Efficient Image Denoising and Semi-Automated Detection Of ACL Injury From MRI Scans

[1]Shubham Patil, [2]Sourish Vaghulade
[1](Electronics Engineering Department, Veermata Jijabai Technological Institute, Mumbai, India)
[2](Computer Science Department,Thadomal Shahani College of Engineering,Mumbai, India)

Abstract : Radiologists have the troublesome job of diagnosing various injuries through MRI and CT scans. This job is subject to errors which can prove costly to a patient's health as well as the radiologist's reputation. This is aggravated by the fact that often times these MRIs are noisy and distorted. An aid in the form of a machine for this job will definitely be beneficial. This paper deals with denoising such images and detecting the injury grade related to Anterior Cruciate Ligament (ACL). This paper tries to examine the possibility of predicting diagnosis based on the detection of the ACL after clearing the initial scan. This process is begun by filtering out the noise from the input images using a DnCnn followed by various detection models such as HOG + SVM, GIST + SVM, HOG + RBF K-Means, GIST + RBF K-Means, HOG + SOM, HOG + Random Forest, GIST + Random Forest, and Autoencoder. These approaches are compared and a single approach is picked out which is the most efficient one in terms of accuracy. Clever machines will make workers more productive more often than they will replace them.

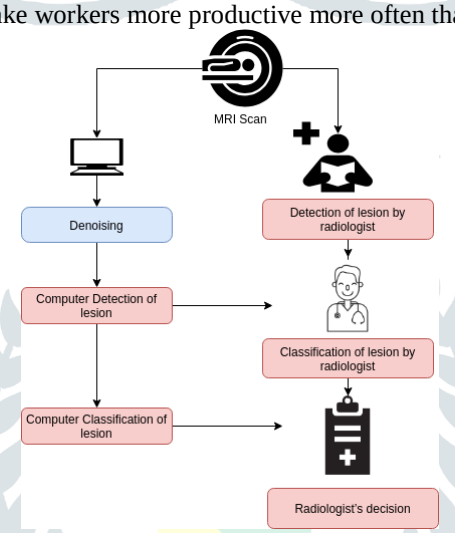


Fig 1 : Workflow of the proposed system

IndexTerms **- Magnetic resonance imaging (MRI),Ligament, Computer-aided detection and diagnosis, Machine learning, Pattern recognition and classification, Feature extraction, Denoising Image.**

I. INTRODUCTION

Anterior cruciate ligament (ACL is found inside the knee joint which is formed by the thigh bone (femur), shinbone (tibia), and kneecap (patella). Cruciate ligaments control the back and forth motion of your knee. The ACL runs diagonally in the middle of the knee. ACL injuries are most frequent in people who play sports which involve a lot of running. These injuries are diagnosed by looking at the MRI or CT scans by the radiologist. There are 3 grades of ACL injuries i.e a mild sprain, a partial tear and a complete tear ref image 1, the latter being the most common. This paper aims to distinguish between a torn and a healthy ACL.

Fig 2: Example of a grade 3 tear of an ACL

As indicated, such a detection requires unwavering attention which is sabotaged by the fact that humans are error-prone. This is common for any medical imaging but especially detrimental for observations that are minute but paramount in diagnosis. These observations could be hindered at times due to blurry and noisy images. Computer-based diagnosis will mitigate these errors even

if the images are unclear[18]. Common issues encountered in such diagnosis will be segmentation of image finding out the exact region of interest, detecting the presence of lesions in that organ. One of the important issues also includes the forming of the independent test case will also turn into a training set if the algorithm is optimized on the test set. Although there is abundant scientific literature available regarding the ACL injury none of the reported work is in the automated detection of this injury. The main problem faced by all the researcher is the data which is required to perform this kind of research needless to say that includes images with a good resolution of both healthy and damaged cases. In this paper, a model is built to initially deblur the corrupted ACL MRI scans accompanied by a predictive model capable of automatically performing the detection of the ACL by pointing out the region of interest i.e the ACL to the radiologist or user. The first model is important because unfortunately, these medical images are affected by a various number of noises such as Poisson, Rician, Gaussian and impulse noise (salt and pepper noise). This creates problems when detecting the lesions from that image. The purpose was to remove this noise and then detect the type of injury whether it is partial or a complete tear of ACL.

II. MATERIALS AND METHODS

2.1 Data

A total of 969 knees sagittal plane DICOM MR volumes in 12-bit grayscale were acquired from Clinical Hospital Centre Rijeka picture archiving and communication system (PACS), along with their respective assigned diagnoses. At the Hospital, a patient's cruciate ligaments conditions are often established by observing only the sagittal plane volumes in fat suppression technique, because they provide details concerning this problem.[1]

Dataset is filled with cases where the ACL: 1) is of physiological shape, but its entire region is saturated with higher pixel intensities; 2) looks dwindled at one point the position of which may be either proximal or distal, exhibiting a wide range of higher pixel intensity region close to it; 3) exhibits a thinning at any of its smaller portions, usually represented by a lack of a low-intensity region, and; 4) is of physiological shape, but is not followed as a completely straight line – rather it looks a bit curved. These visual characteristics are often attributed to a strained or oedematous ligament, meaning partially injured.

Injured ligaments are regarded as "sprains" and the injury is graded on a severity scale.

Grade 1: In this grade, the ligament is just mildly damaged. It has been slightly stretched but is capable of keeping the joint stable

Grade 2: In a grade 2 sprain the ligament is stretched to an extent that it becomes loose and this case is often regarded as the partial tear of the ligament.

Grade 3: In a grade 3 sprain the ligament is stretched so badly that the ligament is split into pieces and the knee joint becomes unstable. This is also known as complete ACL tear injury or ACL is ruptured.0

Sometimes there can be incidences where the radiologist can ignore digital evidence in establishing a diagnosis and stating that ligament is completely healthy[2]. The reason for this discrepancy is very simple because when interpreting the MRI scans radiologist often take into consideration the additional information given by the patient which add a bias to a resultant diagnosis. For example, if a patient came to an emergency room from a football field, then the radiologist would probably be biased towards concluding that the ligament injury is evident, even if the evidence is otherwise inconclusive. Finally, there also exists a possibility that a radiologist can make an error diagnosing an injury.

2.2 Denoising using DnCNN

In [5], Jain and Seung stated that Convolutional Neural Networks (CNNs) are equal or better than Markov Random Field (MRF) models in terms of representation power and also suggested the use of CNNs for image denoising purpose. In [6] and [7], Multi-layer Perceptron was successfully implemented for denoising images and stacked or sandwiched sparse autoencoder methods for denoising purposes were chosen to manage the Gaussian noise removal respectively. The latter type of methods was able to attain commensurate results to K-SVD [3]. In [4], a TNRD i.e a Trainable Nonlinear Reaction Diffusion model, which can be demonstrated in form of a feed-forward Deep Neural Network by flattening out the static number of gradient descent steps, was propounded.

DnCNN is a magnificent research that trains residual noise rather than image mapping parameters[8]. DnCNN model consists of 2 main features - batch normalization, to inculcate faster training and to improve the performance of denoising, and residual learning process, which is incorporated to learn R (y). Like the iterative noise removal strategy used in methods like EPLL and WNNM, DnCNN gradually discretizes the structure of image from the noisy observation by adopting convolution along with ReLU in the hidden layers. It has the exception of an end-to-end training method. Later, the grounds for merging batch normalization and residual learning are discussed.

Batch Normalization adds to the benefit of residual learning. This is straightforward because batch normalization offers some merits for CNNs, such as alleviating internal covariate shift problem

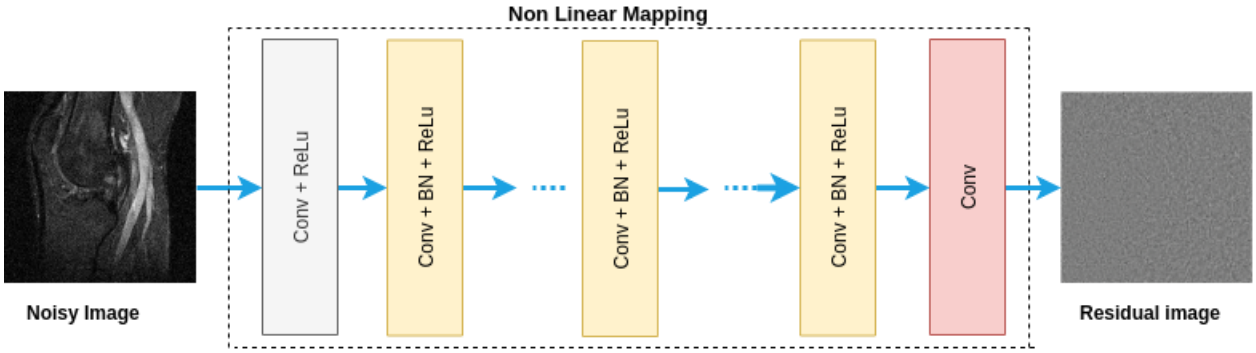


Fig 3: Architecture of Denoising Convolutional Neural Network

2.2.1 Residual Learning

In modern times, Deep Convolutional Neural Networks have shown high performance during image classification. Due to the notion that number of layers is directly proportional to the performance of a CNN, primarily due to experiments, led to the idea of creating deeper networks. A degradation problem was uncovered when convergence occurred in deeper networks which resulted in saturating accuracy followed by a rapid drop upon deepening the network. This was attempted to be fixed by Microsoft in 2015 through their introduction of Deep Residual Learning for Image Recognition. To fit a second underlying mapping $F(x)=H(x)-x$ there is an option of fitting the desired underlying mapping $H(x)$ by a few stacked nonlinear layers. Hence the function can be adjusted to $H(x)=F(x)+x$ where $F(x)$ is the residual function and x is the input. Instead of aiming to fit a stack of layers to a desired underlying mapping function, these layers are directly fitted to a residual mapping. Hence these layers are used to approximate the residual function $F(x)$ instead of fitting $H(x)$. Both forms are capable of fitting the underlying mapping.

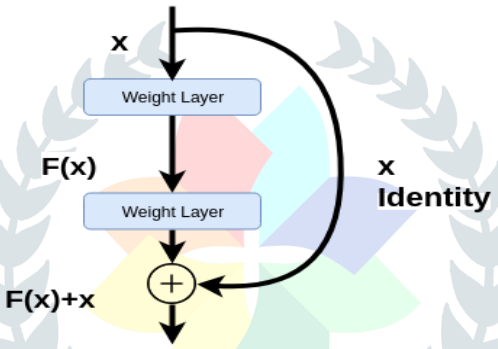


Fig 4: Mapping used by Deep Residual Learning

2.2.2 Batch Normalization

Batch normalization is used to normalize the activations of a given input data before passing it to the next layer in the network. Batch normalization has demonstrated to be extremely effective at reducing the number of epochs it takes to train a neural network. Batch normalization has the advantage of allowing a large variety of learning rates and regularization strengths thus stabilizing training[25]. Batch normalization regularizes output provided by previous layer by subtraction of batch mean and further dividing by batch standard deviation to improve stability of the network. This output is multiplied at each layer by a standard deviation parameter and summed with a mean parameter which are also called gamma and mean parameters respectively. Batch normalization lets Stochastic Gradient Descent (SGD) alter only these 2 trainable parameters instead of all the weights to avoid losing stability for each activation since SGD is responsible for minimizing the loss function which could undo normalization. Batch normalization permits each network layer to learn independently of other layers as well as reduces overfitting due to regularization much like dropout. This will reduce the amount of dropout that is to be used which is a good thing since the network will be less lossy[25]. But it is still advised to use batch normalization along with dropout

This is where residual learning is used for discretizing noise from noisy observation. The combination of batch normalization and residual learning will result in a swift training process along with a significant improvement in denoising performance. DnCNN model has the advantage of having the ability to handle blind Gaussian denoising with unknown noise level over other traditional classification models which train certain models for certain conclusive noise levels[8].

Ground Truth Noisy Image Denoised Image

Fig 5: a) Ideal output that should be obtained b) Image that was feeded to the system along with self-added noise c) Output given by the system

2.3 Feature Extraction Techniques

2.3.1 Hog

HOGs i.e Histogram of Oriented Gradients is feature descriptors used for object detection using edge and color information. It represents objects as single feature vectors and is often used with SVM classifiers like the model presented. A sliding window detector is passed over an image and a HOG descriptor is calculated for each pixel. These pixels are then combined into cell blocks for which gradient directions are computed and grouped into a number of orientation bins usually 9. Gradient magnitudes are added up to give added weight to the stronger gradients and noise reduction in weaker ones.

To calculate a HOG descriptor, the horizontal and vertical gradients need to be first calculated. At every pixel, the gradient holds a magnitude and a direction. Whenever a sharp change in intensity is observed the magnitude of gradient increases. None of them fire when the region is smooth[9].

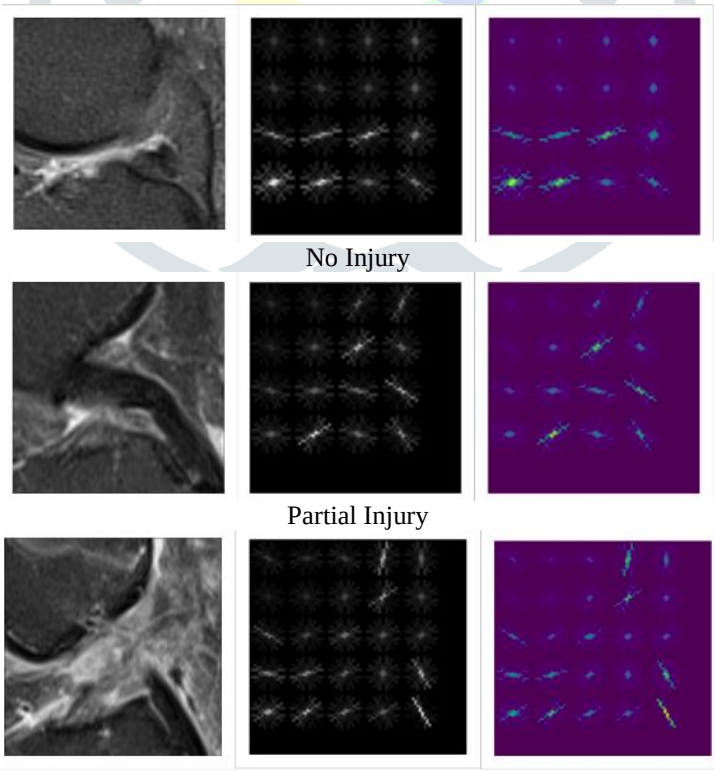
Now, for each of the cells in the image, a histogram of oriented gradients needs to be constructed using the gradient magnitude $|G|$ and orientation θ mentioned above.

The magnitude and direction of the gradient can be found out using the following formula,

$$g = \sqrt{g_x^2 + g_y^2} \quad (1)$$

$$\theta = \tan^{-1} g_y / g_x \quad (2)$$

Normalization is done in order to account for the changes in illumination and contrast. Finally, after all blocks are normalized, the resulting histograms are taken, concatenated, and treated as the final feature vector.



No Injury

Partial Injury

Complete Injury

Fig 6: Visualisation of calculated feature descriptors for a randomly chosen case of each class: (a) not injured, (b) partially injured, and (c) completely ruptured. Left column depicts scaled 3-slice ROIs. Middle column and last column depicts a visualisation of calculated HOG descriptors

2.3.2 Gist

In computer vision, there are different types of feature descriptor. Gist is a feature descriptor or feature extraction technique used for obtaining information from an image. GIST descriptors are a representation of a low-dimensional image that contains enough information to identify the scene in an image[10]. Global Descriptors allow representation of a very small dimensional image. These descriptors were first introduced to the world by Oliva and Torralba in 2001[12,13]. The dominant spatial structure of the scene from a set of perceptual dimensions is represented by gist descriptor. Being global image features they assist in characterizing various crucial information of the scene. Computing these features is done by convoluting the filter with an image at different spatial frequency, scales, and orientations. Thus, high and low frequency repetitive gradient directions of an image can be measured. The scores which are generated at each level of filtering at each orientation and scale are used as Gist features for an image[10]. Gist feature extraction is filtering of the input image into a number of low-level visual feature channels, like intensity, color, orientation, motion, and flicker at multiple spatial scales.

Computation of feature maps for color and intensity channels is done using:

$$M_i(c,s) = |O_i(c) \Theta O_i(s)| = |O_i(c) - interp_{s-c}(O_i(s))| \quad (3)$$

Feature maps are used to detect salient regions in each channel and are merged linearly to yield a prominent map. Information from the orientation channel is integrated by employing Gabor filters to the grayscale image[11].

Computation of feature maps for orientation channels is done using:

$$M_i(c) = Gabor(\theta_i, c) \quad (4)$$

Gist features (Gi k,l(c, s)) are computed per feature map using[11] :

$$G_i^{k,l}(c,s) = \frac{1}{16WH} \sum_{u=\frac{kw}{4}}^{\frac{(k+1)W}{4}-1} \sum_{v=\frac{lH}{4}}^{\frac{(l+1)H}{4}-1} [M_i(c,s)](u,v) \quad (5)$$

where W and H are width and height of the image respectively. k and l are the indices in horizontal and vertical directions respectively.

Finally, the features vectors are calculated for each scale, orientation, and frequency. Those feature vectors are combined to form a global feature descriptor which can be reduced by a principal component analysis (PCA)[10].

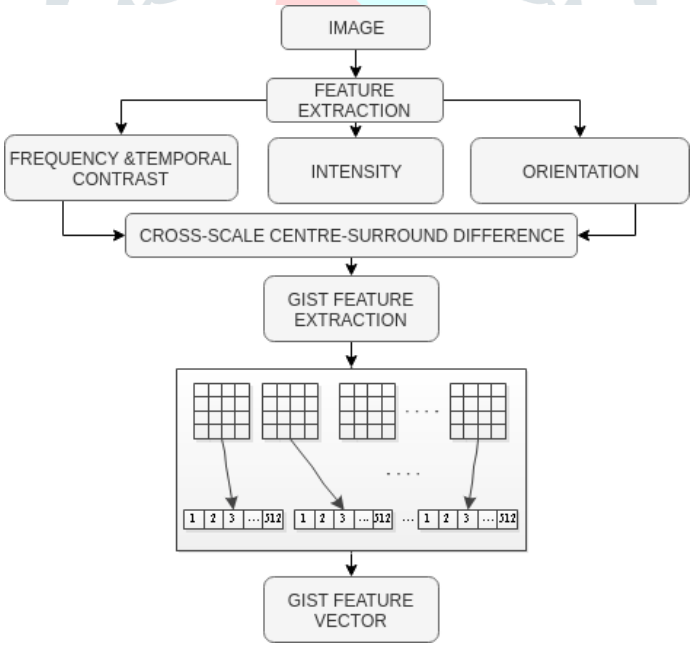


Fig 7: Flow for obtaining feature vector from an image using GIST

2.4 Machine Learning Techniques

2.4.1 Support Vector Machine

A Support Vector Machine (SVM)[14] is a classifier which is represented by a separating hyperplane. It is a form of supervised learning which uses the training data to form the hyperplane and then classifies the new input data based on it. One advantage of SVM is non-linear separability which makes it useful in the case of images and hence relevant to the case presented. The larger the margin, the lower is the generalization error of the classifier. There are 2 tuning parameters in an SVM classifier namely regularization and gamma parameter. A significant amount of nonlinear classification line with more accuracy can be obtained in a rational amount of time by varying these parameters

Minimization error function during training for this SVM is given as follows:

$$\frac{1}{2}W^T W + C\sum_{i=1}^{N}\xi_i \quad (6)$$

subject to:

$$y_i(W^T\emptyset(x_i) + b) \geq 1 - \xi_i \, and \, \xi_i \geq 0, i = 1, \ldots, N \quad (7)$$

C being the capacity constant, w being the vector of coefficients, b a constant, and ξ_i depicts the parameters for supervising conjoined inputs. ϕ is the kernel which is used to convert input data to feature space. i is the index for the N training cases. The class labels and autonomous variables are delineated by $y \in \pm 1$ and xi respectively. The error is corrected in direct proportion to C and hence to avoid overfitting C should be chosen carefully.

Support Vector Machine(SVM) are supplied with the features extracted using both the HOG features and GIST features.

Table 1 HOG+SVM

	precision	Recall	f1-score
avg/total	0.80	0.79	0.73

Table 2 GIST+SVM

	precision	recall	f1-score
avg/total	0.54	0.74	0.63

2.4.2 Radial Based K means

K-means suffers from the fact that it is linearly separable. If the classes are not linearly separable, nonlinear kernels are used, such as quadratic, cubic or radial basis function (RBF) kernels[15], which can be used to absolutely transform the feature space. Hence the data has to be projected onto high-dimensional kernel space and K-means has to be applied. This involves mapping data points in the input space on a high dimensional feature space using a Gaussian Radial based kernel function which increases the computational complexity.

Following is the function in a mathematical form-

$$K(X_i , X_j) = e^{-||Xi - Xj||^2 / 2\sigma^2} \quad (8)$$

Here the Gaussian RBF kernel transformation maps data to a kernel matrix for any 2 points resulting in quality clusters.

Table 3 HOG + RBF Kmeans

	precision	recall	f1-score
avg/total	0.41	0.15	0.19

Table 4 GIST + RBF Kmeans

	precision	recall	f1-score
avg/total	0.59	0.52	0.55

2.4.3 Kohonen's Self Organizing Map

Like SVM and RBK, Self Organizing Map also makes use of HOG features. The aim of Kohonen's SOM[19] is to instill similar behavior in a group of neurons according to a pattern of input. This mimics the processing of sensory information by the cerebral cortex of the human brain. Being an unsupervised form of learning, SOM tries to compare the node weights and the input vector and if they are similar or same then it optimizes the corresponding area of lattice to represent the class more closely to which the input vector belongs to. The way SOM works is by initializing the neuron weights and choosing the neuron vector according to how similar its weights compare to the input vector. The chosen node is called the Best Matching Unit (BMU).

The neurons in the SOM grid transition from being placed in randomly assigned positions to being tempered into a silhouette of inputted data. The BMU along with its neighboring nodes have its weight altered according to the following equation[23]:

$$W(t+1) = W(t) + \Theta * L(t)(V(t) - W(t)) \quad (9)$$

Where the t is the time step, the learning rate is represented by L which decreases with time. The new altered weight is the summation of the old weight (W) and a part of the difference (L) between the old weight and input vector (V). The impact on the learning rate that the node's distance from the BMU has is represented by Θ. Θ(t) is given as follows:-

$$\Theta = exp\left(\frac{dist^{\ 2}}{2\sigma^{\ 2}(t)}\right) \qquad (10)$$

Both the radius of the BMU and learning rate should be adjusted through validation. The neurons will get pushed around continuously and erratically if the values for both are excessive[23]. On the other hand, if the values are too small, then the neurons will move slowly towards their optimal position thus consuming a lot of time. Thus, it is exemplary if the learning rate and BMU radius are gradually decreased after assuming initial high values.

Table 5 HOG + SOM

	Precision	recall	f1-score
avg/total	0.63	0.51	0.56

2.4.4 Random Forest

Random forests (RF) model [16] is an ensemble of decision trees. It is a supervised learning algorithm which can be used for classification and regression purposes which is a big advantage in the case of random forest. A random forest, as the name suggests, is a group of decision trees which is trained using the bagging method which uses the principle that combination of multiple models increases the overall result. It's hyperparameters closely resemble those of bagging classifier or a decision tree. Random Forests can be made use of by using the classifier class for classification and Random Forest regressor for regression problems. It forms multiple decision trees using different decision rules and adds randomness to it for splitting purpose. It hence selects the best feature from amongst a random set of features. The randomness factor can be increased by adding random thresholds instead of calculated thresholds.

Time complexity taken by n instances learned by observing d features for transforming in an m-trees RF model is given by O(m*n*d*logn)[17]

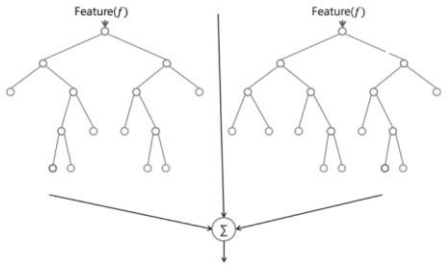


Fig 8: Representation of a collection of decision trees used by Random Forest

Table 6 HOG + Random Forest

	precision	recall	f1-score
avg/total	0.70	0.76	0.69

Table 7 GIST + Random Forest

	precision	recall	f1-score
avg/total	0.74	0.75	0.67

2.4.5 Autoencoder

An autoencoder is an unsupervised learning algorithm. It is a neural network which makes use of backpropagation in which input is same as output [20].

$$y(i)=x(i) \qquad (11)$$

The input is converted into a low level compressed form, also called latent space representation, and then it is again reconstructed to form the output. This process is being represented by the figure below which shows the 3 components of an autoencoder i.e an encoder, code, and a decoder.[24]

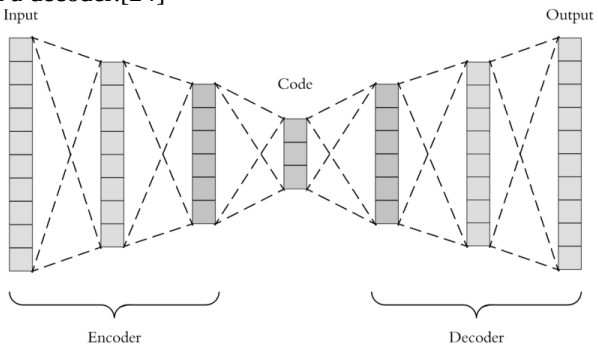


Fig 9: Structural components of an Autoencoder

1. **Encoder:** This is the part of the network that compresses the input into a latent-space representation. It can be represented by an encoding function *h=f(x)*.
2. **Decoder:** This part aims to reconstruct the input from the latent space representation. It can be represented by a decoding function *r=g(h)*.

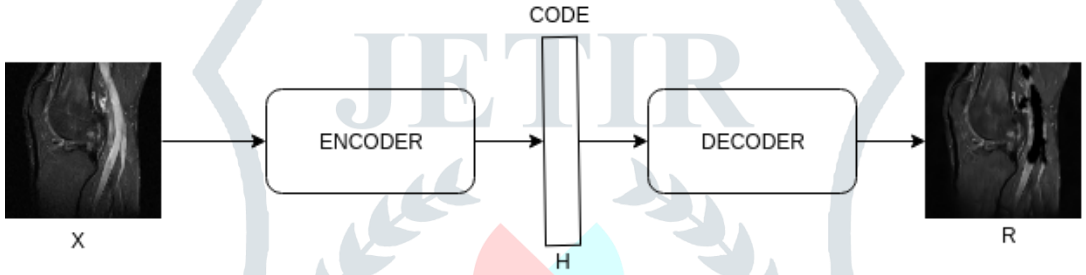


Fig 10: Structural components of an Autoencoder

Autoencoders are used for dimensionality reduction which can be lossy i.e a close but corrupted version of the input. An advantage of autoencoders is that they generate labels for training data on their own. The architecture of the autoencoder involves two fully-connected Artificial Neural Networks; one for the encoder to produce the code and another for the decoder to generate the output using the code.[24] The only essential condition of this architecture is that the input and output must have the exact same dimensions.

Traditionally, in an autoencoder architecture, the fact that a signal consists of a summation of other signals is not considered unlike Convolutional Autoencoders (CAE), which make use of the convolution operator to take adopt this scrutiny. The Convolution operator renders us capable of extracting some part of the content that results from filtering an input signal. Eventually, they get better at encoding the input set of signals and reconstructing the input in the form of output from them.

Due to their property of being stationary, natural images have the same enumerations for all the parts of the image i.e features learnt at one part can be applied to all the parts of an image[21]. For example, when features are extracted from small 8x8 parts sampled randomly over a larger image feature detector can be used anywhere in the image i.e it can be used to convolve with the larger image, thereby getting a disparate value for feature activations at every location in the image.

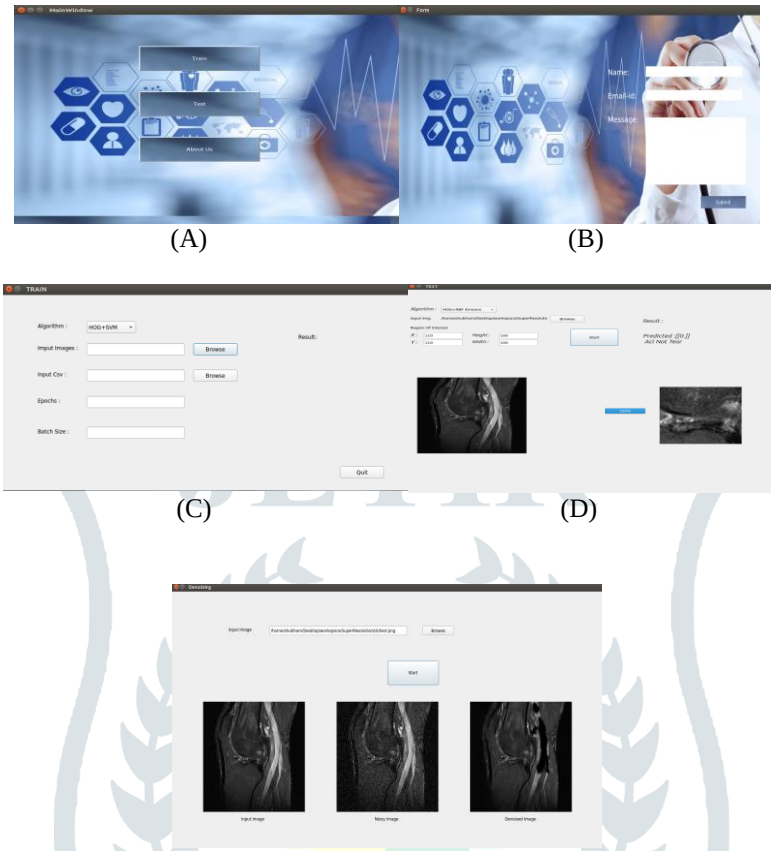
First, the input is given to the encoder which compresses it to a low dimensional representation. Depending on the input, the loss function is set i.e cross entropy for values in the range [0-1] and mean squared error for all other inputs[24]. The autoencoder can have as many layers as preferred. Usually, the number of nodes per layer decreases as the network goes deeper in terms of layers. This is the opposite case for the decoder i.e the number of nodes per layer increases each subsequent layer. The code size can be determined by the number of nodes in the middle layer as it is directly proportional to the amount of compression. The autoencoder can be made more powerful by tuning these hyperparameters. This may result sometimes in overfitting as the autoencoder eventually starts copying the inputs to the outputs thereby perfectly mimicking the identity function[24].

Neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular Neural Networks. Their activations can hence be computed with a matrix multiplication followed by a bias offset[22].The output of this fully connected layer is the class scores.

2.5 About the platform used for UI

In order to make this research available to users and fellow researchers, a user interface has been made for them to try out. PyQt, an open source technology, has been used for developing the GUI. PyQt was developed by RiverBank Computing Ltd. It is a Python interface for Qt, one of the most powerful, and popular cross-platform GUI library. UI consists of five pages - main page, test page, train page, denoising page, feedback page. The main page gives you options for selecting the required action to be

performed. The training page provides the facility to train on your own dataset and after finishing, user will be able to see the results of the same. The test page includes an option for testing an image on the pre-trained model provided by the authors for various algorithms. A denoising page has also been provided which displays three images; the test image, the noisy image which is the test image to which noise has been added for research purposes, and lastly the transitioned image after denoising the noisy image. The last page is the feedback page. If there is any confusion present regarding the algorithm or any other points, then a query can be sent to any one of the authors.



(A) (B)

(C) (D)

(E)

Figure 11- Images of the User Interface provided:
(A)Main Page (B)Feedback (C)Train Page (D)Test Page (E)Denoising Page

III. RESULTS AND DISCUSSION

The significance of the image processing has also increased over a period of time for the valuable and precise analysis of the image for the particular goal. But generally the medical images are noisy and these noises come from various sources ranging from the bit error during the transmission and capturing process like MRI, CT, US etc to the environmental conditions in which the image acquisition took place. This causes a discrepancy in the decision of the doctor because there might a possibility that the lesion to be detected in an image may not be clear. Hence the image has to be denoised first for making the region of interest more informative. For research purposes, artificial noise was added to the images and then fed to the algorithm. Even if the dataset is small an image augmentation technique to proliferate the dataset has been used. The algorithm used was DnCNN, which can handle the gaussian noise of any level, which is not known beforehand ,apart from other algorithms which require to be trained on a certain noise level[8].

Both feature extraction methods were paired with various appropriate algorithms, thus producing a total of 8 independent experimental models whose names are self-explanatory: HOG+SVM, HOG+RBF K-Means, HOG+SOFM, HOG+RF, GIST+SVM, GIST+RBF K-Means, GIST+SOFM, GIST+RF, and autoencoder. Investigation of other methods was done as well but the results were not as expected hence not included in detail.

It is well known that the performance of the predictive model built using an algorithm for learning from the extracted features is dependent on the choice of some hyperparameters. For this purpose, selective important parameters were varied while the others were left as they were present in the programming library.

Hyperparameter ranges and step sizes were estimated from experience in order to cover the most promising scenarios while retaining computational feasibility.

HOG descriptors were extracted from ROI volume data using Skimage, an open source library. For all the slices available in the volume a separate descriptor was calculated. Slice descriptor vectors were then concatenated to form a HOG feature vector for the ROI volume. The entire feature vector was then saved as a pickle file for the ease of the use of different algorithms.

Length of the resulting feature vector depended on the input patch size but for some algorithm, padding was done to satisfy the input size of the feature vector.

Pyleargist equipped us with the required set of tools for Gist descriptors. According to [14], appropriate number of spatial scales i.e 4 and orientations i.e 8 were used which resulted in 32 Gabon filters without boundary extension. The number of blocks that were used to decide the descriptor or grid size coarseness was the only fluctuating value.

The volume descriptors for ROI were formed by initially computing the Gist descriptor per slice for an ROI volume disparately after which they were concatenated. The number of blocks used would be dependent on the length of the feature vector obtained in results.

With the aim of drawing a distinction between various clinical conditions, ML algorithms utilized both of these characteristic features even though some images exist which the naked eye can easily distinguish. Although the nuances between depictions in the example provided in Fig 5 are differentiable through the naked eye, the same cannot be said for the substantial part of used data. Hence, ML algorithms were designated to draw patterns between perceived outcomes and features.

Further, the impact of ROI selection phase on ACL state intrigued us due to 2 significant reasons. Firstly, an observation in terms of ROI was made of a possibility of an impulsive bias being introduced during the extraction phase by a skilled radiologist. Thus it can be assumed that the position or shape of ROI gets affected by the observed ACL condition. Secondly, this examination provides us with an approximate of the properties that an algorithm should possess for establishing a precise ROI position which is a significant step in aiding the creation of a fully-automated CAD detection system.

Table 8

Algorithm	Accuracy
HOG + SVM	78.64
GIST + SVM	74.75
HOG + RBF K-Means	34.29
GIST + RBF K-Means	52.1
HOG + SOM	51.45
Autoencoder	75.77
HOG + Random Forest	76.37
GIST + Random Forest	74.29

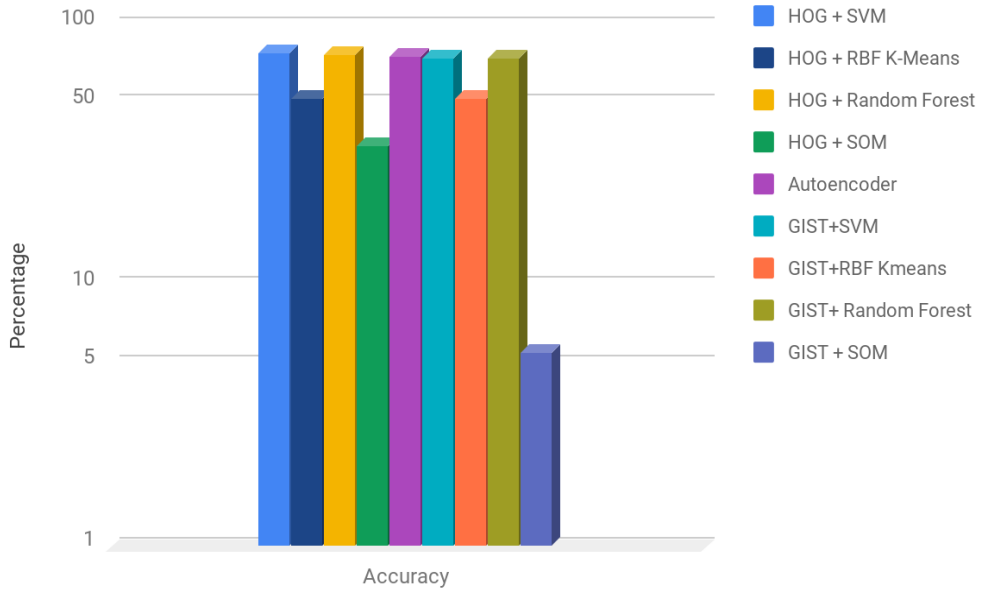


Figure 12

IV. ACKNOWLEDGMENT

This research was possible due to the dataset provided by the Clinical Hospital Centre Rijeka, Croatia

REFERENCES

[1] F. Roemer, M. Crema, S. Trattnig, A. Guermazi, Advances in imaging of osteoarthritis and cartilage, Radiology 260 (2) (2011) 332–354.
[2] T. W. Hash, Magnetic resonance imaging of the knee., Sports health 5 (1)(2013) 78–107. doi:10.1177/1941738112468416.
[3] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," IEEE Transactions on Image Processing, vol. 15, no. 12, pp. 3736–3745, 2006.
[4] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," to appear in IEEE transactions on Pattern Analysis and Machine Intelligence, 2016.
[5] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in Advances in Neural Information Processing Systems, 2009, pp. 769–776.
[6] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2392–2399.
[7] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in Advances in Neural Information Processing Systems, 2012, pp. 341–349.
[8] Zhang K, Zuo W M, Chen Y J, Meng D Y, Zhang L. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Processing*, 2017, 26(7): 3142-3155.
[9] T. Kobayashi, A. Hidaka, and T. Kurita, "Selection of histograms of oriented gradients features for pedestrian detection,"in International conference on neural information processing, pp. 598–607, (2007).
[10] M Oujaoura, B Minaoui, M Fakir and O Bencharef. (2014). Recognition of Isolated Printed Tifinagh Characters. International Journal of Computer Applications. Volume 85, No 1.
[11] Chauhan A, Chauhan D, Rout C, Role of Gist and PHOG Features in Computer-Aided Diagnosis of Tuberculosis without Segmentation, 2014 Nov 12. doi: [10.1371/journal.pone.0112980]
[12] Aude Oliva and Antonio Torralba. Modeling the shape of the scene : A holistic representation of the spatial envelope. International Journal of Computer Vision, 42 : pp. 145–175, 2001.
[13] Aude Oliva , Antonio Torralba, Building the gist of a scene: the role of global image features in recognition, Progress in Brain Research, 2006.
[14] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.
[15] Dhillon, Inderjit S. and Guan, Yuqiang and Kulis, Brian, Kernel K-means: Spectral Clustering and Normalized Cuts, doi : 10.1145/1014052.1014118,ACM 2004, pg 551-556.
[16] L. Breiman, Random forests, Machine learning 45 (1) (2001) 5–32.
[17] I. Witten, E. Frank, M. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition, Morgan Kaufmann, 2011.
[18] Štajduhar, M. Mamula, D. Miletić, G. Unal, Semi-automated detection of anterior cruciate ligament injury from MRI, Computer Methods and Programs in Biomedicine, Volume 140, 2017, Pages 151–164.
[19] T.Kohonen, Self-organizing maps, 2nd edition, Springer, New York, 1995
[20] http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/
[21] http://ufldl.stanford.edu/tutorial/supervised/FeatureExtractionUsingConvolution/
[22] http://cs231n.github.io/convolutional-networks/#fc
[23] http://www.ai-junkie.com/ann/som/som1.html
[24] https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798
[25] https://towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c