorithm – k-nearest neighbors – produced the best result. By selecting $k=10$, we strike a balance between reducing the noise and making prediction using the most relevant data points.
- The strong performance of the supervised learning models suggests that the features we created from the raw image data were useful and perhaps even interpretable.
- For the CNN, we tried different approaches to address the problem of overfit and saw significant improvement in test accuracy after adding dropout layers and reducing the size of the CNN.

Summary / Future Work

- Our logistic regression model provided a pretty strong baseline for classifying positive (anomalous) and negative (normal) bone X-ray images
- Improved accuracy based on regularization and k-NN speaks to the power of feature selection and model simplicity
- Performing more aggressive pre-processing on the images before feeding them into the CNN would be helpful, as well as increasing the size of the training set by applying data augmentation
- Given more time, we would also investigate different types of bones for which data is available and consider other methods of feature selection and hyperparameter optimization

References

- [1] Atul. *K-Nearest Neighbors Algorithm Using Python*. edureka! 2019. URL: <https://www.edureka.co/blog/k-nearest-neighbors-algorithm/>
- [2] Sumit Saha. *A Comprehensive Guide to Convolutional Neural Networks*. Ed. by Towards Data Science 2019. URL: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>