	ChestMNIST	BloodMNIST	BreastMNIST	DermaMNIST	OctMNIST	OrganAMNIST
Run-1	0.94771	0.96229	0.45513	0.70075	0.80500	0.92654
Run-2	0.94797	0.95645	0.80769	0.72865	0.76700	0.91681
Run-3	0.94767	0.95265	0.67308	0.71970	0.77500	0.92682
Run-4	0.94794	0.95557	0.67949	0.74963	0.77500	0.93008
Run-5	0.94739	0.95645	0.81410	0.73367	0.76700	0.92418
Mean	0.94774	0.95668	0.68590	0.72648	0.77780	0.92489
Standard Deviation	0.00021	0.00313	0.13081	0.001613	0.01406	0.00445

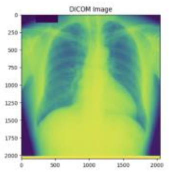
	OrganCMNIST	OrganSMNIST	PathMNIST	PneumoniaMNIST	RetinaMNIST	TissueMNIST
Run-1	0.90628	0.78940	0.91421	0.84135	0.51000	0.70362
Run-2	0.90287	0.78339	0.84345	0.87019	0.51250	0.69860
Run-3	0.89959	0.77943	0.88217	0.85096	0.52500	0.69966
Run-4	0.90202	0.78283	0.90432	0.85897	0.52000	0.70341
Run-5	0.90567	0.77490	0.86769	0.82212	0.52570	0.70207
Mean	0.90348	0.78181	0.88257	0.84492	0.51484	0.699907
Standard Deviation	0.00276	0.00567	0.02616	0.01472	0.00665	0.00220

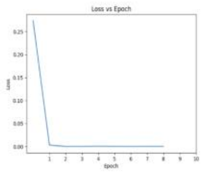
Table 1. Accuracy Metrics for MedMNIST-2D

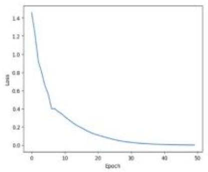
	ChestMNIST	BloodMNIST	BreastMNIST	DermaMNIST	OctMNIST	OrganAMNIST
Run-1	0.77304	0.99788	0.76044	0.88641	0.96913	0.99684
Run-2	0.77255	0.99758	0.83939	0.91156	0.95845	0.99546
Run-3	0.77240	0.99692	0.85589	0.88760	0.95960	0.99503
Run-4	0.77857	0.99703	0.85067	0.91853	0.96215	0.99606
Run-5	0.78028	0.99724	0.85589	0.90858	0.96245	0.99531
Mean	0.77353	0.99733	0.83246	0.902536	0.96235	0.99574
Standard Deviation	0.00336	0.00035	0.03651	0.013091	0.00227	0.00065

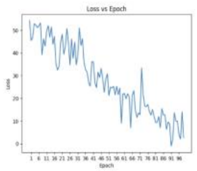
	OrganCMNIST	OrganSMNIST	PathMNIST	PneumoniaMNIST	RetinaMNIST	TissueMNIST
Run-1	0.99222	0.97879	0.99158	0.95560	0.71513	0.93873
Run-2	0.99184	0.97504	0.98551	0.96472	0.71334	0.93755
Run-3	0.99201	0.99201	0.98603	0.95628	0.72290	0.93810
Run-4	0.99194	0.97640	0.99007	0.96397	0.72390	0.93838
Run-5	0.99235	0.97520	0.98505	0.95244	0.72472	0.93810
Mean	0.99215	0.98149	0.98745	0.95860	0.71980	0.93757
Standard Deviation	0.00028	0.00729	0.00271	0.00424	0.00439	0.00064

Table 2. AUC Metrics for MedMNIST-2D


Dicom Image Conversion


Directions


Gender


Age


Lung Segmentation

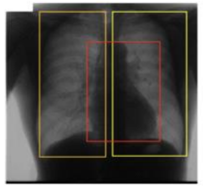
	AdrenalMNIST	FractureMNIST	NoduleMNIST	OrganMNIST	SynapseMNIST	VesselMNIST
Run-1	0.84228	0.57083	0.84839	0.87839	0.73864	0.88743
Run-2	0.82550	0.56250	0.85161	0.88361	0.74716	0.47644
Run-3	0.66779	0.52917	0.85161	0.88033	0.73011	0.51571
Run-4	0.75168	0.63333	0.85161	0.87705	0.73864	0.41623
Run-5	0.67450	0.53333	0.84516	0.90492	0.75000	0.89791
Mean	0.75233	0.56582	0.86980	0.88466	0.74084	0.53874
Standard Deviation	0.0768	0.0373	0.0203	0.0133	0.0071	0.2336

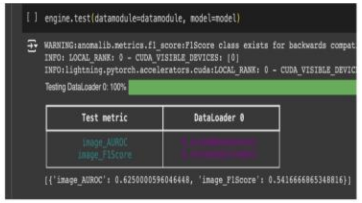
Table 3. Accuracy Metrics for MedMNIST-3D

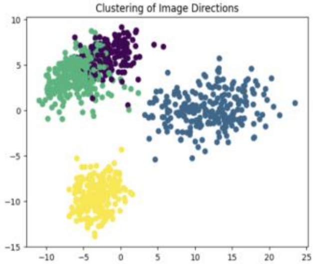
	AdrenalMNIST	FractureMNIST	NoduleMNIST	OrganMNIST	SynapseMNIST	VesselMNIST
Run-1	0.88577	0.73658	0.88910	0.99102	0.67110	0.76628
Run-2	0.88083	0.76245	0.86287	0.99185	0.70649	0.81622
Run-3	0.86222	0.72348	0.88955	0.99073	0.71624	0.78754
Run-4	0.83286	0.77021	0.86935	0.99291	0.70551	0.81889
Run-5	0.85102	0.72861	0.90898	0.99386	0.71763	0.83467
Mean	0.88254	0.744847	0.88385	0.99207	0.70399	0.78412
Standard Deviation	0.02182	0.02045	0.01725	0.00103	0.02131	0.02531

Table 4. AUC Metrics for MedMNIST-3D


Organ Segmentation


Localization


Identify Unusual Flips


Clustering as per Direction

Poster 7: Integrating classification and segmentation for chest X-Rays

Jinal Vyas

Classification

Dataset – ChestXRay-14

ChestXray-14 is a large-scale dataset containing over 112,000 frontal-view X-ray images labeled with 14 different thoracic diseases, such as pneumonia, emphysema, and cardiomegaly. It's widely used for developing automated chest disease classification models.

Model – ConvNeXt (convnext_base from PyTorch)

ConvNeXt is a modern convolutional neural network (CNN) architecture inspired by advancements in vision transformers. It retains the simplicity of CNNs while incorporating key ideas from transformer models, such as larger kernel sizes and layer normalization, making it highly effective for image classification tasks.

SOTA Baseline (AUC) – 0.8429

AUC Score Achieved with Pretrained Weights – 0.7970

Learning Rate	1e-5
Loss Function	BCEWithLogitLoss
Batch Size	16
Image Size	224x224
Weight Decay	1e-4
Optimizer	Adam

Segmentation

Dataset - ChestXDet

ChestXDet is a dataset designed for thoracic disease detection and localization, containing bounding boxes and I created segmentation masks for various abnormalities in chest X-ray images from the coordinates given. It's used to develop models that not only classify diseases but also segment and localize abnormalities with precision.

Model – UperNet (with Swin Backnone; from HuggingFace; Pretrained weights of swin bacbone trained for ImageNet are used)

UPerNet (Unified Perceptual Parsing Network), combined with a **Swin Transformer backbone**, is a powerful architecture for semantic segmentation. The Swin Transformer captures multi-scale hierarchical features, while UPerNet efficiently fuses them for pixel-level predictions.

After training of 60 epochs, it started overfitting between 20-25th epochs and the best testing IoU recorded was 0.42.

Learning Rate	1e-4
Loss Function	IoU Loss
Batch Size	8
Weight Decay	1e-4
Optimizer	Adam
Metrics	IoU

Integrating Classification and Segmentation (Implementation Completed)

UPerNet Overview with Swin Backbone

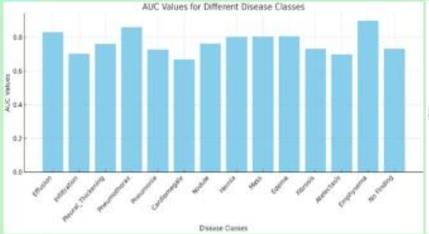
- **Encoder:** The Swin Transformer extracts features at multiple levels.
- **Segmentation Decoder (UPerNet Head):** Combines these features for segmentation.
- Add a **classification head** after the Swin backbone or UPerNet's encoder.

Architecture Modifications

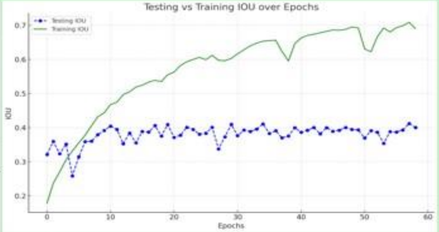
- **Shared Encoder** (Swin Transformer): Extracts multi-scale feature maps.
- Segmentation Head (UPerNet): Produces segmentation masks.
- Classification Head: A fully connected layer or MLP applied to the global pooled features from the last Swin stage

Loss Functions

- **Segmentation Loss**: Use Dice Loss or IoU Loss
- **Classification Loss**: Use BCEWithLogitsLoss (multi-label)



Classification Output AUC of 14 classes



Segmentation Output IoU vs Epochs

Listing 1. Combined Segmentation + Classification Model Code

```
# Combined Model
class CombinedModel(nn.Module):
    def __init__(self, num_classes):
        super(CombinedModel, self).__init__
            ()
        self.segmentation_model =
            segmentation_model
        self.classification_head =
            convnext_base(pretrained=True)
        self.classification_head.classifier
            [2] = torch.nn.Linear(
            in_features=self.
            classification_head.classifier
            [2].in_features, out_features=
            num_classes)
        self.channel_converter = nn.Conv2d
            (2, 3, kernel_size=1)
    def forward(self, x):
        seg_outputs = self.
            segmentation_model(x)
        print('here')
        pooled_output = self.
            channel_converter(seg_outputs['
            logits'])
        class_logits = self.
            classification_head(
            pooled_output)
        return seg_outputs, class_logits
```

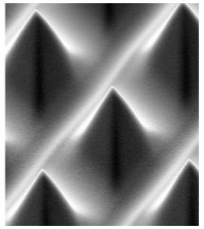


Figure 12. Unprocessed Image


Figure 13. Image Processed Using Otsu

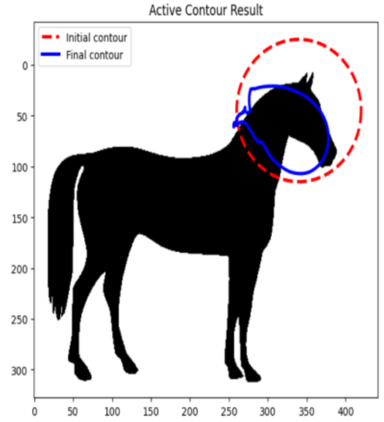


Figure 11. Snake Contours

Poster 7: Integrating classification and segmentation for chest X-Rays

Jinal Vyas

Classification

Dataset – ChestXRay-14

ChestXray-14 is a large-scale dataset containing over 112,000 frontal-view X-ray images labeled with 14 different thoracic diseases, such as pneumonia, emphysema, and cardiomegaly. It's widely used for developing automated chest disease classification models.

Model – ConvNeXt (convnext_base from PyTorch)

ConvNeXt is a modern convolutional neural network (CNN) architecture inspired by advancements in vision transformers. It retains the simplicity of CNNs while incorporating key ideas from transformer models, such as large kernel sizes and layer normalization, making it highly effective for image classification tasks.

SOTA Baseline (AUC) – 0.8429

AUC Score Achieved with Pretrained Weights – 0.7970

Learning Rate	**1e-5**
Loss Function	BCEWithLogitLoss
Batch Size	16
Image Size	224x224
Weight Decay	1e-4
Optimizer	Adam

Segmentation

Dataset - ChestXDet

ChestXDet is a dataset designed for thoracic disease detection and localization, containing bounding boxes and I created segmentation masks for various abnormalities in chest X-ray images from the coordinates given. It's used to develop models that not only classify diseases but also segment and localize abnormalities with precision.

Model – UperNet (with Swin Backnone; from HuggingFace; Pretrained weights of swin bacbone trained for ImageNet are used)

UPerNet (Unified Perceptual Parsing Network), combined with a **Swin Transformer backbone**, is a powerful architecture for semantic segmentation. The Swin Transformer captures multi-scale hierarchical features, while UPerNet efficiently fuses them for pixel-level predictions.

After training of 60 epochs, it started overfitting between 20-25th epochs and the best testing IoU recorded was 0.42.

Learning Rate	**1e-4**
Loss Function	IoU Loss
Batch Size	8
Weight Decay	1e-4
Optimizer	Adam
Metrics	IoU

Integrating Classification and Segmentation (Implementation Completed)

UPerNet Overview with Swin Backbone

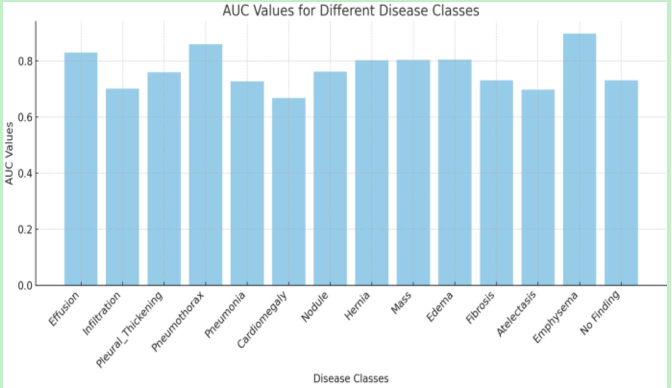
- **Encoder**: The Swin Transformer extracts features at multiple levels.
- **Segmentation Decoder (UPerNet Head)**: Combines these features for segmentation.
- Add a **classification head** after the Swin backbone or UPerNet's encoder.

Architecture Modifications

- **Shared Encoder** (Swin Transformer): Extracts multi-scale feature maps.
- Segmentation Head (UPerNet): Produces segmentation masks.
- Classification Head: A fully connected layer or MLP applied to the global pooled features from the last Swin stage

Loss Functions

- **Segmentation Loss**: Use Dice Loss or IoU Loss
- **Classification Loss**: Use BCEWithLogitsLoss (multi-label)



Classification Output
AUC of 14 classes

Segmentation Output
IoU vs Epochs