Table Of Content

List Of Figures

Chapter-1: Introduction

This research delves into the utilization of transfer learning on datasets like the MedMNIST dataset to enhance the efficiency of 3D scanning processes. The characteristics of this dataset for 3D scans are less size, low resolution scans and imbalance of classes.We are using pre-trained 2D models as base model for our transfer learning approach. Pre-training ensures that the model can be fine-tuned with less size of dataset. We have used 2D models because most of the benchmark 3D models have a large number of parameters which takes more computational resources and while classifying 3D scans efficient use of computational resources is just as important as the results.
One key consideration in our approach is addressing data loss during the transition from 3D to 2D representation. We conceptualize the 3D volume as a stack of 2D images, each with dimensions of 28x28 pixels[2]. By augmenting the channel size to 3, we align the input format with the requirements of the base model used for transfer learning, facilitating seamless integration and processing. Additionally, we employ upsampling techniques to enhance the dimensionality of the image, thereby refining resolution and making the images dimensionally suitable as input for base model.

In Center to our methodology is the utilization of the EfficientNetV2 model for transfer learning. Pretrained on the expansive ImageNet dataset, EfficientNetV2 serves as a robust foundation for feature extraction and classification tasks. Leveraging transfer learning with EfficientNetV2[3], we harness pre learned representations and fine-tune them to suit the intricacies of medical imaging data. Through a pipeline involving flattening, linear encoding, and output generation, we translate the base-model's predictions into our model's predictions according to our dataset.
Our findings indicate promising outcomes, with our approach surpassing seven out of eight benchmark models on average across varying data sizes. Our research trajectory is geared towards refining model architectures, optimizing training protocols, and exploring novel techniques to enhance accuracy and robustness. By pushing the boundaries of medical imaging technology, we aspire to empower clinicians with superior diagnostic capabilities, ultimately advancing patient care and outcomes.

Chapter-2: Background and Motivation

In biomedical imaging and 3D image classification, using 2D models and traditional machine learning algorithms poses several challenges:

Complex Spatial Information: Biomedical scans often contain intricate spatial structures that 2D models struggle to capture fully. This limitation results in loss of crucial 3D information, impacting classification accuracy.
Limited Data Representation: Traditional machine learning algorithms may fail to represent the multidimensional features present in 3D biomedical images effectively. This leads to suboptimal performance as these algorithms are designed for simpler, lower-dimensional data.
Scalability: As the complexity and volume of biomedical data increase, scalability becomes a significant concern for 2D models and traditional algorithms. They may struggle to process and analyze large-scale datasets efficiently, affecting both speed and accuracy.
Generalization: 2D models and traditional algorithms may face challenges in generalizing patterns across different types of biomedical images. This lack of robustness can hinder their performance in real-world scenarios where diverse image variations are encountered.
Computational Efficiency: Processing 3D images using 2D models and traditional algorithms often requires extensive computational resources. This inefficiency can lead to longer processing times and higher costs, limiting their practical applicability in large-scale biomedical imaging tasks.

Biomedical image classification and scanning have been pivotal in medical research and diagnosis. Early methods primarily relied on 2D models and traditional machine learning algorithms. These approaches often faced challenges in handling complex spatial information and achieving high accuracy due to limited data representation.

3D Model Developments: As the need for better accuracy grew, 3D CNNs emerged as a more suitable approach for handling volumetric biomedical images. These models could capture spatial relationships across multiple dimensions, improving classification accuracy significantly.
Transfer Learning: With the advent of transfer learning, researchers found a powerful tool to enhance classification efficiency. Transfer learning[9] leverages pre-trained models on large datasets and fine-tunes them on smaller, domain-specific datasets. This approach reduces the training time while improving model generalization.
Application of Transfer Learning on MedMNIST Dataset which is presented in this paper.

The MedMNIST dataset, with its diverse collection of 2D and 3D biomedical images, provides an ideal platform for applying transfer learning techniques to boost classification efficiency. Here's how transfer learning can be utilized effectively:

Pre-trained Models: Begin by selecting pre-trained CNN models, such as **VGG, ResNet**, or Inception, trained on large-scale image datasets like ImageNet. These models have learned rich hierarchical features that are transferable across domains.
Fine-tuning: Next, fine-tune the selected pre-trained models on the MedMNIST dataset. Since MedMNIST shares similarities with general image datasets, fine-tuning allows the model to adapt to domain-specific features while retaining the learned representations.
Enhanced Efficiency: Transfer learning on MedMNIST enhances efficiency by reducing training time, minimizing the need for labeled data, and improving classification accuracy. The model inherits knowledge from pre-trained networks, leading to faster convergence and better generalization.

2.1 Literature Survey

Initially, conventional 2D models and traditional machine learning algorithms were employed for these tasks, facing limitations in handling complex spatial information and achieving high accuracy due to data representation constraints.

To enhance image classification efficiency, We have explored various strategies, including the utilization of advanced models like EfficientNetV2 alongside transfer learning techniques. EfficientNetV2[3], known for its superior performance in terms of accuracy and computational efficiency, can be integrated into the image classification pipeline to improve overall efficiency.
EfficientNetV2 Integration: EfficientNetV2, with its optimized architecture and parameter efficiency, can replace or complement traditional 2D models in biomedical image classification tasks. By leveraging the compound scaling method of EfficientNetV2, which efficiently balances model depth, width, and resolution, we can achieve better performance while reducing computational resources.

Transfer Learning: Integrating transfer learning with EfficientNetV2[3] further enhances classification efficiency. We can leverage pre-trained EfficientNetV2 models on large-scale image datasets like ImageNet and fine-tune them on biomedical datasets such as MedMNIST. This approach capitalizes on the learned representations from general image datasets, accelerating convergence and improving model generalization on domain-specific biomedical images[9].

Benefits of EfficientNetV2 and Transfer Learning Integration:

Improved Accuracy: EfficientNetV2's[3] advanced architecture and transfer learning's knowledge transfer lead to enhanced classification accuracy on biomedical images.
Reduced Training Time: Leveraging pre-trained models and fine-tuning them on target datasets decreases training time significantly, making the classification process more efficient.
Minimal Data Annotation: Transfer learning reduces the dependency on extensive labeled data, making it feasible to work with limited annotated biomedical images.
Scalability and Versatility: EfficientNetV2[8] and transfer learning techniques can scale to handle diverse classification tasks within the biomedical domain.

By combining EfficientNetV2 with transfer learning strategies, researchers can revolutionize biomedical image classification, achieving higher efficiency, robustness, and scalability essential for advanced medical diagnostics and research endeavors.

2.2 Problem Statement

This project aims to explore transfer learning techniques to efficiently classify 3D medical scans from the MedMNIST dataset using base models designed for 2D image classification. The goal is to achieve high accuracy and AUC with less computational cost and limited dataset.

Why Transfer Learning?

Transfer learning is crucial and relevant for biomedical scanning and classification of 3D datasets using 2D models due to its ability to enhance efficiency and accuracy. Traditional 2D models often struggle to capture complex spatial information present in volumetric biomedical images. Transfer learning[9] addresses this challenge by leveraging pre-trained models on large-scale datasets[9], adapting them to domain-specific tasks like biomedical image analysis. By fine-tuning pre-trained 2D models on 3D datasets such as those in the MedMNIST collection, researchers can benefit from learned hierarchical features, faster convergence, and improved generalization. This approach reduces the need for extensive data annotation and training time while enhancing classification performance, making transfer learning a valuable tool for advancing medical diagnostics and research in a cost-effective and practical manner.

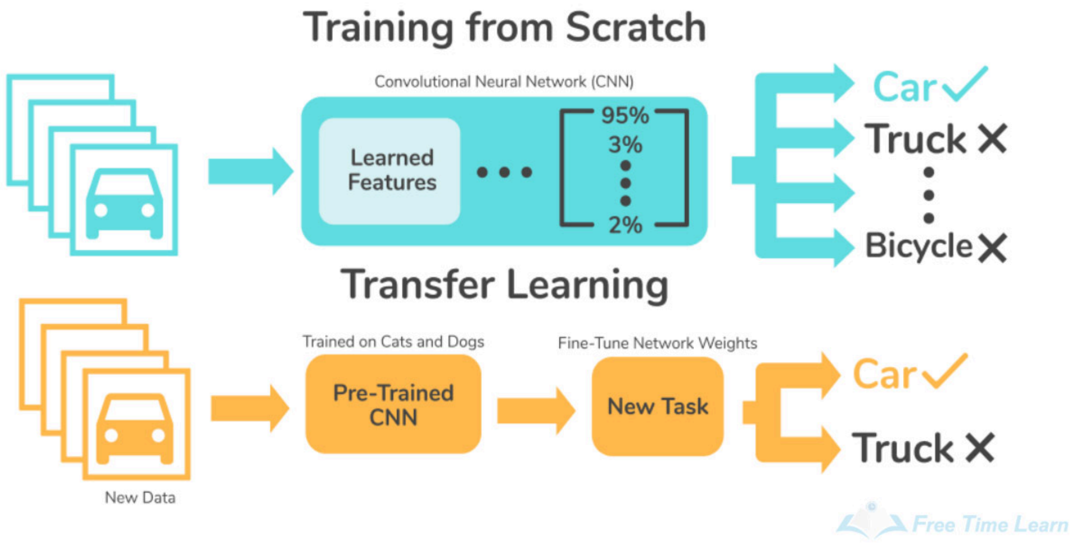


Figure 2.1

Why the MedMNIST Dataset?

With recently launched and not been worked on immensely, This paper employs use of MedMNIST[2] dataset which is efficient in many ways:

MedMNISTv2[1] is a comprehensive dataset meant for evaluating 2D and 3D biomedical image classification tasks. It encompasses 12 2D datasets containing 708,069 images and 6 3D datasets comprising 9,998 images. This dataset is crafted to offer the following advantages for efficient 3D classification:

Diversity: MedMNIST[1] v2 covers a wide range of data modalities, dataset scales, and tasks, including binary/multi-class, multi-label, and ordinal regression. This diversity mirrors datasets like VDD4 and MSD5, enabling a robust assessment of machine learning models across various scenarios. It provides both 2D and 3D biomedical images for a comprehensive evaluation.
Standardization: Each sub-dataset within MedMNIST v2 undergoes preprocessing into a uniform format, eliminating the need for users to possess specialized background knowledge. Additionally, it supplies standardized train-validation-test splits, facilitating easy comparison of algorithms.
Lightweight: The compact size of images in MedMNIST v2, either 28×28 for 2D or $28 \times 28 \times 28$ for 3D, makes it convenient for evaluating machine learning algorithms without excessive computational overhead.
Educational Utility: Biomedical image analysis often demands expertise in multiple domains like computer vision, machine learning, biomedical imaging, and clinical science. MedMNISTv2[1], with its Creative Commons License, is accessible and suitable for educational purposes, aiding researchers and learners from diverse backgrounds to engage with biomedical image analysis tasks effectively.

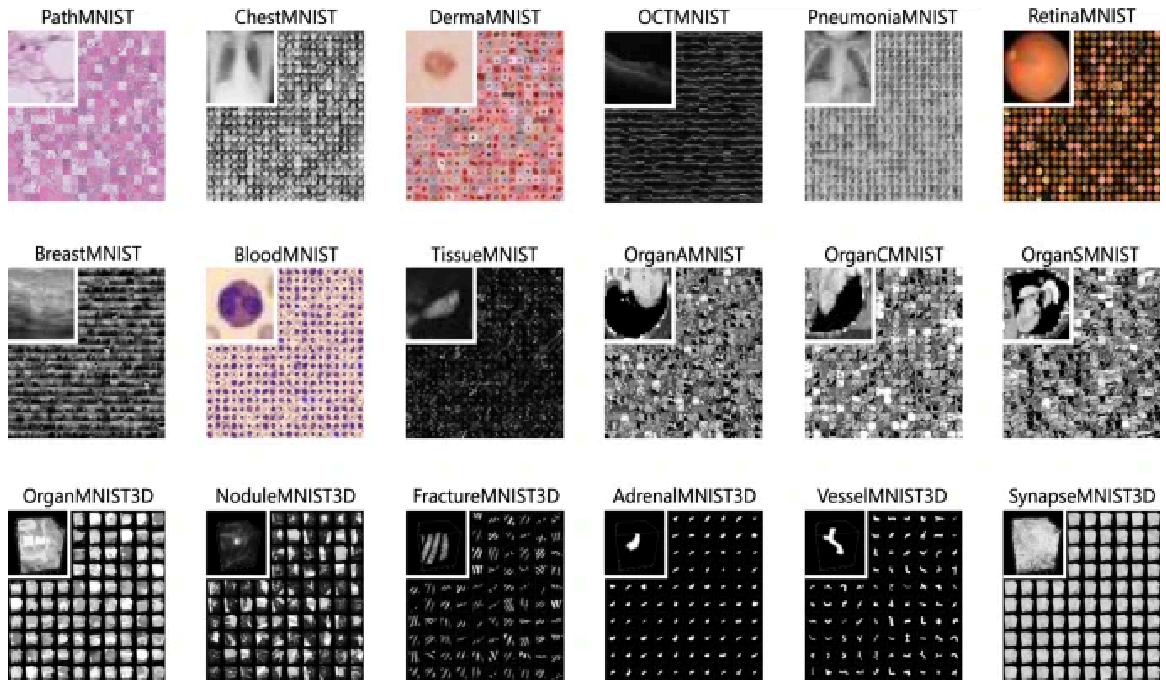


Figure 2.2
An overview of MedMNIST v2. MedMNIST is a large-scale MNIST-like collection of standardized 2D and 3D biomedical images with classification labels.

Chapter-3: Methodology

Dataset selection: We have used the MedMNIST[3] dataset, it contains 6 datasets for 3D scans which are OrganMNIST3D, NoduleMNIST3D, AdrenalMNIST3D, FractureMNIST3D,VesselMNIST3D and SynapseMNIST3D. Each of these datasets contains scans with dimensions of 28x28x28 pixels, which is a common size for biomedical images. Furthermore, each dataset is split into standard subsets for machine learning tasks: training, testing, and validation datasets. The OrganMNIST3D dataset has a more diverse classification task with 11 classes, including various organs, while the FractureMNIST3D dataset is simpler with only 3 classes related to fractures. The other datasets primarily focus on binary classification tasks, such as detecting nodules, vessels, synapses, or abnormalities in adrenal scans.
Additionally, the number of samples in each dataset falls within the range of 1000 to 2000, indicating a moderate-sized dataset suitable for training and evaluation without overwhelming computational resources or requiring extensive data preprocessing.
This detailed dataset selection process ensures that our study covers a range of biomedical imaging tasks, from organ identification to more specialized tasks like nodule detection and fracture identification. The variety in classes and sample sizes allows for a comprehensive evaluation of our classification models across different scenarios and complexities within the MedMNIST[1] dataset.

Handling imbalanced dataset:To handle the inherent imbalance in the dataset, we employed a weighted random sampler during training. This technique ensures that each class is sampled uniformly, mitigating the bias caused by class imbalances. First, we determined the number of instances for each class and computed the inverse of these instances as class weights. For instance, if Class A has fewer samples, it would receive a higher weight to balance its representation during training. Sample weights were then assigned based on the class weight of each sample, ensuring that rarer classes contribute proportionately to the training process. This approach helps prevent the model from favoring classes with more instances and improves its ability to generalize across all classes, leading to more accurate and reliable classification results.
Model Selection: In selecting our base models, we prioritized two key factors: the number of parameters in the model and its performance on pre-training datasets. Initially, we considered 3D models due to their potential in capturing spatial relationships within biomedical images. However, these models typically have a large number of parameters, which can be computationally intensive and may require substantial resources for training.
To address this challenge, we transitioned to 2D models, particularly focusing on EfficientNet models known for their high accuracy with fewer parameters compared to other architectures. We specifically utilized EfficientNet-V2-S, which is optimized for small images and strikes a balance between model complexity and performance.

EfficientNet models are designed based on compound scaling, where the model's depth, width, and resolution are scaled simultaneously. This scaling approach allows EfficientNet models to achieve high accuracy while maintaining computational efficiency, making them well-suited for tasks like biomedical image classification.
By selecting EfficientNet-V2-S[8] as our base model, we aimed to leverage its parameter efficiency and proven performance in handling small images, optimizing our model architecture for the MedMNIST dataset. This decision not only helps reduce computational complexity but also enhances the model's ability to generalize and achieve accurate classifications across diverse biomedical imaging tasks within the dataset.

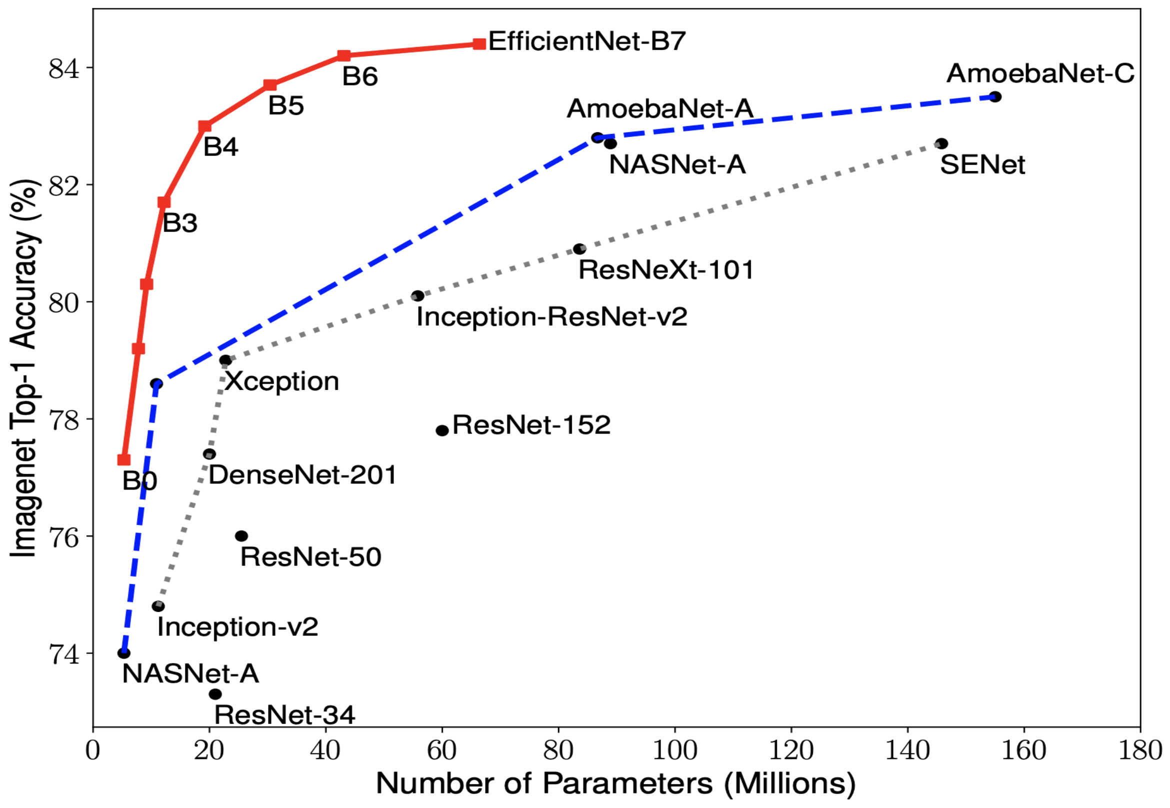


Figure 3.1

Hyperparameter tuning: The process of hyperparameter tuning involves optimizing various parameters and settings of the machine learning model to achieve the best possible performance. Given the dataset's relatively small size, we approached hyperparameter tuning with the goal of slower convergence, focusing on finding the right balance between training speed and model accuracy.
One critical aspect of hyperparameter tuning was determining the appropriate batch size for training. We experimented with different batch sizes and found that a batch size of 32 offered the best trade-off between convergence speed and training time. A larger batch size could potentially speed up training but might also lead to slower convergence or

suboptimal generalization. On the other hand, a smaller batch size might improve convergence but could result in longer training times.
For handling the imbalance in the dataset, we employed the Focal loss function. Focal loss is particularly effective for imbalanced datasets as it down-weights the loss assigned to well-classified examples, focusing more on hard-to-classify instances. This helps the model prioritize learning from challenging samples, improving overall classification performance.
Regarding optimization algorithms, we chose to use the Rectified Adam (RAdam) optimizer. RAdam is known for its robustness and effectiveness, especially in scenarios with limited data samples. Given the dataset's size constraints, RAdam can help the model reach convergence more efficiently and effectively navigate the training process.

Additionally, we incorporated an Exponential learning rate scheduler into our training pipeline. This scheduler gradually decreases the learning rate over time, allowing the model to fine-tune its parameters more delicately as training progresses. This approach can prevent the model from overshooting optimal parameter values and stabilize training, ultimately improving convergence and model performance.
Moreover, we adjusted the number of training epochs based on the specific dataset being trained. Different datasets may require varying amounts of training time to reach convergence and achieve peak performance. By saving the models with the highest accuracy on the validation dataset during training, we ensured that we retained the best-performing models for subsequent evaluation and deployment.

3.1 Focal loss

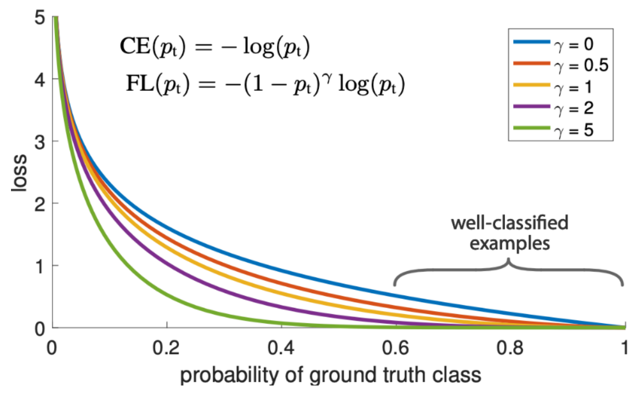


Figure 3.2

In this paper[3] as we are dealing with the Pre trained model and using Transfer learning approach to train and prepare our model for this predictions and computation , there is one problem which arries that is imbalance between different classes in our dataset samples, so if we use loss functions like Cross-Entropy Loss to be dominated by

well-classified examples from the majority class, resulting in suboptimal performance, particularly for minority classes.
Let's look into how it happens :

$$\text{Cross Entropy Loss} = -Y_{act} \ln ln \left(Y_{pred}\right) - (1 - Y_{act}) \ln ln (1 - Y_{pred})$$

The fundamental concept of Cross-Entropy loss is to penalize incorrect predictions more heavily than rewarding correct ones. To address this and achieve better results by maintaining dataset consistency, we have the Focal loss function. Focal Loss is a loss function developed primarily to tackle the challenge of class imbalance in both binary and multi-class classification tasks. It was first proposed by Lin et al. in their 2017 paper "Focal Loss for Dense Object Detection."[10]
In many classification scenarios, the distribution of classes within the dataset is uneven. Certain classes may have a significantly larger number of samples compared to others, leading to this class imbalance issue. Traditional loss functions can be affected by this imbalance. Focal Loss introduces a new approach by reducing the importance of easy-to-classify examples (i.e., those that are well-classified) during training. Instead, it prioritizes hard-to-classify examples, effectively reducing the impact of class imbalance.

How Focal Loss Works:
Modulation of Loss for Well-Classified Examples: Focal Loss modulates the standard Cross-Entropy Loss (CE Loss) by introducing a modulating factor that down-weights the loss contribution from well-classified examples.
It uses a term (1 - pt) ** gamma, where pt is the probability of the true class. This term reduces the loss contribution.

Tuning Focusing Parameter:
The focusing parameter gamma[3] controls the rate at which the modulating factor decreases as the probability of the true class (pt) increases.
A higher gamma puts more emphasis on hard-to-classify examples, making the loss more focused on correcting misclassifications for minority classes.
By down-weighting the loss contribution from well-classified examples, Focal Loss effectively handles class imbalance without the need for re-sampling techniques or class weights. It enables the model to focus more on learning from challenging examples, resulting in improved performance, particularly for minority classes.

Finding the Optimal Gamma for Focal Loss

To determine the optimal gamma value, we conducted experiments using different gamma values ranging from 0.5 to 1.0 with increments of 0.5. The experiments were conducted on our dataset, and the model performance was evaluated using various metrics such as accuracy (ACC) and Area under curve (AUC).

The results indicated that a gamma value of 0.7 yielded the best performance for our model, achieving the highest average ACC and AUC across different metrics compared to other gamma values. Specifically, at gamma=0.7, we observed an average ACC of 0.873 and an average AUC of 0.805, which outperformed other gamma values. We have used the average value of AUC and ACC scores to calculate the optimal gamma.

Table 1 Comparison of model performance with change in gamma

	OrganMNIST 3D		NoduleMNIST 3D		FractureMNIST3D		AdrenalMNIST3D		VesselMNIST3D		SynapseMNIST3D	
gamma	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
6	0.9967	0.924	0.9101	0.87	0.6318	0.491	0.888	0.815	0.93	0.912	0.811	0.745
6.5	0.9967	0.924	0.911	0.88	0.6401	0.503	0.888	0.815	0.961	0.92	0.81	0.74
7	0.9965	0.923	0.9132	0.89	0.6495	0.513	0.888	0.815	0.981	0.94	0.811	0.752
7.5	0.9965	0.923	0.9134	0.89	0.651	0.514	0.888	0.815	0.981	0.94	0.79	0.713
8	0.9964	0.923	0.9135	0.9	0.6495	0.513	0.839	0.754	0.983	0.943	0.73	0.65

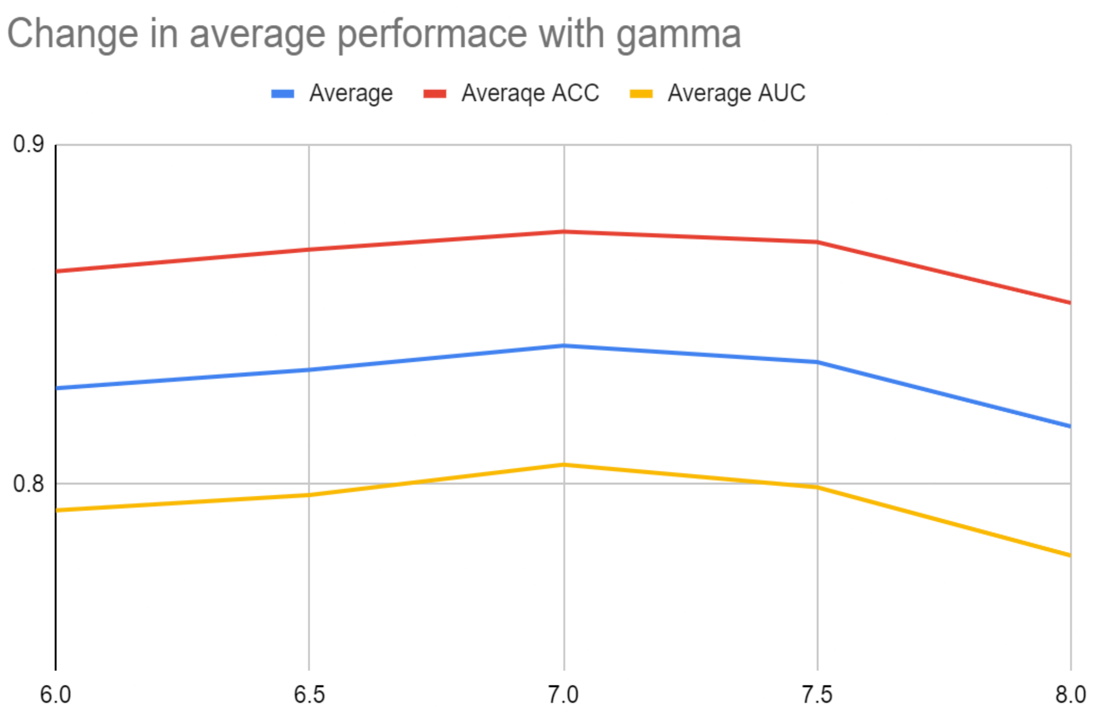


Figure 3.3 Change in performance with gamma

Architecture

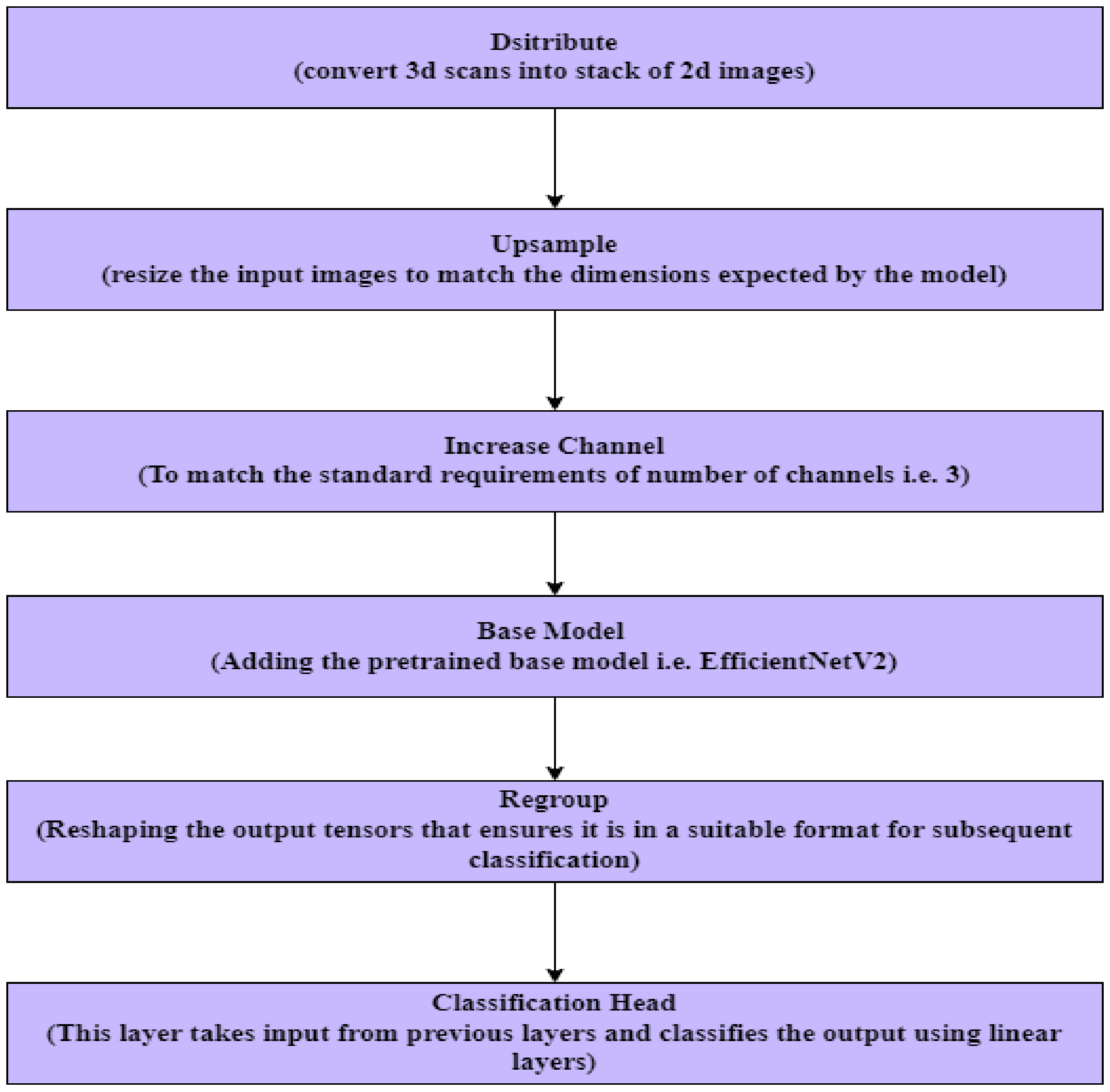

Dsitribute
(convert 3d scans into stack of 2d images)
Upsample
(resize the input images to match the dimensions expected by the model)
Increase Channel
(To match the standard requirements of number of channels i.e. 3)
Base Model
(Adding the pretrained base model i.e. EfficientNetV2)
Regroup
(Reshaping the output tensors that ensures it is in a suitable format for subsequent classification)
Classification Head
(This layer takes input from previous layers and classifies the output using linear layers)

Figure 3.4

Let's look into the each layer of our architecture:

1) Distribute Layer

The input tensor has additional dimensions that need to be flattened or distributed before passing through the pre-trained model. By rearranging the dimensions, we ensure that the input tensor is in a format suitable for processing by succeeding layers in the neural network.

The Distribute layer[3] rearranges the dimensions of the input tensor to prepare it for processing by the pre-trained model. It reshapes the tensor from 3D scans to 28 2D images which can be processed by a pre-trained base model.

Let X be the input tensor with dimensions

(B,C,H,W,D) where B is the batch size, C is the number of channels, H and W are the height and width of the image, and D represents additional dimensions if any. After reshaping, the tensor becomes
(B×D,C,H,W). This step ensures compatibility between the input tensor and the pre-trained model, allowing us to leverage the feature extraction capabilities of the model effectively.

2) Upsample
The spatial dimensions of the input tensor may not match the expected input size of the pre-trained model. Upsampling is necessary to resize the input images to match the dimensions expected by the model.
The Upsample layer resizes the spatial dimensions of the tensor to match the specified image_dimension. It performs interpolation to increase the resolution of the image. Let X be the input tensor with dimensions (B×D,C,H,W). After upsampling, the spatial dimensions become
(new_H,new_W).
Properly resizing the input images ensures that the pre-trained model can process the images without distortion or loss of information, improving the accuracy of feature extraction. We have used bilinear mode for upsampling.

3) Increase Channel
The number of channels in the input tensor may need to be adjusted to match the input requirements of the pre-trained model. Increasing the number of channels using convolutional operations allows us to extract richer features from the input data.
Having an appropriate number of channels ensures that the pre-trained model can effectively capture the complexity and variability of the input data, leading to better feature representation and classification performance.
Our pretrained model requires a 3 channel input of image while in MedMNIST each dataset has a different number of channels in image.
4) Model
The pre-trained model serves as a feature extractor, leveraging its learned representations to extract high-level features from the input data. By using a pre-trained model, we benefit from its ability to capture complex patterns and structures in the input data, even if our dataset is small or limited. This saves computational resources and time compared to training a model from scratch.
the pre-trained model that processes the input tensor and extracts useful features. It can be any convolutional neural network architecture pretrained on a large dataset such as ResNet, VGG, etc. the pre-trained model applies a series of convolutional, pooling, and fully connected layers to the input tensor to extract hierarchical features.

We have used EfficientNetV2 as our base pretrained model. Reason for selecting EfficientNetV2 is that it gives one of the best accuracy on imageNet with less number of parameters when compared to other benchmark models like ResNet. It is also compatible for Transfer learning.

5) Regroup

After processing by the pre-trained model, the output tensor may need to be reshaped to its suitable format for further processing or interpretation.

The output from model is of size 1000 so the dimension of the tensor is (B*D) x 1000, which we reshape to B x 1 x D x1000 so that all the features for 3D scan are together.

6) Classification Head

The Classification head layer represents the final output layer that produces the predicted class probabilities for each input sample.

It consists of a flatten layer and two linear layers. First linear layer takes input of size 28000 and output tensor of size 1000 and second linear layer takes input of size 1000 and output tensor of size of number of the classes in the dataset.

\

Chapter-4: Results

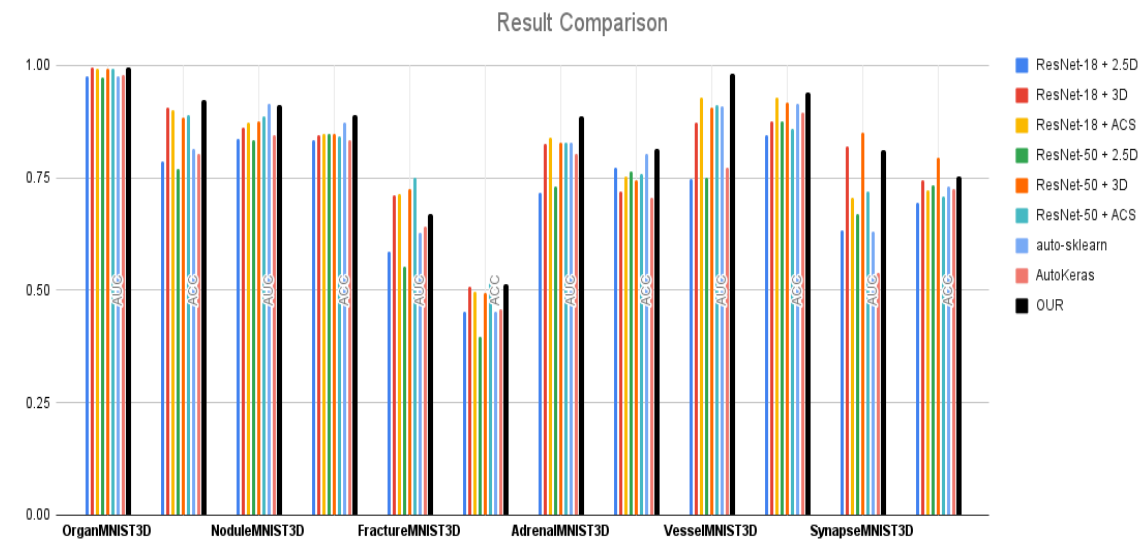


Figure 4.1

We have used Accuracy and AUC(Area Under the Curve) to compare the results with all different models and out transfer learning model, As we have six different types of 3D scans OrganMNIST3D , NoduleMNIST3D, FractureMNIST3D, AdrenalMNIST3D, VesselMNIST3D, SynapseMNIST3D , so we calculated accuracy and AUC for all these data samples.
accuracy and AUC are evaluation metrics commonly used in binary and multi-class classification tasks to assess the performance of machine learning models.
Accuracy is the number of correct predictions made by the model.

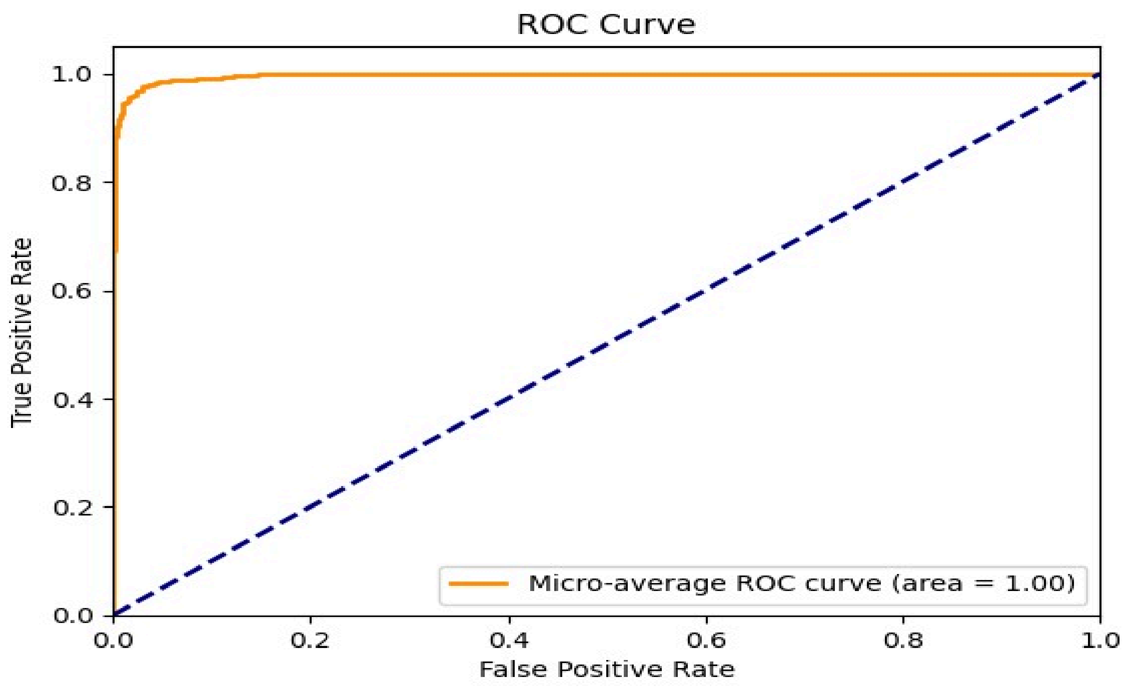
$$\textbf{\textit{Accuracy}} = \frac{TP + TN}{TP + TN + FP + FN}$$

The ROC curve is a graphical representation of the performance of a classification model across different threshold values. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) for various threshold settings. TPR (Sensitivity): The ratio of correctly predicted positive instances to all actual positive instances.

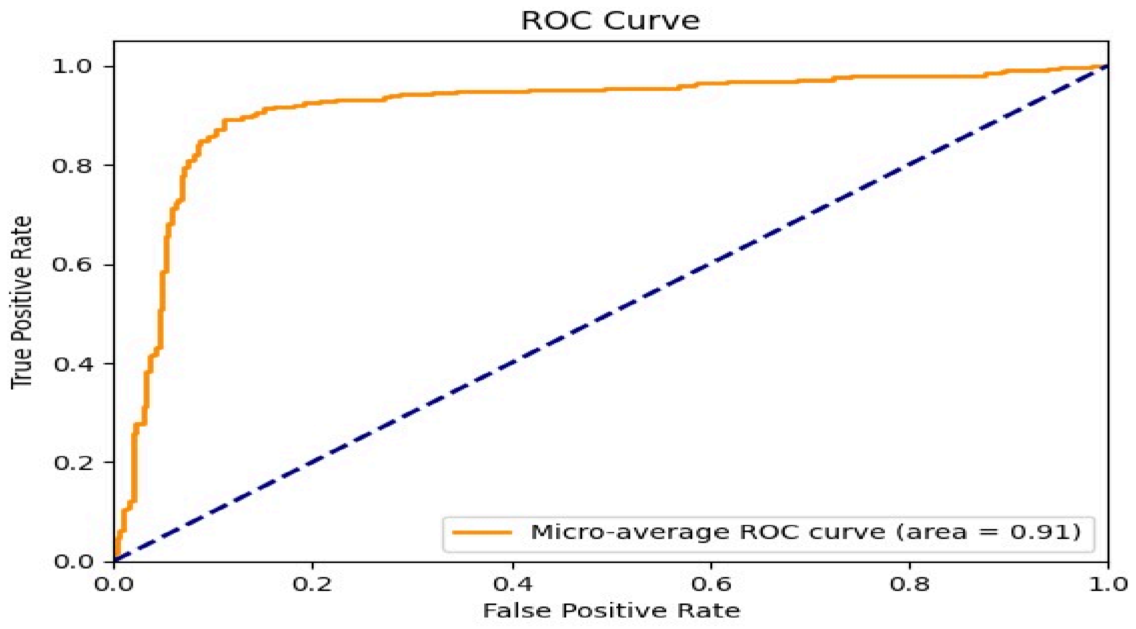
$$True\ Positive\ Rate = \frac{TP}{Actual\ Positive} = \frac{TP}{TP+FN}$$

FPR (1 - Specificity): The ratio of incorrectly predicted positive instances to all actual negative instances.

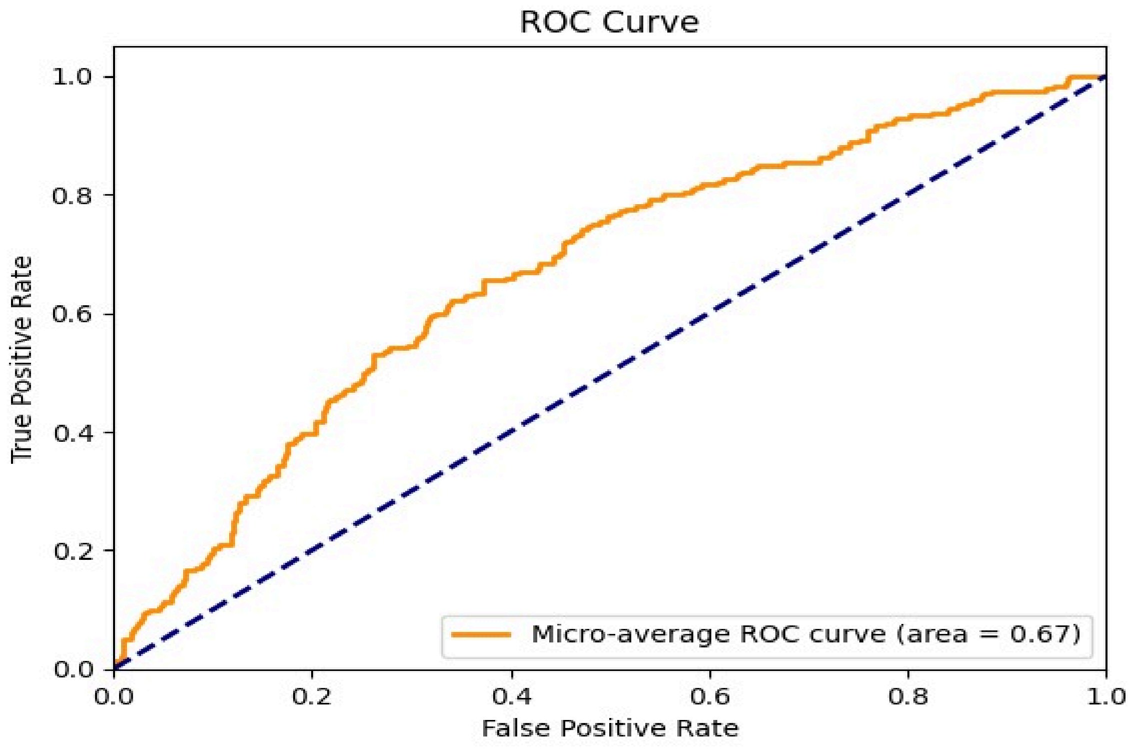
$$False\ Positive\ Rate = \frac{FP}{Actual\ Negative} = \frac{FP}{TN+FP}$$



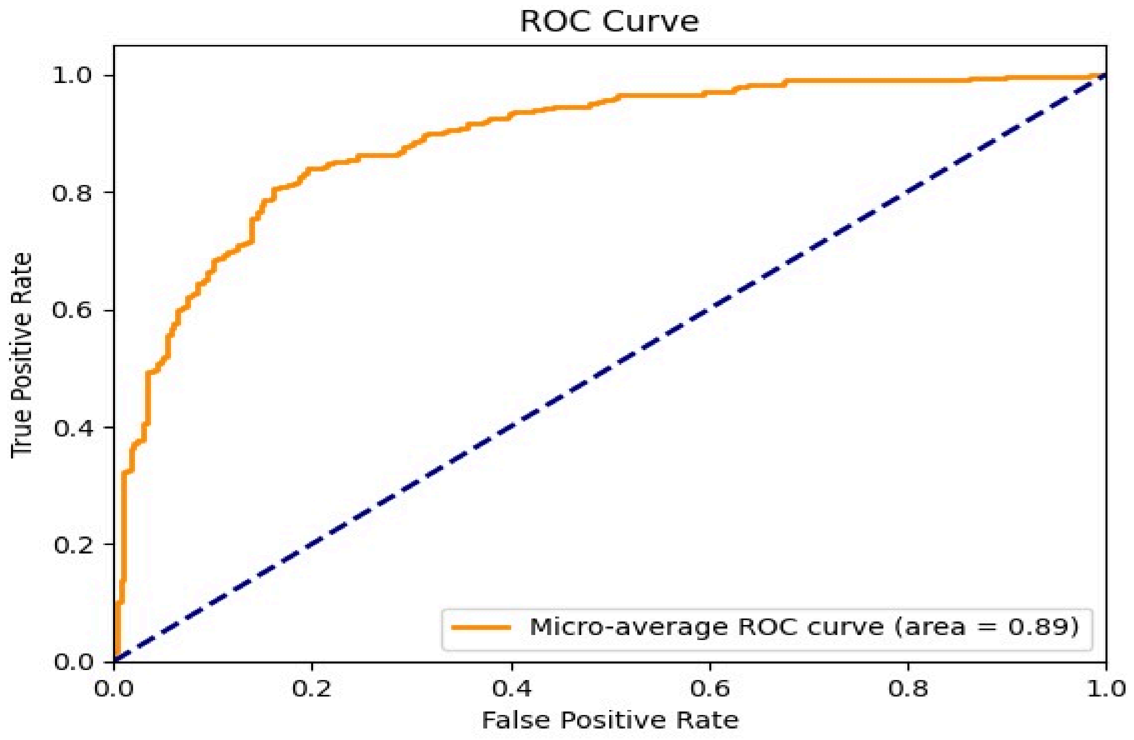
OrganMNIST3d (Figure - 4.2)



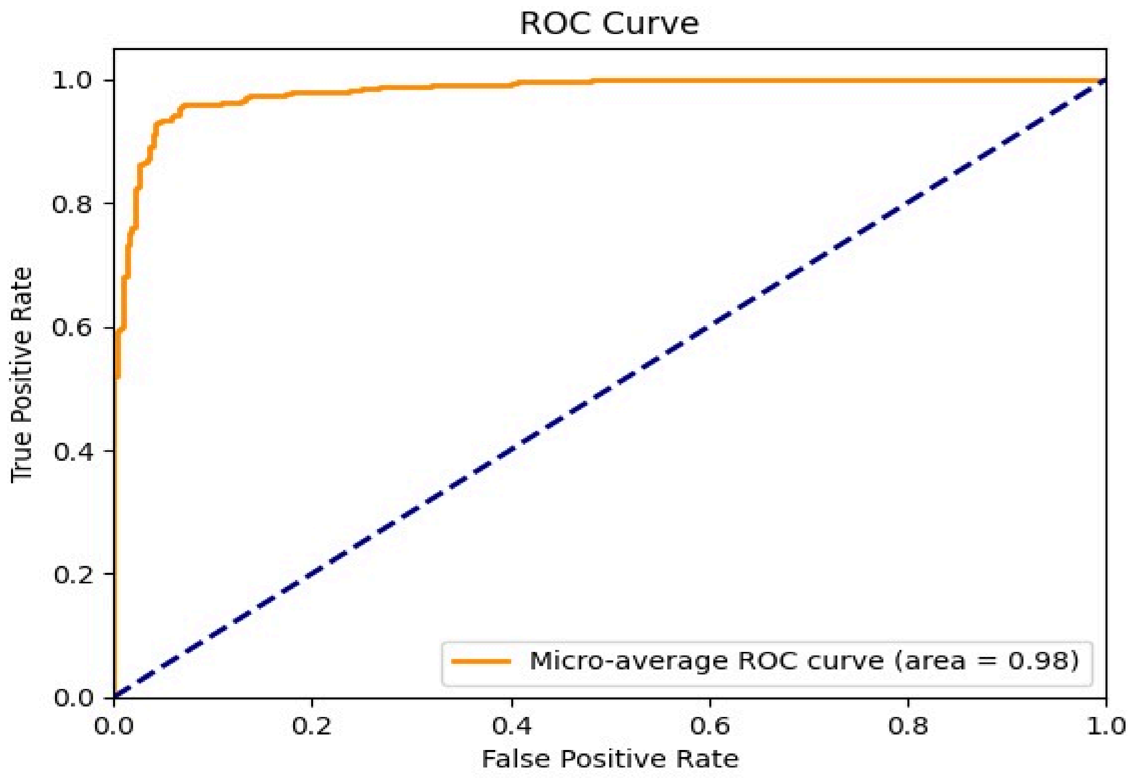
NoduleMNIST3d (Figure - 4.3)



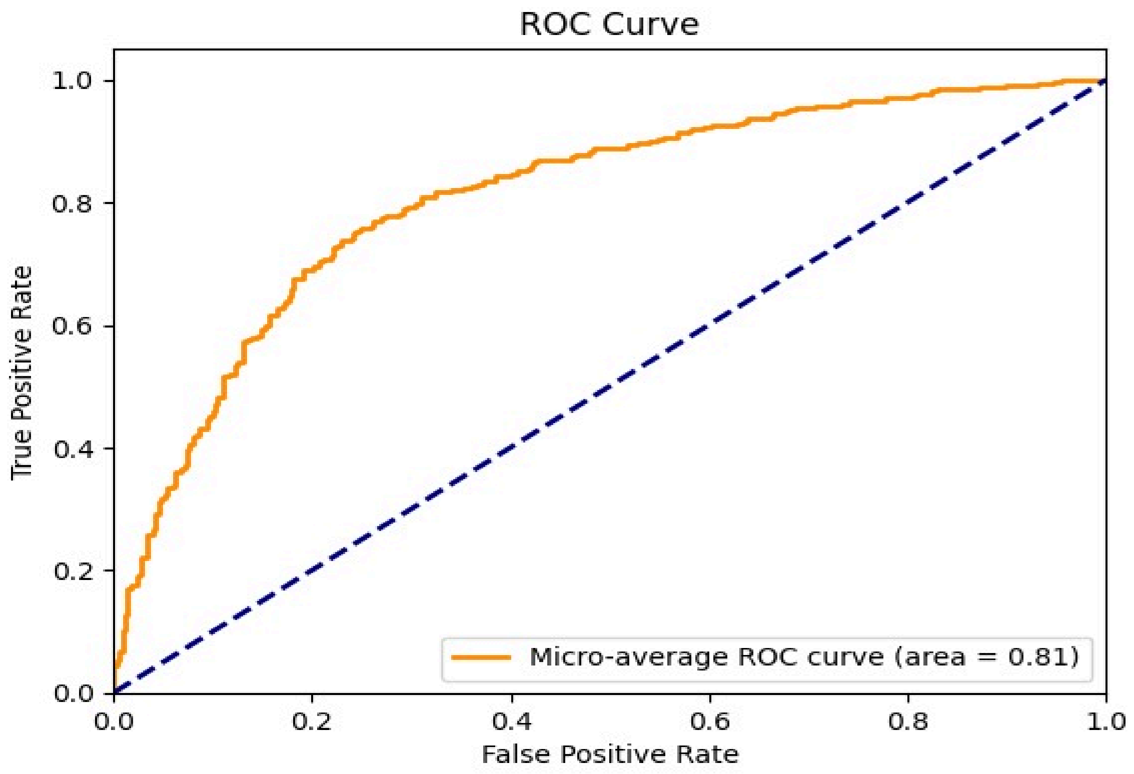
FractureMNIST3d (Figure - 4.4)



AdrenalMNIST3d (Figure - 4.5)

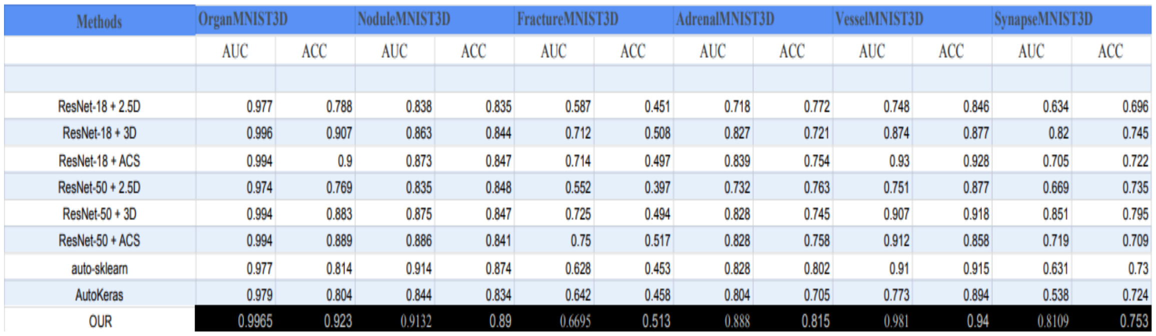


VesselMNIST3d (Figure - 4.6)



SynapseMNIST3d (Figure - 4.7)

AUC refers to the area under the ROC curve, which serves as a summary measure of a classification model's performance across various classification thresholds. It signifies the likelihood of the model prioritizing a randomly selected positive instance over a randomly chosen negative instance. AUC values fall between 0 and 1, with higher values indicating superior performance.
AUC = 1: Perfect classifier (all positive instances ranked higher than negative instances).AUC = 0.5: Random classifier (no discrimination between positive and negative instances).AUC < 0.5: Inverted classifier (worse than random, misclassified positive and negative instances). ROC curves help visualize the trade-off between sensitivity and specificity, aiding in the selection of an optimal classification threshold.



Methods	OrganMNIST3D		NoduleMNIST3D		FractureMNIST3D		AdrenalMNIST3D		VesselMNIST3D		SynapseMNIST3D	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
ResNet-18 + 2.5D	0.977	0.788	0.838	0.835	0.587	0.451	0.718	0.772	0.748	0.846	0.634	0.696
ResNet-18 + 3D	0.996	0.907	0.863	0.844	0.712	0.508	0.827	0.721	0.874	0.877	0.82	0.745
ResNet-18 + ACS	0.994	0.9	0.873	0.847	0.714	0.497	0.839	0.754	0.93	0.928	0.705	0.722
ResNet-50 + 2.5D	0.974	0.769	0.835	0.848	0.552	0.397	0.732	0.763	0.751	0.877	0.669	0.735
ResNet-50 + 3D	0.994	0.883	0.875	0.847	0.725	0.494	0.828	0.745	0.907	0.918	0.851	0.795
ResNet-50 + ACS	0.994	0.889	0.886	0.841	0.75	0.517	0.828	0.758	0.912	0.858	0.719	0.709
auto-sklearn	0.977	0.814	0.914	0.874	0.628	0.453	0.828	0.802	0.91	0.915	0.631	0.73
AutoKeras	0.979	0.804	0.844	0.834	0.642	0.458	0.804	0.705	0.773	0.894	0.538	0.724
OUR	0.9965	0.923	0.9132	0.89	0.6695	0.513	0.888	0.815	0.981	0.94	0.8109	0.753

Table 2. Comparison of proposed model with benchmark model

As we are using a 2D image classification model , we have fewer parameters in our model so it will take much less resources and time to train and test our model on the dataset of MedMNIST.
And because of transfer learning we can work with great precision even with small datasets.
Our model gives better performance compared to most of the benchmark models of MedMNIST. When averaged across 8 benchmark models and 6 datasets, our model gives better accuracy 95% of time and better AUC 85% of time. Our model gives best accuracy in four out of six datasets.

For OrganMNIST3D, our model outperforms all other benchmark models in terms of accuracy and AUC. For NoduleMNIST3D, our model outperforms every model except auto-sklearn in terms of accuracy and outperforms every model for AUC. For FractureMNIST3D, our model outperforms 4 models for accuracy and 7 models for AUC. Resnet-50 + ACS outperformed our model for AUC in FractureMNIST3D. For AdrenalMNIST3D, our model outperforms every other model in terms of accuracy and AUC. For VesselMNIST3D, our model again outperforms every benchmark model in terms of accuracy and AUC. For SynapseMNIST3D, our model outperforms 6 models for accuracy and 7 models for AUC. Resnet50+3D and Resnet18+3D outperforms our model in this dataset.

4.2: Conclusion

In summary, our approach demonstrates the effectiveness of transfer learning using a 2D image classification model for classifying 3D medical scans in the MedMNIST 3D dataset. By using transfer learning, we achieved reductions in computational resources and training time while maintaining accuracy. Since we are using transfer learning we have overcome the challenge caused by less data size. Our model consistently outperforms most benchmark models, achieving better accuracy and AUC across various datasets. Although there are instances where other models outperform ours, our approach shows robust performance and provides valuable insights for future work in medical scan classification.

References

1. Yang, J., Shi, R., Wei, D. *et al.* MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Sci Data* 10, 41 (2023).

2. Jiancheng Yang1,2 Rui Shi1 Bingbing Ni1,2 MedMNIST CLASSIFICATION DECATHLON: A LIGHTWEIGHT AUTOML BENCHMARK FOR MEDICAL IMAGE ANALYSIS 20 Jan 2020

3. Mingxing Tan, Quoc V. Le EfficientNetV2: Smaller Models and Faster Training Wed, 23 June 2021

4. Geert Litjens, Thijs Kooi, et al., "A survey on deep learning in medical image analysis," Medical image analysis, vol. 42, pp. 60–88, 2017.

5. Zheng, Z. and Jia, X., 2023. Complex Mixer for MedMNIST Classification Decathlon. arXiv preprint arXiv:2304.10054.

6. C. A. C. F. da Silva, P. B. C. Miranda and F. R. Cordeiro, "A New Grammar for Creating Convolutional Neural Networks Applied to Medical Image Classification," 2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Gramado, Rio Grande do Sul, Brazil, 2021, pp. 97-104, doi: 10.1109/SIBGRAPI54419.2021.00022.

7. Mingxing Tan, Quoc Le Proceedings of the 38th International Conference on Machine Learning, PMLR 139:10096-10106, 2021.

8. Zhou, Feng, Shijing Hu, Xiaoli Wan, Zhihui Lu, and Jie Wu. 2023. "Diplin: A Disease Risk Prediction Model Based on EfficientNetV2 and Transfer Learning Applied to Nursing Homes" Electronics 12, no. 12: 2581.

9. Torrey, Lisa, and Jude Shavlik. "Transfer Learning." In Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques, edited by Emilio Soria Olivas, et al., 242-264. Hershey, PA: IGI Global, 2010.

10. Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollár, P., 2017. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).