

# Patellar Ligament Tear Detection using Machine Learning

Salik Syed  
CS 229A Stanford University  
ssyed@cs.stanford.edu

## Abstract

*In this paper we show a method for identifying ligament tears in MRI scans. We present a new algorithm that builds a spatial pyramid over the tendon invariant to scale and orientation changes of the tendon. We analyze the performance of both LLC encoded texon features and Intensity histograms for building extracting features from the pyramid regions.*

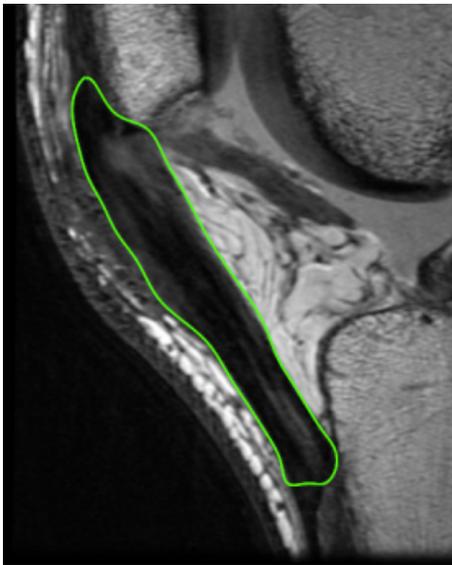


Figure 1. A cropped MRI of a damaged Patellar Tendon (highlighted in Green)

## 1. Introduction

The widespread availability of Magnetic Resonance Imaging (MRI) now creates vast amounts of detailed high resolution imagery of the human body. While Radiologists are quite good at identifying anomalous regions in these scans they still lack quantitative metrics on how to precisely classify the extent of damage. Furthermore radiologists may overlook anomalies that were not expected given

the purpose for the scan (e.g a scan may have been done to identify a broken bone but a cancer may be visible as well).

For these reasons we believe computer vision may be beneficial in assisting doctors with exploring and analyzing MRI datasets in a quantitative fashion and in providing an additional safety net to further reduce the probability of missing potentially harmful anomalies.

The number of different anomalies that can be present in the human body are extremely vast and building a system to handle all of these would require expansive amounts of training data. For this reason we choose to instead focus upon a single type of ligament tear with the idea that our method could later be generalized. For our project we focus specifically upon detecting ruptures in the Patellar tendon - a tendon connecting the patella to the tibia. This tendon allows the knee to flex and extend enabling basic functions such as walking and standing, thus it is critical to treat the tendon early when it is injured. Furthermore damage to the tendon is fairly common giving us a large body of training data.

In this paper we outline a novel algorithm for classifying ligament tear regions inspired by the spatial pyramid approach of Lazebnik et. al. We test our algorithm on a dataset of 43 patients. Each patients MRI scan provides us with anywhere from 7-10 different slices each with varying degrees of patellar tendon rupture labeled by students from Stanford Medical School. Each tear was classified on a level from 1-5. With 1 being normal and healthy and 5 being highly torn. Our algorithm was given the task to classify any tendon with a rating of 1.5 or greater as "Positive".

### 1.1. Prior Work

Diagnostic reasoning with MRI using Computer Vision has shown promise since the mid 1990s. Academic research has mainly focused upon automated mammography and breast cancer detection. Giger et. al presented a framework known as Computer Aided Detection and Diagnosis (CAD). CAD uses the standard computer vision pipeline of segmentation, feature extraction and classification. The CAD system is meant to provide a second-opinion that augments previous diagnosis from a trained Radiologist. Many

feature extraction and learning methodologies have been used including rule-based methods, discriminant analysis, Bayesian methods, artificial neural networks, and/or fuzzy logic. CAD systems have just recently passed through FDA regulation and studies have shown the systems can often detect anywhere from 50 to 90 % of cancers that were missed by Radiologists. While progress has been made in automated diagnosis the vast majority of computer vision research has focused upon the segmentation of MRI slices in various regions of the body (the brain in particular). We believe there is an opportunity to apply state of the art vision techniques to improve classification further than prior art.

## 2. Our Approach

To develop our algorithm we began by meeting with an expert radiologist and learning the process for how patellar tendon tears are diagnosed and analyzed by humans. Radiologists primarily use an indicator known as *signal abnormality* which indicates how bright a region is relative to the baseline signal strength throughout the tendon. Tendon ruptures result in fluid flow that shows up as bright spots.

However there are a variety of other causes for signal abnormality that are not indicative of a tendon tear and are easily discounted by a trained radiologist. One such example is the fact that MRI slices sample a non-discrete slice of the tendon (each pixel in a slice is an average over a volumetric region of the patients leg). This means that if the volume contains tissue from neighboring structures the region will appear to have signal abnormality. This commonly occurs due to the neighboring Hoffas fat pad. To distinguish this from a tendon rupture it is necessary to reason about the high level shape of the tendon and neighboring structures.

Our initial approach was to represent the signal abnormality using a spatial hierarchy of signal abnormality throughout the tendon. By doing this we hope to capture global relationships between different regions of the tendon. To do this the tendon region was first subdivided recursively. For each region we computed a histogram of intensity values within that region then concatenated these histograms to produce the final feature vector. Finally we fed the feature vector into a Random Forest classifier for labelling.

### 2.1. Dataset Preparation

For each patient we received 7-10 slices of MRI data. Each slice represented a different depth relative the side of the leg. For each slice a medical student had roughly outlined the patellar tendon using a green pen. Using the outline we generated a segmentation mask. Once we had isolated the tendon we used our recursive subdivision algorithm (outlined below) to divide the region further.

### 2.2. Region Sub-Division

Traditional spatial pyramid methods use a fixed grid to subdivide regions as shown in Figure 1. We believe this method would not work well for our problem. The primary reason is that there is a large variation in the size, shape and orientation of the tendon from patient to patient (Figure 2). We wanted a method that would be invariant to changes in length, width or rotation of the tendon. Thus we developed a novel algorithm inspired by least squares line fitting.

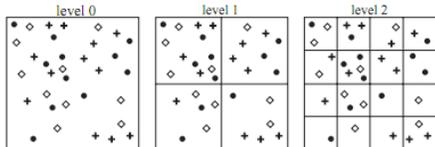


Figure 2. The standard spatial pyramid division scheme. Image features are extracted and pooled over every region for every level then concatenated together to form the final feature vector

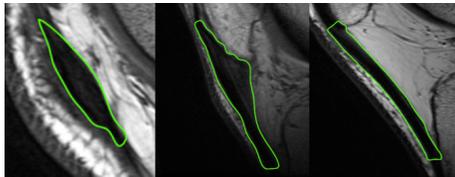


Figure 3. Tendon shape, size and orientation is highly variable based on the patient and depth of the MRI slice

To do the subdivision we treated the pixels contained within the segmented region as a point cloud. We found the best fit line through this point cloud then chose the division line for the region to be the perpendicular bisector of the best fit line:

This process was repeated recursively to generate a spatial pyramid. The primary advantage of this method is that regions in this subdivision correspond to analogous regions relative to the high level structure of the tendon rather than to an arbitrary grid.

### 2.3. Intensity Histogram Features

For our initial attempt we used intensity histograms as our feature vector. For each depth of subdivision we computed an intensity histogram over the resulting region then concatenated the set histograms for all regions. To explain more clearly: suppose we had 3 levels of subdivision. We would compute a histogram for subdivision Level 0 (the entire tendon region). Level 1 subdivision would result in two regions (so a histogram for each half of the tendon). For Level 2 we would further divide each of the halves resulting in 4 total regions. Finally for Level 3 we would divide each of the 4 regions in half resulting in 8 regions. Thus

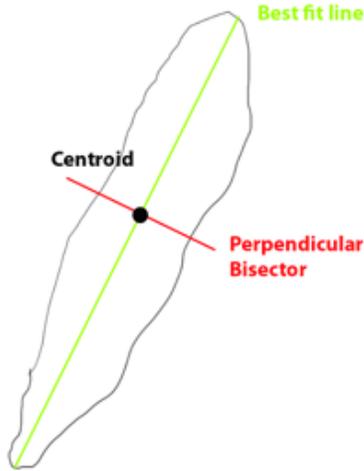


Figure 4. Our method for subdividing the spatial region encompassed by the tendon

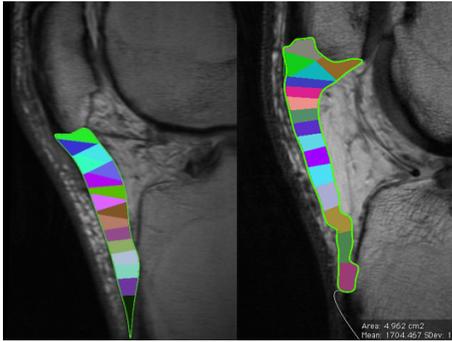


Figure 5. Regions generated at subdivision level 4 on two different patient tendons

our final feature vector would consist of histograms concatenated from  $1+2+4+8 = 15$  regions. For our experiments we tried histograms with 8 and 16 buckets and both 3 and 4 levels of subdivision.

## 2.4. Texton Features

Our initial method using simple intensity histograms gave fairly good results. However we believe that these histograms are discarding many of the nuances of the surface texture of the tendon. To combat this we augmented our approach taking inspiration from Locality-constrained linear coding for image classification

Rather than computing a simple histogram of intensity for a given sub-region we instead compute a representation of each patch within the region using a texton dictionary then pool the results to generate our final feature vector. Our method is broken down into three steps: Dictionary Generation, Feature Coding and Feature Pooling.

## 2.5. Dictionary Generation

To generate our dictionary we used all images in the dataset and extracted 100  $32 \times 32$  pixel patches from each image using random sampling. These patches were normalized then clustered using the K-Means algorithm with  $K=200$ . This resulted in 200  $32 \times 32$  basis patches. We wanted to experiment with other dictionary sizes but the process was too slow for our time constraints.

## 2.6. Feature Coding

Because we did not use a grid for our spatial hierarchy we have non-rectangular sub-regions. Each region was represented using a list of pixel coordinates. We random sampled a set of coordinates within each region proportional to the total number of pixels contained in the region. Each coordinate sampled was then used as the center of a  $32 \times 32$  patch. We used the locality constrained linear coding scheme to sparsely encode this patch to the 200 dimensional basis of our dictionary. To do this we build a KD-Tree containing all of the dictionary (basis) patches. We use the KD-Tree to find the 5 nearest neighbors of the patch to be encoded. We then find the linear combination of these 5 neighbors which results in the lowest total error. This linear combination is used as our feature vector for that patch. This method is known as *Approximate LLC* coding and more details can be found in the appendix.

## 2.7. Feature Pooling

Within each region we have sampled a set of coordinates. Each coordinate corresponds to a 200 dimensional vector. To get the final feature vector corresponding to the region we simply sum all vectors together then normalize (sum pooling). The summed vectors for each region in the spatial hierarchy is concatenated to give us our final feature vector. Thus if we have 15 regions in our spatial hierarchy we will have a  $15 \times 200 = 3000$  dimensional feature vector.

## 2.8. Training

We had skewed classes so we normalized the number of instances by random sampling the class that had a higher number of examples. This gave us a total of 80 training samples for each class. We believe we could achieve better performance with a larger dataset but this was impossible given the difficulty of obtaining patient information so we instead focused upon using learning methods that would be robust even with little training data. In our tests we found that our total accuracy and F-Measure was highest using a Random Forest classifier with 10 trees. We believe this is because the feature space is highly non-linear. We also experimented with using an SVM and both Gaussian and Polynomial Kernels but experienced overfitting even with regularization.

### 3. Results

#### 3.1. Intensity Histograms

We tried using intensity histograms with both 8 and 16 buckets (less than 8 yielded very poor results) and subdivision levels of 3 and 4. Our best accuracy was 74% using 3 levels of subdivision and 8 histogram buckets. Higher dimensionality histograms and subdivision levels result in higher dimensional feature spaces that are much more susceptible to overfitting – which is problematic given we only have 80 examples of each class.

Subdivisions, Histogram Buckets				
	3, 8	3,16	4,8	4,16
Accuracy	74.68%	70.88%	72.15%	69.62%
Precision	75.1 %	71.6%	72.4%	70.3%
Recall	74.7 %	70.9%	72.2%	69.6%
F-Measure	74.6%	70.6%	72.1%	69.4%

#### 3.2. Texton Features

Unfortunately texton based feature extraction did not yield improvement to our results. We used 3 subdivision levels and 200 texton basis patches giving us a very high dimensional feature vector. While we were still able to classify above chance the results were worse than using simple intensity histograms.

Texton Feature Performance (Sum Pooling)	
Accuracy	64.66%
Texton Feature Performance (Max Pooling)	
Accuracy	63.26%

### 4. Conclusion

In this paper we have presented an approach to classifying ligament tears. Ligament tears are fairly subtle and difficult to accurately quantify even by trained radiologists. The problem posed to our algorithm was also more difficult than it may first seem because our threshold for classifying a tear as positive was a tear rating of 1.5 or higher out of a total maximal score of 5 – thus even subtle tears needed to be detected. Our algorithm was able to accurately classify 74% of tears. We believe our technique holds more promise with further training data to better utilize the texton features.

### References

- [1] S.Lazebnik, C.Schmid, and J.Ponce, *Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories*. CVPR 2006
- [2] D. Bodne, SF. Quinn, WT. Murray, T. Bolton *Magnetic resonance images of chronic patellar tendinitis* Skeletal Radiology 1988
- [3] JS. Yu, JE. Popp, CC. Kaeding *Correlation of MR imaging and pathologic findings in athletes undergoing*

*surgery for chronic patellar tendinitis*. American Journal of Radiology Vol. 165 No. 1 1995

- [4] M Giger, H MacMahon *Image processing and computer-aided diagnosis..* Radiologic clinics of North America 1996 (Volume 34)
- [5] J. Wang, J. Yang; K. Yu, F. Lv, T. Huang, Y. Gong; *Locality-constrained Linear Coding for image classification* CVPR 2010
- [6] K. Yu, T. Zhang, Y. Gong *Nonlinear Learning using Local Coordinate Coding*. NIPS 2009

### 5. Appendix

#### 5.1. The Locality Constrained Linear Coding Model

The objective of LLC is to represent a feature as a linear combination of its neighbors. To do this we can create a cost function of the weights  $W$  that we then minimize. Our exposition of the LLC optimization differs from the original paper but we feel our notational convention is slightly clearer. Assume we have  $N$  features to encode and a dictionary  $B$ . Let  $\vec{x}_i$  denote the  $i$ th feature. Let  $\vec{b}_j$  denote the  $j$ th codeword in our dictionary, and let  $w_{ij}$  be the weight given to  $b_j$  for feature  $i$ . For a single feature  $\vec{x}_i$  we have the following cost:

$$c_i = \|\vec{x}_i - \sum_{b_j \in B} w_{ij} \vec{b}_j\| \quad (1)$$

We also have the constraint that  $\sum_{j \in B} w_{ij} = 1$ . To get the total cost  $C$  over all features we then have:

$$C = \sum_{i=1}^N \|\vec{x}_i - \sum_{b_j \in B} w_{ij} \vec{b}_j\| \quad (2)$$

Unfortunately this equation alone does not enforce locality – we need to create a penalty term for using a codeword that is very far from the feature we are coding. To do this we add a spatial cost term  $s_i$ :

$$s_i = \lambda \sum_{b_j \in B} w_{ij} \exp\left(\frac{\|\vec{x}_i - \vec{b}_j\|}{\sigma}\right) \quad (3)$$

Thus we get the following equation for the final cost:

$$\sum_{i=1}^N \|\vec{x}_i - \sum_{b_j \in B} w_{ij} \vec{b}_j\| + \lambda \sum_{b_j \in B} w_{ij} \exp\left(\frac{\|\vec{x}_i - \vec{b}_j\|}{\sigma}\right) \quad (4)$$

By minimizing this cost we get our final set of LLC encoding weights  $W$ .

## 5.2. Analytic Solution to the LLC Optimization

The LLC optimization problem can be solved analytically using a simple matrix formulation. Let  $\mathbf{B}$  be the dictionary matrix whose  $i$ th column is given by  $\vec{b}_i$ . For each feature create the data covariance matrix  $C_i$  as follows (We denote a vector of 1's as  $\mathbf{1}$ ):

$$C_i = (B - \mathbf{1}x_i^\top)(B - \mathbf{1}x_i^\top)^\top \quad (5)$$

(This matrix corresponds to the  $c_i$  term of our optimization). Next we create a matrix  $S_i$  for the spatial cost term. This matrix will be diagonal with each diagonal entry  $d_{ij}$  having the following value:

$$d_{ij} = \lambda \exp\left(\frac{\|\vec{x}_i - \vec{b}_j\|}{\sigma}\right) \quad (6)$$

Now to solve for our set of weights  $\vec{w}_i = \{w_{i1} \dots w_{iB}\}$  we have the following formulation

$$\vec{m}_i = (C_i + S_i)^{-1} \mathbf{1} \quad (7)$$

$$\vec{w}_i = (\mathbf{1}^\top \vec{m}_i)^{-1} \vec{m}_i \quad (8)$$

## 5.3. Approximate LLC

Intuitively most codewords that are very far away from the feature will have a very large cost term and will probably have near zero weights. Thus by including them in our  $B$  matrix we are introducing additional computation for no benefit. To prevent this we can simply discard these codewords and limit to the  $k$ -nearest neighbors. This means we use a much smaller matrix for  $B$  consisting of only the codebook entries that are in the  $K$ -nearest neighbors lowering our optimization time from  $O(|B|^2)$  to  $O(|B| + K^2)$ .