Data255 Deep Learning

Breast Tumor Classification on PCam

Mavis Wang, Xiaocen Xie, Coco Yu,
Matthew Guzman, Siying Wu

Team 1

Intro

- Motivation
- Data Collection
- General Approach
- Preprocessing
- Training 3 models
- Compare the results
- Discussion

Motivation

- Breast cancer is the second leading cause of cancer-related death among women with a one-in-39 probability.

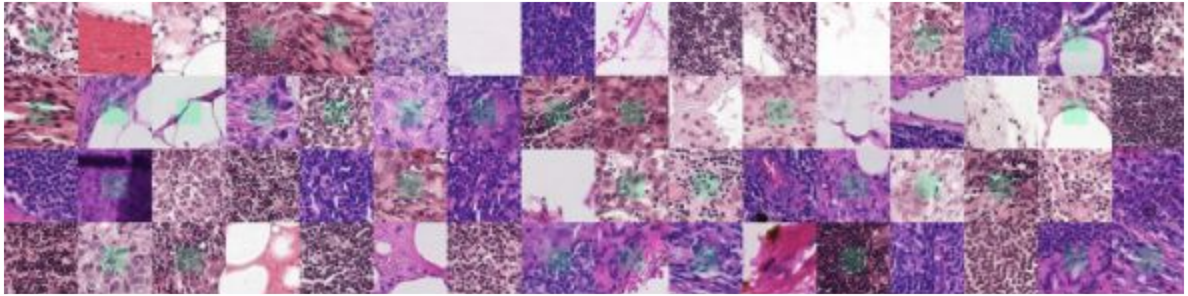
- Relying solely on doctors to determine disease characteristics takes time and has a certain error rate.

- Proposed Solution: Compare 3 models MobileNetV2, DenseNet and ResNet to find the best model has detection accuracy

Data Collection

PatchCamelyon (PCam)

- Images extracted from histopathologic scans of lymph node sections
- 327,680 color images with sizes 96 pixels *96 pixels
- Each image has a binary label that indicates the existence of metastatic tissue.



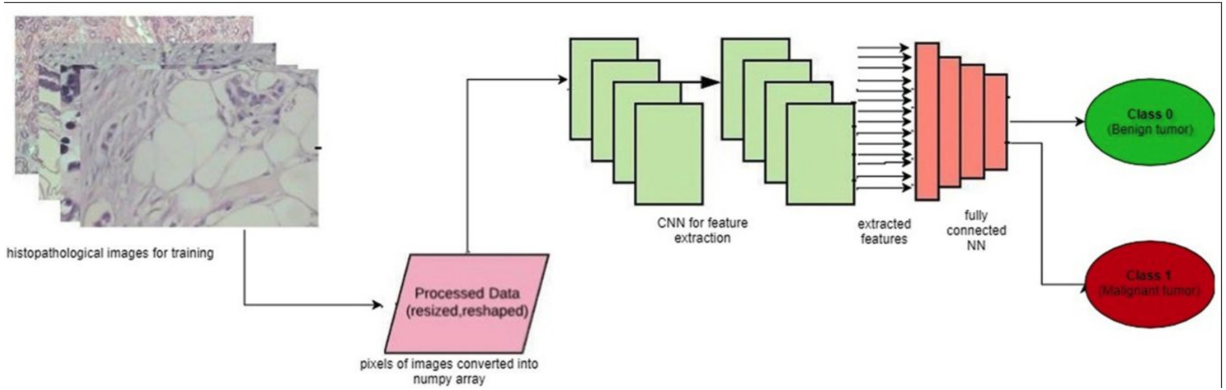
Green boxes indicate tumor tissue in center region, which dictates a positive label.

General Approach

- Use the partition of the PCam validation set as our main training dataset for the experiment due to the computational limit

- Train image classifiers by using convolutional neural network with transfer learning

- Evaluate the performance of each classifier



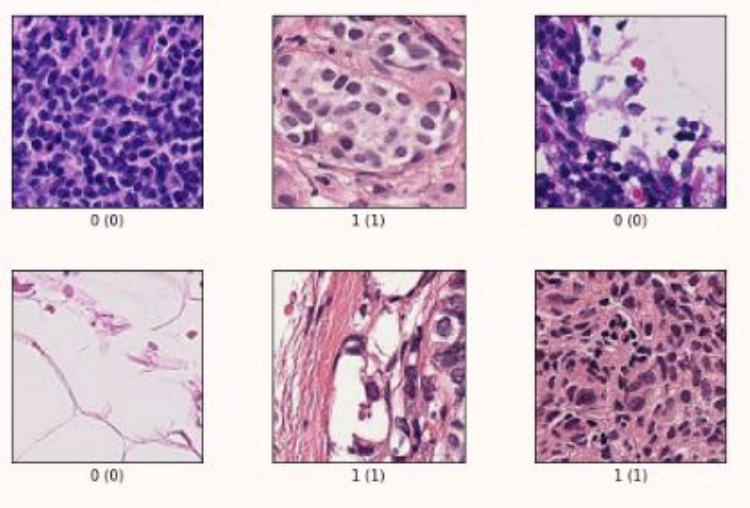
Preprocessing

Load using Keras' TDFS module to import pre-made Patch Camelyon dataset

One-hot encode images from int8 to float 32

No **resizing** necessary as imported models accepted 96x96x3 shape.

Different models had **different train/validation/test sizes**



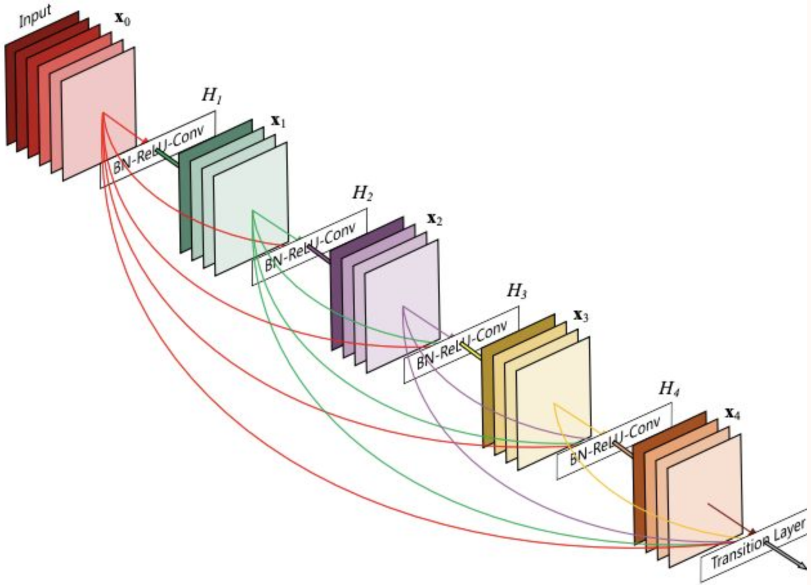
Modeling

01 DenseNet

02 MobileNetV2

03 ResNet-50

DenseNet 121 & DenseNet 201



DenseNet Architectures

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201
Convolution	112×112	7×7 conv, stride 2		
Pooling	56×56	3×3 max pool, stride 2		
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv		
	28×28	2×2 average pool, stride 2		
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 1$[illegible]	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv		
	14×14	2×2 average pool, stride 2		
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 3$[illegible]	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Transition Layer (3)	14×14	1×1 conv		
	7×7	2×2 average pool, stride 2		
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 3$[illegible]	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Classification Layer	1×1	7×7 global average pool		
		1000D fully-connected, softmax		

DenseNet Result

Optimizer: SGD

LR: 0.01

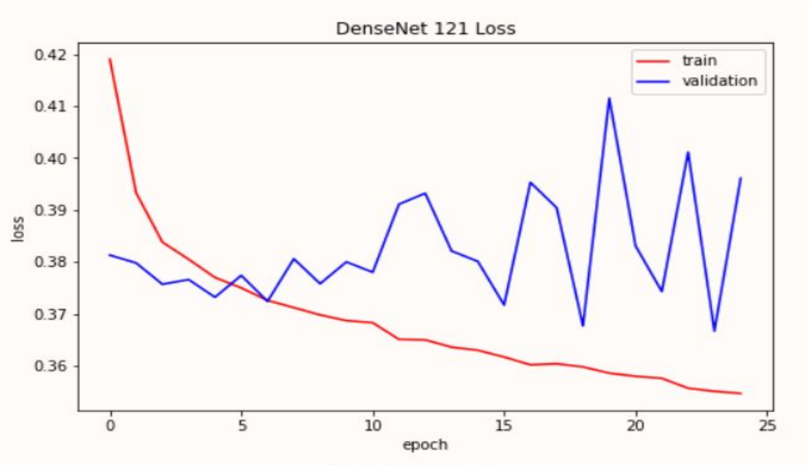
Loss: Binary Cross Entropy

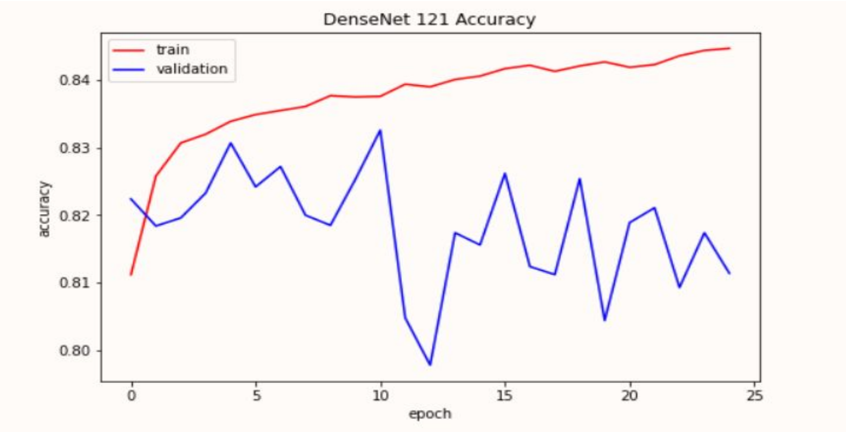
Metric: Accuracy

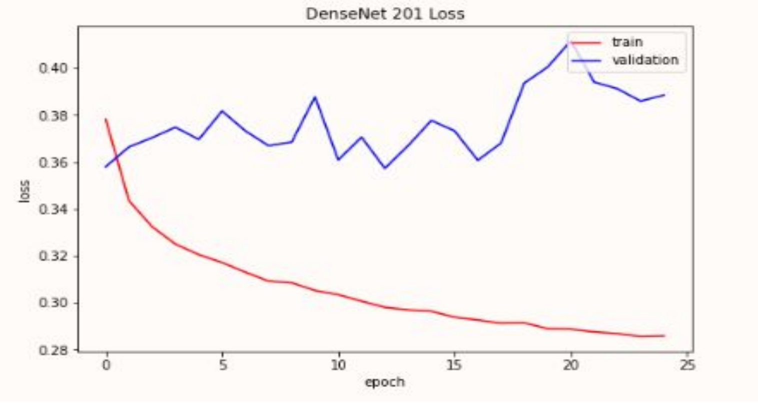
Activation: Sigmoid

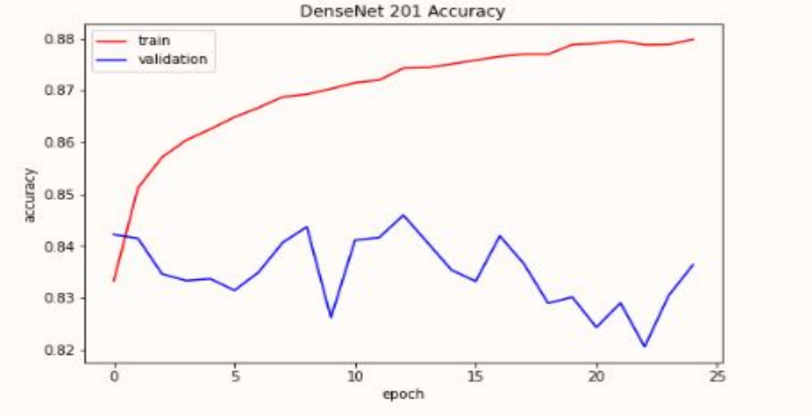
	DenseNet 121	DenseNet 201	S-Densenet w/ Equivariance (Worrall & Welling 2019)
Total Parameters	7,627,522	19,428,098	N/A
Run Time per Epoch	**122.2s**	160.5s	N/A
Num Epochs	25	25	100
Test Accuracy	79.78%	**81.67%**	88.1%

DenseNet Result







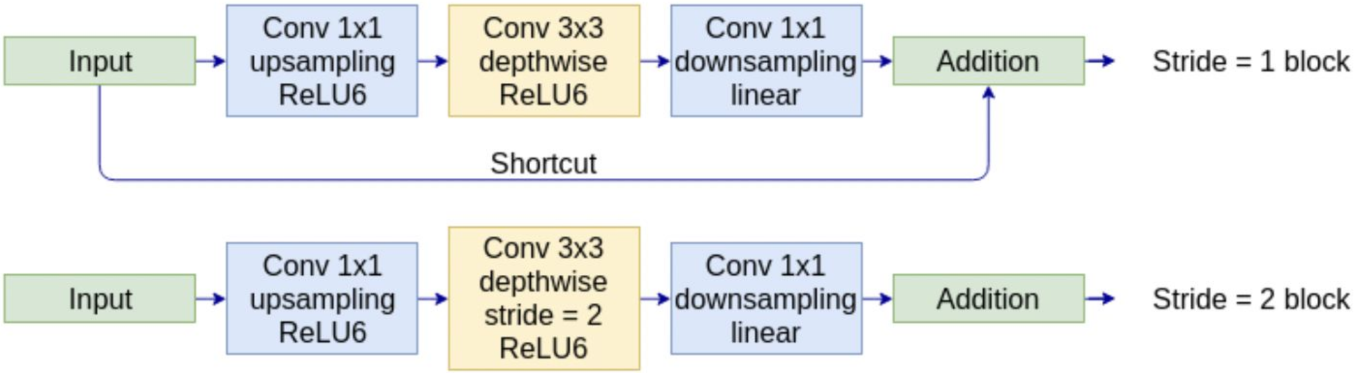


MobileNetV2

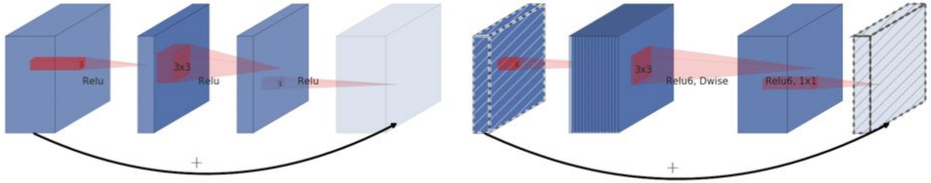
Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	0.790	0.945	22,910,480	126	109.42	8.06
VGG16	528	0.713	0.901	138,357,544	23	69.50	4.16
VGG19	549	0.713	0.900	143,667,240	26	84.75	4.38
ResNet50	98	0.749	0.921	25,636,712	-	58.20	4.55
ResNet101	171	0.764	0.928	44,707,176	-	89.59	5.19
ResNet152	232	0.766	0.931	60,419,944	-	127.43	6.54
ResNet50V2	98	0.760	0.930	25,613,800	-	45.63	4.42
ResNet101V2	171	0.772	0.938	44,675,560	-	72.73	5.43
ResNet152V2	232	0.780	0.942	60,380,648	-	107.50	6.64
InceptionV3	92	0.779	0.937	23,851,784	159	42.25	6.86
InceptionResNetV2	215	0.803	0.953	55,873,736	572	130.19	10.02
MobileNet	16	0.704	0.895	4,253,864	88	22.60	3.44
MobileNetV2	14	0.713	0.901	3,538,984	88	25.90	3.83

https://keras.io/api/applications/

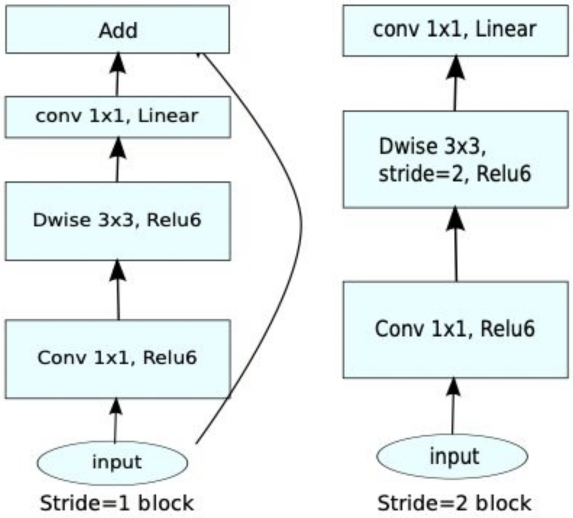
MobileNetV2 **Tricks**



(a) Residual block (b) Inverted residual block



MobileNetV2 **Architecture**



Bottleneck Residual Block

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

MobileNetV2 **Training Setup**

Data Generator = 32,768

Split: 80% train, 20% test, reserve 6,500 from training set as validation set

Distribution: 9,774 negative, 9940 positive (tumor)

Batch size = 64

Data augmentation: {sheer, zoom, flip, rotation, contrast}

Framework: Tensorflow Keras

Environment: Google Colab w/ GPU

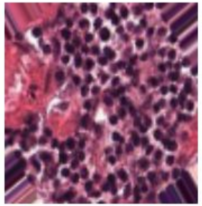
Training Epochs = 20

Opt = Adam (lr=0.0001)

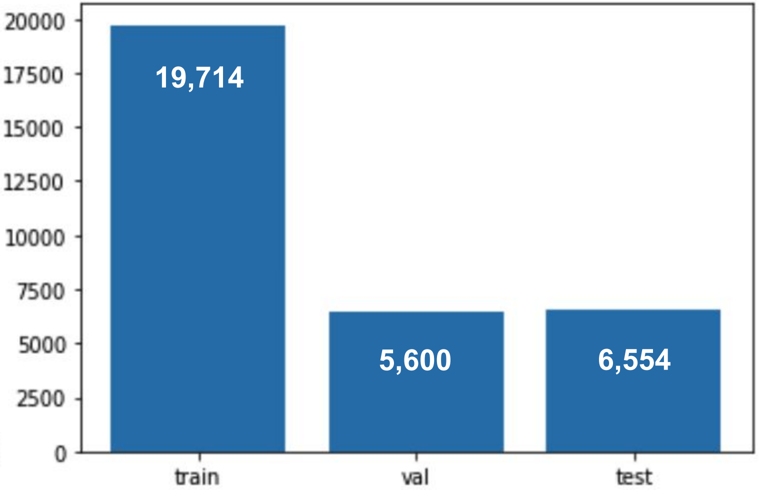
Act = elu

Loss = binary_crossentropy

Evaluation = accuracy



(64, 96, 96, 3)

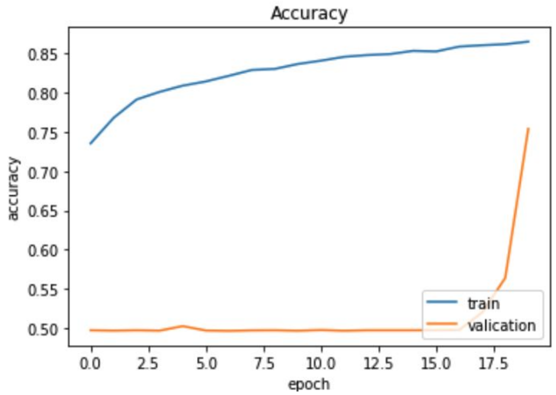


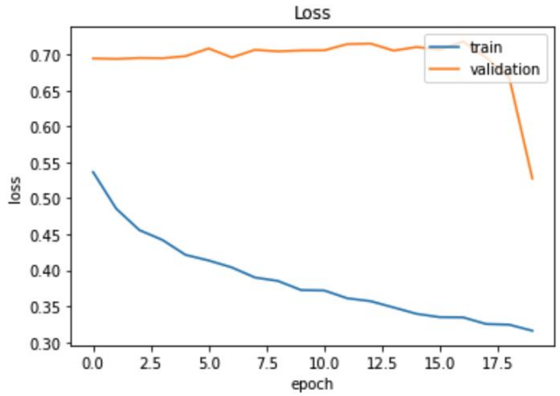
MobileNetV2 **Train From Scratch**

Exp: use MobileNetV2 architecture only

NN: added additional 2 dense layers

```
Model: "mobileNet2NN"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_37 (InputLayer)       [(None, 96, 96, 3)]       0

 mobilenetv2_1.00_96 (Functi  (None, 3, 3, 1280)       2257984
 onal)

 global_average_pooling2d (G  (None, 1280)             0
 lobalAveragePooling2D)

 flatten_5 (Flatten)         (None, 1280)              0

 dense (Dense)               (None, 1024)              1311744

 dense_1 (Dense)             (None, 512)               524800

 predictions (Dense)         (None, 2)                 1026

=================================================================
Total params: 4,095,554
Trainable params: 4,061,442
Non-trainable params: 34,112
```





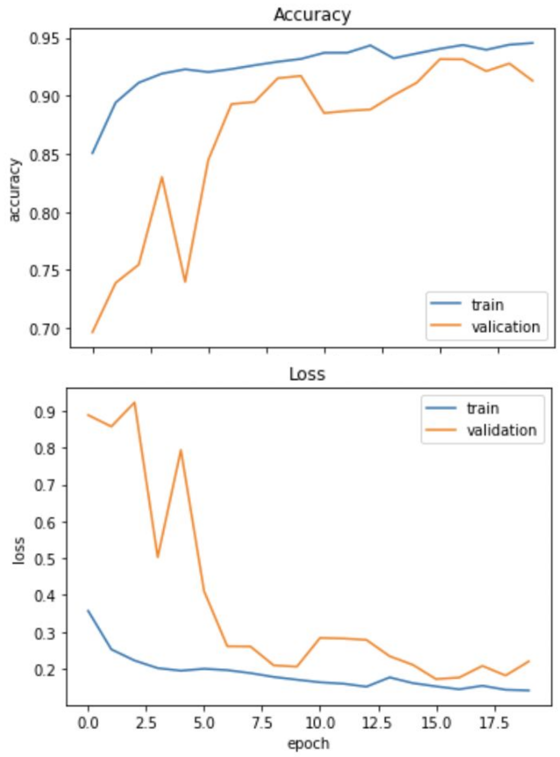
MobileNetV2 **Extract Features**

Exp 1: train with the pre-trained ‘imagenet’ weights.

Exp 2: train with ‘relu’, ‘elu’, ‘selu’

NN: + applied Batch Norm after each dense layers.

```
 dense_10 (Dense)            (None, 1024)              1311744

 batch_normalization_13 (Bat  (None, 1024)             4096
 chNormalization)

 dense_11 (Dense)            (None, 512)               524800

 batch_normalization_14 (Bat  (None, 512)              2048
 chNormalization)

 predictions (Dense)         (None, 2)                 1026

=================================================================
Total params: 4,106,818
Trainable params: 4,067,074
Non-trainable params: 39,744
```

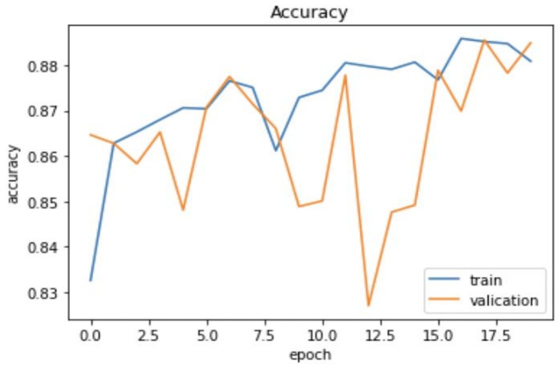


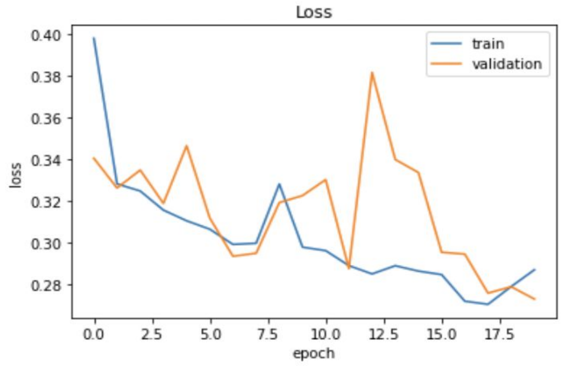
MobileNetV2 **CNN-LSTM**

Exp: add lstm after Conv blocks

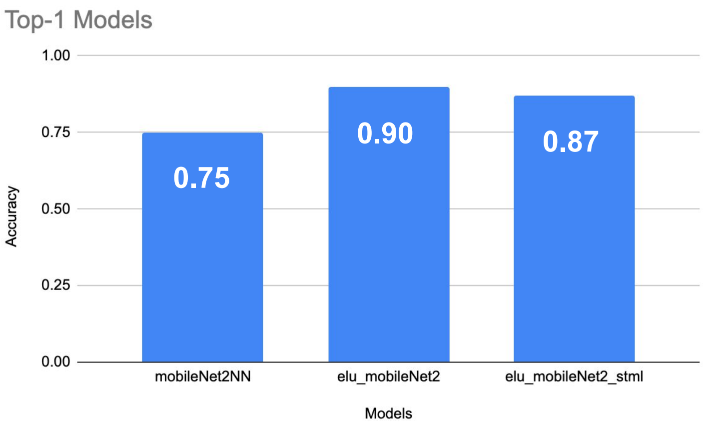
NN: pre-trained mobileNetV2 + 2 Conv layers + lstm

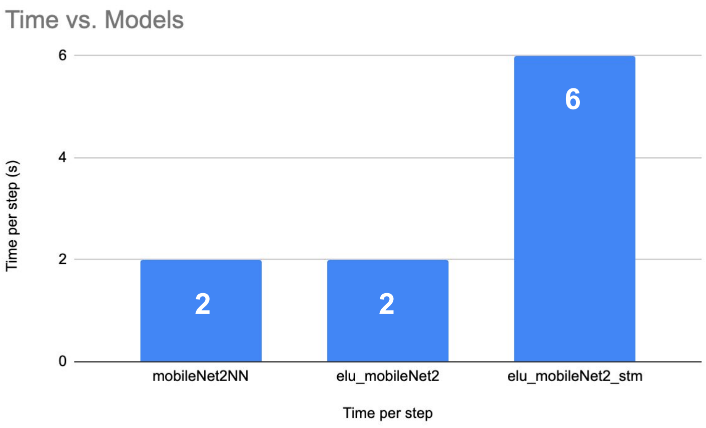
```
 conv2d_7 (Conv2D)           (None, 94, 94, 64)        1216

 conv2d_8 (Conv2D)           (None, 92, 92, 64)        36928

 max_pooling2d_3 (MaxPooling  (None, 46, 46, 64)       0
 2D)

 reshape_7 (Reshape)         (None, 2116, 64)          0

 lstm_3 (LSTM)               (None, 32)                12416

 flatten_3 (Flatten)         (None, 32)                0

 pred (Dense)                (None, 2)                 66

=================================================================
Total params: 4,146,180
Trainable params: 4,112,068
Non-trainable params: 34,112
```

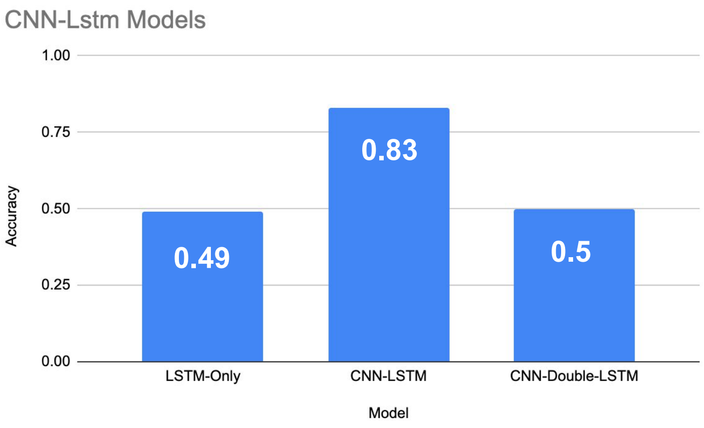


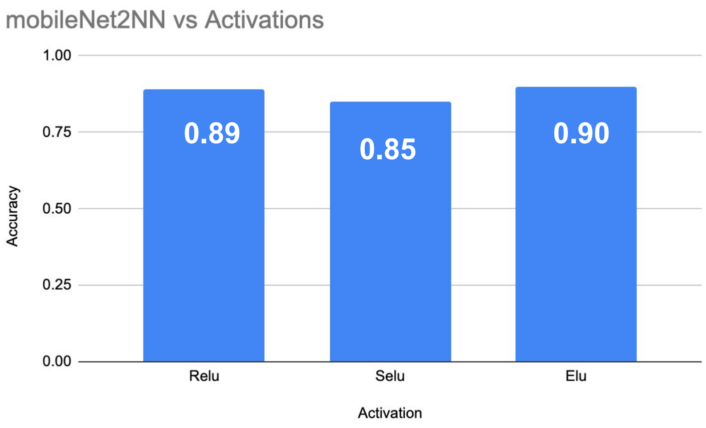


MobileNetV2 **Result**



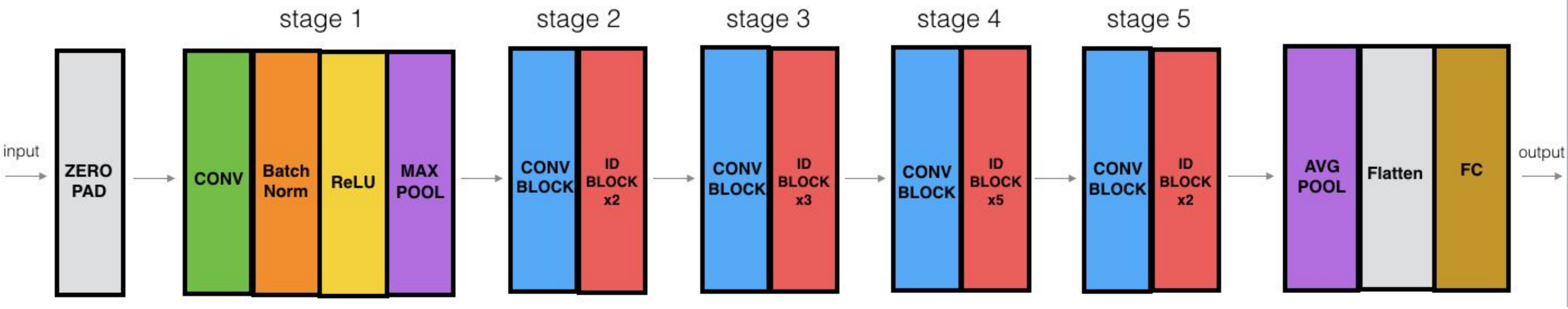




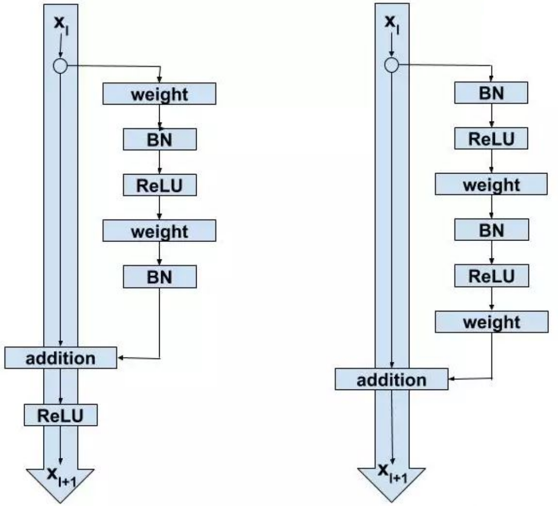


ResNet-50

- ResNet: Residual Networks
- First introduced skip connection
- ResNet-50: Pre-trained model in Keras



ResNet-50 V1 & V2



(a) ResNet - V1 (b) ResNet - V2

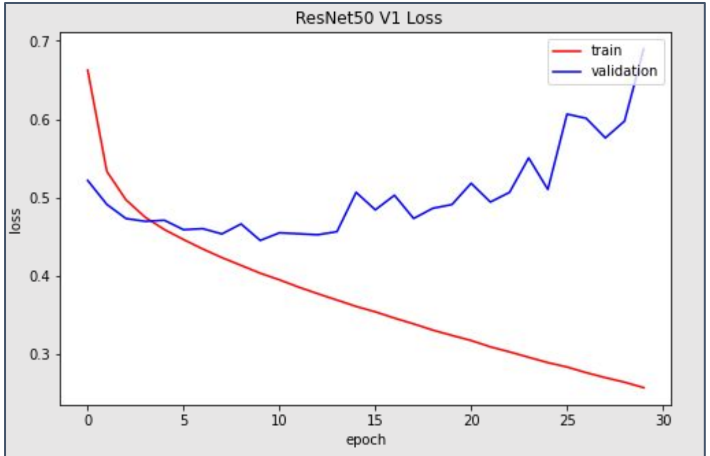
	Version 1	Version 2
Total Parameters	23,718,978	23,696,066
Run Time per Epoch	~270s	~275s
Test Accuracy	76.70%	81.65%

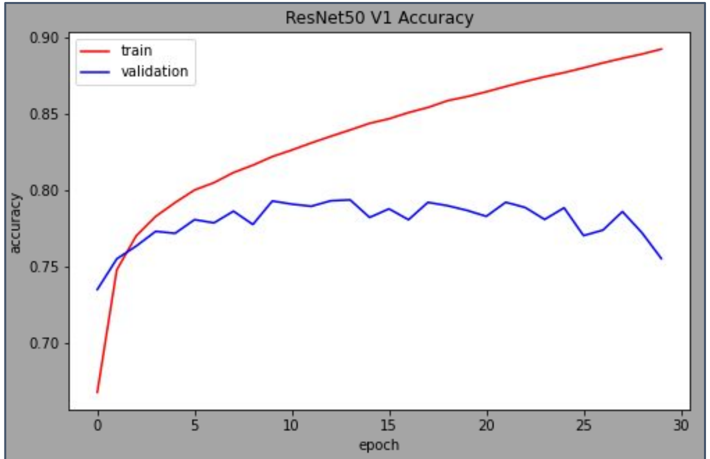
```
Layer (type)                 Output Shape              Param #
=================================================================
resnet50 (Model)             (None, 2048)              23587712
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
dropout (Dropout)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 64)                131136
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 130
=================================================================
Total params: 23,718,978
Trainable params: 22,215,874
Non-trainable params: 1,503,104
```

```
Layer (type)                 Output Shape              Param #
=================================================================
resnet50v2 (Model)           (None, 2048)              23564800
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
dropout (Dropout)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 64)                131136
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 130
=================================================================
Total params: 23,696,066
Trainable params: 22,204,610
Non-trainable params: 1,491,456
```

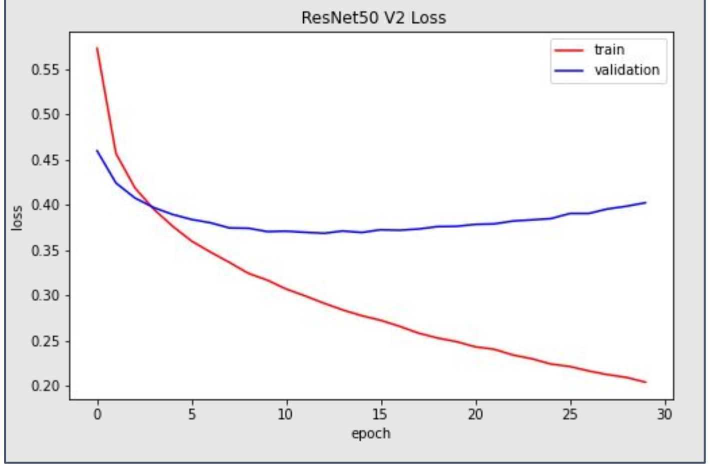
ResNet-50 Result

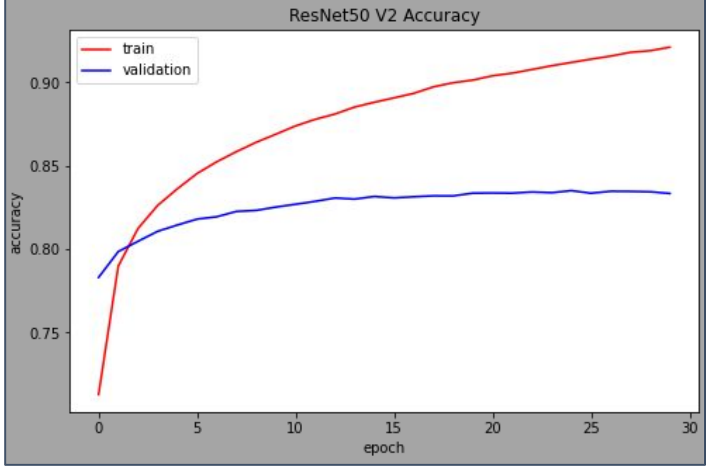
- ResNet-50 Version 1





- ResNet-50 Version 2





Discussion

Data: partition, start with small patches

Training: try to tune one hyperparameter at one time

(Activation, number of hidden layers, etc)

Future work: Fine-tune

Increase data for training

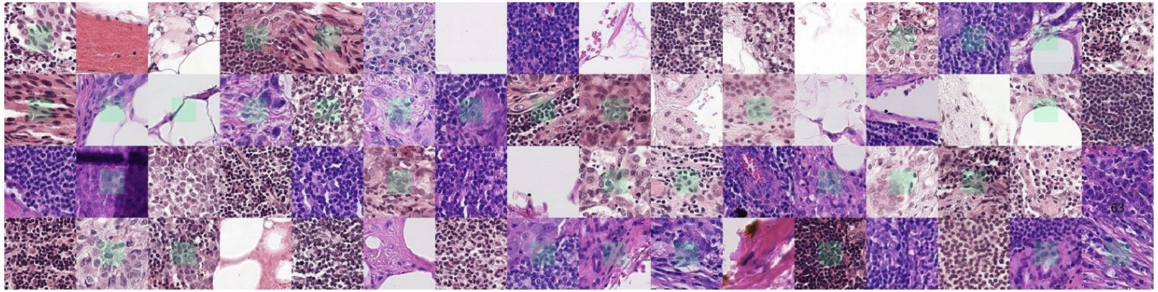
Reference

1. A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *arXiv [cs.CV]*, 2017.
2. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” *arXiv [cs.CV]*, 2018.
3. S. S. Yadav and S. M. Jadhav, “Deep convolutional neural network based medical image classification for disease diagnosis,” *J. Big Data*, vol. 6, no. 1, 2019.
4. B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling, “Rotation Equivariant CNNs for Digital Pathology,” *arXiv [cs.CV]*, 2018.
5. M. Z. Islam, M. M. Islam, and A. Asraf, “A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images,” *Inform. Med. Unlocked*, vol. 20, no. 100412, p. 100412, 2020.
6. https://cv-tricks.com/keras/understand-implement-resnets/
7. https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33
8. https://www.kaggle.com/drscarlat/melanoma-resnet50-fine-tune
9. S. Dabeer, M. M. Khan, and S. Islam, “Cancer diagnosis in histopathological image: CNN based approach,” Inform. Med. Unlocked, vol. 16, no. 100231, p. 100231, 2019.
10. https://github.com/basveeling/pcam/blob/master/README.md

Contribution

Team 1: Mavis Wang, Xiaocen Xie, Coco Yu, Matthew Guzman, Siying Wu

Breast Tumor Classification on PCam



Dataset Source

Tasks	Intro	Data Collection	Data Preprocing	DenseNet	MobileNetV2	ResNet-50	Discussion
Presentation	Xiaocen Xie	Xiaocen Xie	Matthew Guzman	Matthew Guzman	Mavis Wang	Siying Wu	Coco
Project	All	Mavis Wang	Matthew Guzman	Matthew Guzman	Mavis Wang	Siying Wu	All
IEEE Paper	Coco Yu, Siying Wu	Mavis Wang	Matthew Guzman	Matthew Guzman	Mavis Wang, Xiaucen Xie	Siying Wu	Coco

https://github.com/SJSUMS/PCam