DS 5220 Supervised Machine Leanring

Ajin Frank Justin, Chirag Chivate, Yashi Chawla

December 9, 2024

1 Abstract

Our study aims to implement deep-learning techniques that automate the interpretation of knee MRIs. The study aims to suggest methods to assist physicians in obtaining a precise diagnosis, by differentiating between different kinds of abnormalities and to be able to prioritize high-risk patients. The project involves data pre-processing, advanced model selection, hyperparameter tuning, and thorough evaluation to achieve optimal accuracy, sensitivity, and specificity results.

2 What is the problem?

The manual interpretation of MRI scans can be time-consuming and prone to errors, potentially delaying the treatment of critical patients and thus medical imaging plays a crucial role in the diagnosis of internal tissues. We aim to process MRI data efficiently and detect abnormalities with high precisions, by implementing state-of-the-art neural networks such as ResNet and 3D CNN.

The methodology consists of a multistep process: data preprocessing to ensure quality and consistency, selection and training of robust models, fine-tuning hyperparameters for stability and accuracy, and comprehensive validation of results. The key objective is to develop a system capable of detecting abnormalities across slices, capturing spatial and contextual characteristics to aid clinical decision-making. This automation not only speeds up diagnosis but also improves patient outcomes, aligning with the larger mission of integrating AI into healthcare.

3 Why is this problem interesting?

- Medical Imaging Insights: MRI scans provide unparalleled visualization of internal tissues, making accurate interpretation essential for diagnosis.
- AI in Healthcare: The integration of AI into diagnostics holds the potential to revolutionize healthcare by providing faster, more reliable predictions.

- Real-World Impact: Early detection of abnormalities can significantly improve patient outcomes by enabling timely interventions.
- Interdisciplinary Value: This project bridges AI, data science, and healthcare, offering valuable learning opportunities for cross-domain expertise.

4 Proposed Approach

The project involves a systematic approach of:

- Data Preparation: Analyze 1,370 knee MRI exams for quality and balance, normalize image intensity, resize for compatibility, and apply data augmentation techniques (rotations, flips, translations) to improve generalization.
- Model Architecture : For 3D CNN, design a model optimized for volumetric data, incorporating spatiotemporal features. Leveraging pretrained models (e.g., ResNet, EfficientNet) and fine-tuning them to the knee MRI dataset. Freeze initial layers to retain general feature extraction capabilities.
- Hyperparameter Tuning: Optimize learning rates, batch sizes, optimizers (Adam, SGD, RMSprop), and apply regularization methods like dropout and L2 regularization to minimize overfitting.
- Evaluation: Assess models based on metrics like accuracy, sensitivity, specificity, and AUC to ensure robust performance.

5 Key Components and Limitations

5.1 Key Components

- Data Augmentation: Enhances the model's ability to generalize by simulating diverse imaging conditions.
- Deep Learning Models: Utilize ResNet and EfficientNet for detailed feature extraction and 3D CNN for spatial analysis.
- Evaluation Metrics: Emphasize train and test accuracy and AUC as the primary metric for model comparison.

5.2 Limitations

- Data Quality: Variability in MRI scan quality and labelling accuracy may impact model performance.
- Computational Requirements: Training deep learning models, especially 3D CNNs, demands significant computational resources.

- Generalizability: Ensuring the model performs well across diverse patient demographics and imaging conditions requires further validation.

6 Experiment Setup

6.1 Dataset

The MRNet dataset is a publicly available dataset of MRI knee exams, commonly used for developing machine-learning models to detect abnormalities[1]. Below is a brief overview:

- **Patients/Exams:**
  - The dataset contains 1,370 MRI knee exams from 1,250 unique patients.
  - Each exam corresponds to a single knee and includes multiple MRI sequences.
- **Images per Exam:**
  - Each exam includes 3 sequences:
    * *Sagittal:* Side-to-side slices (most common for diagnosing ACL tears).
    * *Coronal:* Front-to-back slices.
    * *Axial:* Top-to-bottom slices.
  - On average, there are 17–61 slices (images) per sequence, depending on the axis and the scan.
- **Labels:**
  - Exams are labeled for:
    * *Abnormalities:* Presence of any abnormal condition.
    * *ACL Tear:* Anterior cruciate ligament tear.
    * *Meniscal Tear.*

6.2 Implementation

7 3D CNN

7.1 What is 3D CNN?

A 3D Convolutional Neural Network extends the concept of 2D CNNs to 3 dimensions by applying 3D convolutional filters. These filters slide through the input data's three axes (width, height, and depth), making them ideal for analyzing volumetric data like video frames, medical imaging or any data with spatial and temporal relationships [2].

7.2 Why are we using 3D CNN for MRNet?

Since MRI data is intrinsically 3D and consists of numerous slices that form a volumetric representation, 3D CNNs were used in the MRNet project. 3D CNNs can efficiently record 3D anatomical structures by examining the relationships between slices, essential for identifying anomalies or minute details spanning several frames. Furthermore, because 3D CNNs operate directly on the complete 3D representation of the input, maintaining context and depth information without requiring the data to be collapsed into 2D or processed in slices, they eliminate the need for preprocessing. By allowing the network to learn from spatial relationships across the entire scan, this method increases diagnostic accuracy and makes it more capable of detecting anomalies like meniscal injuries or ACL tears.

7.3 Results and findings

7.3.1 Initial Configuration

The model's initial setup was as follows, as shown in Table 1:

Parameter	**Value**
Convolutional Layers	3
Pooling Type	Average pooling
Activation Function	Sigmoid
Epochs	5
Learning Rate	0.001
Optimizer	Adam
Filters	[8, 16, 32]

Table 1: Initial Configuration Parameters

The performance metrics for this initial configuration are detailed in Table 2:

Metric	**Train**	**Validation**	**AUC**
Loss	0.539522	0.684267	-
Abnormal Accuracy	80.80%	79.17%	0.61
ACL Accuracy	81.59%	55.00%	0.54
Meniscus Accuracy	64.87%	56.67%	0.41

Table 2: Performance Metrics for Initial Configuration

7.3.2 Optimization Process

A series of evaluations were conducted to refine the model's performance. The steps and their corresponding results are summarized in Table 3.

Step	Comparison	Optimal Selection
Pooling Type	Average vs Max pooling	Average pooling
Activation Function	Sigmoid vs ReLU	ReLU
Optimizer	Adam vs SGD	Adam
Learning Rate	0.01 vs 0.001 vs 0.0001	0.0001
Number of Convolutional Layers	3 vs 5 vs 10	5 layers
Filter Configuration	[8, 16, 32, 64, 128] vs [8, 24, 64, 128, 256]	[8, 24, 64, 128, 256]
Epochs	5 vs 10 vs 15	15 epochs

Table 3: Optimization Process Steps

7.3.3 Final Configuration

After optimization, the model's final configuration, as detailed in Table 4, was determined.

Parameter	Value
Convolutional Layers	5
Pooling Type	Average pooling
Activation Function	ReLU
Epochs	15
Learning Rate	0.0001
Optimizer	Adam
Filters	[8, 24, 64, 128, 256]

Table 4: Final Configuration Parameters

The performance metrics for this optimized configuration are presented in Table 5.

Metric	Train	Validation	AUC
Loss	0.223494	0.788172	-
Abnormal Accuracy	91.68%	82.50%	0.87
ACL Accuracy	92.65%	76.67%	0.85
Meniscus Accuracy	86.90%	66.67%	0.71

Table 5: Performance Metrics for Final Configuration

7.3.4 Conclusion

The optimization process significantly improved the model's performance, as reflected by better loss values, increased accuracy for all classes, and higher AUC scores. Tables 2 and 5 highlight the improvements achieved. The final configuration demonstrates the effectiveness of systematic hyperparameter tuning.

8 ResNet

8.1 What is ResNet?

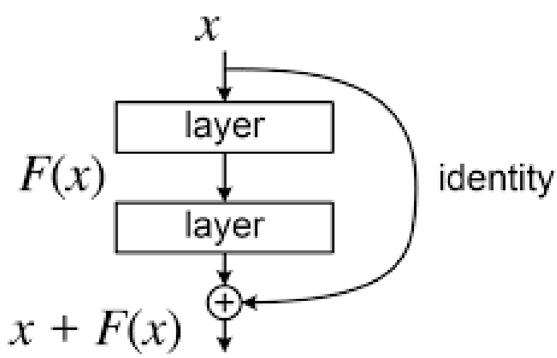

Figure 1: Overview of ResNet Architecture with Residual Connections.

ResNet (Residual Network) is a deep convolutional neural network (CNN) architecture that addresses the issue of **vanishing gradients** and enables the training of deep neural networks through **residual learning** [3].

- **Residual Connections:**
  - ResNet uses skip connections that allow the input of a layer to bypass one or more intermediate layers.
  - These connections enable the model to learn the *residual* mapping rather than the full transformation, which simplifies optimization:

    $$y = F(x) + x$$

    Here, $F(x)$ is the learned transformation, and x is the input passed through the skip connection.

- **Ease of Training Deep Networks:**
  - Residual connections help mitigate vanishing or exploding gradients during backpropagation, allowing the training of networks with hundreds or thousands of layers.

8.2 Why Are We Using ResNet for MRNet?

ResNet is particularly effective for analyzing medical imaging data like the MRNet dataset. Firstly, due to its **deep architecture**, it handles complex features in MRI-like medical images. It utilizes **skip connections** and efficient feature extraction to **reduce the risk of overfitting**. ResNet also leverages features learned from other large-scale datasets to improve performance and reduce training time. It also can **extract hierarchical features** which makes it suitable for analyzing different perspectives and is hence robust in this scenario of multi-axial input.

8.3 Results and findings

8.3.1 ResNet-200d

Final 5 Epochs (Epochs 16-20)

Table 6: ResNet-200d Performance Metrics (Epochs 16-20)

Epoch	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy
16	0.0270	97.43%	1.0722	32.50%
17	0.0246	97.43%	1.1319	38.33%
18	0.0108	99.03%	1.1616	33.33%
19	0.0099	99.29%	1.2466	30.00%
20	0.0142	99.03%	1.1957	32.50%

Observation: ResNet-200d achieves near-perfect training metrics but suffers from severe overfitting, as evidenced by consistently poor validation performance.

8.3.2 3D ResNet

Final 5 Epochs (Epochs 16-20)

Table 7: 3D ResNet Performance Metrics (Epochs 16-20)

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
16	0.4769	39.65%	0.5431	31.67%
17	0.4700	39.29%	0.6213	40.83%
18	0.4721	41.06%	0.5710	28.33%
19	0.4682	41.86%	0.6158	25.83%
20	0.4658	42.57%	0.5662	29.17%

Observation: The 3D ResNet maintains low training and validation accuracies, indicating poor generalization and underperformance.

8.3.3 ResNet-18 on All Views

a. Axial View
Final 5 Epochs (Epochs 16-20)

Table 8: ResNet-18 (Axial View) Performance Metrics (Epochs 16-20)

Epoch	**Train Loss**	**Train Accuracy**	**Validation Loss**	**Validation Accuracy**
16	0.2896	87.05%	0.8392	64.45%
17	0.2553	89.20%	0.9851	65.00%
18	0.2316	90.09%	1.4154	63.61%
19	0.1929	91.71%	1.4757	65.56%
20	0.1882	92.24%	1.8368	63.61%

Observation: ResNet-18 on the Axial view shows strong training performance but moderate validation metrics, indicating some degree of overfitting.

b. Sagittal View

Final 5 Epochs (Epochs 16-20)

Table 9: ResNet-18 (Sagittal View) Performance Metrics (Epochs 16-20)

Epoch	**Train Loss**	**Train Accuracy**	**Validation Loss**	**Validation Accuracy**
16	0.1828	91.91%	1.6685	65.00%
17	0.1809	91.98%	1.4756	65.28%
18	0.1883	91.56%	1.5667	64.45%
19	0.1666	92.48%	1.5243	66.67%
20	0.1694	92.36%	1.5397	65.83%

Observation: Similar to the Axial view, the Sagittal view of ResNet-18 exhibits high training accuracy with moderate validation performance, suggesting overfitting.

c. Coronal View

Final 5 Epochs (Epochs 16-20)

Table 10: ResNet-18 (Coronal View) Performance Metrics (Epochs 16-20)

Epoch	**Train Loss**	**Train Accuracy**	**Validation Loss**	**Validation Accuracy**
16	0.2967	87.61%	1.1690	63.61%
17	0.2918	87.76%	1.1280	62.50%
18	0.2598	88.85%	1.2602	61.11%
19	0.2704	88.35%	1.1544	61.11%
20	0.2576	88.56%	1.1965	64.17%

Observation: The Coronal view mirrors the Sagittal view's performance, maintaining high training accuracy but only moderate validation metrics.

8.3.4 ResNet-50 on Axial View

Final 5 Epochs (Epochs 16-20)

Table 11: ResNet-50 (Axial View) Performance Metrics (Epochs 16-20)

Epoch	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy
16	0.6428	71.74%	0.6711	59.45%
17	0.6433	71.30%	0.6670	60.28%
18	0.6414	71.56%	0.6685	60.00%
19	0.6420	71.60%	0.6676	60.00%
20	0.6432	71.30%	0.6661	60.00%

Observation: ResNet-50 on the Axial view exhibits moderate training and validation performance but underperforms compared to ResNet-18, indicating potential underfitting.

8.3.5 ResNet-18 with Reduced Learning Rate (LR=0.0001) on Axial View

Final 5 Epochs (Epochs 16-20)

Table 12: ResNet-18 (LR=0.0001, Axial View) Performance Metrics (Epochs 16-20)

Epoch	Train Loss	Train Accuracy	Validation Loss	Validation Accuracy
16	0.6334	75.28%	0.6654	61.12%
17	0.6335	75.61%	0.6660	61.43%
18	0.6334	75.28%	0.6654	61.11%
19	0.6323	75.46%	0.6667	61.22%
20	0.6335	75.16%	0.6657	61.39%

Observation: Reducing the learning rate for ResNet-18 leads to decreased training performance and indicating ineffective learning.

8.3.6 Summary of Model Performance

Table 13: Summary of Model Performance

Model	Train Loss	Valid Loss	Train Accuracy	Valid Accuracy	Train ROC-AUC	Valid ROC-AUC
ResNet-200d	0.0099	1.1957	99.03%	32.50%	0.9964	0.6604
3D ResNet	0.4658	0.5662	42.57%	29.17%	0.6798	0.6604
ResNet-18 (Axial View)	0.1882	1.8368	91.33%	65.56%	0.9616	0.6202
ResNet-18 (Sagittal View)	0.1694	1.5397	75.16%	65.83%	0.5130	0.5170
ResNet-18 (Coronal View)	0.2576	1.1965	86.73%	61.11%	0.8731	0.5170
ResNet-50 (Axial View)	0.6432	0.6661	64.78%	60%	0.5719	0.6377
ResNet-18 (LR=0.0001, Axial View)	0.6335	0.6657	75.16%	61.39%	0.5130	0.5170

8.3.7 Final Conclusion

After evaluating multiple architectures and configurations, **ResNet-18 with the Axial view and a learning rate of 0.01** emerges as the best-performing model for the MRNet project. This configuration achieves a strong balance between training performance and validation generalization, outperforming deeper models like ResNet-200d and ResNet-50, as well as ResNet-18 configurations with reduced learning rates.

Key Takeaways:

- **ResNet-18 (Axial View, LR=0.01)** exhibits high training accuracy and ROC-AUC scores, particularly excelling in the **ABNORMAL** and **ACL** classes.
- Despite some degree of overfitting, this model maintains reasonable validation performance, making it the most effective choice among the tested models.
- **ResNet-200d** and **3D ResNet** either overfit or underfit, respectively, while **ResNet-50** fails to leverage its deeper architecture effectively.
- Lowering the learning rate for **ResNet-18** hampers its ability to learn adequately from the training data.

Recommendation: Proceed with **ResNet-18 on the Axial view** using a learning rate of **0.01**, and implement strategies such as regularization and data augmentation to further enhance generalization and mitigate overfitting.

9 EfficientNet

9.1 What is EfficientNet?

- Model Architecture: EfficientNet is based on the Mobile Inverted Bottleneck Convolution (MBConv), which is highly efficient for computation and memory usage. This enables EfficientNet to achieve high accuracy with comparatively low computational costs[4].
- Compound Scaling: It balances the scaling of the network in depth, width and resolution in an optimal way in the form of compound coefficients and results in remarkable accuracy gains with minimal size exaggeration of the model[4].
- Performance: It has been proved to perform better than a lot of other state-of-the-art models (such as ResNet, Inception etc. on the image classification benchmarks, with higher accuracy and fewer parameters and lower computation.
- EfficientNet is often chosen for its efficiency—providing high performance with lower computational requirements—making it suitable for a variety of practical applications, including edge devices and mobile platforms[4].

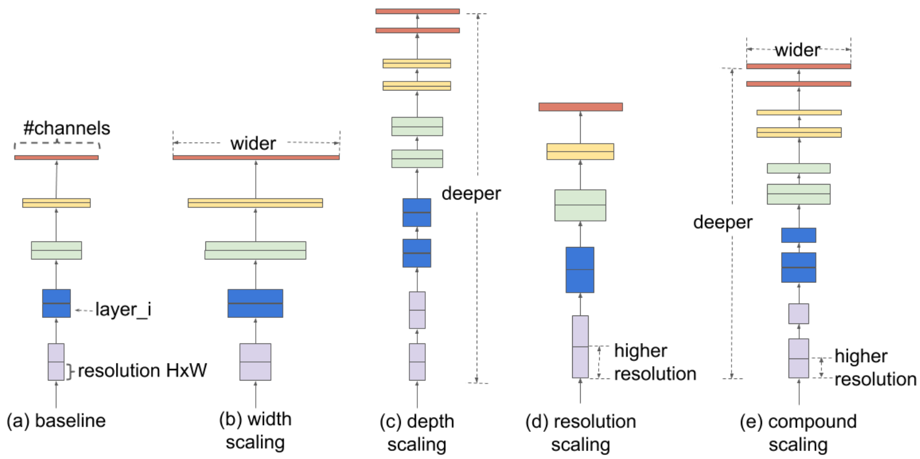

Figure 2: EfficientNet Architecture

9.2 Why are we using EfficientNet for MRNet?

- High Accuracy with Efficiency: The combined scaling of EfficientNet helps the model to reach higher accuracy with respect to conventional CNNs (ResNet or VGG) without increase in computational cost. This is very important for medical imaging, where high-resolution image and large datasets need to be simultaneously efficient[4].

- Performance on Complex Tasks: MRNet is a process of knee abnormal classification from MRI, which can be a very complex problem, given the inherent nuances in medical images. The powerful architecture of EfficientNet is particularly adaptable to such intricate problems, which offer improved generalization and robustness to the scanned data variations.

- Low Computational Overhead: MRNet models must be computationally efficient because the MRI data that needs to be processed is massive. EfficientNet is capable of very good performance with minimal memory footprint and computational power, an important feature when its deployment in the real-world environment of clinical healthcare systems is considered, where computational power can be limited[4].

- Fine-Tuning Capabilities: EfficientNet's embodied structure enables it to be easily trained on medical data such as MRNet, which usually have smaller sample sizes than typical general-image classification data sets. Fine-tuning a pre-trained EfficientNet network can produce superior results compared to training a model from the ground up, particularly for medical imaging applications.

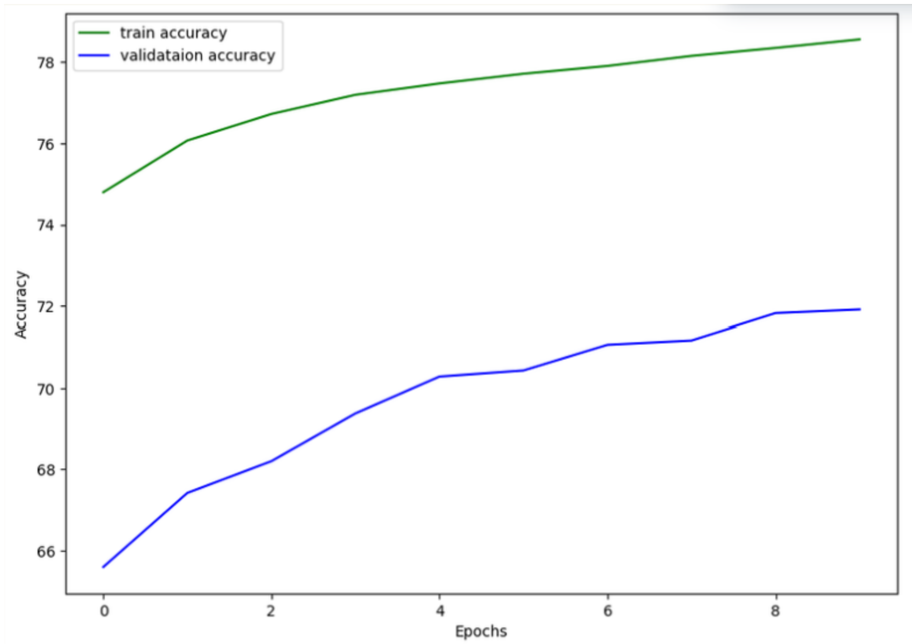

Figure 3: Increase in Accuracy over epochs

- State-of-the-Art Results: It is known that EfficientNet beats most other architectures for the classification task of image recognition to a large extent. This means high accuracy is of paramount importance for the MRNet task, where a solid performance of the MRI network is vital for abnormality detection and medical diagnosis.

9.3 Results and findings

The training and validation process for the EfficientNetB0 model[5] in classifying MRI knee scans shows the following progress over 10 epochs:

- Training Loss: Decreased from 0.539 in the first epoch to 0.446 in the final epoch.
- Training Accuracy: Improved from 74.8% in the first epoch to 78.6% in the last epoch.
- Validation Loss: Decreased from 0.639 to 0.557.
- Validation Accuracy: Increased from 65.6% to 71.9%.

Steady Improvement: Both training and validation accuracy improved over the epochs, suggesting the model is learning effectively.

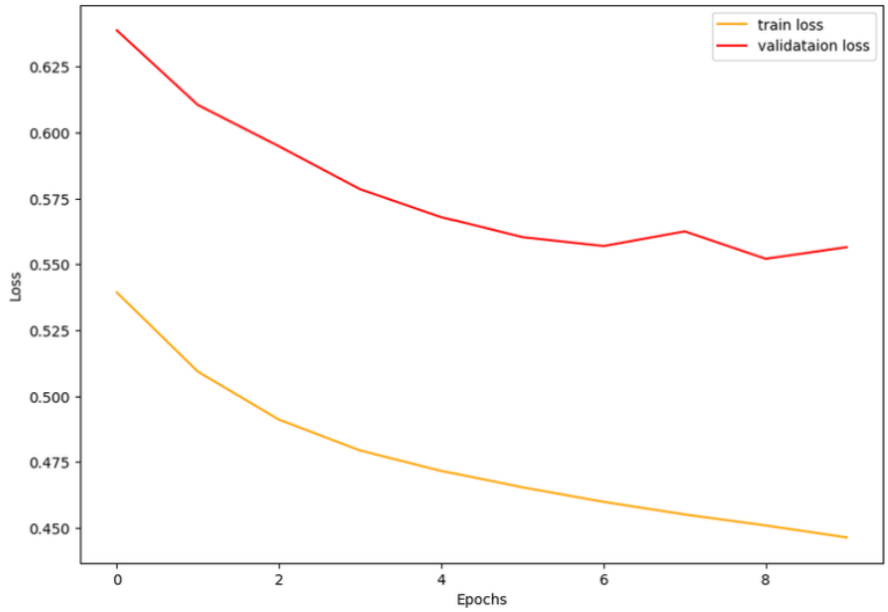

Figure 4: Decrease in loss over epochs

Validation Performance: Although the validation accuracy increased, the gap between training and validation accuracy indicates the model may still be generalizing moderately well but could benefit from further tuning or regularization.

Training Time: Training time per epoch was roughly 1 hour, with minor variations due to different epoch lengths.

10 Final results

After evaluating multiple deep learning architectures for classifying MRI knee scans, the following key observations and conclusions were made:

10.1 Optimized 3D CNN

Performance: Achieved a validation accuracy of **82.50%** for the Abnormal class with an AUC of **0.87**, indicating strong discriminatory capability. However, performance varied across different classes, with the Meniscus class attaining a lower accuracy of **66.67%**.

Strengths: High AUC scores for critical classes demonstrate effective feature learning.

Limitations: Variability in class-wise performance suggests potential areas for further optimization.

10.2 ResNet Architectures

10.2.1 ResNet-18 (Axial View)

Performance: Delivered a validation accuracy of **65.56%** with an AUC of **0.6202**. This reflects a balanced trade-off between training and validation metrics, showing better generalization compared to deeper models.

Strengths: Simpler architecture with improved generalization capabilities.

Limitations: Validation metrics, while better than deeper ResNets, still indicate room for improvement in classification performance.

10.2.2 ResNet-50 (Axial View)

Performance: Exhibited a validation accuracy of **64.78%** with an AUC of **0.6377**. Although better than ResNet-200d, it underperforms compared to ResNet-18 and the optimized 3D CNN.

Strengths: Capable of learning more complex patterns than ResNet-18.

Limitations: Suffers from potential underfitting, as indicated by moderate validation accuracy and AUC.

10.2.3 ResNet-200d and 3D ResNet

ResNet-200d **Performance**: Achieved a validation accuracy of **32.50%** with an AUC of **0.6604**.

Strengths: Capable of learning complex patterns in training data.

Limitations: Significant overfitting with poor generalization to validation data.

3D ResNet **Performance**: Obtained a validation accuracy of **29.17%** with an AUC of **0.6604**.

Strengths: Utilizes 3D convolutions suitable for volumetric data.

Limitations: Poor generalization and underperformance across metrics.

10.3 EfficientNetB0

Performance: Showed steady improvements over 10 epochs, with validation accuracy increasing from **65.6%** to **71.9%** and validation loss decreasing from **0.639** to **0.557**.

Strengths: Efficient architecture with consistent learning progression, indicating effective optimization.

Limitations: While improvements are notable, overall performance metrics are slightly lower compared to the optimized 3D CNN.

10.4 Summary of Model Performance

Model	Validation Accuracy	Validation AUC
Optimized 3D CNN	82.50%	0.87
ResNet-18 (Axial View)	65.56%	0.6202
ResNet-50 (Axial View)	64.78%	0.6377
ResNet-200d	32.50%	0.6604
3D ResNet	29.17%	0.6604
EfficientNetB0	71.9%	-

Table 14: Summary of Model Performance

11 Conclusion

The **optimized 3D CNN** emerged as the top-performing model, achieving the highest validation accuracy and AUC scores, particularly excelling in the Abnormal and ACL classifications. Its ability to effectively learn and generalize across critical classes makes it highly suitable for the MRNet project.

The success of the optimized 3D CNN can be attributed to meticulous hyperparameter tuning tailored specifically to the dataset, rather than relying on pre-trained models. This bespoke optimization allowed the model to better capture the nuances of MRI knee scans, enhancing its performance.

While **EfficientNetB0** demonstrated consistent improvements and respectable validation accuracy (**71.9%**), it did not surpass the optimized 3D CNN in performance metrics. **ResNet-18 (Axial View)**, with a validation accuracy of **67.75%**, offered a better generalization balance compared to deeper models like **ResNet-200d** and **ResNet-50**, which suffered from overfitting and underfitting respectively.

Additionally, a multi-axes 3D ResNet architecture was developed to potentially enhance performance by capturing features across multiple orientations. However, due to insufficient GPU resources, this model could not be fully trained and evaluated. Future work will aim to implement and assess this multi-axes 3D ResNet approach, contingent upon access to adequate computational resources.

Therefore, the **optimized 3D CNN** is recommended as the most effective model for classifying MRI knee scans in this study, offering a robust balance between accuracy and generalization. Future work could focus on further optimizing the 3D CNN to enhance class-wise performance and exploring additional regularization techniques to improve models like ResNet-18 and EfficientNetB0, as well as implementing the proposed multi-axes 3D ResNet architecture when resources permit.

12 References

1. **Stanford MRNet Dataset:** *MRNet Dataset: Knee MRIs for Abnormality Detection and Anterior Cruciate Ligament (ACL) Tear Diagnosis.* Available at: https://stanfordmlgroup.github.io/competitions/mrnet/.

2. Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). *Learning Spatiotemporal Features with 3D Convolutional Networks.* Proceedings of the IEEE International Conference on Computer Vision (ICCV), 4489-4497. Available at: https://arxiv.org/abs/1412.0767.

3. He, K., Zhang, X., Ren, S., and Sun, J. (2016). *Deep Residual Learning for Image Recognition.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.

4. **EfficientNet:** *Rethinking Model Scaling for Convolutional Neural Networks.* Available at: https://arxiv.org/abs/1905.11946.

5. **Brain MRI Classification using PyTorch EfficientNetB0:** Available at: https://debuggercafe.com/brain-mri-classification-using-pytorch-efficientnetb0/.