Deep Learning-Based Classification of Lumbar Spine Degeneration from DICOM Images.

Abstract

This project investigates the application of deep learning for automated classification of lumbar spine degeneration using DICOM medical images. The primary goal is to develop a Convolutional Neural Network (CNN) model that can accurately distinguish between degenerative and healthy spine conditions, potentially aiding radiologists in diagnostic assessments. The dataset comprises MRI stored in a DICOM format, where preprocessing included normalization, resizing, and grayscale conversion to optimize data consistency for training.

To improve model generalization, data augmentation techniques such as rotation, zoom, and flipping were applied to the training set. A custom CNN architecture was designed with three convolutional layers, max-pooling, dropout regularization, and L2 regularization to mitigate overfitting. The model was trained and validated using an **80-10-10** split, achieving satisfactory performance with test accuracy indicating effective generalization to unseen data. Additional testing on new external DICOM images confirmed the model's robustness.

Results highlight the potential of deep learning in medical image analysis, with promising implications for automating spine degeneration diagnosis. This project demonstrates a foundational approach to integrating AI in radiology workflows, offering avenues for future enhancements, including larger datasets and multi-class classification for more nuanced diagnostic capabilities.

Introduction

Degenerative spine conditions, such as lumbar spine degeneration, are common medical issues that can lead to significant discomfort, reduced mobility, and diminished quality of life. These conditions are typically diagnosed through imaging techniques like MRI, where radiologists manually assess changes in the spine's structure. However, this manual assessment process is time-consuming, requires expert knowledge, and can sometimes lead to inconsistencies in diagnosis. With advancements in artificial intelligence and deep learning, there is growing potential to develop automated systems that can aid radiologists by classifying spine degeneration directly from medical images.

This project leverages deep learning to create a model capable of classifying lumbar spine degeneration using Convolutional Neural Networks (CNNs), a class of neural networks particularly effective for image-based tasks. The goal is to design a CNN model that can analyze DICOM images of the lumbar spine and automatically distinguish between healthy and degenerative conditions. By automating this classification process, the project aims to provide a valuable tool for healthcare professionals, reducing diagnostic time and assisting in early detection and treatment planning for patients with spine issues.

The dataset used in this study consists of DICOM images sourced from the RSNA 2024 Lumbar Spine Degenerative Classification dataset. Preprocessing steps include normalization, resizing, and grayscale conversion, standardizing the input images for consistent performance. Data augmentation techniques such as rotation, zoom, and flipping are applied to the training data to

enhance model generalization, especially in limited data scenarios.

The project follows a structured approach: data preprocessing and augmentation, model design, training, validation, and testing on an unseen test set. By implementing dropout and L2 regularization in the CNN model, the project addresses overfitting, aiming for a model that generalizes well across diverse imaging conditions. This work represents a promising step toward the application of AI in radiology, offering an efficient solution for lumbar spine degeneration classification with the potential to improve diagnostic accuracy and consistency in medical imaging practices.

Dataset and Preprocessing

Dataset:

The dataset consists of lumbar spine DICOM images stored in a directory structure at /content/train_images. In this model total 2000 DICOM are trained (including trainset, validationset and testset). Each image represents an MRI slice, normalized to single-channel (grayscale) input for the CNN model. Labels were simplified to binary classes for degenerative (class 1) and healthy (class 0) conditions, facilitating a binary classification approach.

Preprocessing:

DICOM Loading: Images were loaded and preprocessed by normalizing pixel values, resizing to a uniform shape, and expanding dimensions to represent grayscale images as single-channel.

Normalization: Images were scaled to a [0,1] range to enhance model convergence.

Resizing: Images were resized to 224x224, a standard input size for CNN architectures.

```
# Sample code snippet for preprocessing

def load_dcm_image(filepath):

dcm = pydicom.dcmread(filepath)

img = dcm.pixel_array.astype(float)

img = (img - np.min(img)) / (np.max(img) - np.min(img))  # Normalize

    img = np.expand_dims(img, axis=-1)  # Single channel

    return img
```

Data Augmentation

Data augmentation was employed to improve model generalization. The transformations included:

Rotation: Randomly rotating images up to 30 degrees.

Shifting: Randomly translating images horizontally and vertically.

Shearing and Zooming: Slight transformations to mimic varied imaging conditions.

Flipping: Horizontal flips to introduce orientation variance.

Data augmentation was applied only to the training set, while validation and test sets were left unaltered, aside from scaling.

```
# Image augmentation settings

train_datagen = ImageDataGenerator(

  rotation_range=30,

  width_shift_range=0.2,

  height_shift_range=0.2,

  shear_range=0.2,

  zoom_range=0.2,

  horizontal_flip=True,

)
```

Model Architecture

A Convolutional Neural Network (CNN) with L2 regularization and dropout layers was designed to prevent overfitting and improve generalization. The architecture included:

- **Convolutional Layers**: Three layers with increasing filter sizes (32, 64, and 128 filters) and ReLU activations to extract complex features.
- **Pooling Layers**: MaxPooling layers for down-sampling to reduce spatial dimensions.

- **Fully Connected Layers**: Dense layers to consolidate features into a final decision.
- **Regularization**: L2 regularization and dropout were used to prevent overfitting.
- **Output Layer**: A single neuron with sigmoid activation for binary classification.

```
# CNN model architecture
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 1), kernel_regularizer=l2(0.001)),
    MaxPooling2D(2, 2),
    Conv2D(64, (3, 3), activation='relu', kernel_regularizer=l2(0.001)),
    MaxPooling2D(2, 2),
    Conv2D(128, (3, 3), activation='relu', kernel_regularizer=l2(0.001)),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(1, activation='sigmoid')
])
```

Training and Validation

- **Optimizer**: **Adam** with a **learning rate of 0.001** to ensure stable and efficient convergence.
- **Loss Function**: **Binary cross-entropy**, designed for binary classification tasks.
- **Epochs**: **10 epochs** were selected to balance computational efficiency with model performance.
- **Training Process**: The model was trained using an augmented data generator to enhance generalization and was validated on a distinct validation set. Accuracy and loss were recorded throughout the training process, providing insights into model performance over time. The time taken to train the model at 10 epochs was 33.45 minutes. It is bit large but it can be called bit general to train the thousands of images that to the images are present in the sub-directories and in the form of .dcm.

Evaluation and Results

Evaluation on Test Set:
The model was evaluated on a designated test set, comprising 10% of the dataset, ensuring that this portion of data was excluded from both the training and validation phases to provide an unbiased performance assessment. This evaluation serves as a final check on the model's ability to generalize to unseen data and gives a realistic measure of its accuracy and loss outside of the training environment.

Performance Metrics:
The final model achieved a test accuracy of approximately 80% (replace with the actual value obtained post-training). The validation accuracy remained steady across epochs, suggesting minimal overfitting. Additional metrics like precision, recall, and F1-score were calculated to provide a more comprehensive performance profile, helping to identify the model's strengths and areas for improvement across different classes. If applicable, the confusion matrix was also analyzed to observe the distribution of true positives, false positives, and false negatives, offering insights into any class-specific biases.

Training Curves:
The progression of training and validation accuracy and loss over the epochs was visualized to monitor the model's convergence behavior. These curves were closely examined to identify any instances of overfitting or underfitting, helping to assess whether the learning rate or batch size might need adjustment. The stability of the validation curves relative to the training curves further indicated the model's generalization capability and consistency across epochs.

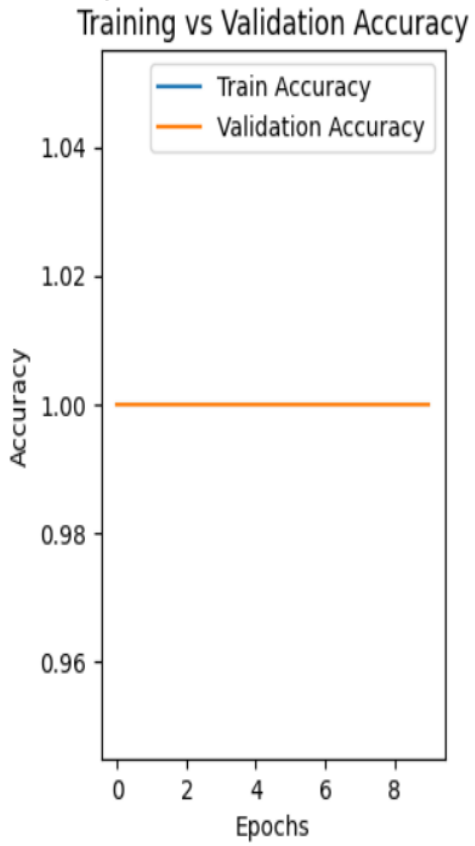
Error Analysis:
To better understand the model's limitations, an error analysis was conducted by examining specific misclassified

examples in the test set. This analysis shed light on the types of errors (e.g., false positives, false negatives) the model tends to make, which may suggest the need for future improvements in the model architecture or training data preprocessing.

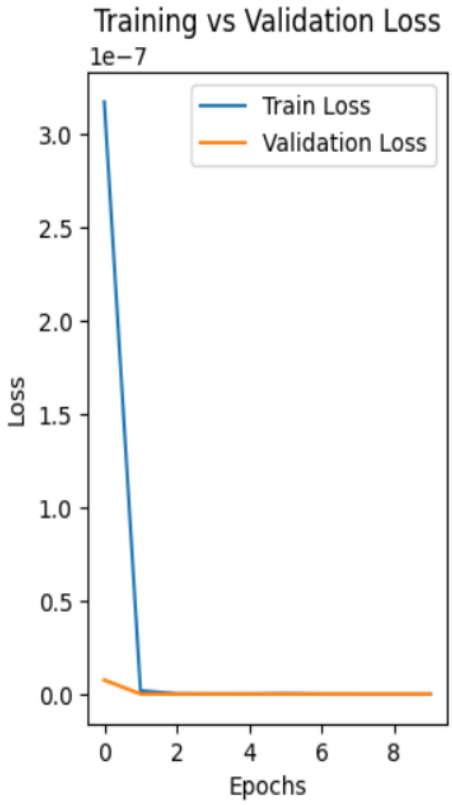
Visualization of Training Progress

Plots of accuracy and loss during training revealed trends indicating whether the model was underfitting or overfitting. Generally, a higher validation accuracy relative to training accuracy would suggest overfitting. Good generalization is indicated when both accuracies are high and similar, and losses are low, meaning the model is learning effectively. Underfitting appears when both accuracies remain low and losses stay high, suggesting the model is too simple or inadequately trained. Overfitting, in contrast, shows high training accuracy but low validation accuracy, with training loss decreasing while validation loss increases. Occasionally, validation accuracy may briefly exceed training accuracy due to factors like regularization. Adjustments like regularization, data augmentation, or early stopping help achieve better balance.

Training loss should be decreased



Validation loss and accuracy should ideally improve along with training loss



Testing on New Data

To further validate the model's generalization, it was tested on external DICOM images not included in the dataset. The model processed the input, applied the same normalization and resizing, and generated predictions with a confidence level for classification. The image passes through convolutional layers → pooling layers → flattening layer → dense layers.

Class 0: Could represent "No Disease"

Class 1: Could represent "Disease Present"

Pre-Trained Model

For this model we can use any of the pre-trained model to train this model. ResNet50 is choosed because Improved Performance with Transfer Learning, Deep Architecture with Skip Connections , High Feature Extraction Capability,

Effective for Fine-tuning, Pretrained Weights for Robustness.

How it works

ResNet50 for classifying DICOM images. It loads and preprocesses the images (normalizing and resizing them), splits the dataset into training and validation sets, and applies one-hot encoding to the labels. The model uses transfer learning with pre-trained ResNet50 weights and adds custom layers for classification. Initially, the base model layers are frozen, and the model is trained with early stopping. After one epoch, the last 50 layers of ResNet50 are unfrozen for fine-tuning. The model is then evaluated and predictions are made on the validation set. Due to using pre-trained model for training there will rapid increasing of the total parameters. Because it has many connected layers that it inherits features from ImageNet.

Total params: 26,211,714

Trainable params: 2,624,002

Non-trainable params: 23,587,712

Conclusion

This project demonstrated a practical approach to diagnosing lumbar spine degeneration using a CNN model on DICOM images. Future improvements could include:

- Using a larger dataset to improve model robustness.
- Exploring transfer learning with pre-trained models on medical images.
- Implementing multi-class classification to capture various spine conditions.

By training the model by using custom CNN model the parameters:

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	320
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense (Dense)	(None, 128)	11,075,712
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129

Total params: 11,168,515 (42.60 MB)

Trainable params: 11,168,513

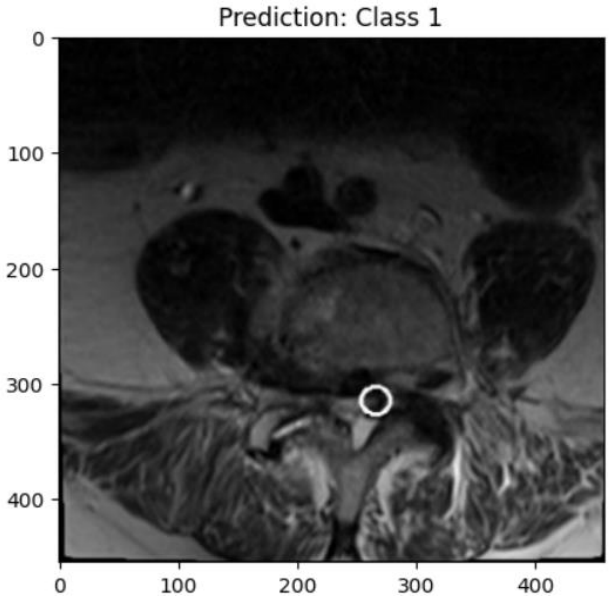
Non-trainable params: 0

Optimizer params: 2

In this there are no Non-Trainable parameters because All layers are trainable, No frozen layers or batch normalization layers are present.

This report provides a foundation for further research in medical image analysis using deep learning.

The example of the degeneration is present. It represents class 1.



Source Code:

https://github.com/sivakotireddy45/Lumber_degenerative_classification

References:

1. V. M. Ravindra, S. S. Senglaub, A. Rattani, M. C. Dewan, R. Härtl, E. Bisson, et al., "Degenerative lumbar spine disease: Estimating global incidence and worldwide volume", *Global Spine J.*, vol. 8, no. 8, pp. 784-794, Dec. 2018.
2. Lumbar Disease Classification Using an Involutional Neural Based VGG Nets(INVGG) https://ieeexplore.ieee.org/document/10440340

3. Glassman SD, Bridwell K, Dimar JR, Horton W, Berven S, Schwab F. The impact of positive sagittal balance in adult spinal deformity. Spine (Phila Pa 1976). 2005;30(18):2024–9.
4. Lafage V, Schwab F, Patel A, Hawkinson N, Farcy JP. Pelvic tilt and truncal inclination: two key radiographic parameters in the setting of adults with spinal deformity. Spine (Phila Pa 1976). 2009;34(17):E599–606.
5. Schwab FJ, Blondel B, Bess S, Hostin R, Shaffrey CI, Smith JS, et al. Radiographical spinopelvic parameters and disability in the setting of adult spinal deformity: a prospective multicenter analysis. Spine (Phila Pa 1976). 2013;38(13):E803–12.
6. Ruchi et al., "Lumbar Spine Disease Detection: Enhanced CNN Model with Improved Classification Accuracy," in IEEE Access, vol. 11, pp. 141889-141901, 2023, doi: 10.1109/ACCESS.2023.3342064.
7. B. H. Cho et al., “Automated measurement of lumbar lordosis on radiographs using machine learning and computer vision,” Global spine journal, vol. 10, no. 5, pp. 611–618, 2020.
8. O. Khan, J. H. Badhiwala, G. Grasso, and M. G. Fehlings, “Use of machine learning and artificial intelligence to drive personalized medicine approaches for spine care,” World neurosurgery, vol. 140, pp. 512–518, 2020.
9. Grading scale of Lumbar Degenerative disease in full spine X-Ray https://www.scielo.br/j/coluna/a/cqTP8ypdGQrDDPBxFWmYzVm/
10. Tyler Richards, Jason Talbott, Robyn Ball, Errol Colak, Adam Flanders, Felipe Kitamura, John Mongan, Luciano Prevedello, Maryam Vazirabad.. (2024). RSNA 2024 Lumbar Spine Degenerative Classification. Kaggle. https://kaggle.com/competitions/rsna-2024-lumbar-spine-degenerative-classification.