AUTOMATIC KNEE CARTILAGE SEGMENTATION

Chaitra V. Hegde
Center for Data Science
New York University
New York, NY 10011, USA
cvh255@nyu.edu

Aakash R. Kaku
Center for Data Science
New York University
New York, NY 10011, USA
ark576@nyu.edu

Sreyas Mohan
Center for Data Science
New York University
New York, NY 10011, USA
sm7582@nyu.edu

ABSTRACT

The first step in determining how severe the progression of osteoarthritis (OA) requires the segmentation of knee cartilages. Currently, the procedure involves a human expert manually performing the segmentation of a diffusion weighted Magnetic Resonance Imaging (MRI) image of the knee of the individual. The manual segmentation is a painstaking task which usually takes, on an average, about a day for the human expert to complete. We use machine learning and deep learning methods to automate this segmentation task. We benchmark our best performing model against a human expert and conclude that our model is superior for the tissues Patella and Tibia when using dice score as the comparison metric. In the end, we do a perturbation analysis to understand the sensitivity of our model to the different components of our input. We also build confidence maps for the predictions that can help radiologists to tweak the model predictions as required.

1 PROBLEM MOTIVATION

Osteoarthritis (OA) is the most prevalent knee joint disease in the United States which eventually leads to chronic disability(CDC, 2001). OA is characterized in its early stages by degeneration of articular cartilage. The degeneration of articular cartilage is a primary sign of progress of OA in an individual. To assess the integrity of the cartilage, its biochemical properties needs to be studied (Raya et al., 2012; 2013). The biochemical constituents can be measured using MRI. A major limitation for the broad use of these advanced MRI techniques are the lengthy image processing time(Raya et al., 2011). As an initial step towards studying the biochemical properties of the cartilage, the cartilage needs to be segmented, which is usually done by a human expert, a trained Radiologist, and takes this extremely skilled person about a day to complete segmenting the MRI Images from a patient. This way of segmentation is not scalable and extremely slow. We propose to use Machine Learning and Deep Learning techniques to solve this segmentation problem.

2 DATASET

2.1 DATA SOURCE AND DESCRIPTION

With the help of Dr. José Raya from NYU Langone medical center, we procured the dataset of diffusion-weighted MRI of knees of OA patients. The study is funded by NIH. 71 MRI scans/volumes for several patients whose OA severity ranges from mild to severe are included in the data. Each diffusion MRI includes 15 spacial 256 × 256 images with a resolution of $0.6 \times 0.6 \times 3\ mm^3$ covering the entire knee. Each image is obtained 7 times with different diffusion directions and orientations. Each diffusion-weighted dataset is a 256 × 256 × 7 × 15 matrix associated with an individual patient. A musculoskeletal radiologist has segmented all cartilage plates (lateral and medial tibia, femur and patella) in each diffusion-weighted acquisition in the form of a binary mask. These will be considered as the ground truth. A sample image is shown below.

In addition to the 7 contrast (diffusion direction and orientation) images, we have also used additional two maps which were calculated using the 7 contrast images. These maps are referred to as the mean diffusivity maps and fractional anisotropy maps. These maps helps a radiologist distinguish between articular cartilage and fluid. Since fluid and articular cartilage have similar voxel intensity

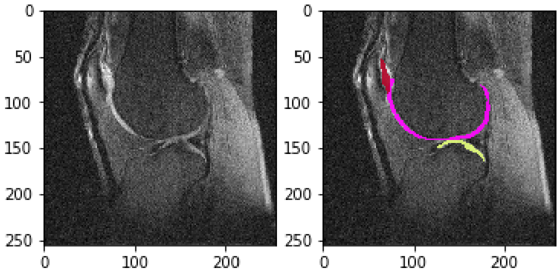

Figure 1: A sample data with the ground truth articular cartilages marked in different color. Red: Patella, Yellow: Tibia, Pink: Femur

Table 1: Distribution of labels for each segment

Segment	Ave. Voxel Count	%
Femur	1083	1.659%
Patella	260	0.397%
Tibia	186	0.284%
None	64007	97.66%
Total	256 × 256	100%

in the images, these images were useful to distinguish fluid from the articular cartilage. So each MRI volume has a size of 256 × 256 × 9 × 15.

Labels: For each MRI volume of 256 × 256 × 9 × 15, we have labels in form of binary mask of size 256 × 256 × 3 × 15. The three channels correspond to the three tissues in the knee. Therefore, for the segmentation task, it can be considered as a 4-class classification problem at voxel level. The 4 class being femur, patella, tibia and none. To understand the sparsity of each class, some descriptive statistics would be helpful. The table below shows the average count of voxels for each segment in a 256 × 256 image (average calculated only for training set images)

As it can be seen from the statistics in the table, the problem is highly imbalanced distribution of labels across different classes. It also sometimes seen that some tissues are not at all present in a particular slice of image. Hence, during predicting the binary mask, the model should not predict any segmentation for such tissues else the dice score would be zero if it does predict even one voxel as tissue. The model should, therefore, be able to generate sparse predictions and grainy or noisy predictions would be severely detrimental for model performance. One way induce such sparsity would be to use a loss function that encourages such sparse prediction. The details of loss function will be discussed in the Loss Function section.

2.2 Train - Validation - Test Split

Among the 71 MRI volumes, there were many follow-up scans. In other words, there could be several volumes for one particular patient but these volumes are obtained over a certain time. Therefore, while splitting the data, it has to be ensured that each patient and all the MRI volumes belonging to him/her should be in one particular set (either it can be training, validation or testing set). The training set contained 57 MRI volumes, while validation and testing set had 7 MRI volumes each.

2.3 Data Preprocessing

Our models used 15 spacial images as a separate 2D image with nine channels. Nine channels include the seven contrast images, one mean diffusivity map and one fractional anisotropy map. Therefore, the 3D MRI volume of 256 × 256 × 9 × 15 were split into 15 2D images of size 256 × 256 × 9.

Preprocessing for 2D MRI image: The values in each channel of the 2D MRI image was normalized to be between 0 and 1 using min-max normalization method. The minimum and maximum voxel intensity was taken across that particular channel of that particular image for which the normalization was supposed to be done.

Converting Image Into Patches: For the purpose of training a classifier which is not convolutional (SVM or Random Forest), we convert one image into several 15×15 patches which are extracted with a stride of one. The exact way in which this patches are used is discussed in the modeling section. For the purpose of training our convolutional neural networks, we extract 50×50 patches with a stride of 15. We note that, conceptually, training on patches is equivalent to training on the whole image due to the spatial invariance underlying convolution. In addition to this, training on patches provide us with the advantage of using a larger batch size as more samples can fit into GPU and also doing much more gradient updates in just one batch as the patches are overlapping.

2.4 Data Augmentation

To compliment the available limited amount of data we have, the training dataset was augmented using random clockwise and anticlockwise rotation and random horizontal and vertical shift of the images. The range of degree of rotation was -5° to 5°. The range of horizontal and vertical shift was -10 to 10 voxels. This augmentation was justified as perturbations of this small magnitude could happen during the procurement of MRI Images because of small movements of the patient. We decided against using the other common augmentation techniques used in natural images, like vertical or horizontal flipping, as such a sample cannot naturally arise in the data generation process. In addition to this, we considered adding noise to our samples as another way of data augmentation and regularization. However, we decided against it considering the fact the images we currently have already has a substantial amount of noise.

3 Model Building Process

3.1 Assumptions Made

For the machine learning model, we make a naive assumption that each patch is independent of the other. We are aware that this assumption is not justified, but we couldn't find a better way to deal with it.
For our deep networks, we assume that each spatial slice is independent of other, or in other words, the location of the slice in the 3-D cube doesn't matter. While it intuitively looks like this might not be a valid assumption, we evaluated it by building a model which took this 3-D structure into account, but the performance remained the same.

3.2 Loss Function

We model the segmentation task as a multi-class classification problem. Here, since we have 3 tissues of interest, this is a 4-class classification problem, where the last class is background.

Since we know that cross entropy would just give the maximum likelihood estimate over our dataset, and since our dataset is highly imbalanced, we use a weighted cross entropy loss for our task. The weighted cross entropy loss could be defined as:

$$\text{Weighted-CEL} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{4} w_j(y_{ij}log(p_{ij})) \tag{1}$$

Where w_j = Weight of the jth segment. The weight can be calculated using many approaches that are based on the voxel count for that particular image/volume. Here, w_j = $\frac{N-\sum_{i=1}^{N} p_{ij}}{\sum_{i=1}^{N} p_{ij}}$, N = number of voxels in the MRI volume or 2D MRI image, p_{ij} = probability of voxel i to be belonging to segment j, y_{ij} = label of voxel i to be belonging to segment j = 1 or 0. (Sudre et al., 2017))

Even though we use weighted-CEL as the loss function, the accepted metric for measuring the quality of segmentation is dice score (discussed below). In our experiments, we find that the segmentations that come out of weighted cross entropy is very noisy. We propose to directly optimize the dice loss which is defined below. Weighted Dice Loss: The weighted dice score is defined as:

$$\text{Weighted-Dice Loss} = 1 - 2\frac{\sum_{j=1}^{4} w_j \sum_{i=1}^{N} y_{ij} p_{ij}}{\sum_{j=1}^{4} w_j \sum_{i=1}^{N} y_{ij} + p_{ij}} \tag{2}$$

Weights are calculated using the same approach as mentioned for weighted-CEL.

We experimented with local(using only a specific batch) and global (using the entire train set) estimation of w_j. In the local case, w_js were adapted for each batch, and hence the loss function for each batch was slightly different. However, we found that a global w_j gave us the best out of sample performance.

3.3 Evaluation Metric

Dice score (DSC) as a primary measures of quality of the segmented images, since they provide direct information on how well the images are segmented.

$$\text{DSC} = \frac{2||PT||_2^2}{||P||_2^2 + ||T||_2^2} \tag{3}$$

where P = Predicted Binary Mask, T = True Binary Mask, PT = element-wise product of P and T, $||X||_2$ is the L-2 norm. Dice score can equivalently be interpreted as the ratio of the cardinality of $(T \cap P)$ with the cardinality of ($(T \cup P)$+$(T \cap P)$).

DSC ensures that the predicted segments are as close to ground truth. From the above metric, it can be seen that it penalizes over prediction and under prediction. Hence, this metric tends to be well-suited for segmentation task, particularly in the field of medicine.

3.4 Machine Learning Models

We tried to solve the problem of segmentation using traditional machine learning models. The ideal way would be to extract features/values from the image and represent it as one dimensional vector of values and then train SVM model with RBF kernel and try to predict the center pixel as belonging to either of the 4 classes. RBF kernel is better when compared to linear or polynomial because its very flexible. But most of the available implementations of SVM are suitable for dataset within 10,000 data points. Hence, we had to settle with non-svm models. Best non-linear model other than svm is Random Forrest as it is non linear and is ensemble of number of models.

During first phase, as mention earlier we extract patches of size 15 × 15 with stride of 1 and feed it to Random Forrest model. But it was evident that the model wasn't learning properly. So, we decided that the position from where the patch was taken might be informative but the model was highly overfitting on this location variable. So, with machine learning models, we barely got the dice score of 0.4.

During second phase, we decided to use computer vision algorithm, Histogram Oriented Gradients (HOG), to extract the features of these 15 × 15 patches rather than directly using the values of the voxels itself. Each of the 15 × 15 image generated 256 dimensional feature vector (variable) and random forest and SVMs were trained on these features. But it turns out that, this was performing worse than the model using raw pixel values. The reason could be that HOG was designed to get gradients of natural images in different directions(hence, more general features of image) and natural images have greater gradient than the gradient for MRIs and hence, HOG couldn't capture the specific features of MRI.

Hence, the conclusion is that the machine learning model is not best fit for this task because of the assumption that all the patches are independent. It fails to capture the spatial relations within image. Rather, convolutions are best way to deal with image dataset.

3.5 Deep Learning Models

3.5.1 Deep CNN Models

We trained a deep convolutional neural network which consisted of $3x3$ convolution kernels, followed by a relu and batch norm which was repeated 17 times. This model wasn't performing very well and this lead us to believe that a multi resolution approach is necessary. A recent approach towards building deep learning models for segmentation is the fully convolution encoder-decoder approach (Ronneberger et al., 2015; Lin et al., 2016). The encoder encodes the high-resolution input to a low-resolution output and a decoder decodes this low-resolution output back to high-resolution output. The up-sampling and down-sampling is done using max-pooling or strided convolutions (or strided transposed Convolutions). The low-resolution output efficiently captures the high level semantics (object/class specific features) whereas the high-resolution input captures the low level semantics (like edges, corners, spatial information etc.). Therefore, while reconstructing the segmentation maps, the low level semantics which gives spatial information and the high level semantics which recognizes the object class are important. We, therefore, will use the hyper-columns (skip connections) that helps us to directly connect high-resolution inputs with high-resolution decoded outputs. This is a basic structure of many state-of-the-art segmentation models like U-Net and RefineNet (Ronneberger et al., 2015; Lin et al., 2016).

We try a basic U-Net type of architecture and many improvements over it are tried to get better dice scores. The details of different architectures are described below:

3.5.2 U-Net

A Baseline U-Net model was built for 2D images. The architecture can be seen in the figure 2.

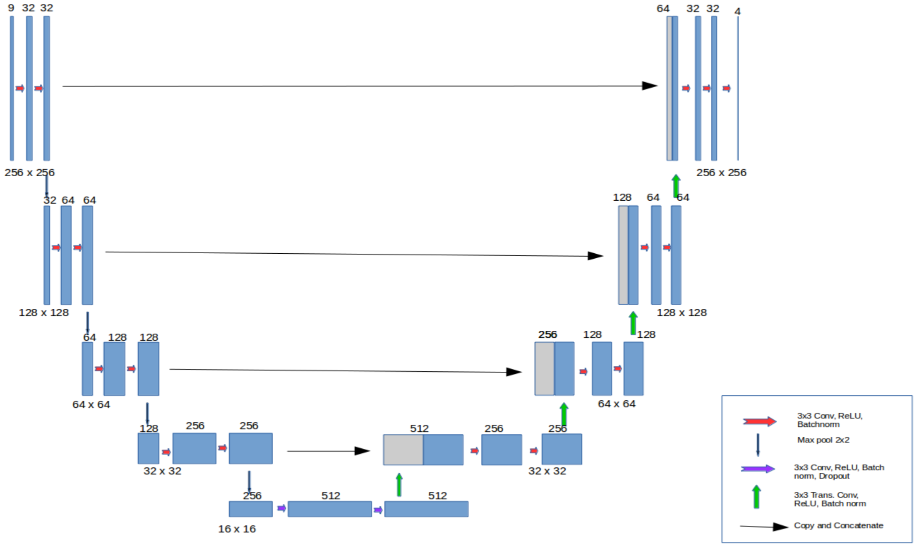

Figure 2: Baseline U-Net for 2D Images

The baseline U-Net model for 2D images had 9.8 million learn-able parameters.

The prediction was done by taking a argmax on the probability maps across 4 segments. In other words, the probability for each voxel belonging to 4 classes is calculated and voxel is assigned to class with highest probability.

The details of the training procedure and performance of both the models are discussed in the experiment and results section.

3.5.3 Dilated U-Net

The baeline 2D U-Net was modified to obtain better performance. Atrous or dilated convolutions were introduced in the architecture because it has shown that dilated convolutions can help the model to have larger field of view and helps the model to have multi-resolution properties (Chen et al., 2017). Intuitively, if the model is able to derive features from multiple resolutions and learns to combine them, then the predictions would be much more accurate. The architecture of the dilated

U-Net, therefore, encourages similar properties that allows the model to extract features at multiple resolution and learns to combine them to give better performance.

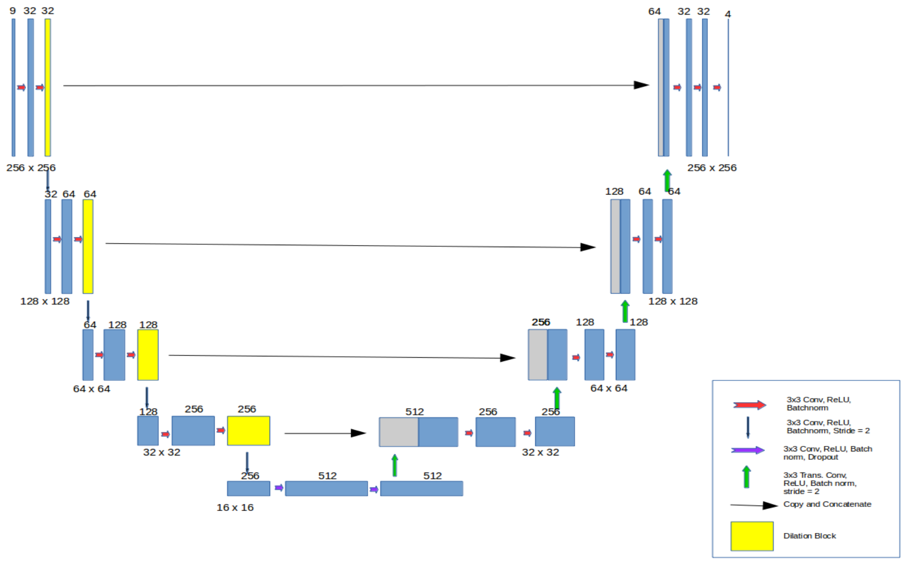

Figure 3: Dilated U-Net for 2D Images

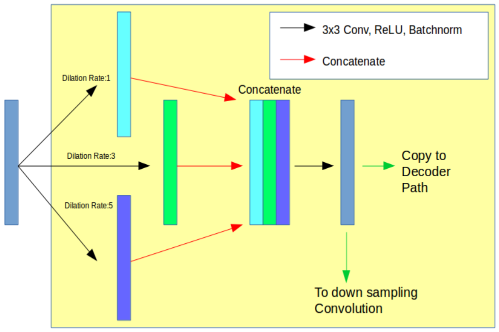

Figure 4: Dilation Block

The complete Dilated U-Net has 48 million learn-able parameters. This could be potentially lead to over-fitting. Hence, the paper also tries smaller dilated U-Net and remove the unnecessary parameters and make the models leaner. This would be discussed in smaller U-Net and Dilated U-Net.

The model performance and training details are discussed in the Experiments and Results Section.

3.5.4 Reducing the Network Size as a form of Regularization

With the models discussed above, we had reached a point where the performance has saturated and we were not able to improve it any further. We did some analysis on our network and realized that we might be overfitting. Also, given than we had about 9 million parameters, this was enough to memorize all the training data we had. We reduce the size of the network in order to limit its capacity.

3.5.5 Smaller U-Net and dilated U-Net

The smaller dilated U-Net has architecture same as dilated U-Net but all the channels equal to 32 channels. The smaller U-Net has same architecture as the base line U-Net but with only 40 channels for each intermediate convolution. The schematic diagram demonstrates the smaller architectures.

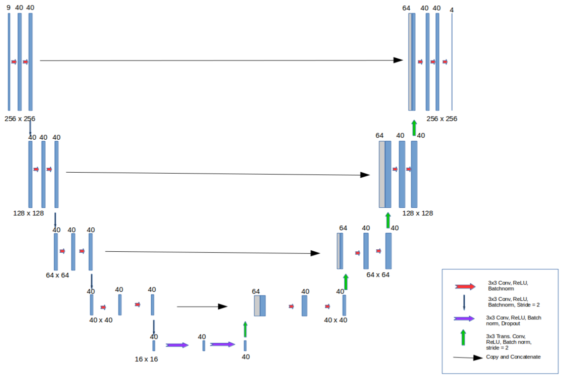

Figure 5: Smaller U-Net

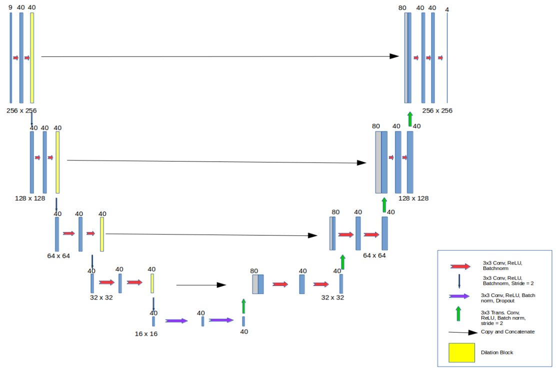

Figure 6: Smaller Dilated U-Net

The smaller dilated U-Net has 367K parameters and smaller U-Net has 410K parameters. The performance of both the models are discussed in the Experiment and Result section.

3.5.6 Ensemble of models

The two best models were ensembled to obtain a superior model. Instead of having a simple weighted linear combination of predictions, the ensembling was done using a simple convolution net with input being the prediction of the individual models and output being the final prediction. The intermediate layers involve three successive convolution followed by ReLU and Batchnorm. All throughout the network, the resolution is maintained at 256 $\times$ 256 and filter size is 3 $\times$ 3. The performance and training details are mentioned in the Experiment and Result section.

4 Experiments and Results

4.1 Training Procedure for all the model training

All the models were trained using Adam optimizer (Kingma & Ba, 2014) with learning rate in the range of 1×10^{-4} to 1×10^{-5}. Validation score was monitored and the step size was adjusted accordingly. If the validation loss stopped decreasing, the learning rate was reduced to half or

Table 2: Performance of the models

Model name	Dice Score (Femur)	Dice Score (Patella)	Dice Score (Tibia)
Baseline U-Net	0.671	0.745	0.573
Dilated U-Net	0.681	0.764	0.580
Dilated U-Net L2 Regularized	0.683	0.751	0.552
Small U-Net	0.678	0.773	0.593
Small Dilated U-Net	0.670	0.771	0.621
Ensemble of Small U-Net and Small Dilated U-Net	**0.689**	**0.783**	**0.640**
Human Expert (Re-segmentation)	0.711	0.743	0.629

one fifth depending on the model's learning progress. The best model was selected based on the lowest validation loss. Hence, an implicit early stopping was done to ensure the model with best performance on the out-of-box sample was chosen. While training the model we experimented with different L2 regularization parameter and drop-out rates. In the performance section, only the best performing model's dice scores are reported.

4.1.1 Performance of all the models

The table below give performance of all the models on the validation set.

A line for Human expert can be seen the performance table. The human expert corresponds to a musculoskeletal radiologist. An interesting experiment was performed where the same radiologist who segmented the original MRI volumes was asked to re-segment them after some time (like 6 months). The difference between the original MRI volumes and the re-segmented MRI volumes was captured using the dice score which can be seen in the table. The discussion regarding the performance of all the models and comparison of the best model with human expert performance is done in the Discussion Section.

The best model (ensemble model) was tested on the test dataset. It yielded dice scores of 0.6902, 0.7789, and 0.6814 for Femur, Patella and Tibia respectively.

5 Discussion

5.1 Visualizing the predicted segmentation and ground truth

Below we visualize some sample predictions on the validation set images:

1. Case where the model correctly predicted ground truth:
2. Case where the model correctly predicted a segment not present in the ground truth:

   There were quite a few cases where the human expert missed the segmentation and the model correctly picked it up. Hence, this model has a potential use as the radiologist can run the MRI volumes through the model and get a first cut of the segmentations which they can tweak a bit.
3. Case where the model was more conservative than the human expert:
4. Case where the human expert was more conservative than the model:

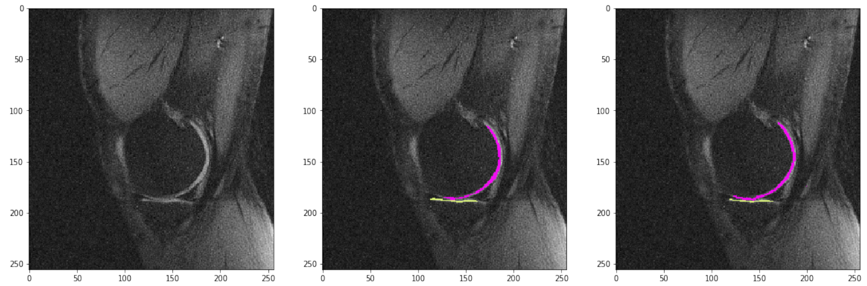

Figure 7: Case where the model correctly predicted ground truth. Left: Original Image, Center: Ground Truth, Right: Model Prediction. Femur = Pink, Patella = Red, Tibia = Yellow

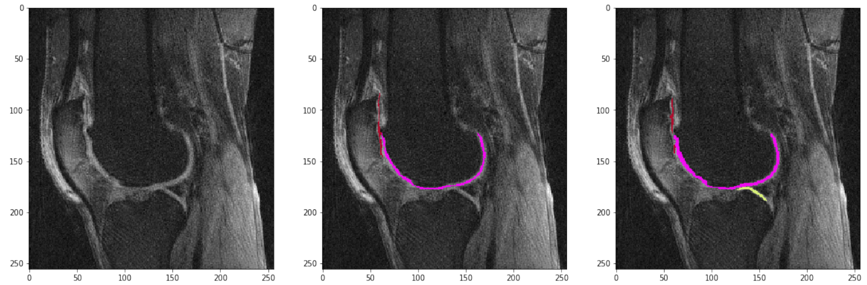

Figure 8: Case where the model correctly predicted a segment not present in the ground truth. Left: Original Image, Center: Ground Truth, Right: Model Prediction. Femur = Pink, Patella = Red, Tibia = Yellow

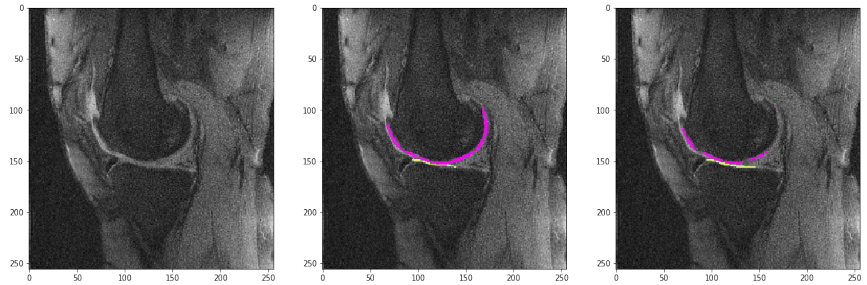

Figure 9: Case where the model was more conservative than the human expert. Left: Original Image, Center: Ground Truth, Right: Model Prediction. Femur = Pink, Patella = Red, Tibia = Yellow

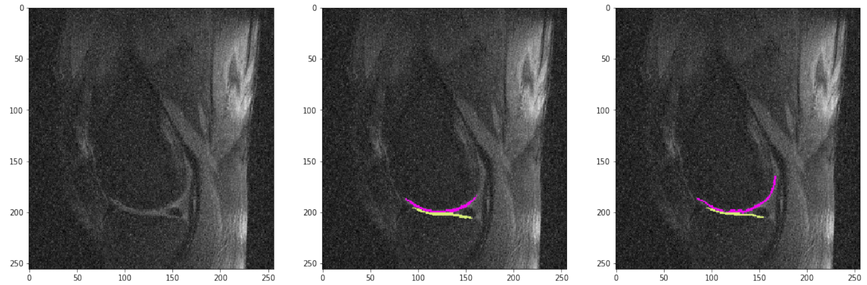

Figure 10: Case where the human expert was more conservative than the model. Left: Original Image, Center: Ground Truth, Right: Model Prediction. Femur = Pink, Patella = Red, Tibia = Yellow

5.2 Analyzing Sensitivity of Trained Model to Input

Our input consists of 9 channels out of which 7 channels are the segmentation of the knee captured at different intensity levels and 2 channels are some maps which are actually calculated using the these original 7 channels. Since, in theory, these 2 maps could have been recreated from the original 7 channels, we decided to analyze the sensitivity of each of the input channel to the output. The metric we look at is the average dice score over the entire validation dataset.

5.2.1 Selectively setting each channel to zero

We set each channel to zero, one at a time. We find that dice score drops drastically irrespective of which channel we dropped. We were expecting to see a high drop in dice score when we drop any of the 2 maps which had included, but since each of the 7 intensities encoded very similar information, this result was very unexpected.

5.3 Permuting the input channels

The only possible explanation we could come up with to explain the phenomena we saw in the section above was that more or less, all these 7 channels should have equal weight. To test this, we permute the order of these 7 intensity channels we give as input. We observe only a small decrease in dice score. We again permute the order of the two maps or one of the maps in an intensity channel and we see a large decrease in intensity. We think this phenomena is very similar to what would happen to a linear model with L2 regularization when you have very highly correlated features.

5.4 Comparison between the model and the human expert

5.4.1 Why human re-segmentation dice scores are not close to one?

The segmentation of tissue from the knee MRI volumes involves a lot of subjectivity. Around the borders of the tissue, it becomes extremely subjective since the voxels are blurry and hence a conservative radiologist might not include those as the part of tissue whereas a less conservative one might include those questionable voxels. Since the overall size of the tissue is small, inclusion or exclusion of the borderline voxels can lead to significant change in the tissue size and in turn, can result in significantly different dice scores.

One way to avoid such subjectivity is to use an MRI volume of high resolution. But a high resolution MRI requires a lot more human effort to segment. Hence, there is a trade-off between having more noisy ground truth versus having few clean ground truth MRI volumes.

We visualize the distribution of the dice score of the resegmented samples by the human expert and the dice score achieved by our model. We see that, the distribution of scores for our model is peaked more towards right, consistently for all tissues. We can also draw from this that, given the limitations of the noisy human generated we have during the training process, we might have reached a saturation point.

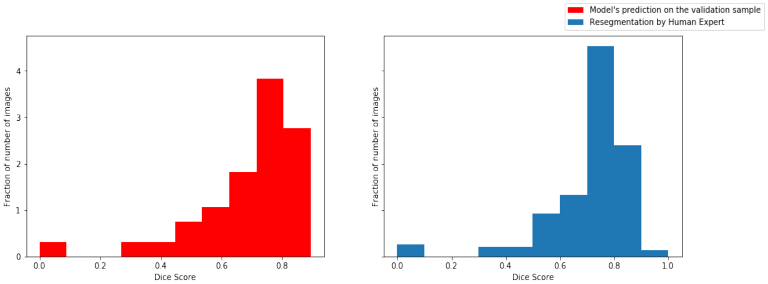

Figure 11: A histogram showing the distribution of dice scores for human re-segmented images and model's prediction of the validation images (Femur Segment)

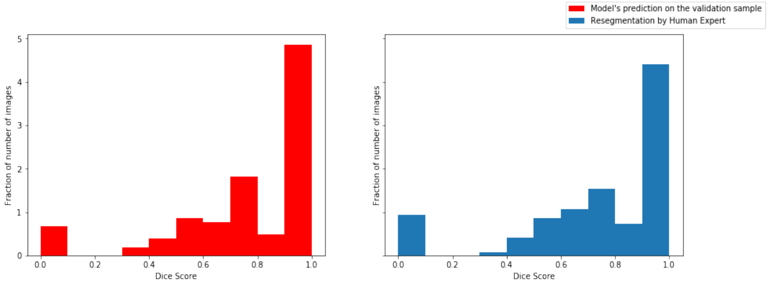

Figure 12: A histogram showing the distribution of dice scores for human re-segmented images and model's prediction of the validation images (Patella Segment)

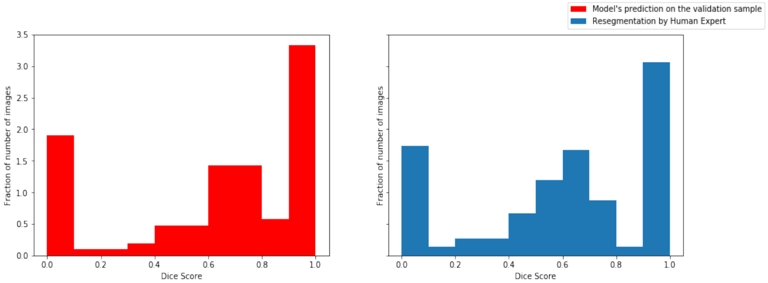

Figure 13: A histogram showing the distribution of dice scores for human re-segmented images and model's prediction of the validation images (Tibia Segment)

5.4.2 Noisy Ground Truth

There are few ways to deal with noisy labels. One way is to way is to acquire data that is segmented by multiple radiologists so that individual biases and noises would cancel out each other and the ground truth would be less noisy. Similar practices are followed when a more standard dataset like PASCAL VOC (Everingham et al., 2015) or The Berkeley Segmentation Dataset are generated (Martin et al., 2001). Another way is to train the model by introducing noise in the existing ground truth labels. This is also termed as label smoothing and frequently done in the context of classification problem but never tried for segmentation task. We tried to perform this analysis but it apparently requires a lot of fine tuning and we were not able to complete the implementation. This is something we would like to focus as a future work on this project.

6 Quantifying Confidence of Prediction

Since the entire process of segmenting involves some subjectivity, it would be a good idea to incorporate certainty metric or confidence metric for each prediction. This would help radiologist to understand which predictions they have might have to tweak a bit and which ones they might use it as it.

We incorporated the certainty metric in the form of log of odds ratio. The output scores of the network were normalized using softmax function to give probability of each voxel belonging to one of the four classes. Hence, there are 4 probability values associated with each voxel which adds up to one. Based on these four values, the maximum probability value is taken as the predicted probability and an odd ratio is calculated using this maximum probability value p_{max}.

Therefore, for each voxel, $log(\frac{p_{max}}{1-p_{max}})$ is the level of confidence. Higher the value, higher the confidence of the prediction.

As seen from the experiments, this metric actually gives low values to incorrectly predicted voxels. Below is a sample image, where the stray voxels (which are incorrectly classified) have low values of this confidence metric. Therefore, such certainty maps can be really helpful for the radiologist to fine-tune the predictions as the predictions can at times have stray incorrect voxels.

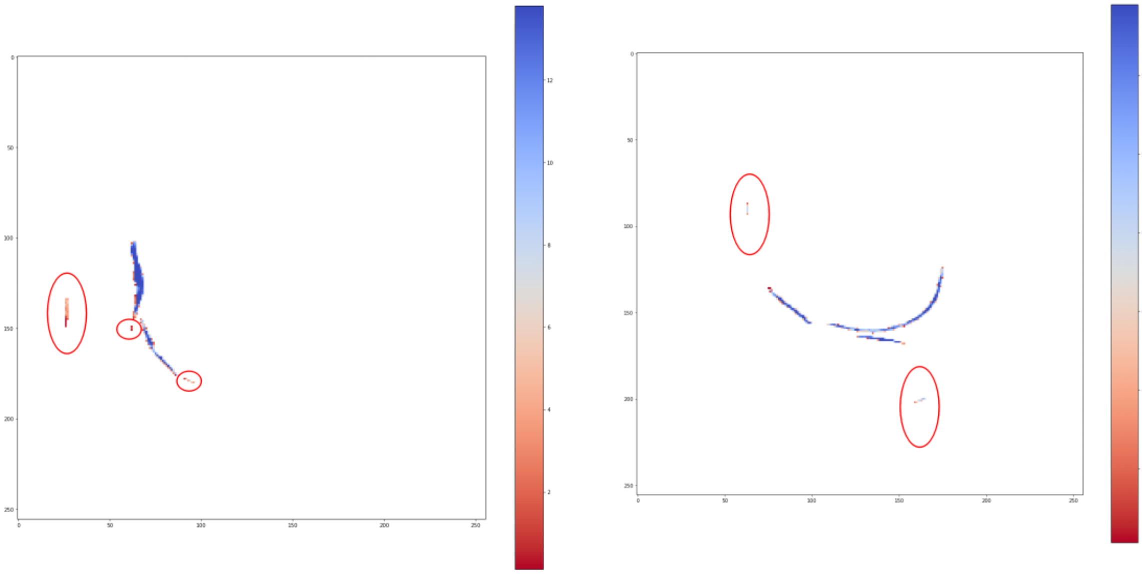

Figure 14: A confidence map for two example validation images can be seen. The circled stray voxels are incorrectly classified as one of the cartilage. It can be seen that incorrect voxels have low confidence

7 Future work

As discussed above, if the dataset is acquired from multiple radiologist the problem of noisy ground truth can be resolved. It is an expensive way to solve the problem of noisy ground truth but it would be the most robust way to tackle it. Another approach can be to train model using label smoothing. As a future work, we would like to optimally train the model using label smoothing and also simultaneously gather data from multiple radiologists.

8 Conclusion

In this paper, we tried to automate the knee cartilage segmentation task using the deep learning models. There was a severe issue of class imbalance which was tackled using weighted dice loss. A novel architecture that makes use of the dilated convolutions and U-Net architecture was found to be more informative as it extracted features from different resolution. We also successfully stripped the unnecessary parameters and made the models leaner and efficient. Finally, to construct a prediction model, an ensemble of best models was used and which performed at an human expert level for few of classes of segmentations.

Code

All the code can be found at `https://github.com/aakashrkaku/knee-cartilage-segmentation`. The executed notebooks, namely `Implementation_3d.ipynb` and `Traditonal_ML_models.ipynb`, and all the ipython notebooks in the `Experimentations` folder is where all the experiments and its results can be seen readily. Since the data is the proprietary NYU Langone Medical School data, it would be difficult to share the data. Hence, if the code is tried to be executed, it may fail. Therefore, pre-executed and extensively commented ipython notebooks are shared so that it would become convenient for the reader to read the code and understand the motive behind the steps of code.

Acknowledgments

This project would not have been possible without the data which was provided by Dr. José Raya from NYU Langone medical center. All MRI volumes was painstaking labeled by Dr. Alejandra Duarte. Without their help and guidance, this project would not have been possible. We would also like to thank our advisor, Dr. Bonnie Ray, for constantly helping and guiding us along the entire project.

References

CDC. Prevalence of disabilities and associated health conditions among adults–united states 1999. *Centers for Disease Control and Prevention (CDC)*, 50:149, 2001.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017. URL `http://arxiv.org/abs/1706.05587`.

M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL `http://arxiv.org/abs/1412.6980`.

Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. *CoRR*, abs/1611.06612, 2016. URL `http://arxiv.org/abs/1611.06612`.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pp. 416–423, July 2001.

JG. Raya, Gerd Melkus, Silvia Adam-Neumair, Olaf Dietrich, Elisabeth Mtzel, Bart Kahr, Maximilian F. Reiser, Peter M. Jakob, Reinhard Putz, Christian Glaser, and et al. Change of diffusion tensor imaging parameters in articular cartilage with progressive proteoglycan extraction. *Investigative Radiology*, 46(6):401409, 2011. doi: 10.1097/rli.0b013e3182145aa8.

JG. Raya, Annie Horng, Olaf Dietrich, Svetlana Krasnokutsky, Luis S. Beltran, Pippa Storey, Maximilian F. Reiser, Michael P. Recht, Daniel K. Sodickson, Christian Glaser, and et al. Articular cartilage: In vivo diffusion-tensor imaging. *Radiology*, 262(2):550559, 2012. doi: 10.1148/radiol.11110821.

JG. Raya, Gerd Melkus, Silvia Adam-Neumair, Olaf Dietrich, Elisabeth Mtzel, Maximilian F. Reiser, Reinhard Putz, Thorsten Kirsch, Peter M. Jakob, Christian Glaser, and et al. Diffusion-tensor imaging of human articular cartilage specimens with early signs of cartilage damage. *Radiology*, 266(3):831841, 2013. doi: 10.1148/radiol.12120954.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL `http://arxiv.org/abs/1505.04597`.

Carole H. Sudre, Wenqi Li, Tom Vercauteren, Sébastien Ourselin, and M. Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. *CoRR*, abs/1707.03237, 2017. URL `http://arxiv.org/abs/1707.03237`.