MURA Classification with VGG16

10801128 陳俊鴻

Implementation Details

PL	Python 3.8
API	Tensorflow 2.4.0
Packages	os, numpy, matplotlib
GPU	M1
Github	None

Dataset

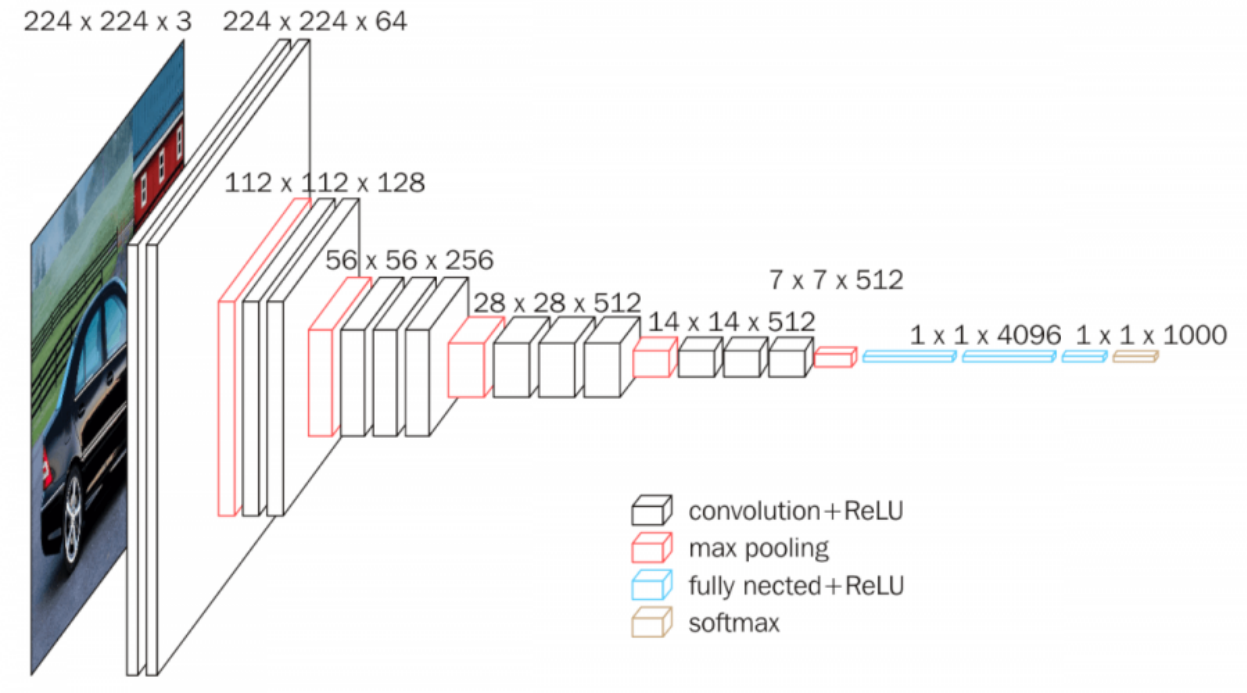
Name	Musculoskeletal radiograph(MURA)
Content	Bone X-ray images
Source	https://stanfordmlgroup.github.io/competitions/mura/
Size	224x224 pixels, channel number=3
Classes	ELBOW_negative, SHOULDER_negative, WRIST_negative
Distribution	Train: 9032, validation: 1289, test: 2580

Origin size: 512x512 pixels; origin classes: ELBOW_positive, ELBOW_negative, FINGER_positive, FINGER_negative, FOREARM_positive, FOREARM_negative, HAND_positive, HAND_negative, HUMERUS_positive, HUMERUS_negative, SHOULDER_positive, SHOULDER_negative, WRIST_positive, WRIST_negative

Experimental Design

VGG16 is a convolutional neural network proposed by K. Simonyan and A. Zisserman in the paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. Its architecture contains several convolutional layers for extracting features with kernel K from input images I $S(i,j) = (I * K)(i,j)$ and max pooling layers for refining information like down sampling. After conducting stacks of convolution and max pooling, all the feature maps is flattened to

undergo full connection layers for classifying objects to 1000 classes $\overrightarrow{o} = \sigma(W \cdot \vec{i} + \overrightarrow{b})$. However, our trail had merely 3 classes to train; hence, the final full connection layers deserved 3 outputs only.



Modified Parameters

Due to the limitation of hardware, I implemented the following architecture with attempt to refine the structure of model, improve computing speed, and maintain similar accuracy.

VGG16:

Layers	# Filter	Size	Activation
Conv2D	64	3, 3	ReLU
Conv2D			
MaxPool2D		2, 2	
Conv2D	128	3, 3	ReLU
Conv2D			
MaxPool2D		2, 2	
Conv2D	256	3, 3	ReLU
Conv2D			

Conv2D			
MaxPool2D		2, 2	
Conv2D	512	3, 3	ReLU
Conv2D			
Conv2D			
MaxPool2D		2, 2	
Conv2D	512	3, 3	ReLU
Conv2D			
Conv2D			
MaxPool2D		2, 2	
Flatten			
Dense	4096		ReLU
Dense			
Dense	3		Softmax

Trial 1:

Layers	# Filter	Size	Activation
Conv2D	64	3, 3	ReLU
Conv2D			
MaxPool2D		2, 2	
Conv2D	128	3, 3	ReLU
Conv2D			
MaxPool2D		2, 2	
Conv2D	256	3, 3	ReLU
Conv2D			
Conv2D			
MaxPool2D		2, 2	
Conv2D	512	3, 3	ReLU
Conv2D			

Conv2D			
MaxPool2D		2, 2	
Flatten			
Dense	4096		ReLU
Dense			
Dense	3		Softmax

Trial2:

Layers	# Filter	Size	Activation
Conv2D	64	3, 3	ReLU
Conv2D			
MaxPool2D		2, 2	
Conv2D	128	3, 3	ReLU
Conv2D			
MaxPool2D		2, 2	
Conv2D	256	3, 3	ReLU
Conv2D			
Conv2D			
MaxPool2D		2, 2	
Conv2D	512	3, 3	ReLU
Conv2D			
Conv2D			
MaxPool2D		2, 2	
Conv2D	512	3, 3	ReLU
Conv2D			
Conv2D			
MaxPool2D		2, 2	
Flatten			
Dense	4096		ReLU

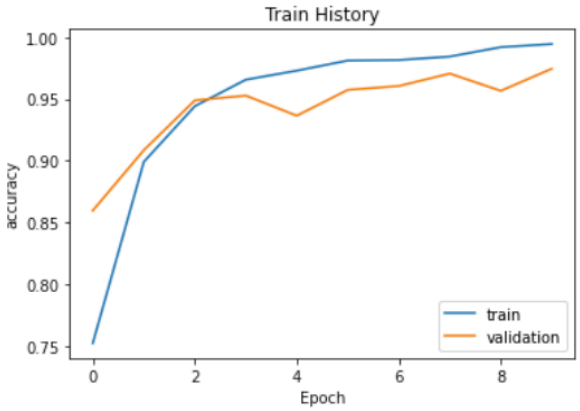
Dense	3		Softmax

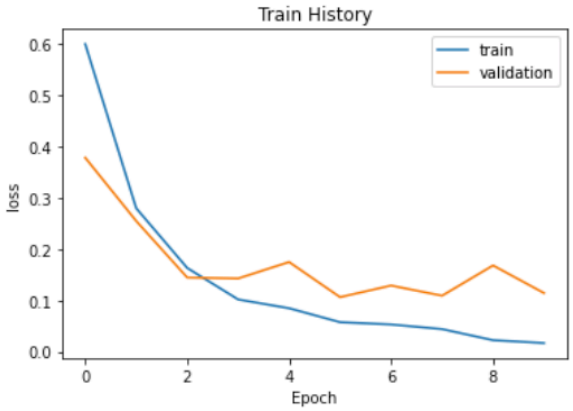
All the trials were trained with Adam optimizer and learning rate=1e-3 for 10 epochs.

Results

VGG16:

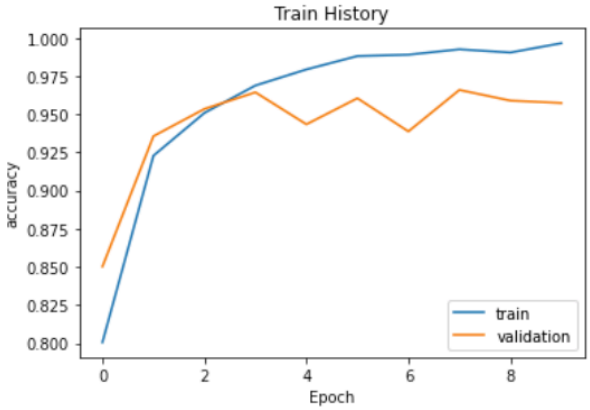
	Accuracy	Confusion metric	1	2	3
Training	99.89%	1	557	9	19
Validation	97.44%	2	16	882	4
Test	97.05%	3	22	6	1125

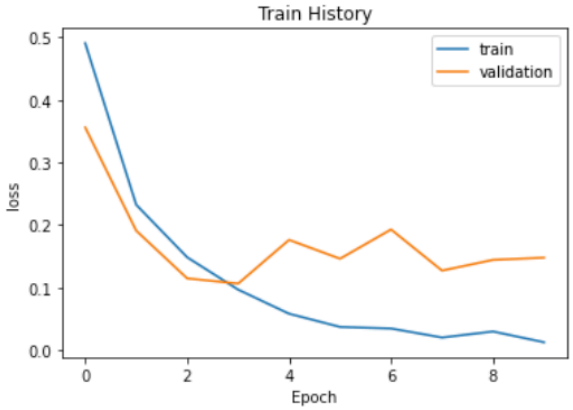




Trial 1:

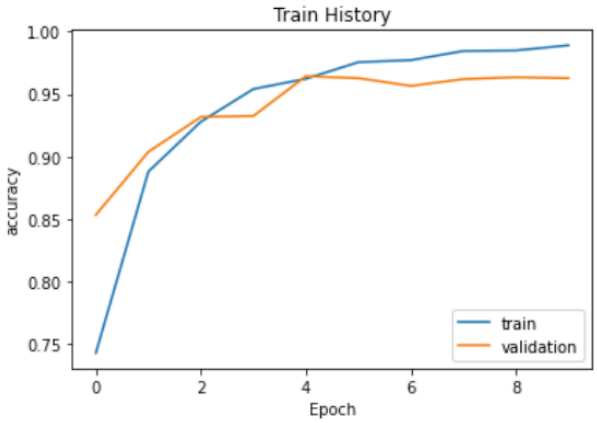
	Accuracy	Confusion metric	1	2	3
Training	99.83%	1	533	15	37
Validation	95.73%	2	15	818	9
Test	96.12%	3	18	6	1129

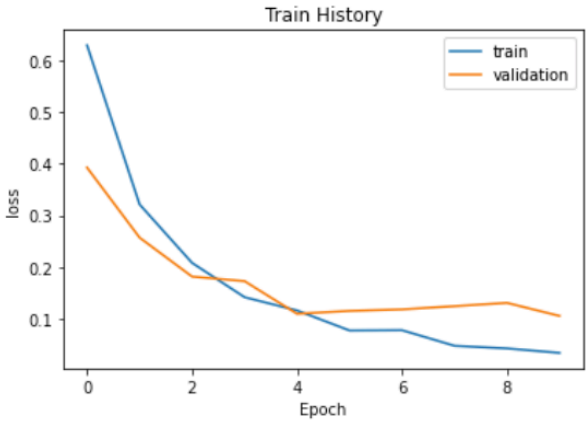




Trial 2:

	Accuracy	Confusion metric	1	2	3
Training	99.34%	1	552	21	12
Validation	96.28%	2	21	818	3
Test	96.16%	3	26	16	1111





Reference

[1] Accelerating Very Deep Convolutional Networks for Classification and Detection

[2] MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs

Bonus

1. Two trials in my report showed that VGG16 exactly is too deep for this classification task; hence, removing redundant layers doesn't matter to accuracy.
2. In my previous but failed trial, classification task for 14 classes in original dataset required too many memory and GPU resources.