Detection and Diagnosing of Spinal Stenosis Based on Faster RCNN

By
Muhammad Osama Jamil (B16158060)
Muhammad Moiz Khan (B16158059)
Muhammad Hamza Temuri (B16158055)

Project Submitted to the
Department of Computer Science (DCS)
In Partial Fulfillment of the
Requirements for the degree of
BACHELOR'S OF SCIENCE
SOFTWARE ENGINEERING
Major Subject: Computer Science

Approved by

Dr. Humera Tariq, Project Supervisor

University of Karachi
Karachi

October, 2020

Abstract

This report is about the spinal diseases called spinal canal stenosis. Spinal stenosis is a narrowing of the spaces within your spine, which can put pressure on the nerves that travel through the spine. The purpose of this project is to automate the process of detecting these diseases. This would help in fasten the process of pointing out a problem and help in making a decision.

We took a dataset contained of MRI's of spine. The MRI's were in dicom format then we converted them in jpeg format. After converting we reduced their size (space) by cropping and applying other methods so it would be faster for machine to learn. After reducing size we trained our model on these MRI's (sagittal view).

After training we gave some inputs to test if the model is working and if working then at what accuracy. The output we got are the marked images with about 85% of accuracy, the marked areas are pointing out where problem occurs and where it does not.

So, this project provides an easy and fast way for diagnosing spinal stenosis in sagittal view.

Acknowledgements

All praise is due to Allah Alone, the Sovereign Lord of all the Universes and Allah's Peace and Blessings be upon him after whom there is no other Prophet.

We would like to express our great gratitude to Miss Humera Tariq for her valuable and constructive suggestions during the planning and development of this project. Her willingness to give her time so generously has been very much encouraging.

Table of Contents

Chapter 1: Introduction

Introduction

The lumbar spine (lower back) consists of five vertebrae in the lower part of the spine, between the ribs and the pelvis. Lumbar spinal stenosis is a narrowing of the spinal canal, compressing the nerves traveling through the lower back into the legs. While it may affect younger patients, due to developmental causes, it is more often a degenerative condition that affects people who are typically age 60 and older.

A description of LSS was published by Sachs and Frankel in 1900, but the first clinical description of LSS is usually attributed to the Dutch neurosurgeon Henk Verbiest, whose report appeared in 1954.

Spinal stenosis began to be recognized as an impairing condition in the 1950s and 1970s. During the 1970s and 1980s, many case reports showed successful surgical treatment rates, but these were based on subjective assessment by surgeons. In 1992, Johnsson, Rosén and Udén described the natural history of LSS, with different conclusions about prognosis and treatment: "70% of patients reported no significant change in symptoms, 15% showed significant improvement, whereas 15% showed some deterioration. The investigators concluded that observation is a reasonable treatment option for lumbar stenosis and that significant neurologic deterioration is rare.

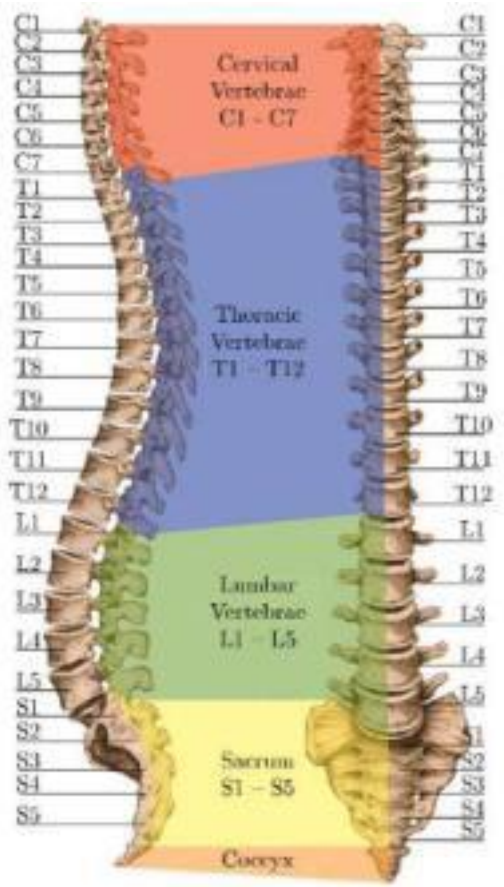
The definitive diagnosis is established by either computerized tomography (CT) or magnetic resonance imaging (MRI) scanning. MRI is the preferred method of diagnosing and evaluating spinal stenosis of all areas of the spine, including cervical, thoracic, and lumbar.

In this project we process the MRI images of patients and output comes with processed images pointing out the problem. This is very helpful in diagnosing a single patient or a number of patients. Through our project you can diagnose the MRI's faster and especially for a large number of patients it saves a lot of time.

Chapter 2: Background

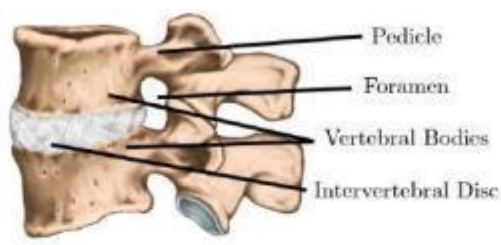
2.1 SPINAL ANATOMY:

Anatomical illustrations detailing the anatomy of the human spine can be seen.



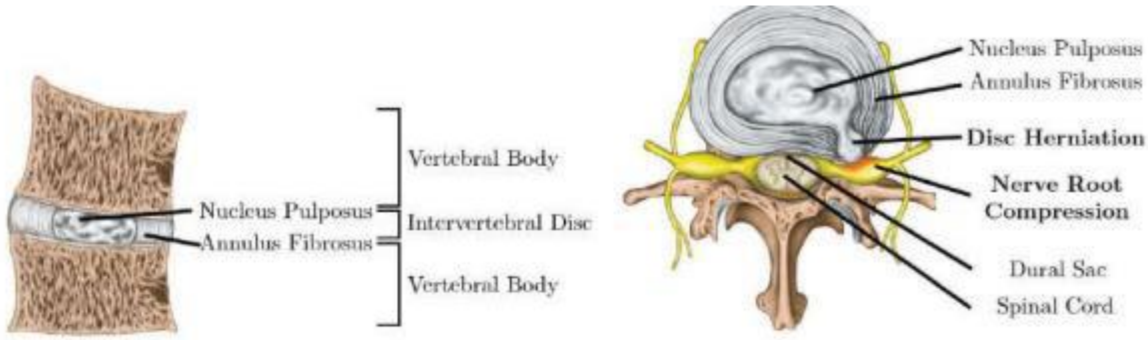
Sagittal view

2.1.1 Vertebrae. A normal human spine contains 33 vertebrae in total, though this number can vary. They are grouped into five main regions corresponding to the curves of the spinal column: cervical, thoracic, lumbar, sacrum and coccyx. There are seven cervical (C1–C7), twelve thoracic (T1–T12), five lumbar (L1-L5), five sacral (S1–S5).



Overview of intervertebral disc

2.1.2 Intervertebral Disc. In between a pair of vertebrae are intervertebral discs. They are labelled according to their neighboring vertebrae e.g. the L5-S1 intervertebral disc is the disc between the L5 and S1 vertebrae. The disc is made up of two main parts: a nucleus pulposus, and an annulus fibrosus. Though these parts are quite distinct, separation of these parts in the MRI is normally unclear. The region connecting the disc and the vertebral body is called the vertebral endplate.



2.1.3 Spinal Cord. The spinal cord is a bundle of nervous tissue normally involved in the transmission of signals from the brain to the body. A specific region of the spine controls a specific motor function e.g. the L1-L4 region controls the flexing of the thigh.

2.1.4 Lumbar MRIs. There are two main types of MRI sequences used in this thesis namely, T1-weighted and T2-weighted MRIs. In short, T1-weighted scans are produced using short repetition and echo times, TR and TE respectively, while T2-weighted scans are produced using longer TR and TE. These distinctions in acquisition times cause visual differences of the tissues in both sequences. In lumbar scans, this means the spinal canal appears brighter in T2-weighted scans than T1-weighted scans. Similarly, inflammations appear darker in T1 and brighter in T2 e.g. marrow changes of the vertebral bodies. These differences result in variation of scanning protocols across difference centers for acquisition of MR scans though T2-weighted scans tend to be more popular.

2.1.4 Lumbar MRIs. In our project we were currently working on **sagittal view** or **frontal view.** We will try to take this project to that level in which all views can be used and also our model detection is from L1 to L5 lumber in which most of the spine disease like stenosis occur.

Chapter 3: Dataset and Pre-processing

3.1 DATASET:

This data set contains anonymized clinical MRI study, or a set of scans, of 515 patients with symptomatic back pains. Each patient data can have one or more MRI studies associated with it. Each study contains slices, i.e., individual images taken from either sagittal or axial view, of the lowest three vertebrae and the lowest three IVDs. The axial view slices are mainly taken from the last three IVDs – including the one between the last vertebrae and the sacrum. The orientation of the slices of the last IVD are made to follow the spine curve whereas those of the other IVDs are usually made in blocks – i.e., parallel to each other. There are between four to five slices per IVD and they begin from the top of the IVD towards its bottom. Many of the top and bottom slices cut through the vertebrae leaving between one to three slices that cut the IVD cleanly and show purely the image of that IVD. In most cases, the total number of slices in axial view ranges from 12 to 15. However, in some cases, there may be up to 20 slices because the study contains slices of more than last three vertebrae. The scans in sagittal view also vary but all contain at least the last seven vertebrae and the sacrum. While the number of vertebrae varies, each scan always includes the first two sacral links. There are a total 48,345 MRI slices in our dataset. The majority of the slices have an image resolution of 320x320 pixels, however, there are slices from three studies with 320x310 pixel resolution.

Medical images are in dicom format. Therefore, dicom viewer is used to view all the images.

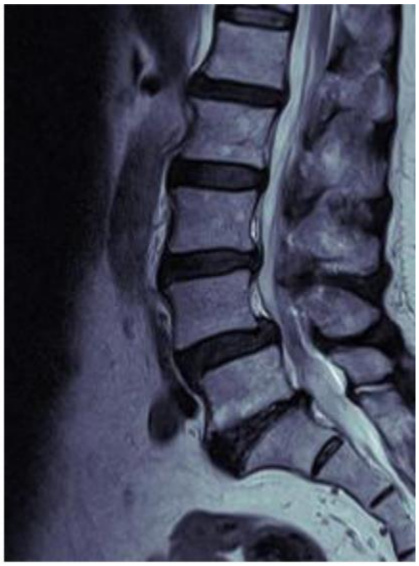
Dataset Link: https://data.mendeley.com/datasets/k57fr854j2/2

3.1.1 Dataset View:

View of our dataset.

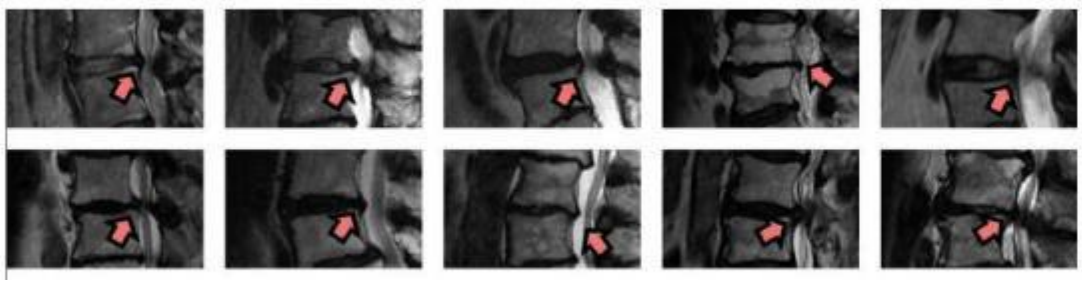
Patient 1

Patient 2

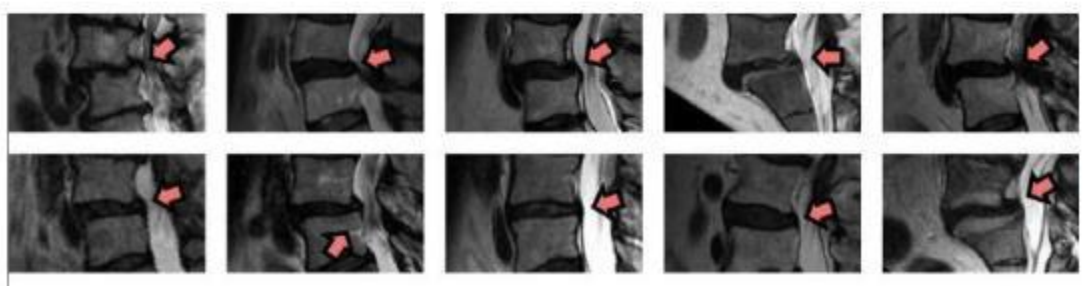


Patient 3

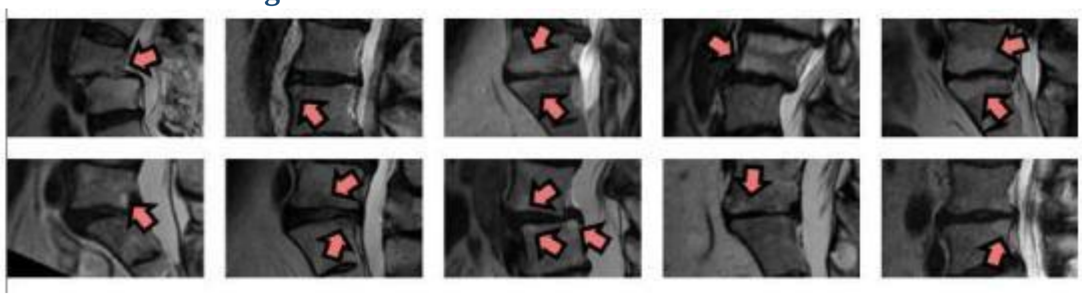
3.1.2 Central Canal Stenosis:



3.1.3 Spondylolisthesis:



3.1.4 Marrow Changes:



3.2 Image Pre-Processing:

3.2.1 Conversion DICOM to JPEG:

As we are working on medical images and the standard format of images are DICOM. DICOM format is not properly supported by python. So, we converted our image to JPEG for further processing.

3.2.2 Image Normalization:

Image normalization is a typical process in image processing that changes the range of pixel intensity values. Its normal purpose is to convert an input image into a range of pixel values that are more familiar or normal to the senses, hence the term normalization. In this work, we will perform a function that produces a normalization of an input image (grayscale or RGB). Then, we understand a representation of the range of values of the scale of the image represented between 0 and 255, in this way we get.

norm = (img - np.min(img)) / (np.max(img) - np.min(img))

3.2.3 Cropping & Image Resizing:

It depends on the context. You might casually speak of resizing a web image by zooming into it on your browser so you can see more of the detail. In that case, the image is not changed at all - Only your view of it. When professionals speak of resizing, they are often talking about changing the number of pixels in an image. As an example, a pro may have shot an image at 1 MB. That is much too big to use on a web page. It takes too long to load or download, and too much room to archive.

- Original resolution of image was (384 x 384)
- After resizing image, the resolution is (215 x 295)
- Reducing size from 200KB to 16KB for faster processing.
- Removing black borders from the image.
- Zoom in picture and focus on Lumbar L1 - L5 and Sacrum S1 - S3.

Annotations:

A note by way of explanation or comment added to a text or diagram.

Size for annotation box width = 35, Height = 32.

1. **xmin:** x-coordinate of the bottom left part of the image
2. **xmax:** x-coordinate of the top right part of the image
3. **ymin:** y-coordinate of the bottom left part of the image
4. **ymax:** y-coordinate of the top right part of the image

Photo filter is used to annotate our images.

CHAPTER 4: Experimentation

4.1 Deep Learning

We heavily used Convolutional Neural Networks (CNNs), which is why we include literature concerning CNN as background. CNN is a type of artificial neural network (ANN) designed to work on data that are spatially connected e.g. an image, a video, an audio file and is a part of the increasingly popular subset of machine learning now called deep learning. A standard CNN is made up of five different layers: convolutional, pooling, dropout, non-linearity (typically ReLU) and fully-connected layers. In training, sets of weights or convolutional filters are learned such that the input becomes linearly separable in the last layer of the CNN.

4.2 FRCNN

In our current context/problem CNN is not a good option to use in detecting certain regions (ROI) in our LUMBER MRI, so FRCNN is used. A convolutional neural network (CNN) is mainly for image classification. While an R-CNN, with the R standing for region, is for object detection. A typical CNN can only tell you the class of the objects but not where they are located.

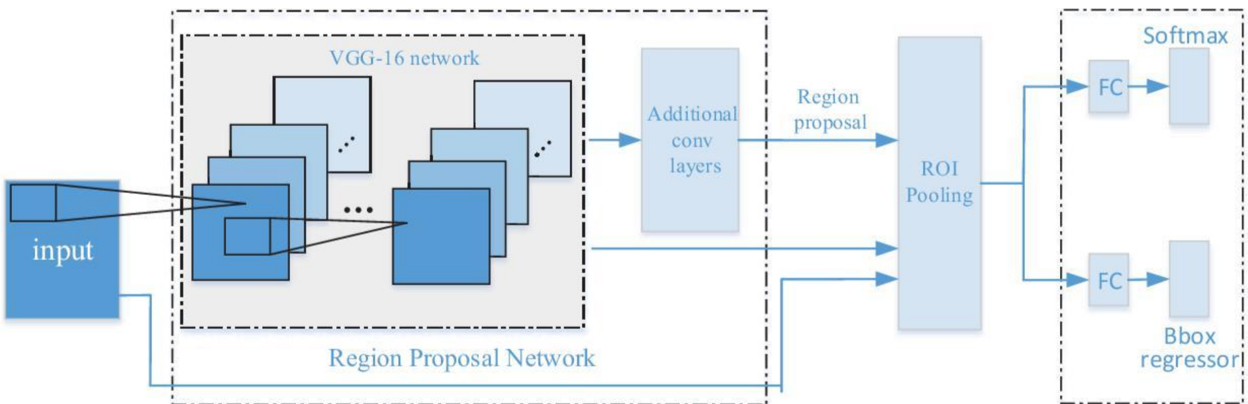


Figure: 4.1

4.3 IMPLEMENTATION

4.3.1 Classification

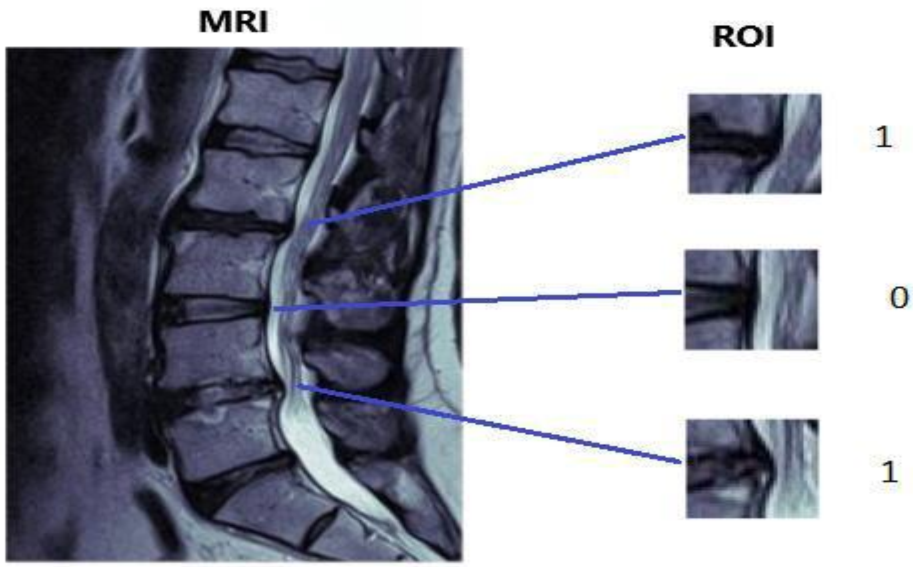


Figure: 4.2

*Figure 4.2: Each MRI image is given to FRCNN with selected inputs each of size **35×32** (in which **0** indicate no problem and **0** indicate defected) which are classes for FRCNN.*

Training

The model is trained on **100** test samples (images) each of size **215 x 295,** the total number of entries (input volumes/ ROIs) are 265 in which **143** indicates defect and **122** with no problem. Model is trained on **100 epochs** with **50 iterations** (epoch length) which takes up about **48 hrs.** to train complete model.

4.3.2 Evaluation Protocols

Evaluation protocols currently are mainly dependent on us, as students we tried our best to learn, observe from various sources on internet about spinal stenosis. We separated 100 samples/images from a huge dataset and identified as per our understanding about defected and normal segment of Lumber Canal Stenosis. Figure 4.3 shows an example of defected and normal portions of discs from our understanding which are given as input to FRCNN for training of model.

[11]

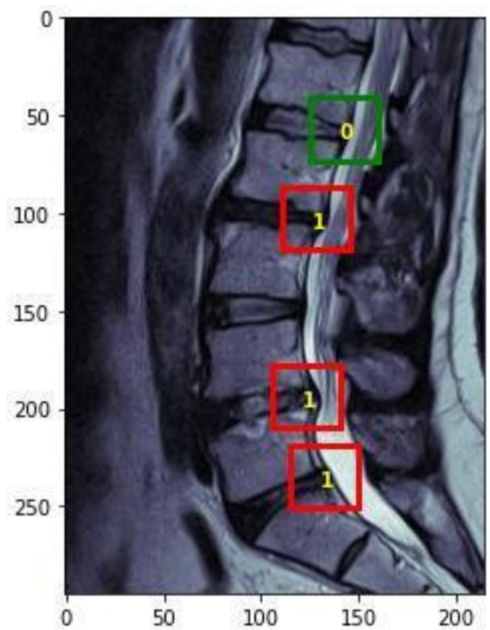


Figure: 4.3

4.3.3 Limitations

We only use T2-weighted sagittal scans in our experiments while certain gradings are normally assessed with the addition of other modalities e.g. axial scans are commonly used to assess central canal stenosis, and Modic changes grading typically requires both T1-weighted and T2-weighted scans. We compensate for the classification of Modic changes by only looking at Modic changes specific to T2-weighted scans. This is why we call our grading, marrow changes. We are able to achieve reasonable performance in central canal stenosis with only sagittal scans.

4.4 WORKING

MRI images of format **jpeg, png, dicom** can be given to the model to predict disease. The model then compresses, crop and convert image to **jpeg**, the final image given to model is of size **12 to 18 kb** and **215 x 295** in resolution. Now our pre trained model predicts and classify the locations on image with problem and normal discs with bounding boxes.

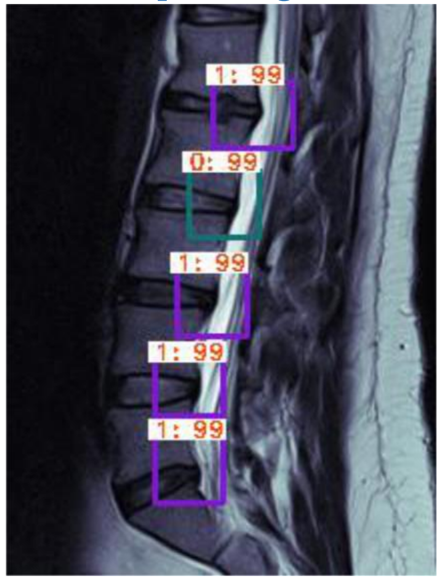
Result

Figure 4.4 shows raw image which is given as input to the model, then figure 4.5 shows the output image with boxes, where **0 = Normal and 1 = Defect** and percentage which shows predicted accuracy percentage of region.

Input Image

Figure: 4.4

Output Image



1 = Error, 0 = No Error

Figure: 4.5

CHAPTER 5: Results

5.1 Confusion Matrix:

n= 100	Predicted: NO	Predicted: YES
Actual: NO	51	12
Actual: YES	03	34

TRUE NEGATIVE: 51
TRUE POSITIVE: 34
FALSE POSITIVE: 12
FALSE NEGATIVE: 03

5.2 Accuracy:

the quality or state of being correct or precise

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Accuracy = (51 + 34)/ (51 + 12 + 03 + 34)
Accuracy = 0.85
Accuracy Percentage = 85%
Our trained model is 85% accurate.

5.3 Precision:

Precision is defined as the fraction of relevant instances among all retrieved instances.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision = (34) / (34 + 12)
Precision = 0.73
Precision Percentage = 73%

5.4 Recall:

Recall, sometimes referred to as 'sensitivity, is the fraction of retrieved instances among all relevant instances.

$$\text{recall} = \frac{\text{tp}}{\text{tp} + \text{fn}}$$

Recall = (34)/ (34 + 03)
Recall = 0.91
Recall Percentage = 91%

5.5 Future Improvement

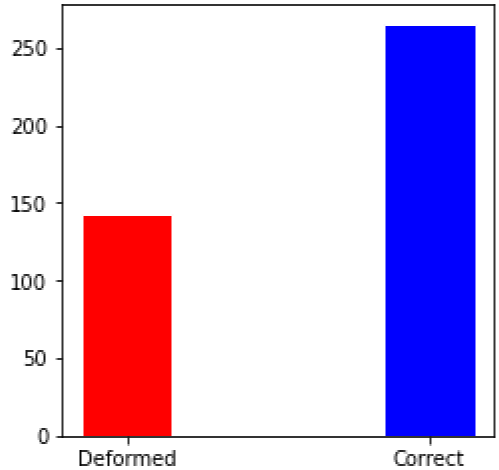
In this project we have focused on Lumber region, in future we will try to make more improvement in this model and will also try to work on other regions as well.

Chapter 6: Graphical Illustrations and System Requirements

6.1 Workflow Diagram:

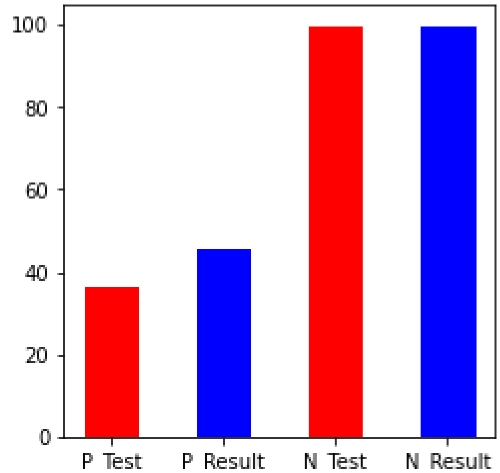
Following diagram shows the flow of work/task from start of project to the end.

6.2 CORRECT VS DEFORMED TRAINING DATA:



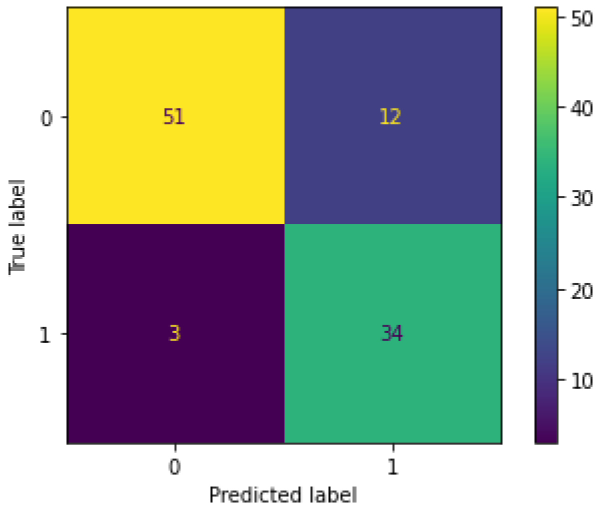
Following Bar chart shows the illustrations of correct and deformed values provided in our training dataset.

6.3 TESTING VS RESULTS:



P_Test are positive ground truth values
P_Result are positive values predicted by our model.
N_Test are negative ground truth values
N_Result are negative values predicted by our model.
As, you see that there is very small difference the shows our model is working well.
Note: **These illustrations are for those who doesn't understand confusion matrix well.**

6.4 Confusion Matrix:



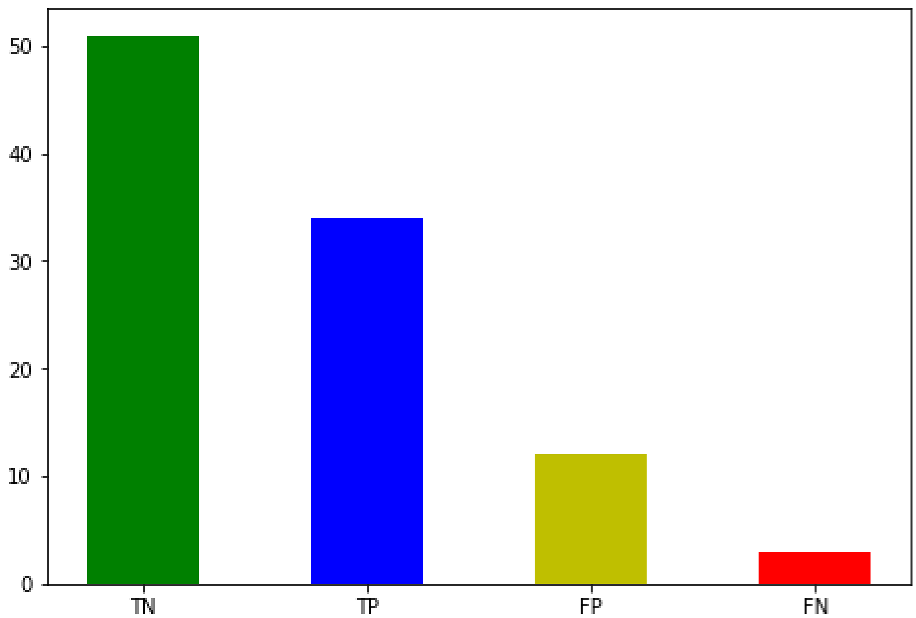
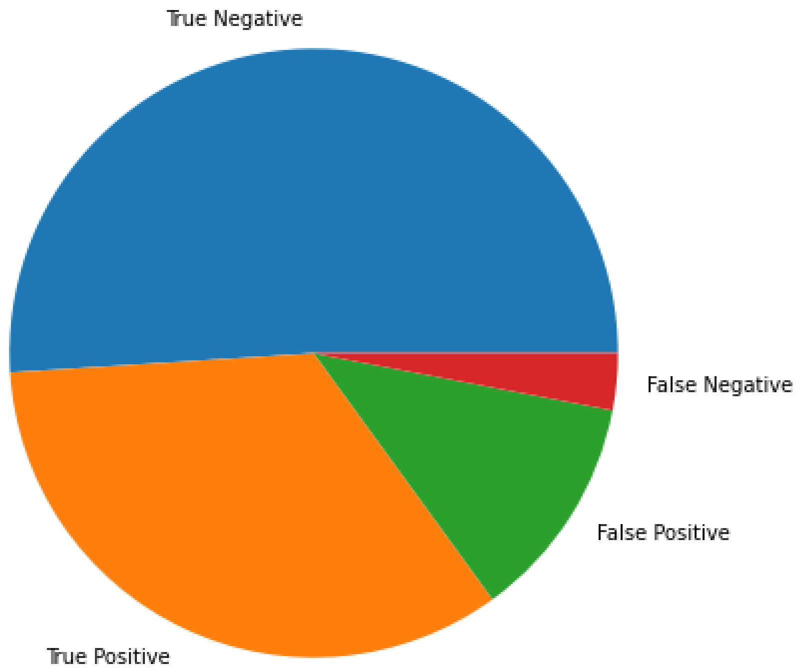
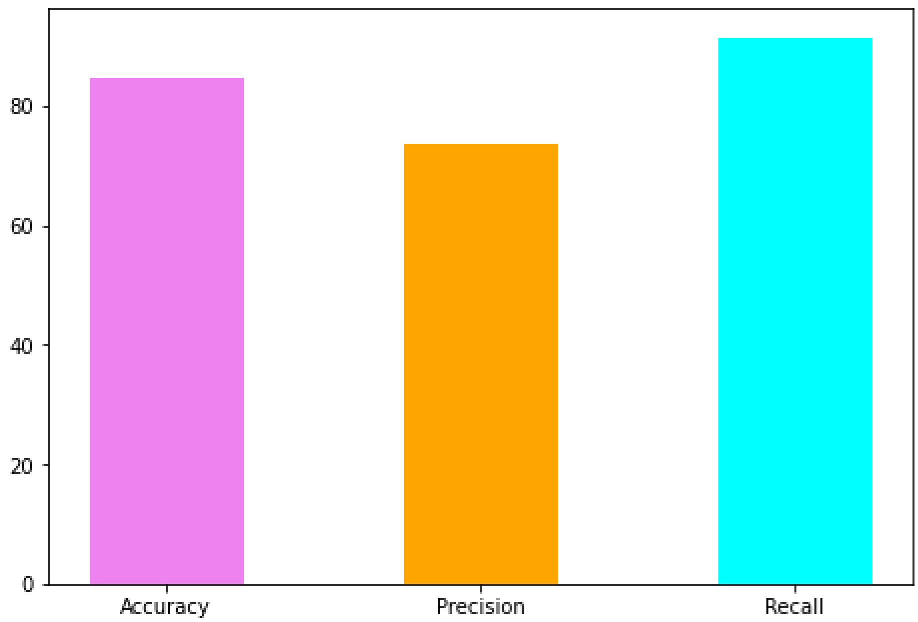


Illustration show TRUE NEGATIVES & TRUE POSITIVE have higher rate than FP and FN.



Above is the PIE CHART of the same TP TN FP FN. Shows most of the area is captured by TN and TP.

6.5 Accuracy Precision Recall:



Above illustration shows ACCURACY 85%, PRECISION 73% and RECALL 91%

6.6 System Requirements:

- i5 2^{nd} gen or above.
- RAM Minimum 4 GB.
- HDD 2GB free space (SSD preferred for faster use).
- Windows 7,8,10.

Summary

In this project we have proposed a method for detecting/diagnosing spinal disease called spinal stenosis.

- In Chapter 1, we have given the introduction about the problem and some previous workings on it.
- In Chapter 2, we have given some basic explanation about human spine for better understandings.
- In Chapter 3, we look at different datasets and the necessary pre-processing steps used. In pre-processing we have defined how we normalized, resized and cropped images.
- In Chapter 4, we described the working of our project. What methods we used and how we trained our model.
- In Chapter 5, we have defined the result like accuracy, precision, recall and confusion matrix.
- In Chapter 6, we have provided the graphical illustration of our project.

.

References

Tonya Hines, CMI, Mayfield Clinic, Cincinnati, Ohio. 2018. Anatomy of the Spine. [https://mayfieldclinic.com/pe-anatspine.htm].

Anatomy of the Spine. [https://www.globusmedical.com/patient-education-musculoskeletal-system-conditions/symptoms/anatomy-of-the-spine/].

kbardool/keras-frcnn, September 10, 2020. Keras Implementation of Faster R-CNN. [https://github.com/kbardool/keras-frcnn].

PULKIT SHARMA, NOVEMBER 4, 2018. A Practical Implementation of the Faster R-CNN Algorithm for Object Detection. [https://www.analyticsvidhya.com/blog/2018/11/implementation-faster-r-cnn-python-object-detection/].

Mobbs, R. J., Phan, K., Malham, G., Seex, K., & Rao, P. J. (2015). Lumbar interbody fusion: techniques, indications and comparison of interbody fusion options including PLIF, TLIF, MI-TLIF, OLIF/ATP, LLIF and ALIF. Journal of Spine Surgery, 1(1), 2-18.
Backstrom, K. M., Whitman, J. M., & Flynn, T. W. (2011). Lumbar spinal stenosis-diagnosis and management of the aging spine. Manual therapy, 16(4), 308-317. [https://www.physio-pedia.com/Lumbar_Spinal_Stenosis].

Donald Corenman, MD, DC. How to Read a MRI of Lumbar Degenerative Spondylolisthesis and Spinal Stenosis. [https://www.youtube.com/watch?v=A9InC8ppa9k].

Dr Mostafa El-Feky and Dr Aviad Lampner et al. [https://radiopaedia.org/articles/spinal-stenosis-1].

Prince Canuma, Oct 11, 2018. Image Pre-processing. [https://towardsdatascience.com/image-pre-processing-c1aec0be3edf].

Yinghan Xu, Nov 20, 2018. Faster R-CNN (object detection) implemented by Keras for custom data from Google's Open Images Dataset V4. [https://towardsdatascience.com/faster-r-cnn-object-detection-implemented-by-keras-for-custom-data-from-googles-open-images-125f62b9141a].

Appendix A: Initial Report/Proposal

1) Submission 1: Anatomy

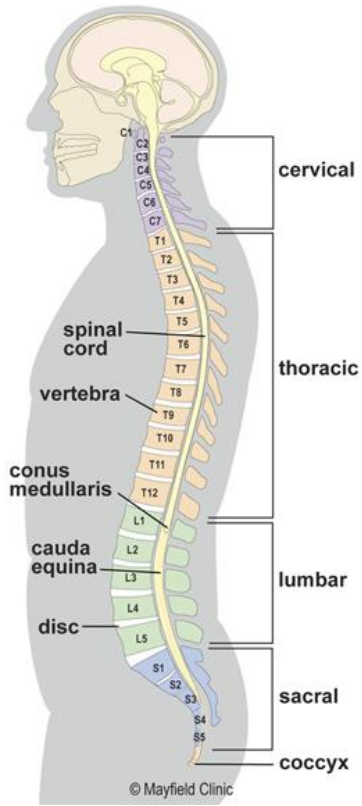
TASK:

(0) Anatomy of spine as you share the links with me
(1) How to load a data set?
(2) Show and display images from you own application
(3) Certain pre-processing like image resizing, normalization.

Question # 0

Anatomy of spine as you share the links with me

The normal anatomy of the spine is usually described by dividing up the spine into three major sections: the cervical, the thoracic, and the lumbar spine. (Below the lumbar spine is a bone called the sacrum, which is part of the pelvis). Each section is made up of individual bones, called vertebrae.

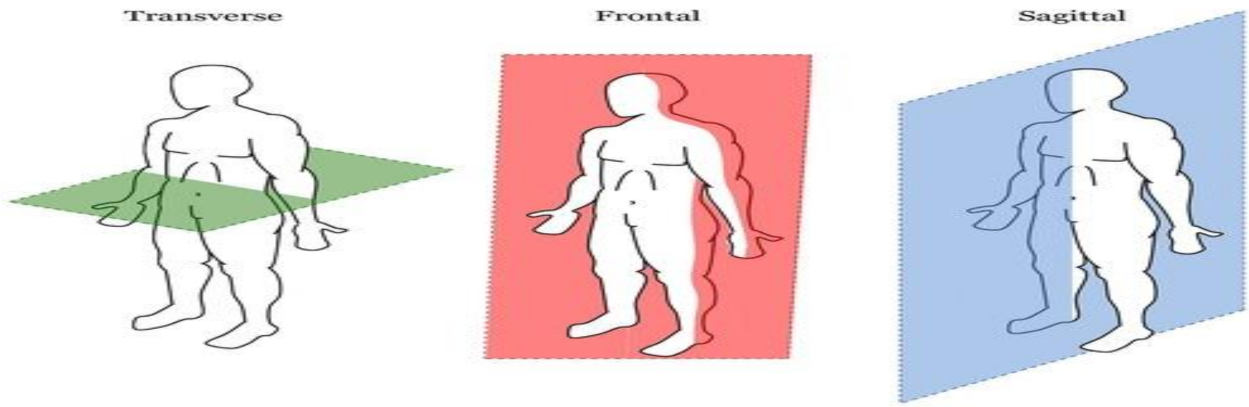


LUMBAR (LOW BACK) - the main function of the lumbar spine is to bear the weight of the body. The five lumbar vertebrae are numbered L1 to L5. These vertebrae are much larger in size to absorb the stress of lifting and carrying heavy objects.

Three types of views in MRI.

1) Sagittal View
2) Transverse View (Axial view)
3) Coronal View (Frontal view)

For now, we are just working on sagittal view.



(This view)

SPINAL STENOSIS:

Spinal stenosis is a narrowing of the spaces within your **spine**, which can put pressure on the nerves that travel through the **spine**. **Spinal stenosis** occurs most often in the lower back and the neck. Some people with **spinal stenosis** may not have symptoms. Symptoms can worsen over time.

Question # 1 & 2

How to load a data set?

Show and display images from you own application

In [2]:
```
import os
import pydicom
import matplotlib.pyplot as plt
```

In [3]:
```
path = r"C:\Users\OSama JamIL\Desktop\Final Year Project\T1_TSE_SAG_320_0003"
```

In [4]:
```
list = []
for file in os.listdir(path):
    if file.endswith(".ima"):
        list.append(file)
```

In [5]:
```
print(list)
```

```
['T1_TSE_SAG__0001_001.ima', 'T1_TSE_SAG__0001_002.ima', 'T1_TSE_SAG__0001_003.ima', 'T1_TSE_SAG__0001_004.ima', 'T1_TSE_SAG__0001_005.ima', 'T1_TSE_SAG__0001_006.ima', 'T1_TSE_SAG__0001_007.ima', 'T1_TSE_SAG__0001_008.ima', 'T1_TSE_SAG__0001_009.ima', 'T1_TSE_SAG__0001_010.ima', 'T1_TSE_SAG__0001_011.ima', 'T1_TSE_SAG__0001_012.ima', 'T1_TSE_SAG__0001_013.ima', 'T1_TSE_SAG__0001_014.ima', 'T1_TSE_SAG__0001_015.ima']
```

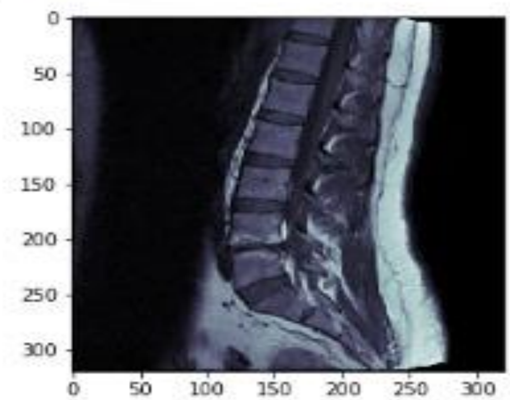
In [17]:
```
def read_(i):
    ds = pydicom.dcmread(i)
    plt.imshow(ds.pixel_array,cmap=plt.cm.bone)
    ds.pixel_array.shape
```

In [18]:
```
read_(list[7])
```

Question # 3

Certain pre-processing like image resizing, normalization

In [97]:
```
plt.imshow(res,cmap=plt.cm.bone)
plt.show()
```



In [118]:
```
cropped = pic[0].pixel_array[:, 90:270]

plt.imshow(cropped,cmap=plt.cm.bone);
```

2) Submission 2: Non-Technical Report

Problem Statement:

The application we have created is about spinal problem (Spinal Stenosis). Spinal stenosis is a narrowing of the spaces within your spine, which can put pressure on the nerves that travel through your spine. It can be detected through MRI of your spine and what doctor's understand of that MRI. We came up with an application/model which take your MRI as an input and process it and predicts if you have spinal problem or not and point out the areas where you have problem. So even a normal person who doesn't have much knowledge of spinal diseases can diagnose his/her MRI through our application. It provide ease to a doctors and a patients as doctors doesn't have to point out the problems as they are pre-defined through our application and save a lot of time for a bigger amount of patients.

Dataset:

The dataset we used for our project is taken from (https://data.mendeley.com/datasets/k57fr854j2/2). The dataset contain about 500 patients MRI's and each patient's folder contained of MRI's from different views/sides. Each view has about 15 to 25 MRI files. We converted each MRI file into jpeg format and then reduced its size by cropping and other methods so that it is easier and faster for machine to learn.

Training:

We trained our model on the basis of above mentioned dataset. We took each MRI and examine it by our understanding of what we learned through internet and other similar topics. We point out in each MRI where it has problem and where it does not, then we trained our machine on the selected MRI's that we have diagnosed.

Testing/Accuracy:

After machine's training is completed we gave it some MRI's (that are not used in training) to test that is it detecting something or it does not detect anything at all. If it detects then at what of level of accuracy it is catching a problem. After testing it is providing results with marked areas where a problem is and where it is not with about 85% of accuracy.

Input	Output
 *Input*	 *Output*
This is the MRI image that was given as input.	This is the processed image with marked areas where problem exist and where it does not. Purple squares define area with problem whereas orange squares define area with no problem.

Appendix B: User Manual

Guide to use SPINAL SCANNER (GUEST):

1. You got 2 buttons on the first screen. GUEST ADMIN
2. If you want to test a single image click on GUEST
3. It will take you to the next screen where you can select your image and press TEST button.
4. It may take a while depending on your CPU processing.
5. After it finished working it will display your RESULT IMAGE.

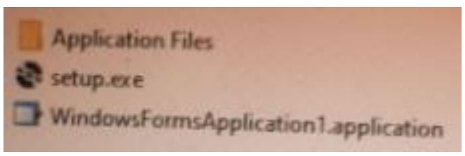
Guide to use SPINAL SCANNER (ADMIN):

1. To use admin option, you need to be an authentic user.
2. Click on the admin button, you need to login first.
3. Clicking next where you to the next screen.
4. Clicking on UPLOAD button will open result images folder where you can paste your images.
5. Clicking NEXT button will start prediction you need to wait for a while.
6. Clicking on RESULTS button will open the location of your result images folder.

Appendix C: Installation/Setup

- Download ANACONDA from the following link.
  https://docs.anaconda.com/anaconda/install/windows/
- Setup Python 3.7
- Open Command Prompt.
- Tensorflow 1.13.1 (pip install tensorflow==1.13.1)
- Keras 2.0.3 (pip install keras==2.0.3)
- Numpy (pip install numpy)
- Pandas (pip install pandas)
- Opencv-python (pip install opencv-python)
- Matplotlib.pyplot (pip install matplotlib)
- Hdf5 (pip install hdf5)

APPLICATION SETUP:



Clicking on the above setup.exe file will install SPINAL SCANNER to your pc.

Appendix D: User Interface

INTERFACE:

The interface is designed simple and easy to use for any personnel. It consists of two parts guest and member. Guests can only use one picture at a time for prediction and member once registered can able to perform prediction on as many images as they can with time for predictions in mind.

Loading Screen

Figure: 5.1

Main Menu

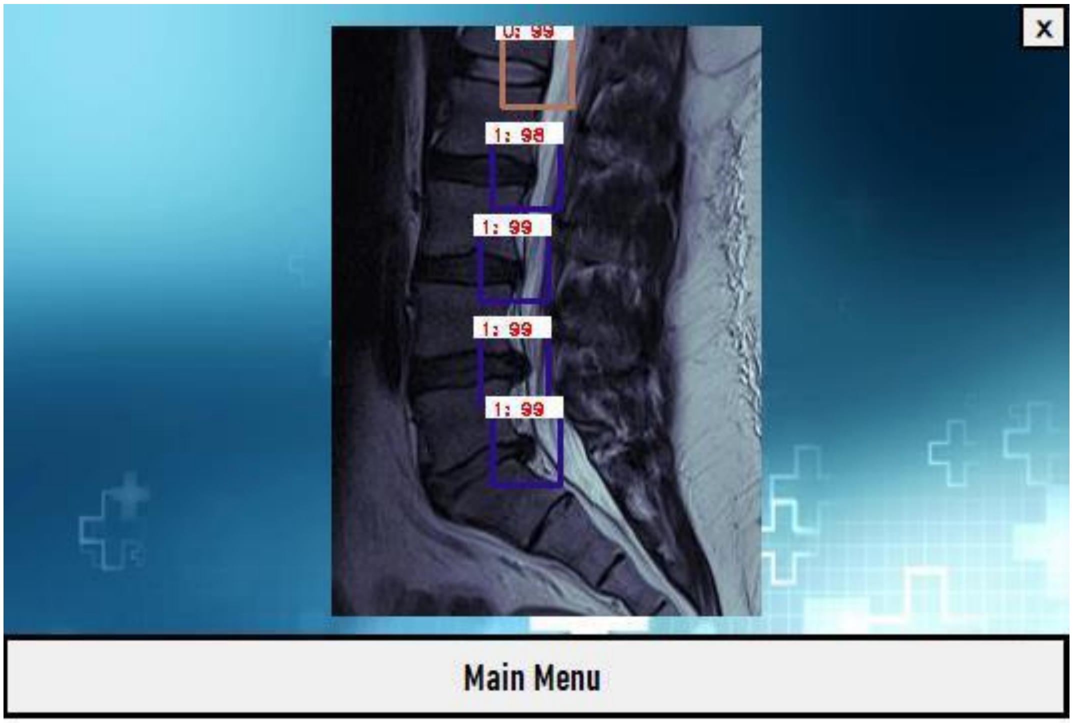
Figure: 5.2

❖ Guest button for single image at a time prediction and Member for multiple images at a time.

Guest Form 1

Figure: 5.3

❖ Browse the image you want to detect problem and press test button to begin testing.



Guest Form 2

Figure: 5.4

❖ This is the final predicted image.

Login Form

Figure: 5.5

❖ After clicking on member, you must have an account or have to register yourself to proceed.

Member Form 1

Figure: 5.6

❖ Now upload multiple images into the folder in the prompt by clicking on upload button.

Member Form 2

Figure: 5.7

❖ Now click on test to begin prediction.

Member Form 3

Figure: 5.8

Now after console is closed click of Result and it will direct you to the folder in which predicted images are stored

Appendix E: Necessary Code Snippets

Note: All code snippets belong to our team.

Crop, Save & Normalize Image:

```
In [102]: def crop_(j):
              return j[50:345, 125:340]
```

```
In [103]: # Save Images Fn
          def save_(img,name,path):
              loc = os.path.join(path, name)
              norm = (img - np.min(img)) / (np.max(img) - np.min(img))
              matplotlib.image.imsave(loc,crop_(norm),cmap=plt.cm.bone)
```

```
In [104]: f_path = []
          p_name = []
          #path = 'C:\\Users\\Hamza\\Downloads\\Compressed\\MRI_Data\\01_MRI
          for name in os.listdir():
              if name.endswith(".jpg"):
                      full_path = os.path.join(path, name)
                      f_path.append(full_path)
                      p_name.append(name)
```

```
In [106]: name = 1
          sv_path = 'C:\\Users\\Hamza\\Desktop\\Crp_Imgs'
          for i in f_path:
              name = name + 1
              save_(mpimg.imread(i),str(name)+'.jpg',sv_path)
```

Convert csv to txt:

```
data = pd.DataFrame()
data['format'] = train['image_names']

# as the images are in train_images folder, add train_images before the image name
for i in range(data.shape[0]):
    data['format'][i] = data['format'][i]

# add xmin, ymin, xmax, ymax and class as per the format required
for i in range(data.shape[0]):
    data['format'][i] = data['format'][i] + ',' + str(train['xmin'][i]) + ',' + str(train['ymin'][i]) + ',' + str(train['xmax'][i

data.to_csv('annotate.txt', header=None, index=None, sep=' ')
```

Single Image testing:

```
fig = plt.figure()

#add axes to the image
ax = fig.add_axes([0,0,1,1])

# read and plot the image
image = plt.imread('train_images/5.jpg')
plt.imshow(image)

# iterating over the image for different objects
for _,row in train[train.image_names == r"C:\Users\Osama\Desktop\Final Year Project\keras-frcnn\train_images\5.jpg"].iterrows():
    xmin = row.xmin
    xmax = row.xmax
    ymin = row.ymin
    ymax = row.ymax

    width = xmax - xmin
    height = ymax - ymin

    # assign different color to different classes of objects
    if row.class_type == 1:
        edgecolor = 'r'
        ax.annotate('P', xy=(xmax-20,ymin+20),color = 'yellow', weight = 'bold')
    elif row.class_type == 0:
        edgecolor = 'g'
        ax.annotate('N', xy=(xmax-20,ymin+20), color = 'yellow', weight = 'bold')

    # add bounding boxes to the image
    rect = patches.Rectangle((xmin,ymin), width, height, edgecolor = edgecolor, facecolor = 'none', lw=3)

    ax.add_patch(rect)
```

For output refer to figure number 4.3.

Confusion Matrix:

Confusion Matrix

In [3]:
```
df.loc[(df['Class'] == 1) & (df['Class_R'] == 1) , 'TP'] = 1
df.loc[(df['Class'] == 0) & (df['Class_R'] == 1) , 'FP'] = 1
df.loc[(df['Class'] == 1) & (df['Class_R'] == 0) , 'FN'] = 1
df.loc[(df['Class'] == 0) & (df['Class_R'] == 0) , 'TN'] = 1
```

In [4]:
```
tp = int(df['TP'].sum())
fp = int(df['FP'].sum())
fn = int(df['FN'].sum())
tn = int(df['TN'].sum())
print('TRUE POSITIVE :',tp)
print('TRUE NEGATIVE :',tn)
print('FALSE POSITIVE:',fp)
print('FALSE NEGATIVE:',fn)
```

```
TRUE POSITIVE : 34
TRUE NEGATIVE : 51
FALSE POSITIVE: 12
FALSE NEGATIVE: 3
```

Accuracy, Precision & Recall:

Accuracy:

In [8]:
```
#ACCURACY:
Accuracy = (tp + tn)/(tp + tn + fp + fn)
print("ACCURACY: {0}%".format(int(Accuracy*100)))
```

```
ACCURACY: 85%
```

Precision:

In [9]:
```
#PRECISION:
Precision = tp/(tp + fp)
print("PRECISION: {0}%".format(int(Precision*100)))
```

```
PRECISION: 73%
```

Recall:

In [10]:
```
#RECALL:
Recall = tp/(tp + fn)
print("RECALL: {0}%".format(int(Recall*100)))
```

```
RECALL: 91%
```

Appendix F: Communication with the SUPERVISOR

Mail 01 By Ma'am Humera:

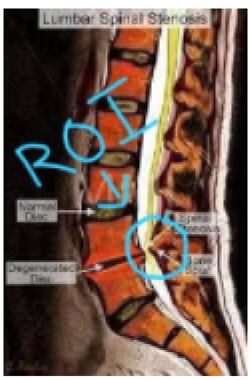
Humera Tariq <humera@uok.edu.pk>
to nabeel.imran852, chandanlalan123, me

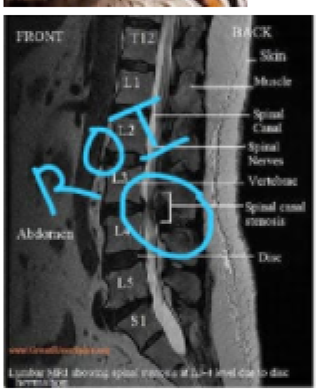
Here are some ideas for FYP. You may discuss if any idea from given list works for you.

- Hand gesture recognition ----------> robotic arm, swipe action controller, dumb/deaf action to text mapping
- Visual Search Engine / Image Retrieval System --------> image search, business forecast, brand ranking
- Multimodal Biometric Facial Feature Fusion based on Periocular Region, and Attentional Feature-Pair Relation Network (AFRN)
- Diabetic Retinopathy
- Glaucoma Diagnosis
- Website for Data Scientist (GPU sharing, data uploading, jupyter, .py files integration, file sharing)
- Human Spine gap measurement from X-ray/MRI
- Symmetry base biomedical image compression
- OpenGL/ Graphics Animation Player for web (C++)

Mail 01 by our team:

Assalam O Alaikum Ma'am.
Miss humne apke pas visit kia tha **FYP** k hawale say apnay **SPINAL CORD** mai gaps ko explore kerne ko kaha tha/anatomy.
Humne select ki hai <u>Disease (SPINAL CANAL STENOSIS (Gap between Vertebrae))</u> ki detection.





SPINAL STENOSIS DISEASE:
https://www.mayoclinic.org/diseases-conditions/spinal-stenosis/symptoms-causes/syc-20352961
UNDERSTANDING MRI:
https://healthcareextreme.com/how-to-read-your-spine-mri-study/
So Ma'am we are interested in doing this project can you please further guide us in making progress.
Jazak'Allah.

Mail 02 By Ma'am Humera: (TASK # 01)

You may start with finding an appropriate dataset to know that:

(0) Anatomy of spine as you share the links with me
(1) How to load a data set ?
(2) Show and display images from you own application
(3) certain pre processing like image resizing, normalization

Please go ahead if you are interested and update me about your progress.

https://data.mendeley.com/datasets/k57fr854j2/2

Regards,

Dr. Humera Tariq
Professional Member ACM, Member SIGGRAPH,
NVIDIA DLI Certified Instructor (USA)
Assistant Professor, University of Karachi,
Karachi, Pakistan
acm's women in computing
acm-w

Mail 02 By our team: (TASK # 01)

On Wed, Apr 8, 2020 at 2:32 AM Hamza Temuri <hamza.temuri97@gmail.com> wrote:

Assalam O Alaikum
Ma'am we tried our best to complete task you've given to us. All related code snaps and anatomy of spine are attached in this mail. We have little bit concern in image pre processing task if you can elaborate, we did as much we understood. Data loading, reading and displaying image tasks are completed.

Can you please further guide us in making progress.
Jazak'Allah.

Mail 03 By Ma'am Humera: (TASK # 02)

TASK 2:

(1) Batch Processing: Handling at-least 1000 images at once for
- **Loading 1000 images from one folder and store them into other**
- **Resizing**
- **Renaming**
- **Normalization**
- **Display as thumbnails / kind of image gallery**
- **Split 1000 images into batches of say 50 or you can perform random split**

(2) Filing, selection and cropping:
- **Save valid coordinates/parameter of rectangle into text file**
- **Show rectangle on image**
- **Crop selection and store them in another folder.**

(3) Start designing mockup for desk-top OR web based application & send wireframe design to me. This clearly helps you in doing input-output analysis.

Hope this helps.

Regards,

Dr. Humera Tariq

Mail 04 By our team: (Progress Report)

Progress Report

Inbox ×

osama jamil <osamajamil45@gmail.com>

to Humera

SPINAL CANAL STENOSIS:

Assalam O Alaikum Ma'am.

1) First we spent time reading PDFs about Spinal Stenosis.(Spinal Stenosis.jpg).
2) Then 100 out of 600 patients ki images alag ki jinme hamein laga k inko spinal stenosis hai.
3) Images ki pre-processing ki crop,resize,normalize. (Pre-Processing.ipynb)
4) Un tamam images ko label kia aur unke annotations csv file mai save karaye. (train.csv)
5) Annotations use kerke image pe rectangle banaye. (Ground truth.jpg / Frcnn.ipynb)
6) UI ki implementation in progress hai. (UI Folder)

Ab hum in images ko Faster RCNN pe chalany ki koshish kerrhe hain.

Google Drive Link:

https://drive.google.com/drive/folders/1SiUbOTd2U4gvG-wsXU2kxoHmzfkEhV2O?usp=sharing

Mail 04 By Ma'am Humera:

Humera Tariq <humera@uok.edu.pk>

to me

Keep working. Thanks for sharing progress. Will discuss with you by the end of this month.

Regards,

Dr. Humera Tariq
Professional Member ACM, Member SIGGRAPH,
NVIDIA DLI Certified Instructor (USA)
Assistant Professor, University of Karachi,
Karachi, Pakistan

acm's women in computing
acm-w

Mail 05 By our team:

Final Year Project Inbox ×

osama jamil <osamajamil45@gmail.com>
to Humera

SPINAL CANAL STENOSIS:

Assalam O Alaikum Ma'am.

1) Faster Rcnn ki implementation ki website ki guide say. Unke code ko apne model k according set kiya aur training kerwai.
2) Model train main 60hrs lagay.
3) Test images select ki 20 aur uske csv save karai. Than model say test kerwaya. (test.csv)
4) Testing k bd images Result image k folder mai save karai (Result_Images)
5) UI banai using python tkinter library, but we want to switch it to windows form c#. we found that tkinter is not a flexible library.

Ab hum UI c# pe implement ker rhe hain.

Google Drive Link:

https://drive.google.com/drive/folders/1x9WWHnpRv4UymzQAIzFXa2OETsqHvPc1

Results we got after testing.

Mail 05 By Ma'am Humera:

Humera Tariq <humera@uok.edu.pk>
to me

Sounds impressive from points in your email.

--

Regards,

Dr. Humera Tariq
Professional Member ACM, Member SIGGRAPH,
NVIDIA DLI Certified Instructor (USA)
Assistant Professor, University of Karachi,
Karachi, Pakistan

acm's women in computing
acm-w

Humera Tariq <humera@uok.edu.pk>
to me

Along with good results also keep track of bad results as well.

Record results for a group of pics as well.

Mail 06 By our team:

Spinal Stenosis UI Inbox ×

osama jamil <osamajamil45@gmail.com>
to Humera

Assaalam o Alaikum Ma'am:
Project: Spinal Stenosis
User Interface

3 Attachments

3.mp4 2.mp4 1.mp4

Mail 06 By Ma'am Humera: (APPRECIATION)

Humera Tariq <humera@uok.edu.pk> Sat, Aug 22, 12:40 PM
to me

Dear students,

I am really happy to see the spirit of working within you. As a supervisor, I told you that your work is too worthy if we are capable of adding or relating it to some existing work or domain expert.

Attached is a 2018 work and I request for a 2020 paper as well through the research gate as well. Your UI is fine as a prototype and shows your understanding of work flow as well.

Kindly pick a few points from attached paper for e.g . grading system , dice coefficient and tables to tabulate results. Let me know if you have any questions.

Regards,

Dr. Humera Tariq

Mail 07 By Ma'am Humera:

Humera Tariq <humera@uok.edu.pk>
to me

Just document it formally at your best and finish the project.

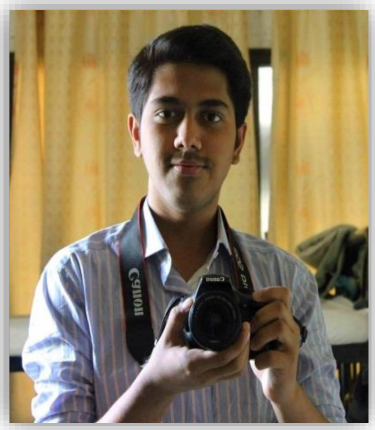
Regards,

Dr. Humera Tariq

Appendix G: Team Profile and Work Distribution

Profiles:

Muhammad Moiz Khan
Contact # 0336-8334399
LinkedIn: https://www.linkedin.com/in/moiz-khan-134a88154/



Muhammad Osama Jamil
Contact # 0333-3175394
LinkedIn: https://www.linkedin.com/in/osama-jamil-490748194/

Muhammad Hamza Temuri
Contact # 0314-9697081
LinkedIn: www.linkedin.com/in/hamza-temuri-21b698137

Work Distribution:

M Osama Jamil:

- Team lead PYTHON development. 50% work of AI part
- Final User Interface contribution Idea & Design 40%.
- Contributed in documentation 30%.
- Excel Annotations TRAIN TEST contributions 33%.
- 50% work of all technical work.

M Hamza Temuri:

- PYTHON development. 50% work of AI part
- Final User Interface contribution 40%.
- Contributed in documentation 30%.
- Excel Annotations TRAIN TEST contributions 33%.
- 50% work of all technical work.

Note: Equal work done by M Osama Jamil & M Hamza Temuri in almost every work.

M Moiz Khan:

- First GUI on Python TKINTER (Rejected)
- 2nd GUI on Windows form contributed 20%.
- Major part of documentation and compiling all work almost 40%.
- Helped a lot in excel annotations contributed about 34%.
- Extraction of 100 patients of data out of 550.
- Contributed in other then technical work.