DLCV FINAL
MEDICAL IMAGING

組別：ebyefan
組員：陳慶豐、洪緯濬、陳芃彣

Abstract

Train a neural network to **classify 5 classes** of cerebral hemorrhage by utilizing data annotated by physicians.

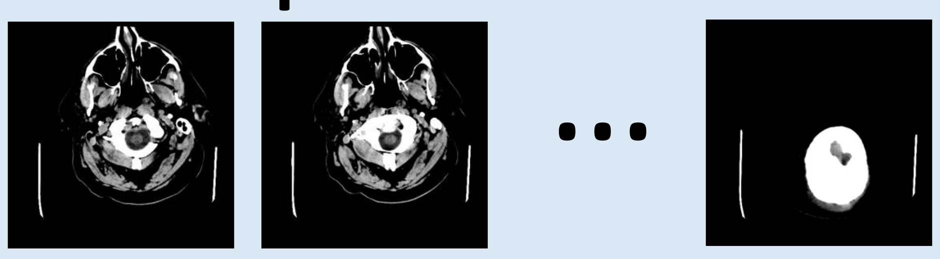
The 5 classes are:

Intracerebral hemorrhage (ICH)	腦內出血
Intraventricular hemorrhage (IVH)	腦室出血
Subarachnoid hemorrhage (SAH)	蜘蛛膜下出血
Subdural hemorrhage (SDH)	硬膜下出血
Epidural hemorrhage (EDH)	硬膜外出血

Data Description

```
└── train
    ├── ID_1038f60d15
    ├── ID_105c1fa4d9
    ├── ID_10b028ace2
    ├── ID_10f3dac8ea
    ├── ID_110b6418fa
    ├── ID_116fd5b262
```

Patients' ID

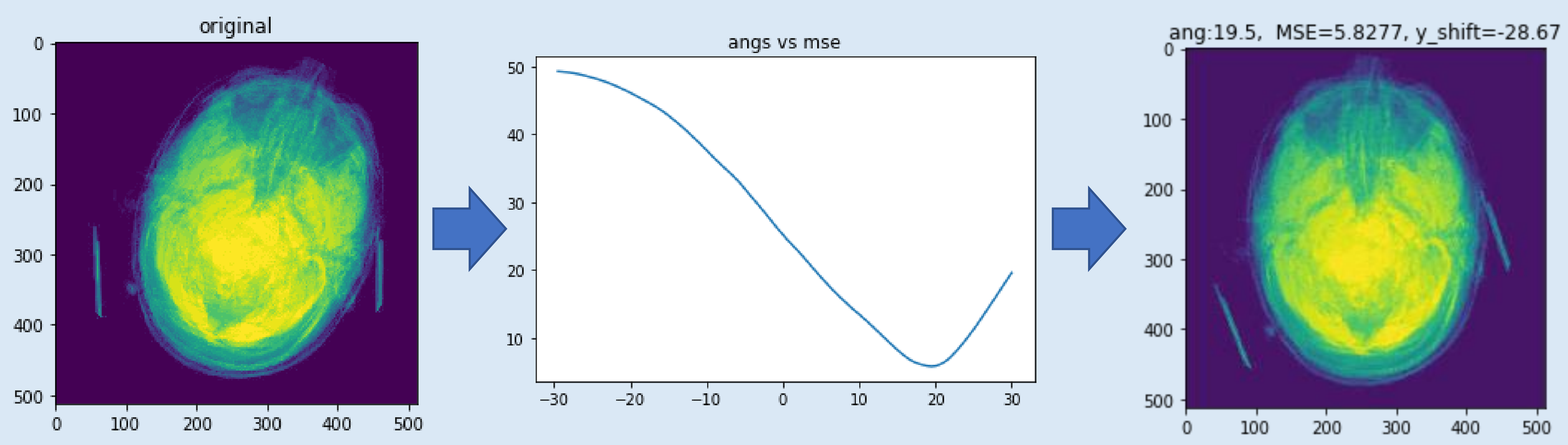


A sequence of CT brain image

Train patients: 1,535 patients, with **50,492** images in total
Test patients: 768 patients, with **25,053** images in total
Image size: 512*512, grayscale
Training Sequence length range in 22~40

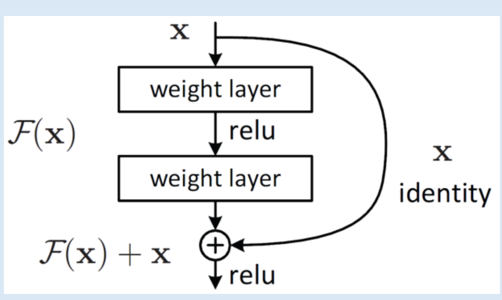
Image Preprocessing

- Data augmentation:
  - Resize all image to (512, 512)
  - RandomHorizontalFlip with probability=0.5
  - RandomAffine with translate=(0.2, 0.2)
  - RandomRotation with 15 degree



Methodology

- Backbone – **Resnet 34**
  - residual block

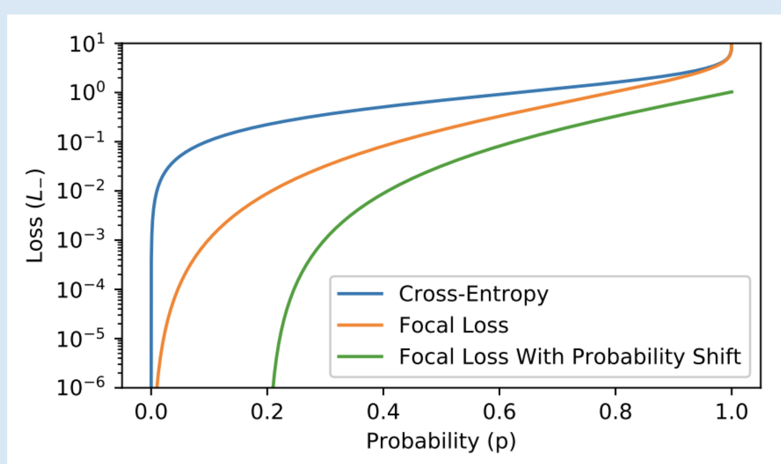


layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64; 3×3, 64] ×	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

- Loss: **Asymmetric Loss For Multi-Label Classification**
  - reducing negative sample loss

$$p_m = \max(p - m, 0)$$

$$ASL = \begin{cases} L_+ = & (1-p)^{\gamma_+}\log(p) \\ L_- = & (p_m)^{\gamma_-}\log(1-p_m) \end{cases}$$

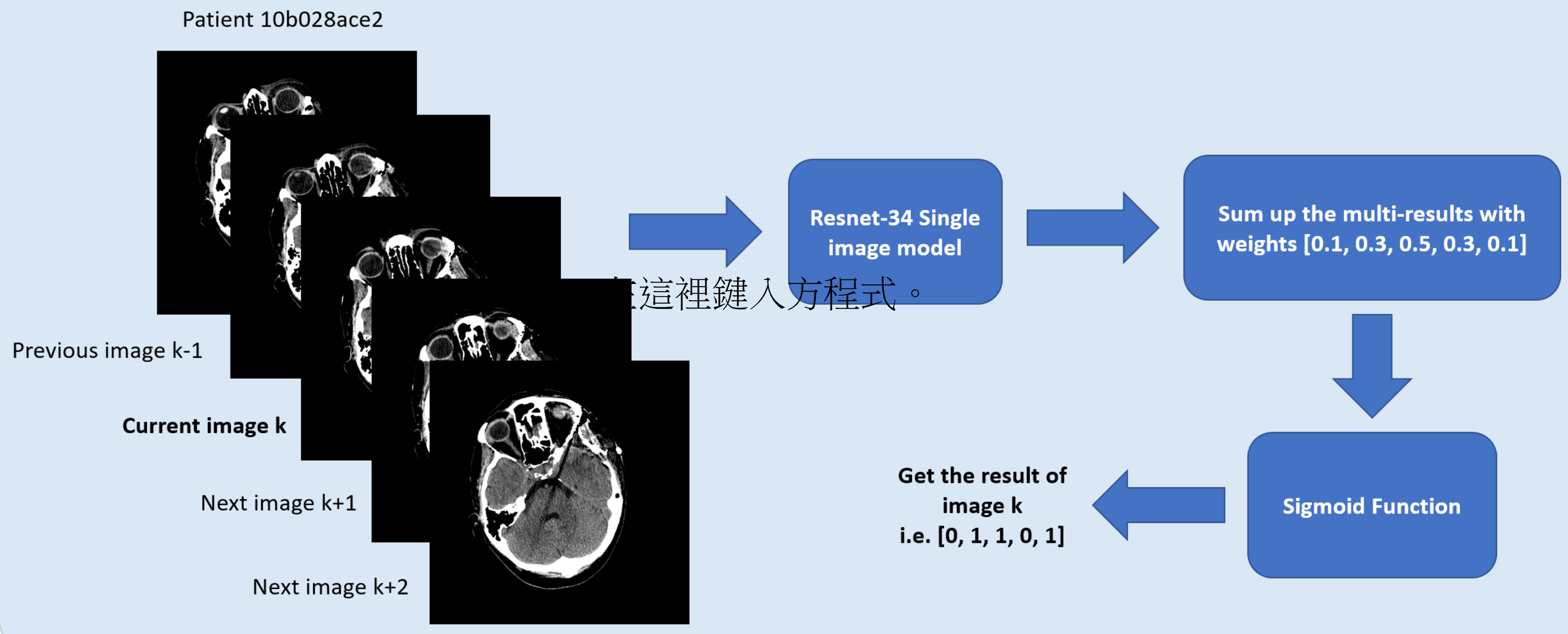


Training Details

- **Train**
  - Training on single image of every patients with

Batch size	lr	Scheduler step	gamma
20	1e-4	10	0.5

- **Refining model:**
  - Refine the model trained in “Train” process by inputting 5 images with lr = 1e-6.
- **Test**
  - Testing with multi images. i.e. 5 images
  - We predict for a single image with the observation of a sequence images.
    1. Predict a sequence of 5 images respectively.
    2. Sum up the 5-image results with weights and filter it with Sigmoid function.



Result

	Best 3 model - inference5	ensemble of 3 model
model 1	0.77632	**0.78966**
model 2	0.77074	
model 3	0.77743	

Inference time (TITAN RTX)
- emsemble models: **368.711 sec**

***Since our dataloader takes a whole sequence data of one patients once.**

***Multipy the weight in foward function.**

```
def forward(self, input, weight):
    num = input.shape[0]
    feature = self.pretrain(input)
    feature = self.fc_1(feature) # i.e. feature.shape = [34, 5]
    head = feature[0].unsqueeze(0)
    tail = feature[-1].unsqueeze(0)
    pad_feature = torch.cat([head, head, feature, tail, tail], dim=0)

    # i.e. weight = [0.1, 0.3, 0.5, 0.3, 0.1]
    out = feature * weight[2]
    out += pad_feature[0:num] * weight[0]
    out += pad_feature[1:num+1] * weight[1]
    out += pad_feature[3:num+3] * weight[3]
    out += pad_feature[4:num+4] * weight[4]
    return out
```

public score

#	Team Name	Notebook	Team Members	Score	Entries	Last
1	VcLD			0.79395	87	14h
2	ebyefan			0.78966	57	3h

private score

#	Δpub	Team Name	Notebook	Team Members	Score	Entries	Last
1	—	VcLD			0.79353	102	20m
2	—	ebyefan			0.79317	65	4h

Conclusion

1. Compare with BCE loss, Asymmetric Loss performs better when dealing with multi-class imbalanced classification.
2. Since we observe the classification result of each slice of CT image only depends on neighbor images, our model which only take 4 neighbors into consideration is easier to train and performs better compared to sequential models such as LSTM, GRU in our experiment.
3. Eventually, we combine simple but well-tuned procedures and lead to a good performance.

Reference

Ben-Baruch, Emanuel, et al. "Asymmetric Loss For Multi-Label Classification." *arXiv preprint arXiv:2009.14119* (2020).
He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.