Knee Osteoarthritis Classification Using Bone Distance

Viswanathan Appakkudal Ramani,

*Seidenberg School of Computer Science and Information Systems*

*Pace University, New York, NY, USA*

***Abstract*—Knee Osteoarthritis (OA) is a disease which commonly occurs to people of varying age groups wherein the permeable membrane called cartilage which holds the femur bone and tibia bone together gets worn out. As the cartilage wears down, the bone distance between the two bones (tibia, femur) decreases. If the bones are very close to each other, it’s highly probable that user can get acute pains on movement of legs. This project deals with a mechanism through which intelligent agents could predict whether a patient has Osteoarthritis or not, using the distance between the femur and tibia bones. Additionally, few features such as pixel counts, and KL grade scores were used to predict the target. This paper will demonstrate several features and models on which the predictions are based. The intelligent machine was able to predict the target with an accuracy of 73% and ROC area of 73.6%.**

***Keywords*—Osteoarthritis, accuracy, ROC area, grade, target, feature, computer vision, distance**

# **INTRODUCTION**

The wearing down of cartilage between the Tibia and Femur bones is called as Knee Osteoarthritis (OA). OA is the most common type of arthritis. Osteoarthritis tends to occur in middle age or due to an injury or obesity. The most common causes of OA are:

* Age
* Obesity
* Knee Injury
* Sporting/ Athletics
* Weak muscles
* Genetics
* Sex

OA occurs commonly for elderly people who are greater than 40 years. People who have a knee pain and are older than 40 years are highly probable of having an OA. Overuse of knees, genetical disorder of OA are also common causes of OA.

The symptoms of OA include knee pain, knee stiffness, less physical activity, clicking or cracking sound when joint bends, swelling around the knee joint, change of knee posture and walking patterns.

Knee Osteoarthritis doesn’t have a cure, however there are therapeutics to mitigate the disorder. Few of them include, physiotherapy, losing weight, controlling blood sugar, healthy physical activity, and a healthy lifestyle.

# **PROJECT REQUIREMENTS**

The project is developed using the following technologies:

* MATLAB v2021
* Weka

MATLAB (an abbreviation of "matrix laboratory") is a [proprietary](https://en.wikipedia.org/wiki/Proprietary_software) [multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language) [programming language](https://en.wikipedia.org/wiki/Programming_language) and [numeric computing](https://en.wikipedia.org/wiki/Numerical_analysis) environment developed by [MathWorks](https://en.wikipedia.org/wiki/MathWorks). MATLAB allows [matrix](https://en.wikipedia.org/wiki/Matrix_(mathematics)) manipulations, plotting of [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) and data, implementation of [algorithms](https://en.wikipedia.org/wiki/Algorithm), creation of [user interfaces](https://en.wikipedia.org/wiki/User_interface), and interfacing with programs written in other languages.

Weka is tried and tested open source machine learning software that can be accessed through a graphical user interface, standard terminal applications, or a Java API. It is widely used for teaching, research, and industrial applications, contains a plethora of built-in tools for standard machine learning tasks, and additionally gives transparent access to well-known toolboxes such as [scikit-learn](https://markahall.blogspot.co.nz/2015/06/cpython-integration-in-weka.html), [R](https://markahall.blogspot.com/2012/07/r-integration-in-weka.html), and [Deeplearning4j](https://deeplearning.cms.waikato.ac.nz/).

# **DATA**

The data contains images of 193 cases, wherein each case corresponds to a patient. Initially a Magnetic Resonance Imaging (MRI) scan is around the knee joint to get a 3D image of the knee joint. MRI uses a strong magnetic field and radio waves to create detailed images of the organs and tissues within the body. An MRI scan uses a large magnet, radio waves, and a computer to create a detailed, cross-sectional image of internal organs and structures. The scanner itself typically resembles a large tube with a table in the middle, allowing the patient to slide in. MRI image of a knee looks as shown in Figure 1.

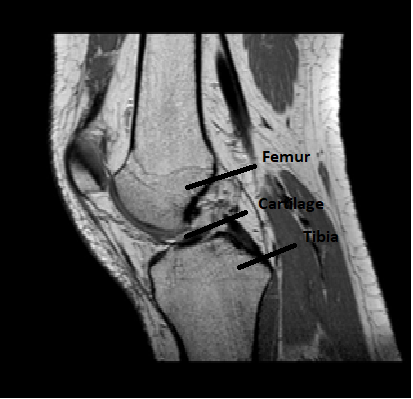


Figure 1. MRI image of knee

Image displayed in Figure 1 is a 2D representation of a 3D image. The image displays Femur and Tibia bones which is separated by a thin layer known as cartilage. A 3D image could be converted to 160 2D images. Hence, each 193 cases have 160 images. Post the conversion each image is segmented to get only the femur bone, tibia bone and the cartilage, as shown in Figure 2.

A picture containing text, nature

Description automatically generated

Figure 2. Segmented image

The segmented image is obtained using an automation segmentation network called U-Net. U-Net is a convolutional network architecture primarily used for bio-medical image processing [1]. U-Net is an architecture that belongs to Semantic segmentation. The goal of semantic image segmentation is to label each pixelof an image with a corresponding class of what is being represented.

# **METHODOLOGY**

As a first step in the machine learning pipeline, I performed data processing to get the desired image. Taking the segmented image, I used Canny Edge Detector to obtain only the edges as shown in Figure 3.

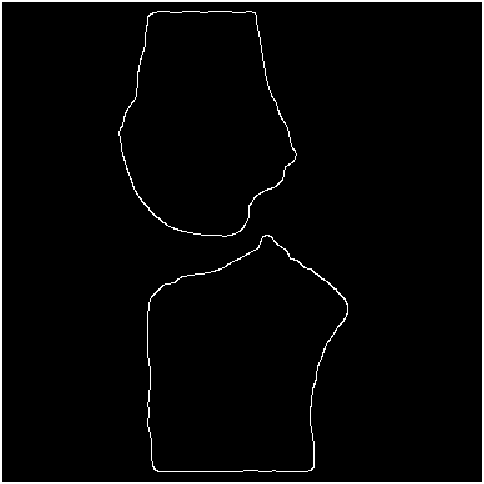


Figure 3. Bone Edges of segmented image

After obtaining the edges of Femur and tibia bones, I find the centroid of the femur and tibia bones as shown in Figure 4.

A picture containing rectangle

Description automatically generated

Figure 4. Centroids of Femur and Tibia

Along the horizontal axis I drew a parallel line crossing the centroid of Femur to identify the points where the line and the edge meets as shown in Figure 4.

A picture containing diagram

Description automatically generated

Figure 5. Horizontal line through Femur Centroid

The points at which the line meets the edges correspond to the range I would be iterating to obtain the distance.

To compute the distance, I first draw a line connecting the femur and tibia centroids as shown in Figure 6.

Diagram

Description automatically generated

Figure 6. Line connecting centroids of two bones

From figure 6, I find the pixels at which the line meets tibia and femur bones. Euclidean distance is computed from obtained pixels. Euclidean distance is represented as:

**Distance = (y2 -y1)2 + (x2 - x1)2**

where,

y1, y2 are the y co-ordinates of tibia and femur bones respectively,

x1, x2 are the x co-ordinates of tibia and femur bones respectively

# **EVALUATION METRIC DEFENITIONS**

* **Accuracy**: the accuracy metric measures the ratio of correct predictions over the total number of instances evaluated [2].
  + Accuracy = total number of correct instances/ total instances
* **Precision**: Precision is used to measure the positive patterns that are correctly predicted from the total predicted patterns in a positive class [2].
  + Precision **=** TP/ (TP + FP)
    - Where, TP – True Positive
    - FP – False Positive
* **Recall**: Recall is used to measure the fraction of positive patterns that are correctly classified [2].
* Recall = TP/ (TP +TN)
  + - where TN = True Negative
* **Specificity:** This metric is used to measure the fraction of negative patterns that are correctly classified [2].
  + Specificity = TN / (TN + FP)
* **ROC** **Area /AUC:** The ROC curve (receiver operating characteristic curve)also known as Area Under ROC curve (AUC)is the plot false positivity rate and true positivity rate. The area between the FP rate and TP rate is the ROC [2]. For two classes, AUC is represented as:
  + AUC = (Sp – (np + 1)/2) / (np\*nn)
    - Where Sp = sum of the all positive examples ranked
    - np and nn denote the number of positive and negative examples

# **EXPERIMENT RESULTS**

**Experiment 1:** The first feature extracted from the data is the minimum distance. The image that were considered for every image ranged between the numbers 60 to 110. Other images either carry less information or missing centroids, hence chose this range. Iterating through images of the given range for every case, found the minimum of the images for all the considered images and stored it in an array. Next computed the minimum of the array and made it the feature. The Euclidean distance was computed for every point between the start and end points as defined in Section IV. The target variable is grade. Later, I appended the dataset to Weka, and modeled using a Random tree to get an accuracy of 64.76% and ROC area of 60.1% as shown in Figure 7.

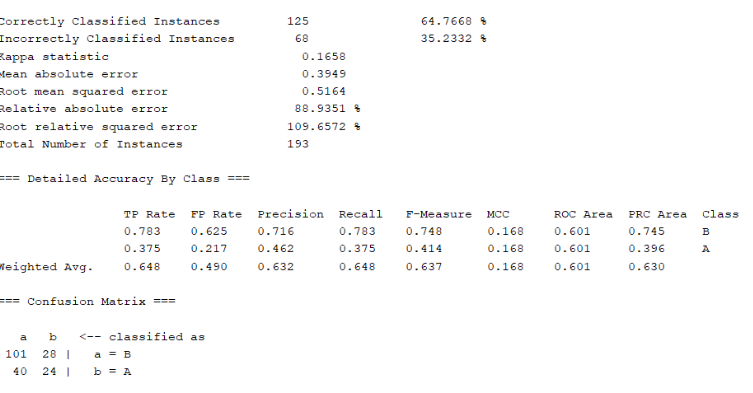


Figure 7. Evaluation metrics of Model 1

**Experiment 2:** The second feature extracted from the data is the normalized average distance and standard deviation. The image that were considered for every image ranged between the numbers 60 to 110. Other images either carry less information or missing centroids, hence chose this range. Iterating through images of the given range for every case, found the average of the images for all the considered images and stored it in an array. Next computed the average of the array and made it the feature. The Euclidean distance was computed for every point between the start and end points as defined in Section IV. The standard deviation is computed for every case measuring the deviation amongst the average array obtained. The target variable is grade. Later, I appended the dataset to Weka, and modeled using a Multi-Layer Perceptron with 10 cross fold validations to get an accuracy of 65.28% and ROC area of 55.5% as shown in Figure 8.

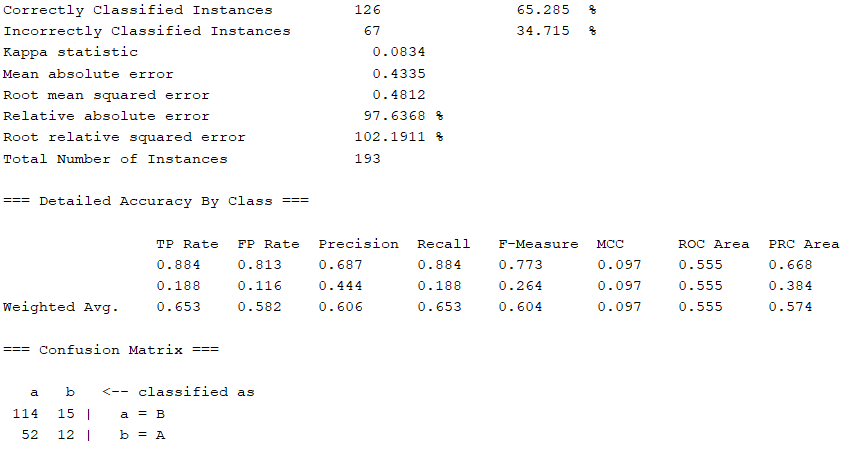


Figure 8. Evaluation metrics of Model 2

**Experiment 3:** The third feature extracted from the data is the Euclidean distance at 3 points (start, mid, end) as shown in Figure 9.

Diagram, schematic

Description automatically generated

Figure 9. Lines representing start, mid and end points to compute distance

The image that were considered for every image ranged between the numbers 25 to 140. Other images either carry less information or missing centroids, hence chose this range. Iterating through images of the given range for every case, found the distance between the corresponding femur and tibia edge points for all the considered images and stored it in an array. The Euclidean distance was computed for every point between the start and end points as defined in Section IV. Also, computed the number of white pixels between the 2 centroids for every image and then summed up the values for every image of the case to obtain the pixel count feature. Total number of features is 350. The target variable is grade. Later, appended the dataset to Weka, and modeled using a Random Forest with 5 cross fold validations to get an accuracy of 67.3% and ROC area of 65% as shown in Figure 10.

Table

Description automatically generated

Figure 10. Evaluation metrics of Model 3

**Experiment 4:** The fourth feature extracted from the data is the Euclidean distance at 5 points (start, 25%-mark, mid, 75%-mark, end) as shown in Figure 11.

Diagram

Description automatically generated

Figure 11. Lines representing start, 25%-mark, mid, 75%-mark and end points to compute distance

The image that were considered for every image ranged between the numbers 27 to 138. Other images either carry less information or missing centroids, hence chose this range. Iterating through images of the given range for every case, found the distance between the corresponding femur and tibia edge points for all the considered images and stored it in an array. The Euclidean distance was computed for every point between the start and end points as defined in Section IV. Then, computed the number of white pixels of every image, and appended it to the array. Also, used the KL grade score as one of the feature Total number of features considered is 674. The target variable is grade. Later, I appended the dataset to Weka, and modeled using a Random Forest to get an accuracy of 70.98% and ROC area of 66.3% as shown in Figure 12.

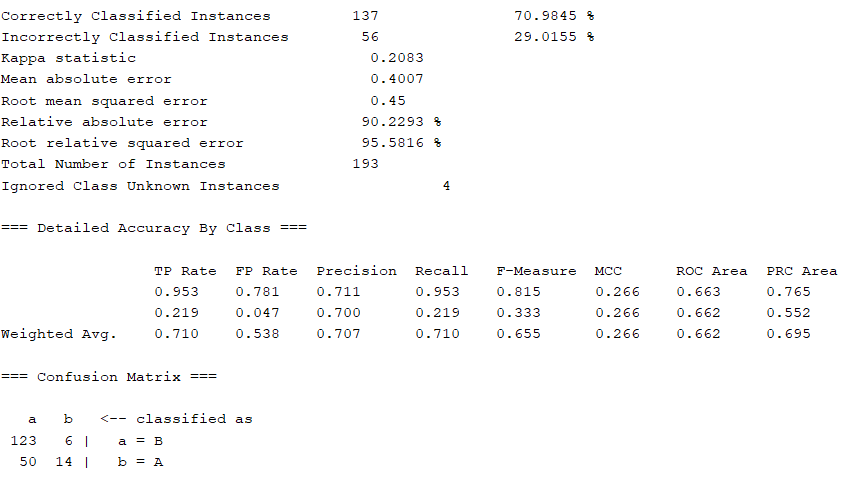


Figure 12. Evaluation metrics of Model 4

**Experiment 5:** The fifth feature extracted from the data is the Euclidean distance at 5 points (start, 25%-mark, mid, 75%-mark, end) as shown in Figure 11. Feature Extraction is the same as described in Experiment 4. However, removed highly correlate features and selected only 13 features (without KL grade score) from 674 features. The features selected are dist1\_58, dist3\_42, dist3\_57, dist3\_58, dist3\_61, dist3\_76, dist3\_79, dist5\_43, pixCount\_29, pixCount\_31, pixCount\_38, pixCount\_46, pixCount\_49. The target variable is grade. Later, I appended the dataset to Weka, and modeled using an Adaboost classifier to get an accuracy of 73.57% and ROC area of 73.7% as shown in Figure 13.

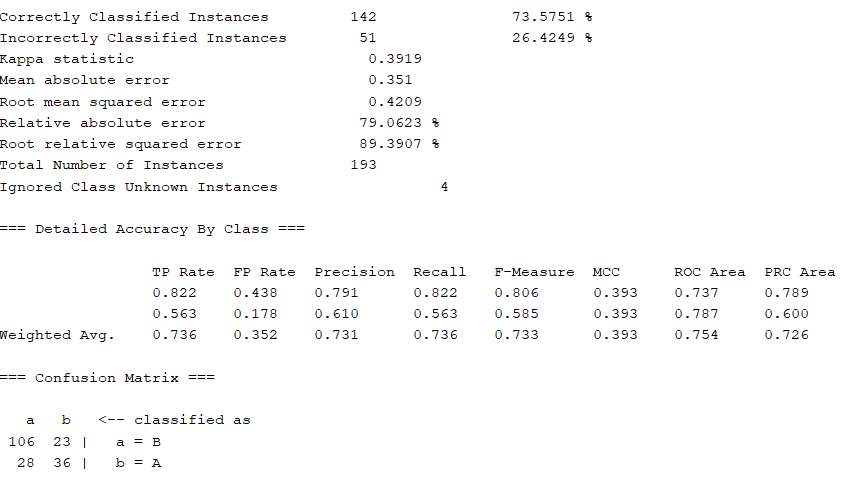


Figure 13. Evaluation metrics of Model 5

# **CONCLUSION**

Feature Engineering and model selection are crucial steps for building a good Computer Vision project. Experimenting different ideas, interpolating multiple features, and testing the features on a Machine Learning algorithm gives way not only to a product but also paves way for new ideas. From the above experiments, we could conclude that experiment 5 with 13 features and AdaBoost model performed better than other experiments. The experiment produced an accuracy of 73.57% and ROC area of 73.7%. We could also infer that a greater number of features (uncorrelated features) produce good results.

# **REFERENCES**

[1] Olaf Ronneberger and Philipp Fischer and Thomas Brox 2015. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. MICCAI 2015. DOI: <https://arxiv.org/abs/1505.04597>

[2] Hossin, M.1 and Sulaiman, M.N. (2015). “A review on evaluation metrics for data classification evaluations”. International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.5, No.2, March 2015. DOI: 10.5121/ijdkp.2015.5201