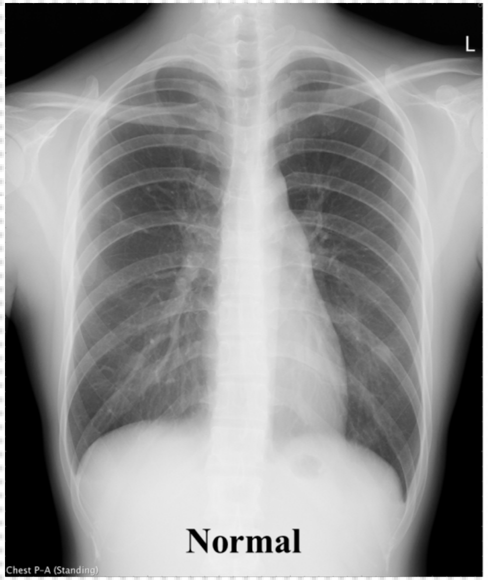
多標籤深度學習分類於胸部X光影像之應用

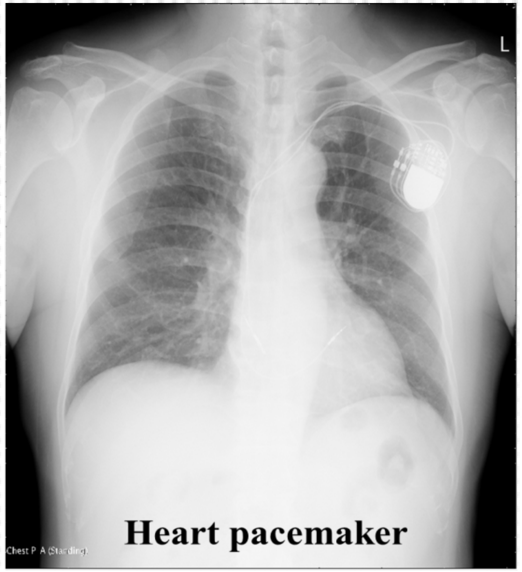
Multi-label deep learning classification of chest x-rays

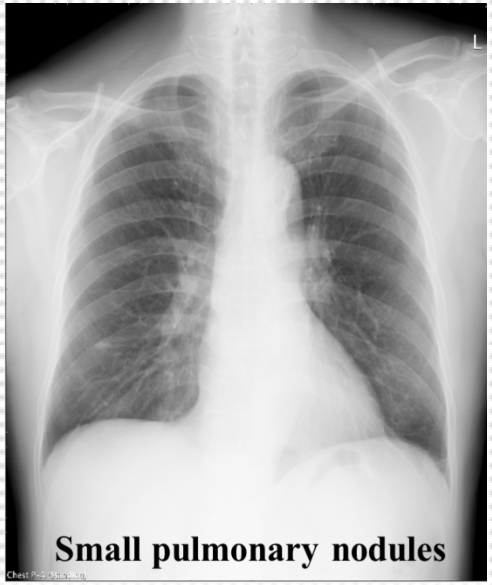
傅琦佳、黃冠華
國立交通大學 統計學研究所

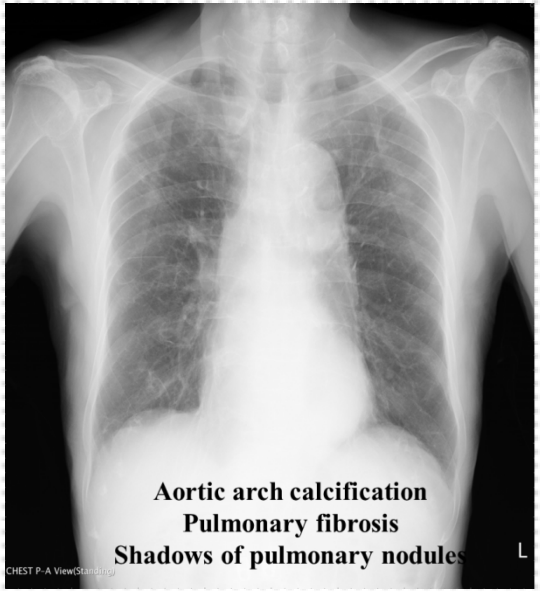
2020/08/21

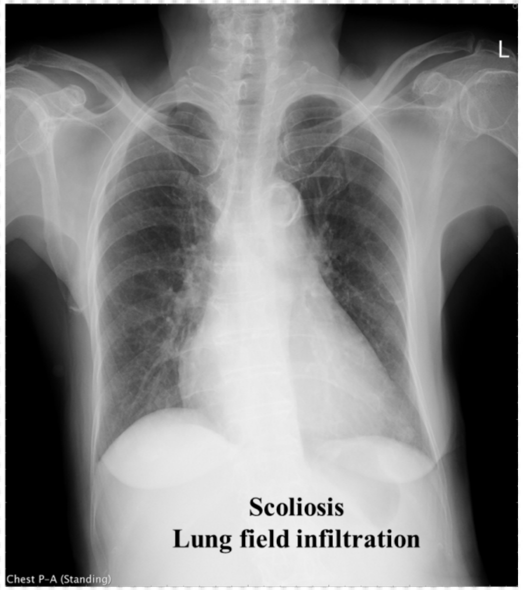
Chest x-ray images











Introduction

Objective To build a computer-aided diagnosis system for chest x-rays

Chance & Challenge

- The demand for medical image analysis is higher and the burden on the medical system is increasing
- Computer-aided diagnosis system is superior to human-based approaches (more efficient, more accurate, regardless of radiologist experience)

- Deep learning is data-hungry, while medical image data is rare. (tough and expensive to collect or label)
- Chest x-rays' size is large (1024x1024) but the lesion area is small, with multiple diseases in one image

Contribution

Use transfer learning technique to borrow information from large publicly available data (ImageNet & ChestX-ray8) to enhance the performance of deep learning prediction in our small-sized data

Introduction

Dataset Target data

Label	Categories	Sample Size	Subcategory	Sample Size
normal	normal	1314	normal	1314
diseases	aortic sclerosis/calcification	91	aortic arch atherosclerotic plaque	28
			aortic arch calcification	16
			aortic atherosclerosis	25
			aortic wall calcification	22
	arterial curvature	96	Aortic curvature	67
			Thoracic vertebral artery curvature	29
	abnormal lung fields	33	small pulmonary nodules	5
			shadows of pulmonary nodules	8
			tuberculosis	5
			pulmonary fibrosis	15
	increased lung patterns	154	increased lung streak	24
			lung field infiltration	85
			obvious hilar	45
	spinal lesions	151	degenerative joint disease of the thoracic spine	76
			scoliosis	75
	intercostal pleural thickening	36	intercostal pleural thickening	36
	cardiac hypertrophy	42	cardiac hypertrophy	42
	heart pacemaker placement	7	heart pacemaker placement	7

Source: E-Da hospital

Sample size: 1924

Sample category: 19 → 9

Image resolution: 0.16 mm per pixel

Image format: DICOM

Image size: 1824~2688 pixels in length

1536~2680 pixels in width

Introduction

Dataset Source data

Name	Source	Size	Class	Feature
ImageNet	Open database	14 million+	20000+	large and diverse
ChestX-ray8	Open database (NIH)	121,010	15	Medium-sized but similar to target data

ChestX-ray8：

Sample size: 121,010

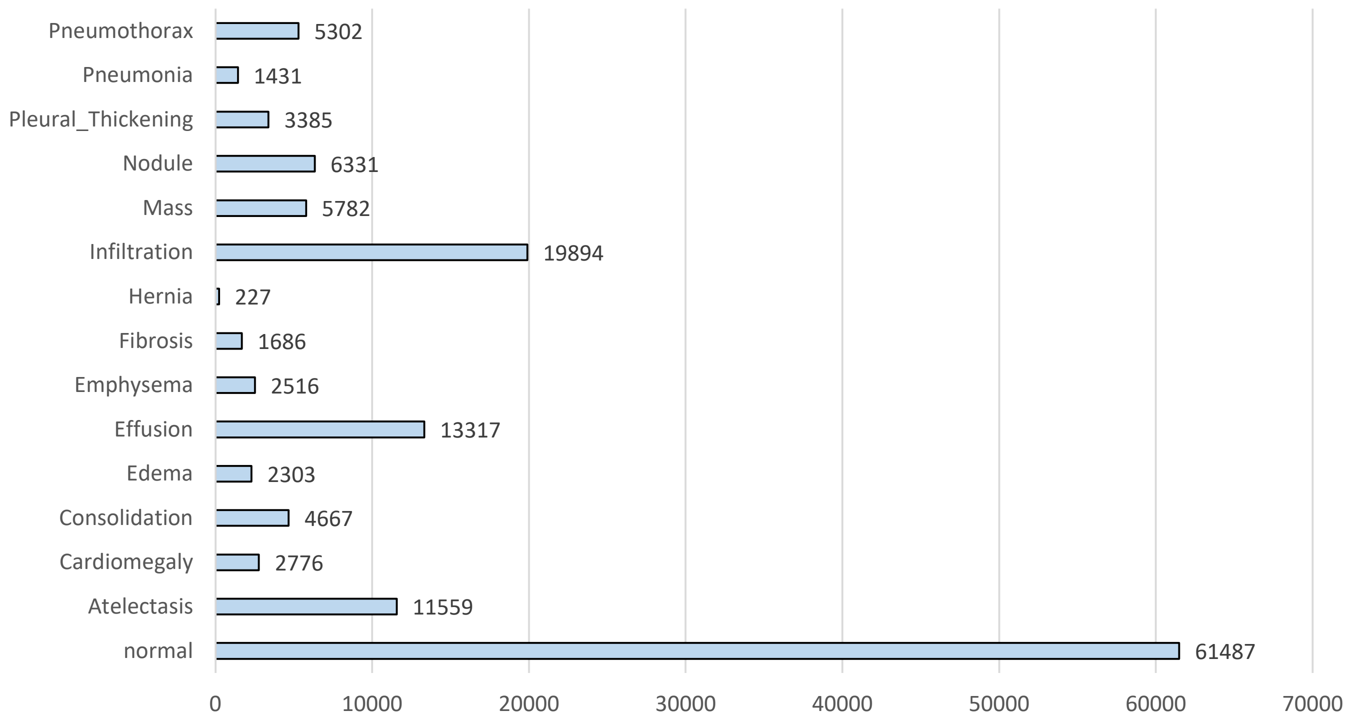
Sample category: normal + 14 diseases

Image format: PNG

Image size: 1024x1024 pixels

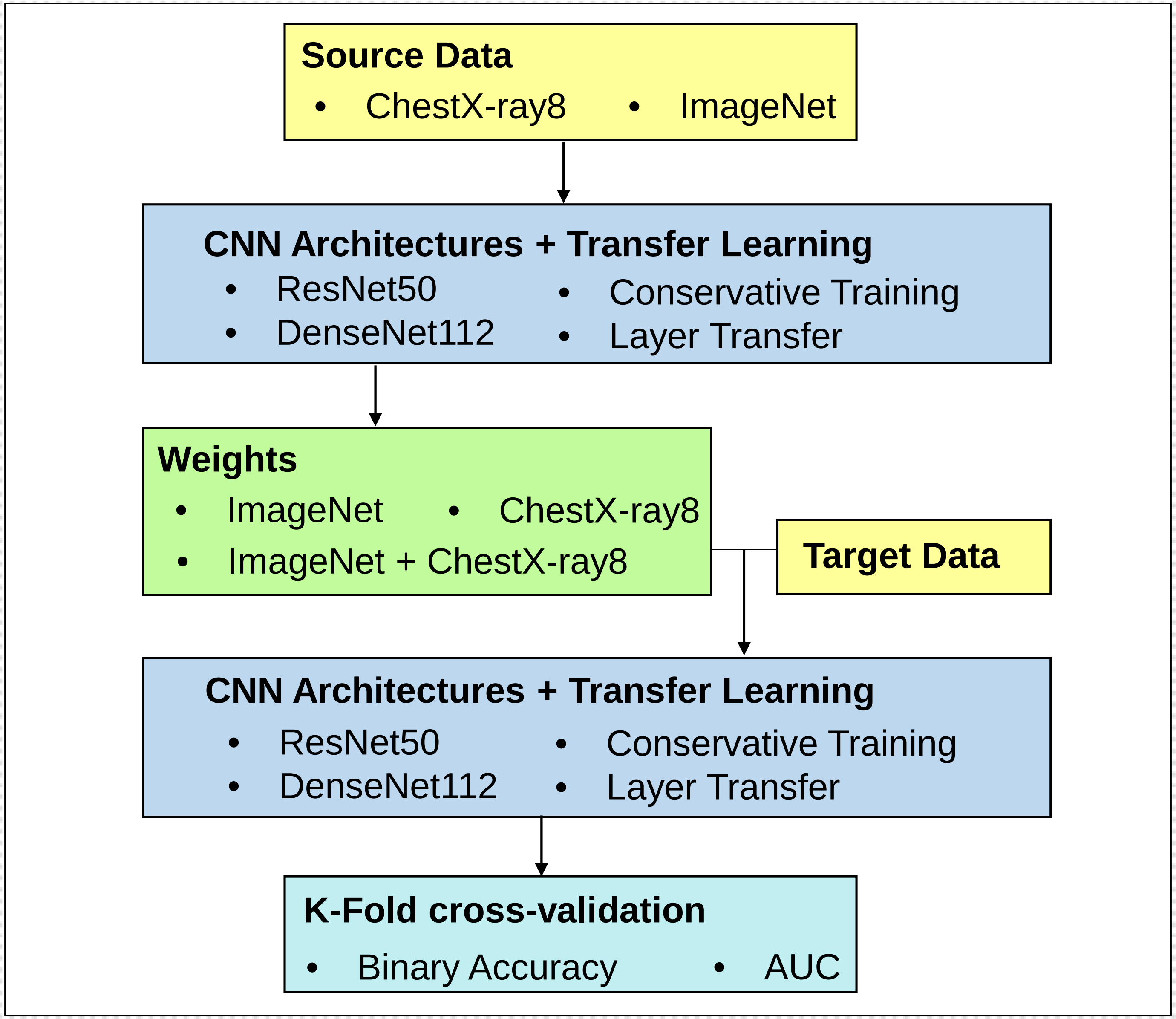
Download source:

https://nihcc.app.box.com/v/ChestXray-NIHCC/folder/36938765345



Methods

Structure



Methods

Preprocessing

Data preprocessing

- Set unique ID for each image
- Discard duplicates and outliers
- Delete the least class
- Use one-hot to encode disease labels

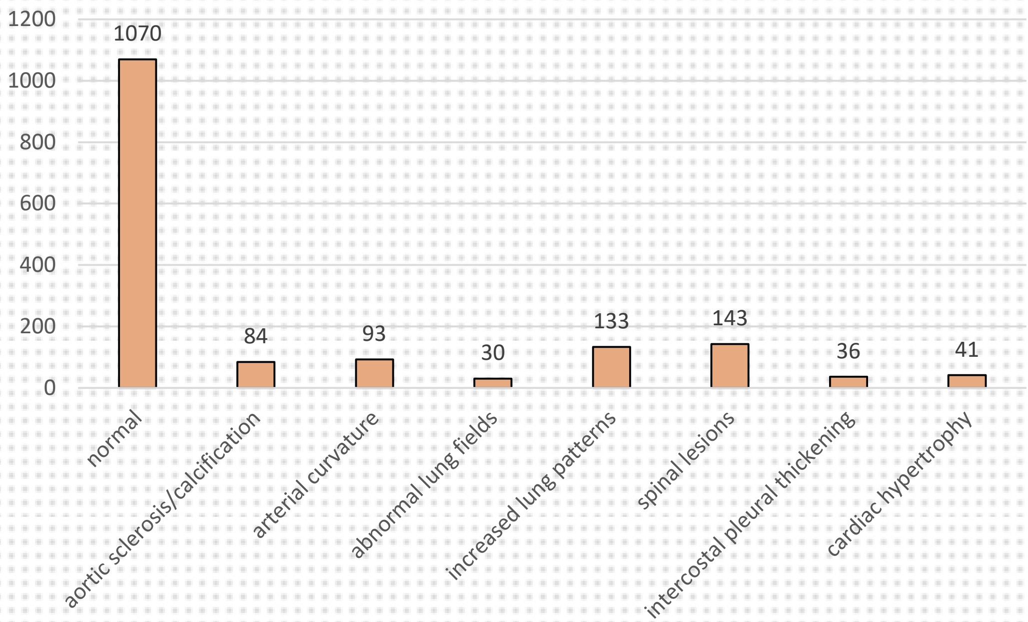


Image preprocessing

For target data

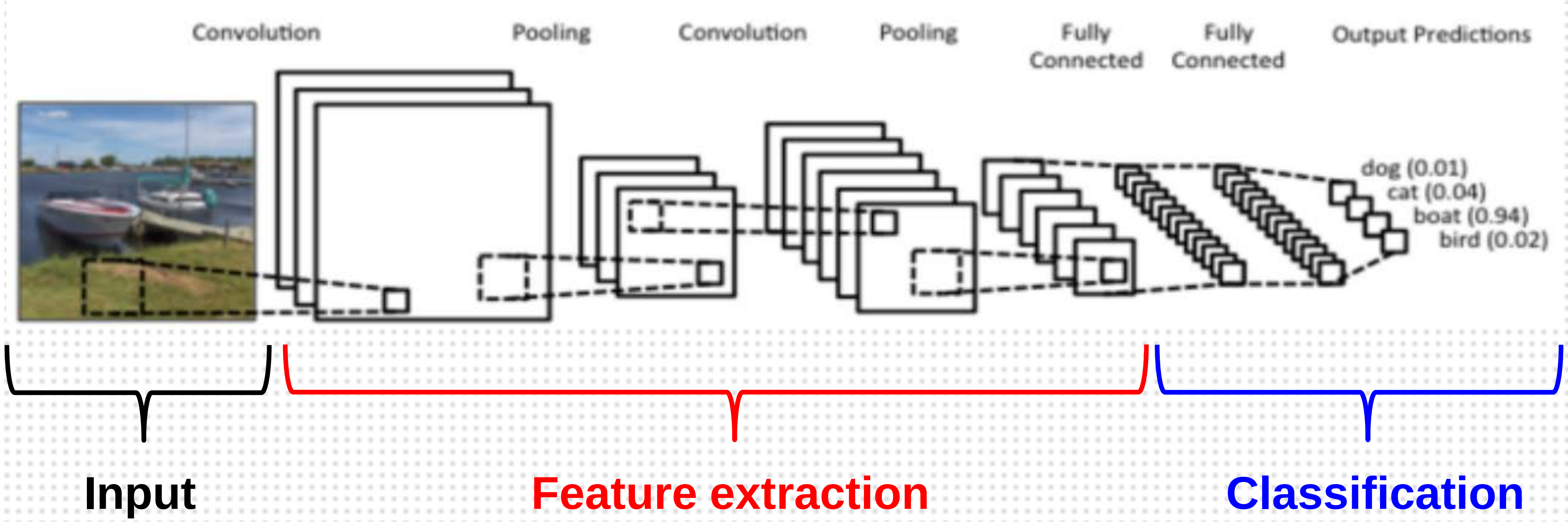
- Convert DICOM format to PNG format
- Resize the images into 512×512 pixels
- Use image augmentation and class weight to deal with insufficient and imbalanced data

For source data (ChestX-ray8)

- Change 2-dimention images into 3-dimensional RGB format
- Wrote Python class ‘MySequence’ to read images in batch

Methods

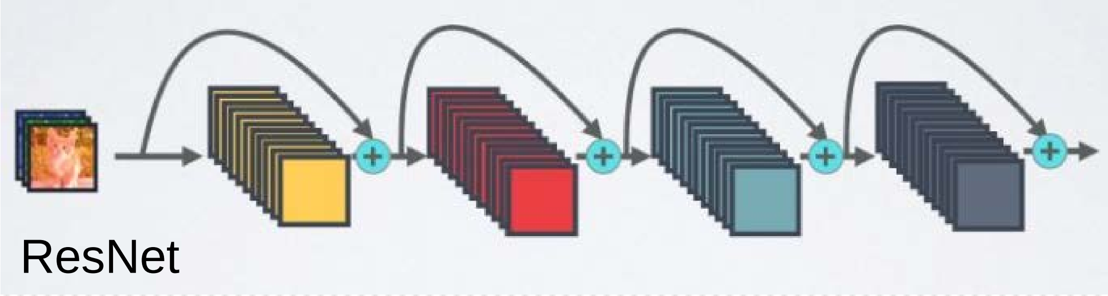
CNN architectures

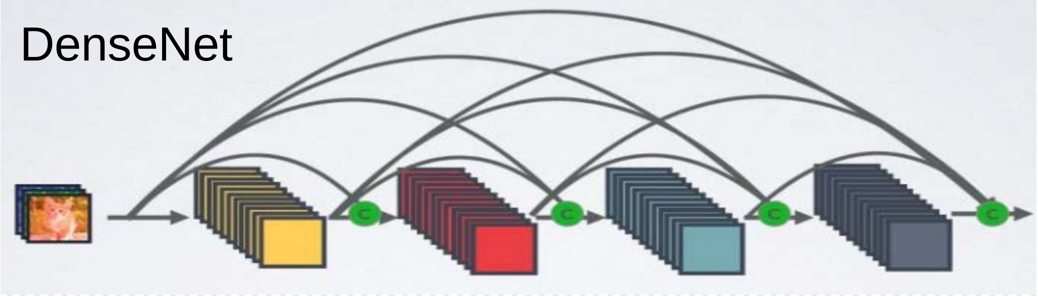


Methods

Modelling

	ResNet	DenseNet
Innovation	residual learning	dense shortcuts
	shortcuts connection	feature reuse
	no degradation	transition layer
Output in L layer	$X_L=H_L(x_{L-1})+ x_{L-1}$	$x_{L-1}= H_L([x_0, x_1, \ldots, x_{L-1},])$
Splicing method	element-wise add	concatenate
training speed	fast	slow
Number of parameters	big	small





Methods

Modelling ResNet50

Layers	Output Size	Structure	50-layers	sublayers in keras
conv1	121 x 121	7x7,64, stride 2	1	7*
conv2_x	56 x 56	3x3 max pool, stride 2 $\begin{bmatrix} 1\times1,64 \\ 3\times3,64 \\ 1\times1,256 \end{bmatrix}$ x 3	3 x 3	12**+10***+10
conv3_x	28 x 28	$\begin{bmatrix} 1\times1,128 \\ 3\times3,128 \\ 1\times1,512 \end{bmatrix}$ x 4	3 x 4	12+10+10+10
conv4_x	14 x 14	$\begin{bmatrix} 1\times1,256 \\ 3\times3,256 \\ 1\times1,1024 \end{bmatrix}$ x 6	3 x 6	12+10+10+10+10+10
conv5_x	7 x 7	$\begin{bmatrix} 1\times1,512 \\ 3\times3,512 \\ 1\times1,2048 \end{bmatrix}$ x 3	3 x 3	12+10+10
classification layer	1 x 1	average pool, 1000-d fc, softmax	1	1

Number of frozen layers

- The first 10 layers (39)
- The first 22 layers (81)
- The first 40 layers (143)

Ex.

```
for layer in res.layers:
    layer.trainable = False
for layer in res.layers[39:]:
    layer.trainable = True
```

Methods

Modelling DenseNet121

Layers	Output Size	Structure	121-layers	sublayers in keras
convolution	121x121	7x7 conv, stride 2	1	9*
pooling	56x56	3x3 max pool, stride 2		
dense block (1)	56x56	$\begin{bmatrix} 1\times1 & conv \\ 3\times3 & conv \end{bmatrix}$ x 6	2x6	7**x6
transition layer (1)	56x56	1x1 conv	1	4***
	28x28	2x2 average pool, stride 2		
dense block (2)	28x28	$\begin{bmatrix} 1\times1 & conv \\ 3\times3 & conv \end{bmatrix}$ x 12	2x12	7x12
transition layer (2)	28x28	1x1 conv	1	4
	14x14	2x2 average pool, stride 2		
dense block (3)	14x14	$\begin{bmatrix} 1\times1 & conv \\ 3\times3 & conv \end{bmatrix}$ x 24	2x24	7x24
transition layer (3)	14x14	1x1 conv	1	4
	7x7	2x2 average pool, stride 2		
dense block (4)	7x7	$\begin{bmatrix} 1\times1 & conv \\ 3\times3 & conv \end{bmatrix}$ x 16	2x16	7x16
Classification layer	1 x 1	7x7 global average pool, 1000-d fc, softmax	1	1

Number of frozen layers

The first 14 layers (55)

The first 39 layers (143)

The first 88 layers (315)

Methods

Modelling Parameter settings

Parameters	Settings
Optimizer	Adam
Learning Rate	1.00E-04
Loss	Weighted Binary Cross Entropy
Metrics	Binary Accuracy
Activation	Sigmoid
Epochs	30
Modify classification layer	global average pooling (✓)
	Dense (x)
	Batch Normalization (x)
	Drop Out (✓)
	Dense (✓)

W-BCE

$$L_{\mathrm{W-BCE}} = \sum_{i} \left\{ \beta_P \sum_{k:\, y_{ik}=1} \left[-\ln\left(\sigma\left(f_k(x_i)\right)\right) \right] + \beta_N \sum_{k:\, y_{ik}=0} \left[-\ln\left(1-\sigma\left(f_k(x_i)\right)\right) \right] \right\},$$

where $f_k(\boldsymbol{x}_i)$ is $\boldsymbol{x}_i$'s kth input for the final fully-connected layer, β_P is set to $\frac{|P|+|N|}{|P|}$ while β_N is set to $\frac{|P|+|N|}{|N|}$. $|P|$ and $|N|$ are the total number of '1's and '0's in all the dataset.

Methods

Modelling Transfer learning

When

- Training data is extremely limited in some emerging professional fields.
- Training data and testing data may follow different distributions

What

- Transfer the trained parameters to a new model in order to accelerate and optimize the process of training
- Inherit the existing neural network and adjust it for new data

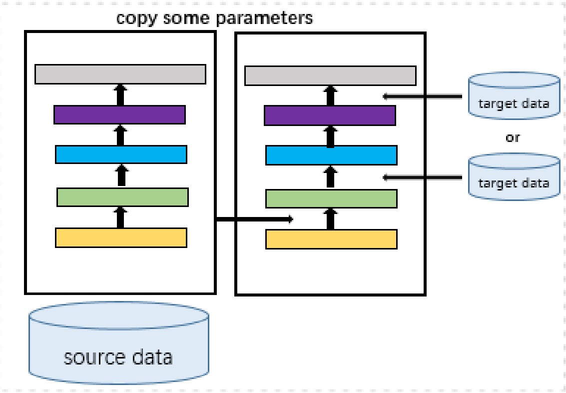
Why

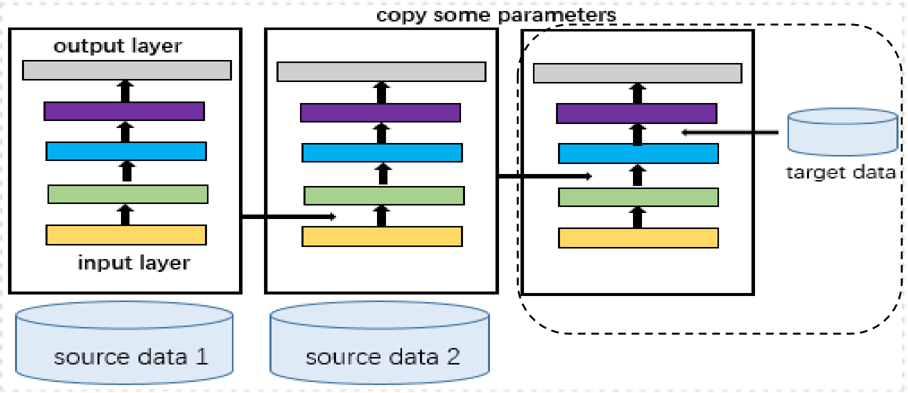
- Standing on the shoulders of giants
- Training cost can be very low
- Suitable for learning tasks in small datasets

Methods

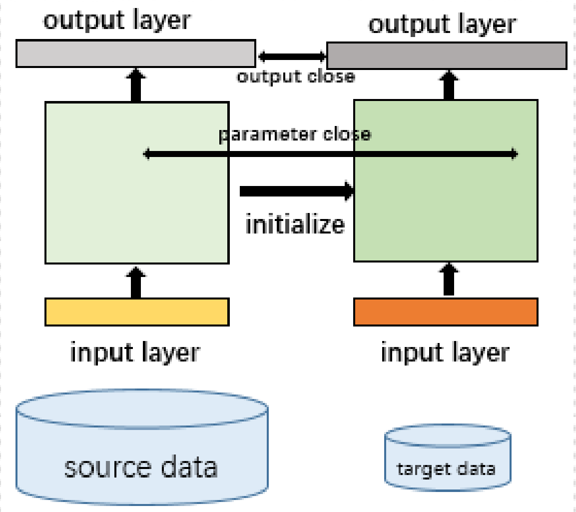
Modelling Transfer learning

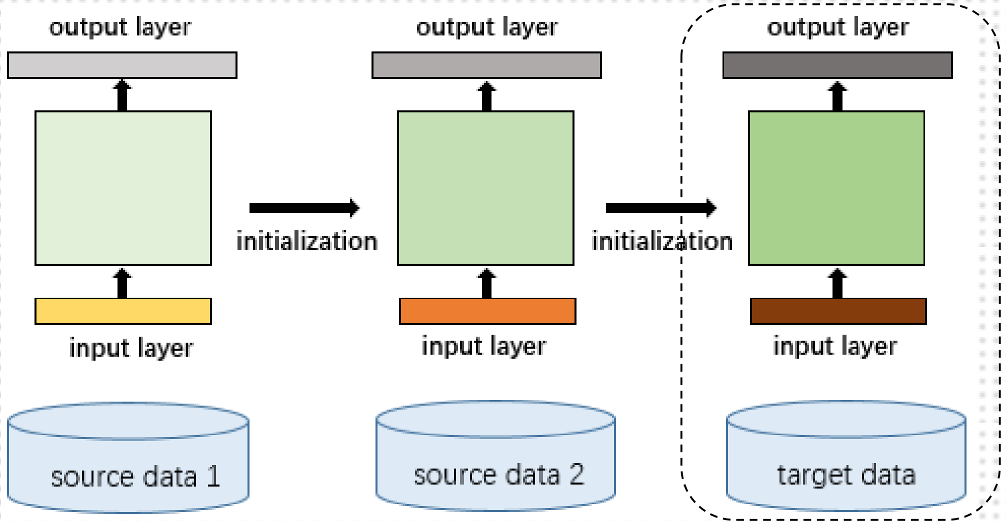
- Layer Transfer





- Conservative Training





Methods

Modelling Weight training methods

A: Modifying the final layer
B: Freezing some layers and retraining the remaining
C: Training all layers with pre-trained initial values
D: Initializing randomly and training from scratch

Target data	**Source data**	**Weight training methods**			
		A	B	C	D
ChestX-ray8	ImageNet	x	✓	✓	✓
E-Da	ImageNet	✓	✓	✓	x
	ChestX-ray8	✓	✓	✓	x
	ImageNet+ChestX-ray8	✓	✓	✓	x

Methods

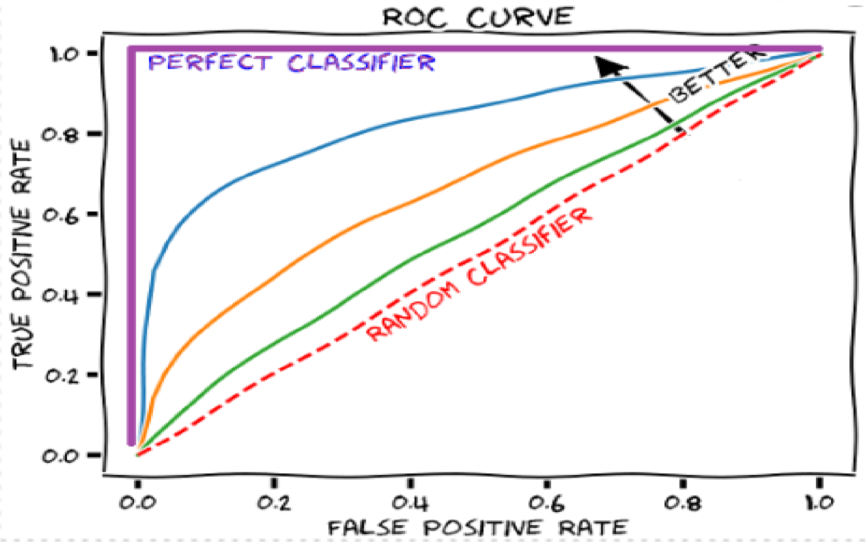
Evaluation

- Metrics

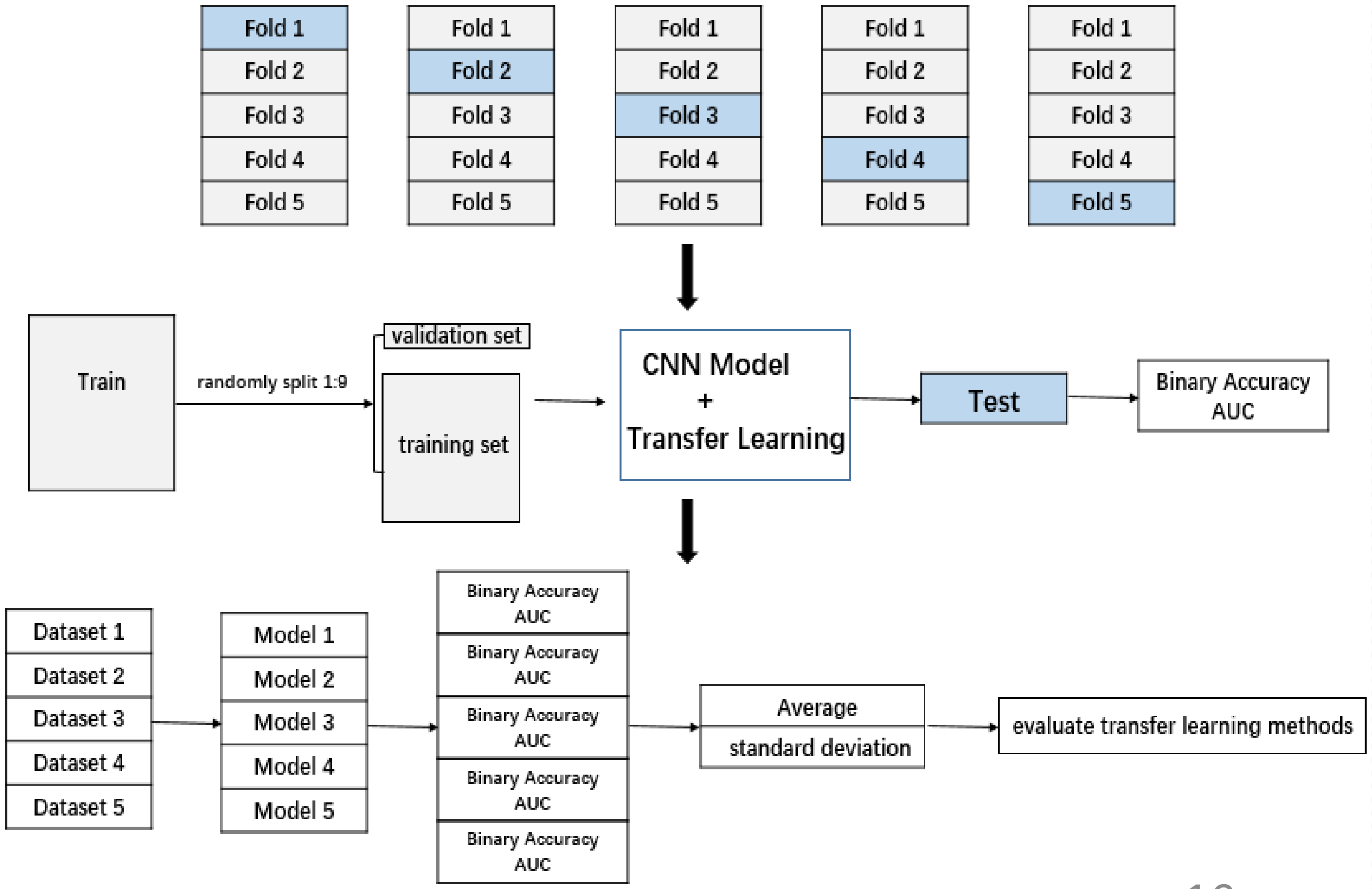
	Real Yes	Real No
Predicted Yes	True Positive (TP)	False Positive (FP)
Predicted No	False Negative (FN)	True Negative (TN)

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\text{TPR} = \frac{TP}{TP+FN} = \text{Recall} \qquad \text{FPR} = \frac{FP}{FP+TN} \qquad \text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$



- 5-fold cross-validation

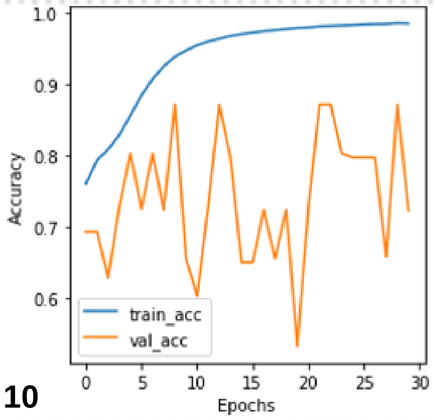


Results

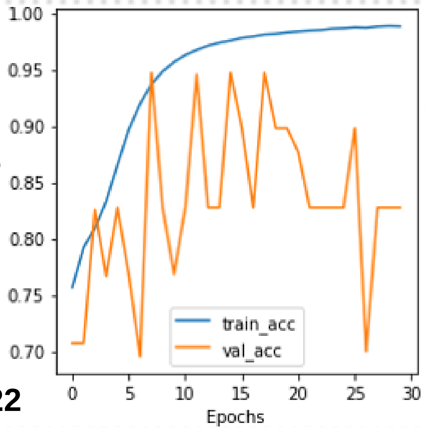
Frozen layers For ChestX-ray8

Model	Frozen Layers	Binary Accuracy in Testing Data	Loss
ResNet50	10	0.724	9.157
	22	0.827	4.281
	40	0.878	3.861
DenseNet121	14	0.765	4.094
	39	0.813	11.572
	88	0.894	7.192

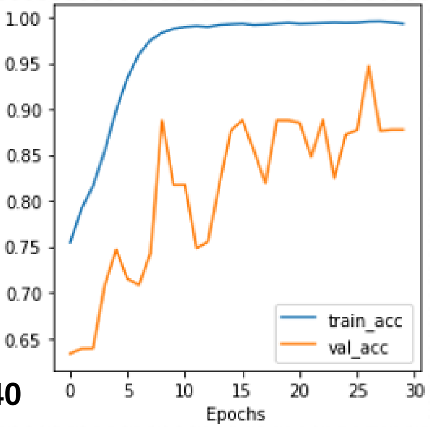
Ex. Accuracy on Training and Validation Data for RsNet50



10



22



40

- ✓ ResNet50 prefers freezing the first 40 layers;
- ✓ DenseNet121 prefers freezing the first 88 layers

Results

Frozen layers For E-Da data

➢ Binary accuracy in testing data

Frozen layers	Pre-trained Weights		
	ChestX-ray8	ImageNet(I**) + ChestX-ray8	ImageNet
B*_10	55.69% (+/- 12.45%)	46.08% (+/- 5.63%)	84.12% (+/- 10.23%)
B_22	74.79% (+/- 13.15%)	74.93% (+/- 9.20%)	82.80% (+/- 8.13%)
B_40	87.08% (+/- 10.59%)	85.12% (+/- 9.22%)	81.41% (+/- 8.37%)

➢ AUC in testing data

Frozen layers	Pre-trained Weights		
	ChestX-ray8	ImageNet (I) + ChestX-ray8	ImageNet
B_10	51.89% (+/- 2.93%)	51.17% (+/- 2.9%)	48.68% (+/-2.34%)
B_22	51.43% (+/-1.52%)	51.05% (+/-3.84%)	48.77% (+/-5.12%)
B_40	49.62% (+/-1.72%)	50.47% (+/-1.43%)	46.85% (+/-5.74%)

- ✓ For ChestX-ray8 and ImageNet(I)+ChestX-ray8, freezing more layers leads to significantly better binary accuracy but vaguely worse AUC.
- ✓ For ImageNet, freezing more layers results in worse binary accuracy and AUC

Notes:： * B refers to the transfer method that is to freeze some layers.

** I means initializing the weight in the beginning to connect ImageNet with ChestX-ray8.

Results

- ✓ Method A prefers ImageNet(F)+ChestX-ray8
- ✓ Method B is less sensitive to pre-trained weight
- ✓ Method C performs better in ImageNet and ImageNet(F)+ChestX-ray8

Methods combination ResNet50

- ➢ Binary accuracy in testing data

Pre-trained Weight	Methods		
	A	B	C
ImageNet	77.09% (+/- 12.75%)	84.12% (+/- 10.23%)	87.30% (+/- 13.96%)
Chest-Xray8	51.20% (+/- 7.02%)	87.08% (+/- 10.59%)	81.11% (+/- 13.08%)
ImageNet (F) + ChestX-ray8	81.81% (+/- 6.34%)	84.01% (+/- 9.09%)	87.14% (+/- 5.65%)
ImageNet (I) + ChestX-ray8	64.59% (+/- 19.36%)	85.12% (+/- 9.22%)	78.90% (+/- 12.02%)

- ➢ AUC in testing data

- ✓ Method C is the best choice

Pre-trained Weight	Methods		
	A	B	C
ImageNet	52.02% (+/- 3.49%)	48.77% (+/-5.12%)	91.07% (+/-12.3%)
ChestX-ray8	49.79% (+/-0.44%)	51.89% (+/- 2.93%)	80.66% (+/-13.8%)
ImageNet (F) + ChestX-ray8	50.1% (+/- 1.03%)	49.58% (+/-1.64%)	82.83% (+/-7.49%)
ImageNet (I) + ChestXray8	49.87% (+/- 0.29%)	51.17% (+/- 2.9%)	77.53% (+/- 14.65%)

Results

Methods combination DenseNet121

- Binary accuracy in testing data

Pre-trained Weight	Methods		
	A	B	C
ImageNet	90.08% (+/- 4.29%)	95.07% (+/- 0.02%)	95.10% (+/- 2.81%)
ImageNet (F) + ChestX-ray8	74.54% (+/- 11.95%)	89.68% (+/- 5.45%)	81.02% (+/- 8.59%)

✓ ImageNet was better than ImageNet(F)+ ChestX-ray8.
✓ Method B took less time and resources than Method C and produced better results than Method A

- AUC in testing data

Pre-trained Weight	Methods		
	A	B	C
ImageNet	67.81% (+/- 2.03%)	71.30% (+/- 2.83%)	95.49% (+/- 6.58%)
ImageNet (F) + ChestX-ray8	52.65% (+/- 3.61%)	57.02% (+/- 1.67%)	78.44% (+/- 10.86%)

✓ The best weight is ImageNet and the best method is C
✓ Combination of ImageNet and C achieved an excellent result

Results

Weights comparison Single weight

ImageNet vs ChestX-ray8 = Sample size vs similarity

Winner → ImageNet Sample size

- ✓ ChestX-ray8 cannot replace ImageNet as the source data but can serve as a bridge between ImageNet and E-Da data
- ✓ Sample size take priority over similarity when choosing source data

Results

Weights comparison Compound weight

Initial values vs Frozen layers

Note：The compound weight comes from ImageNet and ChestX-ray8 through initial values or frozen layers

Trade-off

- ✓ Initializing parameters gives good results but consumes a lots of computing resources
- ✓ Freezing layers is more effective based on its benefits and costs together, but the number of frozen layers is hard to determine

Results

Weights comparison

Single weight vs Compound weight

Two datasets provide more information than one dataset → Compound weight should be superior to single weight

More complex transfer process may produce more noise → Compound weight is demanding and does not necessarily perform better

✓ Specific implementation of transfer learning depends on the research objectives and priorities

Results

Model performance Accuracy

- ResNet50

	Binary Accuracy	AUC
Without Transfer Learning	77.48% (+/- 12.14%)	76.46%(+/-9.14%)
With Transfer Learning	87.14% (+/- 5.65%)	91.07% (+/-12.3%)

By transfer learning, the average AUC value has been raised by 15%, the average binary accuracy was increased by nearly 10% while the standard deviation was reduced by more than half

- DenseNet121

	Binary Accuracy	AUC
Without Transfer Learning	65.72% (+/- 18.12%)	73.60% (+/- 10.50%)
With Transfer Learning	95.10% (+/- 2.81%)	95.49% (+/- 6.58%)

By transfer learning, the average binary accuracy has risen dramatically by nearly 30% with its standard deviation falling to less than 3%, the average value of AUC has grown by more than 20% with its standard deviation going down to around 6.6%.

Results

Model performance Costs

➢ Computing Resources When Training ResNet50 on ChestX-ray8

Methods	Batch Size	Minimum GPUs	TIME/epoch	Trainable parameters
A	128	3	3703s	26,637
B_40	128	3	3745s	15,002,637
B_22			3762s	22,111,245
B_10			4184s	23,334,413
C/D	16	4	5902s	23,561,229

➢ Computing Resources When Training DenseNet121 on ChestX-ray8

Methods	Batch Size	Minimum GPUs	TIME/epoch	Trainable parameters
A	128	3	4166s	13,325
B_88	128	3	3859s	2,172,429
B_39			4184s	5,537,037
B_14			4645s	6,589,069
C/D	16	4	10884s	6,967,181

✓ Methods A and B have clear advantages allowing of bigger batch size and demanding less time and memory.
✓ Under limited hardware conditions and training time, we'd better use transfer learning Method A or B in deep learning tasks.

Results

Summary

Subject	Contents	Results
Frozen layers	10/22/40 layers in ResNet50	freeze 40 layers in ResNet50
	14/39/88 layers in DenseNet121	freeze 88 layers in DenseNet121
Weight training methods	A, B, C	C
Pre-trained weights	ImageNet, ChestX-ray8, ImageNet+ChestX-ray8	ImageNet
Combination of methods and weights		ImageNet + C gets the highest accuracy
		ImageNet+ChestX-ray8 + A gets the lowest costs
		ChestX-ray8 + B is the most cost-efficient

Conclusion

weights

ImageNet performs better than ChestX-ray8
ImageNet+ChestX-ray8 might perform best

- Volume and variety are more valuable for source data
- Compound weight may work better if frozen layers is determined wisely

Initializing parameters may help, but still needs a lot of computing resources

- The initial value is very important
- It’s expected to build the most cost-effective model by freezing some layers

models

DenseNet121 performs better than ResNet50

Transfer learning is helpful to improve models

Different combinations have different strengths

- Trade-off between accuracy and cost based on your goal and available resources
- Explore problems in their specific circumstances and turn to the most suitable methods or tools