SVKM'S NMIMS Deemed to be UNIVERSITY | SCHOOL OF MATHEMATICS, APPLIED STATISTICS & ANALYTICS

Nilkamal School of Mathematics, Applied Statistics & Analytics

DEPARTMENT OF STATISTICS

M.Sc. Statistics and Data Science

Research Treatise – III

Semester – III

Medical Image Analysis through Deep Learning and Report Generation

Group number- 12

- Neha Bahl A003 - 86062300048
- Gaurav Jangid A024
- Garima Joshi A027
- Isha Jadhav A072 - 86062200047

PROJECT GUIDE: Dr. Kavita Jain

Acknowledgement

To all those who have helped in wrapping up this project and especially our respected professors, students of NMIMS Nilkamal School of Mathematics Applied Statistics and Analytics, MSc Statistics and Data Science, Semester III.

First and foremost, we would like to present our dearest respect and gratitude to our mentor, Dr. Kavita Jain, for her inestimable guidance, knowledge, inspiration, and unyielding support down the course of this project. Without her affectionate help and expertise, no one could have undertaken this project.

We would gratefully like to thank our RT Professor, Dr. Pradnya Khandeparkar, for her inspiration and encouragement from the concept to completion of this project.

We certainly express our gratitude to our Dean, Dr. Sushil Kulkarni, for his encouragement and provision of this opportunity to work on such an insightful and challenging project.

We extend our thanks to SVKM's Library for permission to access all these vital research papers, datasets, and study materials without which this project would not have been given the light of day.

Finally, we would like to thank our faculty, parents, and friends for supporting and encouraging us in the completion of this project.

Table of Contents

Motivation

Spinal health is important for overall well being, and spinal stenosis is one of the most common musculoskeletal disorders, especially in the elderly. This condition can cause extreme pain, discomfort and decreased quality of life, and in severe cases may require surgery. Sedentary lifestyle including desk-jobs and no physical activity is one of the main causes of degrading spinal health all over the world. With the increasing incidence of spinal cord injury, this has become a major public health issue, especially with the increasing global ageing population.

Current diagnostic methods such as physical examination and conventional imaging techniques can be subjective and time-consuming. Furthermore, manual inspection of medical images often relies on the expertise of expert radiologists, which can lead to discrepancies and delays and consequently as an accurate, effective and practical method itself is crucial for identifying early spinal cord damage.

The aim of our work is to apply deep learning techniques, specifically the latest EfficientNet deep learning model, to classify spine MRI images to help detect damage in different ways. We will try to help the healthcare professionals diagnose the disease properly in time through a robust and interactive algorithm by developing a classification system.

We might hope to contribute to the constantly expanding state of medical image research by introducing this work and create an accurate instrument for early detection of spinal cord damage.

Abstract

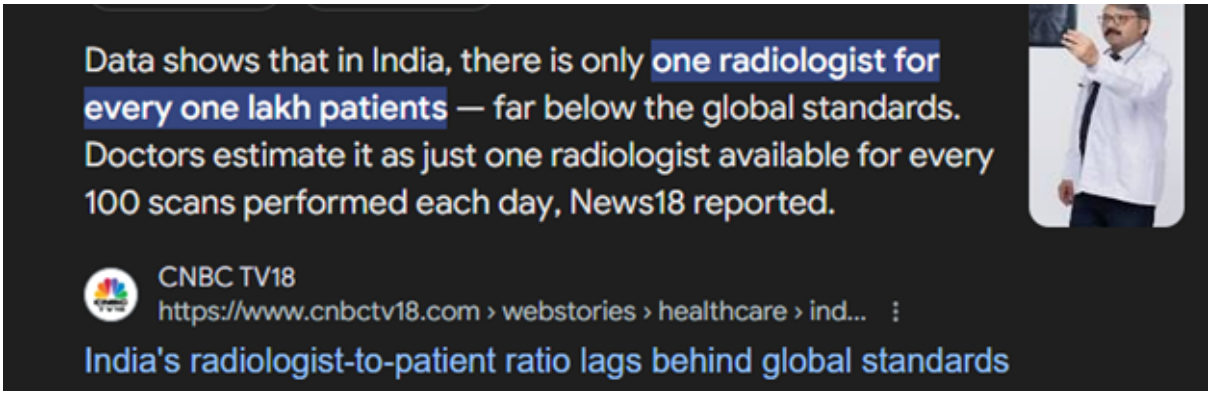
The project introduces an approach that presents a deep learning method for automatically classifying the process of lumbar spine degeneration using MRI images. The efficiency of models of deep learning, which the work considers, applies to its proper handling of medical image data in creating the classification system. Its ability to classify and grade the various conditions of lumbar spine degeneration will determine the model's performance. The results show that the model of EfficientNet is good in classification with adequate reliability, thus helping the traditional manual diagnostic methods. The developed methodology can effectively assist healthcare professionals with early detection of lumbar spine degeneration. It leads to improved diagnosis and, consequently, planning of treatment.

Introduction

Project significance Spinal disorders form one of the significant and formidable health concerns, which are affecting populations all over the world, and in the very context of India itself, it has unique variations and challenges associated with these conditions. Of late, there has been a visibly growing number of elderly people coupled with lifestyle trends drastically different from the modes of previous generations. Such a combination has caused an alarming increase in spinal conditions, which pose severe risks to the lives and health of many individuals. Medically, issues such as foraminal narrowing, subarticular stenosis, and canal stenosis contribute towards chronic pain and a reduction in movement for those that suffer.

The window for diagnosis is critical and, as such, quite short and limited in time; there are nonetheless major and significant barriers which exist to prevent good recognition and effective treatment of the complex conditions within the framework of the Indian health care system.

Challenges in India's Health Care



One of the severe challenges the medical department is actually experiencing today is the seriously inadequate number of skilled radiologists, especially in the rural areas where health care services are usually sparse. For rural areas in particular, it usually takes an extremely long period for diagnosis and subsequent treatment to be carried out, which means that delays in patient care may ensue and complications may further arise. Even in most localities with the availability of radiologists, the MRI scan interpretations and results may be utterly different, highly varying from one practitioner to another, which serves to confuse patients' results.

It is at such times, in some places, when the burden and load of the radiologist's caseload get too overwhelming, sometimes losing sight of small yet minute details. Such oversights have the capacity to make things much worse for patients in their suffering and to make matters even more complicated than ever in their treatment journeys. Thus, the objective of this project will be developing and providing a strong, scalable solution ensuring streamlining of diagnoses while making sure that nationwide consistency is maintained all across India.

Objectives

- **Condition and Severity Prediction:** The main goal of this project is to develop a deep learning model that can identify and classify various spinal conditions. It can accurately detect disorder conditions using MRI images. In addition, the models predict the severity of these conditions at vertebrae levels (from L1/L2 to L5/S1). Its purpose is to assist in the early and accurate diagnosis of lumbar spine disorders, allowing for timely treatment.
- **Deep Learning Architecture Comparison:** This project compares two deep learning architectures: ResNet and EfficientNet. The comparison is carried out to classify spinal conditions from MRI images by training the model on the same data set to attempt to infer the architecture through which the most accurate and efficient detection as well as classification of the severity of the state of the spine can be attained.
- **Generation of clinical report:** To give added value to the deep learning models. We coded the generation of clinical reports depending on the outputs coming from the model. The outcome includes the conditions found, its grade, and which spinal level they are reported at. This helps in involving a minimum amount of manual effort in making reports. Improved process for diagnosis and will provide timely results that are accurate, standard, and easy to interpret for a health worker.

Literature Review

In Yoo et al.'s paper "Deep Learning Model to Identify and Localise Degenerative Changes of the Lumbar Spine Using MRI" (2021), an attempt is made in applying a deep learning model that can identify and classify the degenerative changes seen in the lumbar spine via an MRI image, since there remains the issue of lengthy and subjective interpretations of MRI scans. With deep learning, detection and classification became quite automated and may step up the diagnostic efficiency and accuracy in a clinical setting.

Key Takeaways from the Article

Deep learning application in medical imaging.

This paper handles the expansion role of deep learning, such as automation of medical image analysis, particularly in musculoskeletal pathologies, like lumbar spine degeneration. The authors applied a deep learning model for detecting whether there are degenerative changes and their localization on MRI, with very promising results concerning accuracy and reliability. This piece focuses on the area that deep learning models could actually end up replacing the traditional diagnostic ways because they guarantee consistent and automated results for radiologists in diagnosis.

Classification of Lumbar Spine Degeneration

The study focused on specifically identifying conditions like disc degeneration, spinal stenosis, and other abnormalities in the lumbar spine. The study showcased the capacity of deep learning models to identify different phases of degeneration across more than one spinal level. Having such a large number of images available, the authors were able to train a model that recognized patterns and localised specific degenerative conditions with minimal human interaction.

Challenges for Deep Learning Models

The paper discusses heterogeneity in the evaluation of diagnostics, which, for the most part, depends on the radiologist's experience and expertise. In the development of the automated solution of the study, the aim was to minimise interobserver variability and improve consistency in the diagnosis obtained. Additionally, the authors discussed techniques related to data preprocessing, model training, and hyperparameter tuning technique, which are important for achieving accurate models as well as ensuring their generalizability.

Adapting the Methodology: Applying ResNet and EfficientNet

Inspired by the approach of Yoo et al., we developed a similar deep learning-based project to classify lumbar spine degeneration and related conditions. However, our approach includes a comparative analysis of two advanced architectures—ResNet and EfficientNet—to determine the optimal model for this application.

Overview of the clinical problem

Degenerative spine conditions adversely affect people’s quality of life. Detecting these conditions is crucial for determining therapeutic plans for patients. Therefore, it is essential to develop methods for detecting and assessing the severity of degenerative spine conditions on imaging.

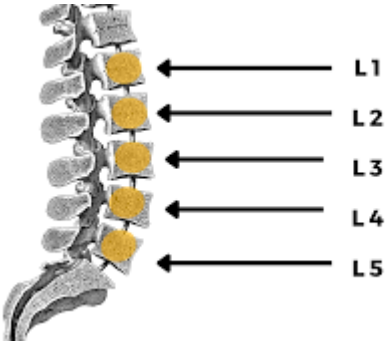
This challenge primarily focuses on identifying three types of conditions in the lumbar region of the spine (refer below for the anatomical overview).

The three conditions we aim to assess are:

1.Foraminal narrowing (on either the left or right foramen at a specified level).
2.Subarticular stenosis (on either the left or right side at a specified level).
3.Canal stenosis (only at a specified level).

Each of these conditions can manifest at various levels within the spine itself, specifically at each vertebral disc (e.g., L4/5 corresponds to the vertebral disc between the L4 and L5 vertebral bodies).

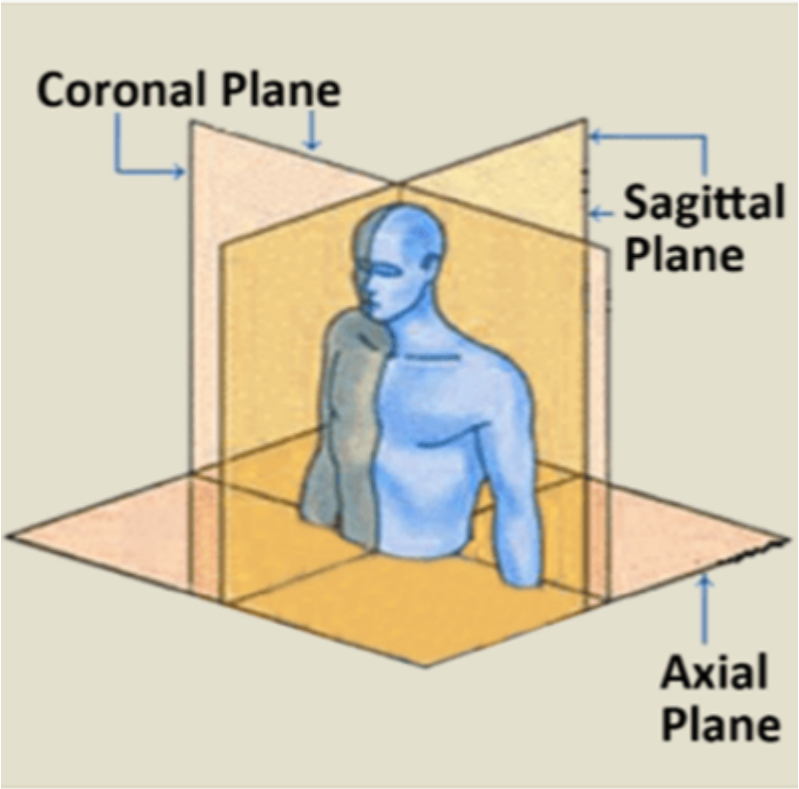
For each of the conditions, you'll need to predict whether the degree of compression is normal/mild, moderate, or severe. You can refer to the example test submission sample_submission.csv to get a better idea for what we're looking for in terms of output. For each case, you'll have to output a score from 0 to 1 representing the probability of the patient having a specific grade (normal_mild, moderate, severe), at the spinal level (l1_l2, l2_l3, l3_l4, l4_l5, l5_s1), for that condition (spinal_canal_stenosis, left_neural_foraminal_narrowing, right_neural_foraminal_narrowing, left_subarticular_stenosis, right_subarticular_stenosis):



The Lumbar is classified into 5 sections labelled as L1- L5 vertebrae. Each of these sections can induce specific symptoms that could affect lower back pain.

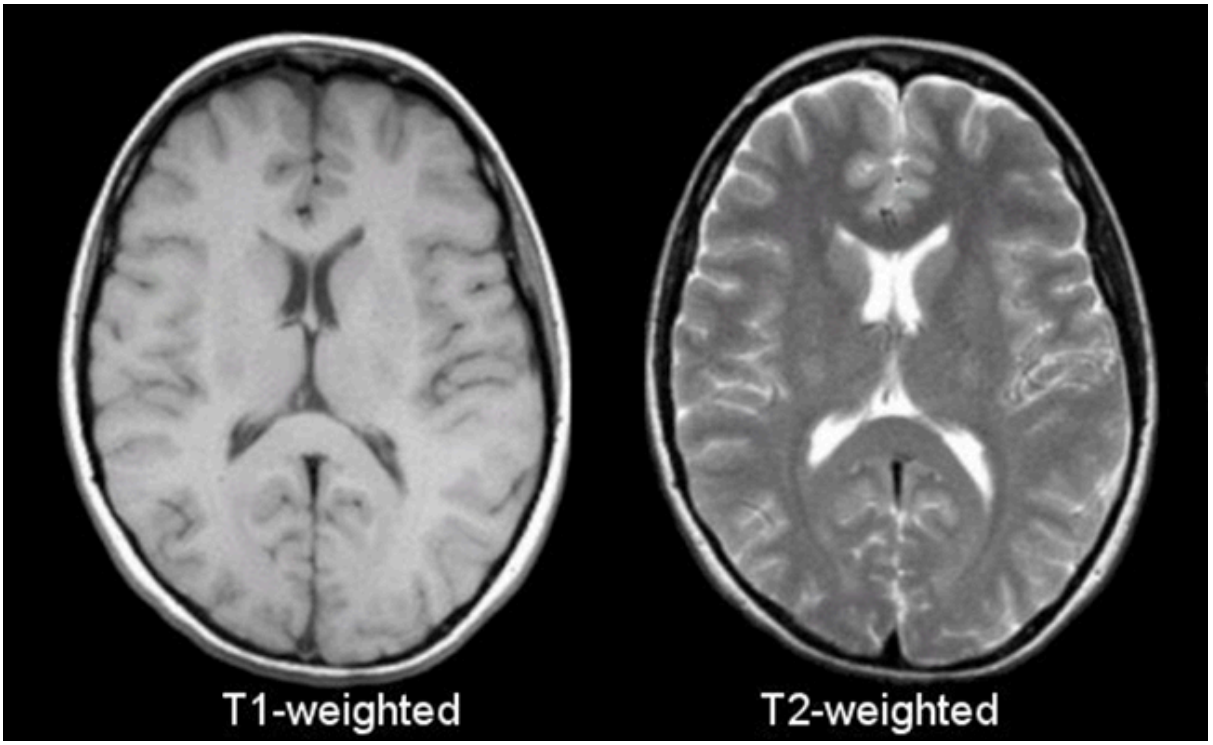
MRI Imaging Overview

MRI imaging of the spine can be taken in three planes: the axial plane, the sagittal plane, and the coronal plane.



The two main image types we'll need for this problem are the axial and sagittal planes. The axial plane takes images of horizontal slices (perpendicular to the spine) across the body from top to bottom. The sagittal plane takes vertical slices (parallel to the spine) going from left to right.

MRI images come in multiple variants. They can generally be classified as either being T1 weighted or T2 weighted. T1 weighted images show fat as being brighter. The inner part of bones would appear brighter on T1 images. T2 images show water as brighter. The spinal canal would appear as bright on T2 images. MRI images are not standardised with regards to the pixel values that are output from it (unlike CT images). So you'll need to figure out how to standardise these images (or maybe you wont need to at all, we'll leave it up to you).



Data Description

Dataset Source
-https://www.rsna.org/rsnai/ai-image-challenge/lumbar-spine-degenerative-classification-ai-challenge

In this project, we are predicting the severity of certain spinal conditions at different vertebral levels using MRI scans. The dataset is split into multiple files that provide different layers of information, which, when combined, allow us to create a training pipeline. Here, we explore how the three primary files are related:

1. train.csv

This file provides the severity labels for the different spinal conditions across multiple vertebral levels. For each study_id, we have information about conditions like Spinal Canal Stenosis or Left Neural Foraminal Narrowing at the levels L1/L2, L2/L3, etc. The severity levels are categorised into:

- **Normal/Mild**
- **Moderate**
- **Severe**

This file provides the ground truth labels, which we will use to train our model. The vertebral levels are specified for each condition, allowing us to track and classify the condition severity for each level.

2. train_label_coordinates.csv

This file gives us the precise coordinates in the MRI images where the conditions are located. For each study_id and condition, we have x and y coordinates that tell us where in the image the region of interest is located. This is crucial for focusing our model's attention on the relevant parts of the scan

Key Columns:

- study_id: Corresponds to the patient or study.
- series_id: Refers to the imaging series where the condition is located.
- instance_number: Refers to the specific slice in the MRI series where the condition is observed.
- x, y: Pixel coordinates of the region of interest (ROI).

This file tells us where the conditions are located within each scan, and we use this information to crop or focus on specific areas of the image during training.

3. train_series_descriptions.csv

This file helps us identify the type of MRI series (scan orientation or modality) used for each study_id. It provides metadata for the different MRI series, such as whether the scan is Sagittal T1, Sagittal T2/STIR, or Axial T2. Each of these gives us different views of the spine, and combining them allows for a richer representation of the data.

Key Columns:

- study_id: Identifies the patient or study.
- series_id: Uniquely identifies a particular series of MRI images within a study.
- series_description: Describes the orientation or modality of the scan (e.g., Sagittal T1, Axial T2).

This file helps us load the correct MRI images corresponding to each condition and level. For example, we know from train_label_coordinates.csv where the condition is located, and we use the series_description from this file to load the relevant series for that study.

Bringing It All Together

These three files are interconnected and together form the foundation for building our training dataset:

- **train.csv** provides the labels (severity of conditions) for different vertebral levels and conditions.
- **train_label_coordinates.csv** provides the precise locations (coordinates) of these conditions within the MRI images.
- **train_series_descriptions.csv** helps us locate the correct series and modality (Sagittal or Axial views) for each study and condition.

Workflow:

Extract Severity Labels: For each study_id in train.csv, retrieve the severity labels based on the vertebral level and condition.

Retrieve Coordinates: Use train_label_coordinates.csv to obtain the coordinates of conditions within the MRI images, focusing on specific areas.

Load MRI Series: Utilise train_series_descriptions.csv to load the appropriate MRI series (e.g., Sagittal T1, Axial T2) corresponding to each study_id.

Preprocess Images: Preprocess the MRI images according to the retrieved coordinates and conditions for focused analysis.

Model Training: Train a deep learning model using the combined data of severity labels, coordinates, and processed MRI images.

Model Evaluation: Validate the model on a separate dataset to assess its predictive accuracy for severity classification.

Prediction: Use the trained model to predict the severity of conditions in new MRI images.

Generate Reports: Automatically create structured reports based on the model's predictions for clinical use.

Data Preprocessing

Data and Image Preprocessing

Spinal cord medical images are analysed with models based on deep learning, which demands preprocessing of data to serve uniformly in the format for training models followed by inference. The processes followed for preprocessing are detailed in the following steps of procedures on the images prepared for this analysis.

1. Preprocessing Dataset

The first step is the combination of data sets based on series_id and study_id, which are the IDs assigned to each image in the dataset. This can sort images by patient studies and actually group images correctly for consistent model training. It is very crucial in the field of medical imaging where one may have divided images of a patient across different studies or series.

2. Scaling Pixel Values

The pixel values from all images are standardised by making normalisation of pixel intensities to scale between 0 and 255. This scaling helps in keeping pixel values within uniform range in order to keep the model behaviours and convergences stable. After scaling, the pixel data is converted to the type uint8. This is essentially the typical data type for image data, which further optimises storage and processing.

3. PIL Conversion

The processed NumPy arrays are then converted into PIL Python Imaging Library objects for image manipulation of properties and compatibility with a variety of image processing libraries and deep learning frameworks. Using PIL objects is helpful to apply transformations in the preparation of images for further preprocessing steps.

4. Resizing All

the images are resized to a standard resolution size of 224x224 pixels. It is done to ensure uniformity in sizes of the inputs fed into the model. Many architectures of convolutional neural networks have a requirement for uniform size, and it's an essential characteristic. The standard resolution used here is that which balances efficiency in terms of computations with a standard amount of details in the structure of the spinal cord being retained.

5. Grayscale Conversion

The images are in grayscale as seen below, but they are converted to three-channel format for being aligned with pre-trained deep learning models like ResNet that expect three-channel input images. This allows grayscale information to be duplicated across three channels, thereby making it amenable to be processed simply within the model:.

6. Tensor Conversion

Last but not least, images are converted into PyTorch tensors, which are native data structures for PyTorch-based models. This means that they will be used as an input within the neural network model. Tensors format is optimised for high-performance computation on GPUs, which is important to train deep learning models on large medical images datasets efficiently.

This preprocessing pipeline ensures that images of the spinal cord are standardised so that deep learning models may learn meaningfully from the information.

Methodology

Deep Learning Models for model training

The use of deep learning models for MRI scanning, specially to detect spinal problems in patients, has led to significant transformations in the domain of medical imaging. The traditional methods of image analysis tend to be difficult due to the complex structure of the spine and sensitivity to disease alterations. In such scenarios, for increasing diagnosis accuracy, automatic segmentation and boosting image quality, deep learning has been found efficient. It helps in detecting issues like malignancies, spinal stenosis, ruptured discs by applying the concept of convolutional neural networks (CNNs). CNN models perform well at identifying the complex patterns in the datasets and thus provide accurate location and characterization of the spinal abnormality.
In our study, we have implemented the vanilla models of ResNet 18 and EfficientNetV2 on our dataset and have studied how these models perform on our data in a generalised way. Using these models will help us to understand and enhance the potential that deep learning holds in the study of spinal MRI by evaluating the performance and accuracy of complex models like EfficientNetV2 with the comparatively simpler models of ResNet18.

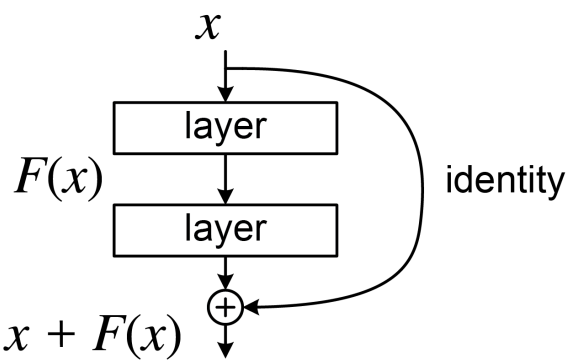
Resnet 18

Residual Function:

H(x) : a complex/underlying function we want few layers of neural network to learn
x: inputs to initial layer
F(x) i.e. difference between input and desired output.

$F(x) = H(x) – x$
Thus, original function for getting desired output becomes:

$H(x) = F(x) + x$



Identity mapping by Shortcuts: (Innovative Solution)

Identity mapping: input is equal to output i.e. function just passes through the input without making changes.

H(x) = x

Shortcut connections: skip one or more intermediate layers and pass inputs from the earlier to the later layer.

Building block: y =F(x, {Wi})+x

For given figure: F = W2*σ(W1*x)

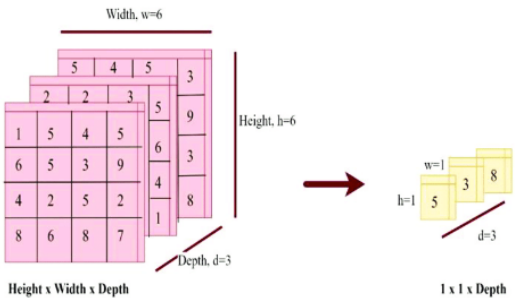
σ(): ReLU function

Here, x and F must have the same dimensions

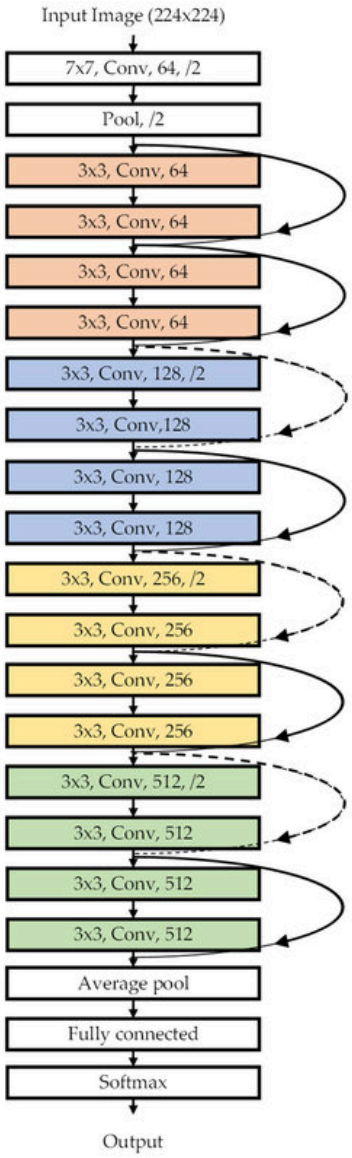
Global Average Pooling (GAP)

Global Average Pooling (GAP) is the method of summarising spatial information inside feature maps. It adds up information from every location in a spatial space across all feature maps and replaces it with a single value per channel. It is very well suited to ResNet-18 because it makes its application right after the last layer of convolution, thus leading to this: For every feature map, it generates one vector of values. These vectors are applied to the final classification task. GAP brings several potential advantages over the traditional max-pooling layer: it preserves spatial information, reduces overfitting, and increases computational efficiency.

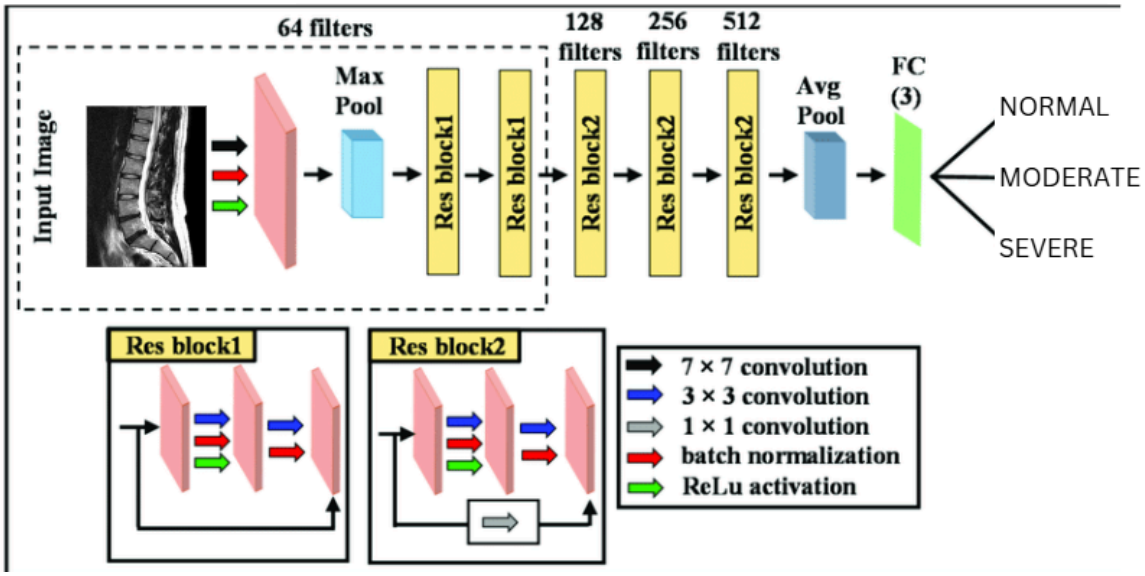
$$y = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{ij}$$



Structure-



ResNet or Residual Network is an architecture of a deep neural network in which inception is aimed at overcoming the problem of vanishing gradients in very deep networks. The first layer that the input image with size 224x224 encounters in ResNet is a first 7x7 convolution layer with 64 filters and stride 2 followed by max-pooling layers to reduce spatial dimensions. Following the image comes a series of residual blocks, each containing several convolutional layers with 3x3 filters. Skip connections or shortcuts are the defining characteristic of ResNet; this is because the network can bypass certain layers by adding the input of one layer to the output of a deeper layer. That addition helps preserve gradient flow as well as makes it possible to train networks much deeper than before. The depth of the network is increased with more blocks containing different numbers of filters 64, 128, 256, and 512 where each transition contains downsampling by convolutional layers with stride 2. After that, the output is then passed into an average pooling layer, then into the fully connected layer, then a softmax for final classification. This architecture helps ResNet to learn complex patterns efficient but keeps away the problems of depth.



EfficientNetV2

EfficientNetV2 is a new family of smaller and faster Convolutional neural networks (CNNs). It was designed to give state-of-the-art performance on image classification cases while also maintaining efficiency of the model. The model has been optimised with train-aware Neural architecture search (NAS) and model scaling and to speed up the training procedure the model implements progressive learning. The model uses MBConv (Mobile inverted residual bottleneck)blocks along with a modified MBConv block i.e. Fused MBConv blocks for further improving the training speed.

Train aware Neural architecture search (NAS):

Neural architecture search (NAS) is a technique used to automate the design of neural networks rather than manually specifying the model. This helps in finding the best or optimal configuration of the model architecture.

The traditional neural network only considers optimising the architecture design.

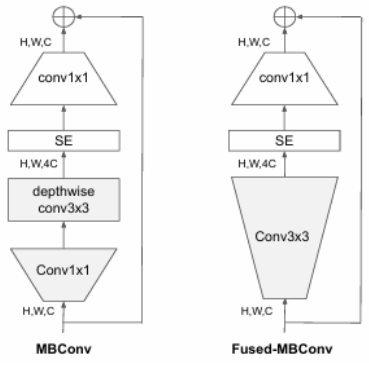
Train - aware NAS takes into consideration the training process of the neural network along with the architecture design. It thus tries to get a deeper understanding of the training dynamics like training curve, convergence speed and robustness of the model to different training conditions and how they affect the performance of the model.

The NAS consists of a search space which is a set of all possible architecture that can be explored by the architecture. For EfficientNetV2 this search space consists of different block types (MBConv, blocks, Fused MBConv blocks, squeeze and EXcitation blocks), block parameters (kernel size, filter size, stride, expansion ratios, etc), the network width, depth and resolution of input images (whether to scale these parameters up or down). The search

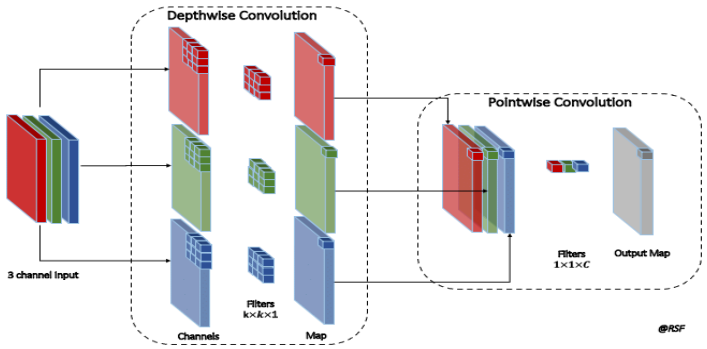
space is explored using methods like Reinforcement learning or Grid - Search to get the optimised architecture.
The training time objective is also included here along with objectives like accuracy and model size.

Objective=Accuracy−λ · Training Time
where:
Accuracy: performance of model on the validation set.
Training Time: time taken to converge during training.
λ: weighting factor that balances the trade-off between accuracy and training time.

MBConv and Fused MBConv blocks:



The Mobile inverted residual bottleneck i.e. MBConv block first goes through a 1x1 pointwise convolution to expand the number of channels. Then the depthwise separable convolution is applied. Here, first a 3x3 size filter is applied on each channel to reduce the number of parameters and also the computation time and then the output goes through pointwise convolution of size 1x1 which creates a linear combination of the depthwise layer.

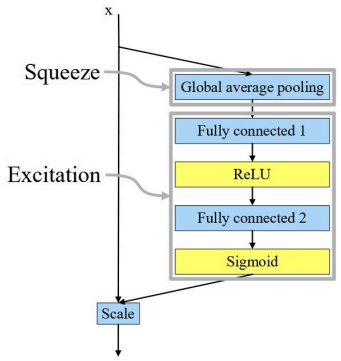


Depthwise separable convolution overall number of operations is the sum of the depthwise and pointwise convolutions:
Total Operations=H×W×Cin×(K×K+Cout)

In comparison, the traditional convolution would require:
Operations (Standard Conv)=H×W×Cin×Cout×K×K

The output of the depthwise separable convolution layer is fed to the Squeeze and Excite block where the channels present in the input to this layer are recalibrated depending on the significance of each channel. This helps the model focus on the important features and suppress the irrelevant features.

Squeeze and Excitation Block:



The SE block consists of 2 phases: 1. Squeeze phase 2. Excitation phase

1. Squeeze phase: Here global average pooling is applied to the feature map obtained as input from the depthwise convolution layer. By the global average activation, the height and width of each channel is compressed into a single value. Such values are obtained for all the channels in the input feature map and a 1D descriptor vector of size C (number of channels) is formed from these values.

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_{i,j,c}$$

where Zc is the average of channel c.

2. Excitation phase: This phase involves passing the 1D vector from the squeeze phase through 2 fully connected layers. The first layer reduces the number of channels by reduction ratio r which is a hyperparameter. This is followed by an activation function like ReLU or Swish. In most cases the Swish activation function is used.

$z' = \text{Swish}(W_1 * z)$

Where, z: output of global average pooling

W_1: weight matrix for the first fully connected layer

After this, the vector goes through the second fully connected layer to restore the number of channels back to C. Here the Sigmoid activation function is applied to get channel-wise attention weights in the range of [0,1].

$s = \sigma(W_2 z')$

where, W_2 : weight matrix for second fully connected layer.

The final output of the SE block is a 1D vector of attention weights for each channel which are multiplied element-wise in each channel of the original input feature map i.e. the feature

map that was given for global average pooling at the beginning of the SE block. This element-wise multiplication gives a feature map in which the significant channels are given more focus than the irrelevant ones.

The output of the SE block passes through another 1x1 convolution layer to restore the number of channels so that the input and output of the MBConv block has same dimensions. As the input and output dimensions match, the input is also added to the output of the block for better training of the networks and making the vanishing gradient problem less severe.

Another type of block seen in the EfficientNetV2 architecture is the Fused MBConv block. In this block, the initial 1x1 convolution layer and the 3x3 depthwise convolution layer are replaced by a single regular 3x3 convolution layer.

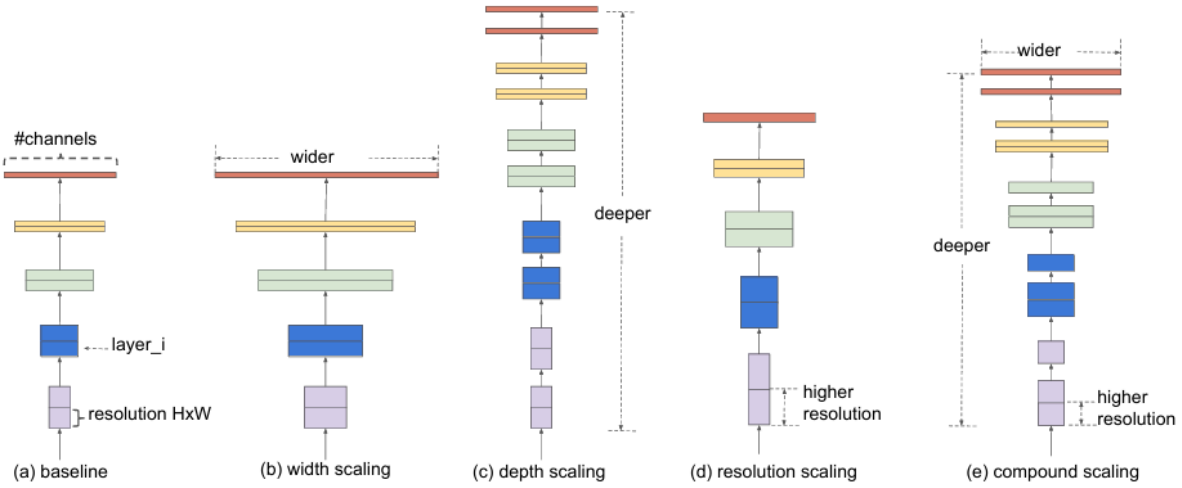
Since the Fused MBConv blocks have combined the depthwise and expansion convolution layers, it reduced the number of computations and thus helped in achieving faster convergence during training. The initial stages of any architecture process the smaller spatial resolutions, so these blocks are applied in the earlier stages of EfficnetNetV2. The later stages of EfficientNetV2 involve extracting the complex features from the data. The depth of the model increases while its spatial dimension decreases. In such situations, the MBConv blocks are more suitable.

Progressive Learning:

In many models, the image size of the data changes during the training ph
ase. These changes in the image size often lead to a drop in the accuracy of the model. The regularisation techniques are fixed here even if the image size changes. Through progressive learning, the input image resolution is gradually increased during training. The model training begins with a smaller input image which gives best accuracy with weak regularisation techniques. As the training progresses, the resolution of the images is increased and the model then performs better with stronger regularisation techniques.

Input: Initial image size S_0 and regularization $\{\phi_0^k\}$.
Input: Final image size S_e and regularization $\{\phi_e^k\}$.
Input: Number of total training steps N and stages M.
for $i = 0$ **to** $M - 1$ **do**
  Image size: $S_i \leftarrow S_0 + (S_e - S_0) \cdot \frac{i}{M-1}$
  Regularization: $R_i \leftarrow \{\phi_i^k = \phi_0^k + (\phi_e^k - \phi_0^k) \cdot \frac{i}{M-1}\}$
  Train the model for $\frac{N}{M}$ steps with S_i and R_i.
end for

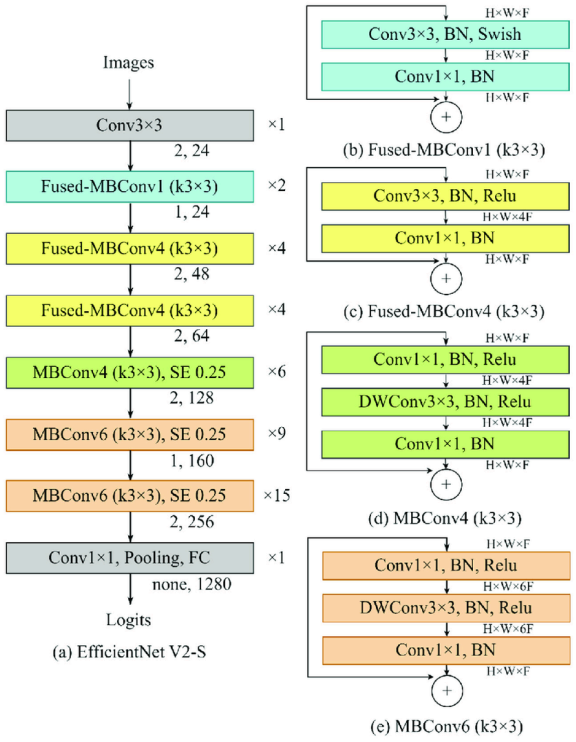
Compound scaling:



To increase the efficiency and accuracy of the model, the data needs to be resized or adjusted to meet the constraints. For this model scaling is performed. This involves resizing the image resolution or the depth of the layers or width of the channels. However, trying to adjust one of the aspects often leads to changes in the other aspects of the model. For eg., increase in the depth of layers leads to vanishing gradient problem.

This indicates that the different scaling dimensions are not independent. Thus, rather than opting for single scale dimension scaling we must opt for coordinating and balancing all the scaling dimensions to get optimised model architechture.

EfficientNetV2 Architecture:



Model Training and Optimization:

For the project, two state-of-the-art vanilla models were used- ResNet18 and EfficientNetV2-S. The two models are less computationally demanding and of lower number of parameters. These models were pretrained on ImageNet, having a thousand classes. The weights from pretrained models were used for modelling and modification was done in output layers based on the number of classes as three, since the problem infers to predict the severity of a spine condition at three levels: moderate, severe, normal/mild. The process of changing the output layers of pretrained models is defined by fine-tuning. In the optimization part models were acquainted with the ADAM Optimizer, Learning Rate Scheduler and Early Stopping. Additionally, we have three models for each View set of MRI Images (Axial T2, Sagittal T1, Sagittal T2/STIR).

Model Evaluation:

The parameter used to evaluate the model was accuracy. Validation accuracy was the yardstick for comparing performance at each epoch, and early stopping was included to halt iterations when the model dipped in accuracy-this improved time complexity of computation. Here, on the next page, a summary of the performance of the ResNet18 and EfficientNetV2-S models is provided in tabular form.

EffNetV2 single-model

Model	Epoch Count	Train Loss	Train Accuracy	Max Val Accuracy	Val Loss	Epoch with Max Val Accuracy
Sagittal T1	9	0.6149	77.70%	78.19%	0.6250	6
Axial T2	9	0.6903	71.73%	72.16%	0.6744	9
Sagittal T2/STIR	9	0.4227	87.57%	86.51%	0.4565	6

ResNet18 Model

Model	Epoch Count	Train Loss	Train Accuracy	Max Val Accuracy	Val Loss	Epoch with Max Val Accuracy
Sagittal T1	5	0.6405	76.95%	78.19%	0.6788	2
Axial T2	5	0.6834	71.75%	73.12%	0.7006	2
Sagittal T2/STIR	4	0.4586	87.63%	87.13%	0.4574	1

After training, both the model performance is checked over testing data, below is the accuracy computed over each view set (Axial T2, Sagittal T1, SagittalT2/STIR) of MRI Images.

EffNetV2 single-model

Model	Best Testing Accuracy
Sagittal T1 (EfficientNet)	76.90%
Axial T2 (EfficientNet)	71.51%
Sagittal T2/STIR (EfficientNet)	86.36%

ResNet18 Model

Model	Test Accuracy
Sagittal T1	76.81%
Axial T2	70.93%
Sagittal T2/STIR	87.63%

Results:

After carefully evaluating the model performance, Iis clear that the EfficentNetV2-S outperforms in all three MRI datasets. The model achieved 86.36% as the highest testing score on the Sagittal T2/STIR dataset. Performance by dataset: Sagittal T1: After 9 epochs, EffNetV2 achieved a maximum validation accuracy of 78.19%. Axial T2: After 9 epochs, the model achieved a maximum validation accuracy of 72.16%. Sagittal T2/STIR: Interestingly, this dataset was the hardest to train, with EffNetV2 achieving 86.51% as the maximum validation accuracy after 9 epochs. Model Based on ResNet18 ResNet18, on the other hand, performed quite decently but always lagged behind EffNetV2 in terms of accuracy. Performance by dataset: Sagittal T1: ResNet18 reached a peak validation accuracy of 78.19% after 5 epochs. Axial T2: The model achieved the highest level of accuracy of 73.12% after 5 epochs. Sagittal T2/STIR: Like EffNetV2, ResNet18 also experienced this dataset as the most difficult, with a maximum validation accuracy of 87.13% being recorded after 4 epochs.

Test Predictions:

Moving forward with the EfficientNet model we generated predictions over test dataset MRI Images and below is the table representing the same.

[68]:

	row_id	normal_mild	moderate	severe
0	44036939_left_neural_foraminal_narrowing_l1_l2	0.819712	0.146381	0.033907
1	44036939_left_neural_foraminal_narrowing_l2_l3	0.855817	0.104550	0.039633
2	44036939_left_neural_foraminal_narrowing_l3_l4	0.812699	0.142418	0.044883
3	44036939_left_neural_foraminal_narrowing_l4_l5	0.833287	0.111888	0.054825
4	44036939_left_neural_foraminal_narrowing_l5_s1	0.826767	0.136919	0.036314
5	44036939_left_subarticular_stenosis_l1_l2	0.798620	0.167362	0.034018
6	44036939_left_subarticular_stenosis_l2_l3	0.806653	0.167035	0.026312
7	44036939_left_subarticular_stenosis_l3_l4	0.778615	0.177743	0.043641
8	44036939_left_subarticular_stenosis_l4_l5	0.749834	0.200944	0.049222
9	44036939_left_subarticular_stenosis_l5_s1	0.801725	0.156951	0.041323

For each spinal condition we have severity and its probability, whichever severity level has high probability, the model predicts the same. These test results were discussed with an expert and it was found that such probability can provide great assistance to training radiologists for careful detection of underlying conditions.

Report Generation

Python codes directly utilise the model's predictions and extracts the necessary conditions and performance measures systematically. These items are then incorporated into the report in an ordered manner so that all the output of the predictions is well explained.Below is the snapshot of the code and report generated

```python
df =  grouped_submission

# Extract the condition type and area for each row
df['condition'] = df['row_id'].apply(lambda x: "_".join(x.split('_')[1:4]))
df['area'] = df['row_id'].apply(lambda x: "_".join(x.split('_')[4:]))

# Function to determine the highest severity level and probability
def determine_severity(row):
    levels = {
        'Normal/Mild': row['normal_mild'],
        'Moderate': row['moderate'],
        'Severe': row['severe']
    }
    highest_severity = max(levels, key=levels.get)
    highest_prob = levels[highest_severity]
    return highest_severity, highest_prob

# Group by condition and generate a report
print("Patient Condition Report:")
print("=" * 40)
for condition, group in df.groupby('condition'):
    print(f"\nCondition: {condition.replace('_', ' ').title()}")
    print("-" * 40)

    # Iterate over each area under the current condition
    for _, row in group.iterrows():
        area = row['area'].replace('_', ' ').title()
        severity, probability = determine_severity(row)

        print(f"  Area: {area}")
        print(f"    - Highest Severity Level: {severity}")
        print(f"    - Probability: {probability:.2%}")
        print("-" * 40)
```

65]

```
Patient Condition Report:
========================================

Condition: Left Neural Foraminal
----------------------------------------
  Area: Narrowing L1 L2
    - Highest Severity Level: Normal/Mild
    - Probability: 81.97%
----------------------------------------
  Area: Narrowing L2 L3
    - Highest Severity Level: Normal/Mild
    - Probability: 85.58%
----------------------------------------
  Area: Narrowing L3 L4
    - Highest Severity Level: Normal/Mild
    - Probability: 81.27%
----------------------------------------
  Area: Narrowing L4 L5
    - Highest Severity Level: Normal/Mild
    - Probability: 83.33%
----------------------------------------
  Area: Narrowing L5 S1
    - Highest Severity Level: Normal/Mild
    - Probability: 82.68%
----------------------------------------
...
  Area: L5 S1
    - Highest Severity Level: Normal/Mild
    - Probability: 85.19%
```

Limitations:

Although there is great promise associated with automated medical image analysis and report generation, there is a considerable number of constraints. First, the model's generalization power to other, already-unseen medical images will be hampered because of dissimilarities in imaging techniques and demographic information of the patients. Secondly, the computational costs incurred during training and fine-tuning deep-learning models are very expensive, and there quite limits scalability and accessibility. In addition, those standard report formats may not fully satisfy all the end-users; hence, a compromise must be struck. Furthermore, classical evaluation metrics may not completely account for the variances found in medical image analysis, and the interpretation of these metrics could be subjective. Lastly, the trustworthiness of the system could be limited by ethical considerations, difficulty with regulations, and the possibility of human mistakes when it comes to data quality.

Future Enhancement:

There are some several future enhancements to address the said limitations and fully realize the benefit of automated image medical analysis. These include development of a robust platform for image upload and report generation, deploying advanced deep learning models such as ResNet50 and efficient models, and improving model interpretability through explainable AI tricks. Moreover, they can seamlessly integrate with LLMs for thus increased quality and flexibility in report generation. By addressing these areas for further work, we can continue to advance the art of medical image analysis while giving valuable tools to physicians in their work.

Conclusion:

EfficientNet V2 has shown a strong promise in the classification of the lumbar spine condition severity, and has the potential to be a useful tool in efficiently diagnosing spinal disorders. This model can be used to examine medical images by using sophisticated deep learning and assist practitioners with their work. Nonetheless, more effort needs to be placed in improving its performance especially with respect to data quality and the model’s interpretability. This would enable us harness the full possibilities of AI-centred medical imaging and image analysis and enhance patient outcomes.

Bibliography