A. MODEL SUMMARY

A1. Background on you

Competition Name: RSNA 2024 Lumbar Spine Degenerative Classification
(https://www.kaggle.com/competitions/rsna-2024-lumbar-spine-degenerative-classification)

Team Name: HLIP
Private Leaderboard Score: 0.401322
Private Leaderboard Place: rank 7th

A2. Background on you/your team

What is your academic/professional background?
I am a contract computer vision and deep learning algorithm engineer. My job includes discovering fracture in x-ray images and implementing visual slam for robotic navigation. Recently,I am helping companies to fine-tune LLM models for their applications.

Did you have any prior experience that helped you succeed in this competition?
I am familiar deep learning and build deep models in my work. I took part in previous RSNA competitions before like RSNA 2023 Abdominal Trauma Detection and RSNA 2022 Cervical Spine Fracture Detection. These are helpful for me.

What made you decide to enter this competition?
The problem of of analyzing 3d MRI volume scan is interesting for me. I am general interested in medical image analysis and computer vision problem.

How much time did you spend on the competition?
About 5 hours per day, over two months.

A3. Summary

In this competition, our task is to predict lumbar spine degenerative conditions from given 3d volume MRI scans. Since my teammate @lihaoweicvch already has very good two-stage model, my task is to design one-stage model to improve his results.

While two-stage model uses "crop and classify" method, one-stage models uses "point-masking and pool" instead. Figure.1 and Figure.2 shows deep net architecture for NFN (neural foraminal narrowing) and SCS (spinal canal stenosis) detection respectively. In summary, the main processing layers for both architectures are:

1. Predict the 5 lumbar level point pixel-wise heatmap for the L1/L2 to L5/S1. For learning the net parameter, we use the Jensen-Shannon divergence loss function.

2. Convert point pixel-wise heatmap to target xyz coordinates using the differentiable spatial-to-numerical transform (DSNT) from in paper [1].

3. Predict the 3 spine degeneration grade pixel-wise probability for the classes Normal/Mild, Moderate, or Severe.

4. Use the point pixel-wise heatmap to pool the grade pixel-wise probability using multiplication and summing over volume or area. The pooled results is the target level-wise grade conditions for submission,

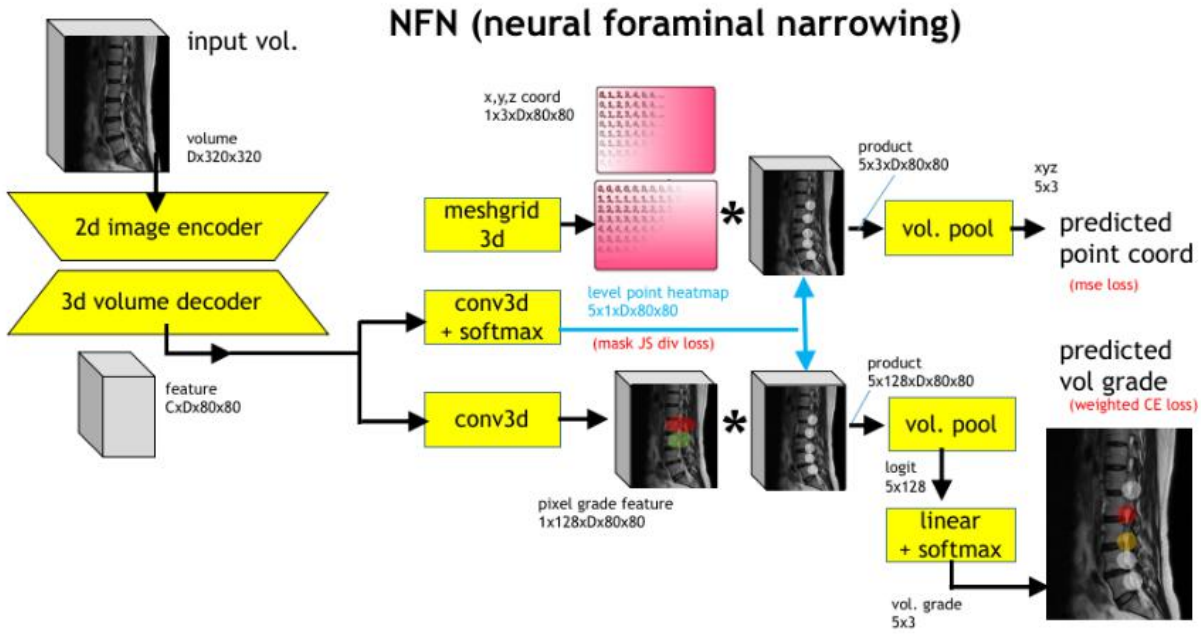

Figure.1 Deep net architecture for NFN (neural foraminal narrowing) detection

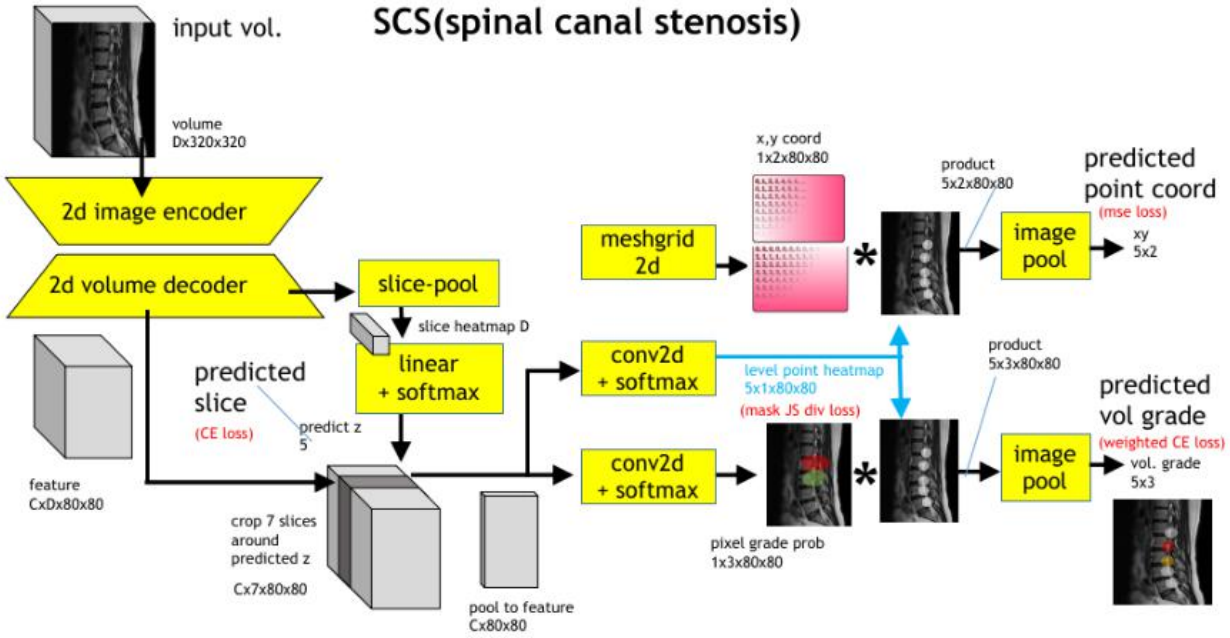

Figure.2 Deep net architecture for SCS (spinal canal stenosis) detection

A4. Features Selection / Engineering

1. 1. Image encoder

We are using pyramid transformer PVTv2 [2] as image encoder backbone for submission. They have very large context which we think are important to get good results. For example, both left and right level are likely to have the same spine degeneration conditions.

2. Point-masking and pooling

This is first mentioned in Section A.3 Summary. The level point heatmap can be seen as learned attention map to select with pixel we should look at for spine degeneration conditions.

A5. Training Method(s)

Training is straight forward with multi-task loss setup for each net in Figure.1 and 2:

- pixel-wise level point : Jensen-Shannon divergence loss
- pixel-wise grade probability : Cross entropy loss
- volume-wise level grade probability : weighted cross entropy loss

We perform back propagation with ADAM optimizer. The training hyper-parameters varies for different net due to the difficulty of the problem. Learning rate is from 1e-4 to 1e-5 and training epochs ranges from 50 to 100. Batch sizes is from 3 to 8 volumes.

For hardware, we use one Nvidia Ada A6000/48GB Ampere GPU. Powerful GPU makes our training fast and large memory enables us to train with whole 3d volume.

Some observations from training are:

- We can temporarily modify the network to use the ground truth level point heatmap and focus solely on learning grade prediction. This serves as an upper bound for grade prediction performance.

- One-stage networks are challenging to train because the optimal parameters for coordinate prediction are not necessarily the best for grade prediction. This can be confirmed by observing the loss for level and grade during training iterations. They seem to be competing with each other.

We believe one reason for this is labeling errors, particularly the significant confusion between L5 and S1 level labels. Additionally, the point labels themselves are not very precise.

A6. Interesting findings

Figure.3 and 4 shows the local validation results of the nets. We use 5-fold validation.

sagittal-t1 (single stage joint grade and point prediction)

NFN	grade				
	bugged #1	fixed			
	320	320	320	384	
	pvtv2-b4	pvtv2-b4	convnext-	effnet-b5	ensemble
fold0	0.487	**0.473**	0.4777	0.4860	0.4664
fold1	0.524	**0.504**	0.5003	0.4960	0.4848
fold2	0.507	**0.500**	0.4950	0.4802	0.4802
fold3	0.521	**0.520**	0.5182	0.5116	0.5014
fold4	0.468	**0.465**	0.4640	0.4641	0.4591
avg	0.501	**0.4923**	**0.4910**	**0.4876**	**0.4784**

	ensemble							
	x<1	x<2	x<5	y<1	y<2	y<5	z<1	z<5
fold0	0.9673	0.9894	0.9998	0.8963	0.9560	0.9771	0.9747	1.0000
fold1	0.9598	0.9847	0.9992	0.8829	0.9537	0.9729	0.9698	0.9987
fold2	0.9693	0.9892	0.9992	0.9054	0.9644	0.9798	0.9706	0.9973
fold3	0.9702	0.9948	1.0000	0.9036	0.9687	0.9876	0.9715	0.9948
fold4	0.9676	0.9903	0.9998	0.9030	0.9656	0.9812	0.9718	0.9973
avg								

with tta (flip left-right condition)

	320	320	384	
	pvtv2-b4	convnext-	effnet-b5	ensemble
fold0	0.4710	0.4743	0.4835	0.4663
fold1	0.5041	0.4992	0.4910	0.4852
fold2	0.4988	0.4939	0.4822	0.4782
fold3	0.5159	0.5166	0.5095	0.5018
fold4	0.4648	0.4603	0.4591	0.4467
	0.4909	**0.4889**	**0.4851**	**0.4757**

Note:
- x,y err is normalised at feature size 80x80
- high threshold (x,y<5) error could be due to label error, e.g. L5/S1 confusion
- z corresponds to dicom slice instance numer

- #1 bug: we forget to reorder the point in left-right flipping augmentation at training. This is discover at last day before submission deadline.
- gray cell: used in submission to get final ranking in private LB

Figure.3 Local validation for NFN (neural foraminal narrowing) detection

sagittal-t2 (single stage joint grade and point prediction)

SCS	grade			
	320	320	384	
	pvtv2-b4	convnext_	effnet-b4	ensemble
fold0	0.2443	0.2498	0.2447	0.2376
fold1	0.2465	0.2520	0.2741	0.2483
fold2	0.2281	0.2233	0.2406	0.2209
fold3	0.3317	0.3244	0.3239	0.3160
fold4	0.2949	0.2866	0.2876	0.2807
avg	**0.2691**	**0.2672**	**0.2742**	**0.2607**

	ensemble							
	x<1	x<2	x<4	y<1	y<2	y<4	z<1	z<3
fold0	0.9721	0.9980	0.9990	0.9477	0.9775	0.9858	0.9941	0.9995
fold1	0.9661	0.9929	1.0000	0.9256	0.9722	0.9833	0.9843	0.9965
fold2	0.9754	0.9936	0.9995	0.9406	0.9770	0.9888	0.9872	1.0000
fold3	0.9683	0.9934	1.0000	0.9371	0.9760	0.9887	0.9882	0.9995
fold4	0.9787	0.9985	0.9995	0.9485	0.9847	0.9921	0.9906	0.9990
avg	**0.9721**	**0.9953**	**0.9996**	**0.9399**	**0.9775**	**0.9877**	**0.9889**	**0.9989**

any severe

	320	320	384	
	pvtv2-b4	convnext_	effnet-b4	ensemble
fold0	0.1877	0.1926	0.2001	0.1877
fold1	0.2162	0.2101	0.2392	0.2191
fold2	0.1788	0.1943	0.2001	0.1842
fold3	0.3182	0.2863	0.3078	0.3008
fold4	0.2971	0.3244	0.3088	0.3019
avg	**0.2396**	**0.2416**	**0.2512**	**0.2387**
		grade + any severe		**0.2497**

Figure.4 Local validation for SCS (spinal canal stenosis) detection

Figure.5 is the leader board submission scores results.

submission			
2stage : @lhw base model : 2 stage crop and classify			
	private	**public lb**	
2stage	0.4046	0.3500	
2stage + 1stage(nfn: pvtv2-v4-bugged)	0.4021	0.3481	
2stage + 1stage(nfn: pvtv2-v4-bugged + two fixed)	0.4013	0.3467	(7-th rank)
post submission			
2stage + 1stage(nfn: pvtv2-v4-fixed)	0.4011	0.3478	
2stage + 1stage(nfn: ensemble)	0.4000	0.3466	

Figure.5 Public and private leader board submission scores

We had a bug in flip augmentation at train. We forgot to reorder the left right level points (i.e. at training the left is mistakenly labeled as right and vice versus for 50% of the time). We discovered the bug just one day before submission and was unable to correct all models for submission. Hence the final submission consisted of both bugged and unbugged models.

We later make post submission after the competition deadline with unbugged models only and showed better results. Next, we summarize our observation:

1. One-stage NFN model indeed improves results. The baseline two stage has private /public score of 0.406/0.3500.

2. Even with bugged models only, the improvement is 0.402/0.3481.

3. For unbugged models, we have the best score of 0.400/0.3466.

4. One stage SCS models did not improve results and are not included for final submission. The reasons are unclear. When alone, one stage SCS model are have comparable accuracy as the two-stage counterparts. However after ensemble, the combined results falls a little.

What was the most important trick you used?
One stage model improves two stage model in ensemble. Blending for submission is:
0.75*2stage + 0.25*1stage

What do you think set you apart from others in the competition?
None of the top winning team uses one-stage model.

A7. Simple Features and Methods

Many customers are happy to trade off model performance for simplicity. With this in mind:
Is there a subset of features that would get 90-95% of your final performance?

From our report it is sufficient just to use only two stage mode.There will be drop of 5% accuracy though.

A8. Model Execution Time

How long does it take to train your model?
Training takes 6 hours for one net for each one fold, using Nvidia Ada A6000 Ampere GPU with memory 48 GB.

How long does it take to generate predictions using your model?
For one stage model, It take less than a minute to process one volume of MRI scan.

How long does it take to train the simplified model (referenced in section A7)?
I expect a reduction of more than 50% in time.

How long does it take to generate predictions from the simplified model?
I expect a reduction of more than 50% in time.

A9. References

[1] “Numerical Coordinate Regression with Convolutional Neural Networks” - Aiden Nibali, Arvix 2018, https://arxiv.org/abs/1801.07372

[2] “PVT v2: Improved Baselines with Pyramid Vision Transformer” - Wenhai Wang https://arxiv.org/abs/2106.13797