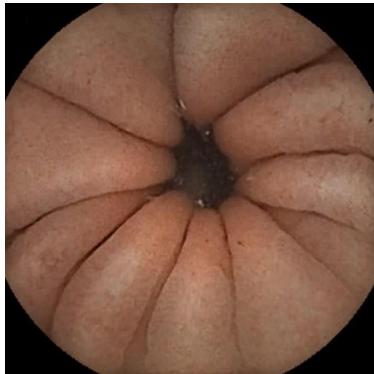


# AI Vision for Endoscopy

Automated Image Analysis

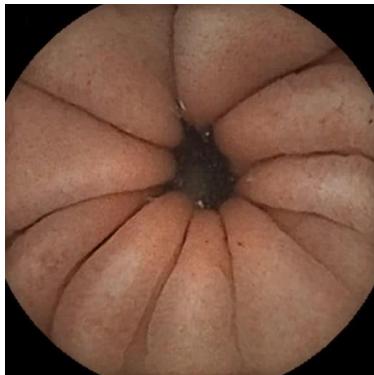


Andrew Forrester

INSIGHT  
AI Fellow

# AI Vision for Endoscopy

Automated Image Analysis

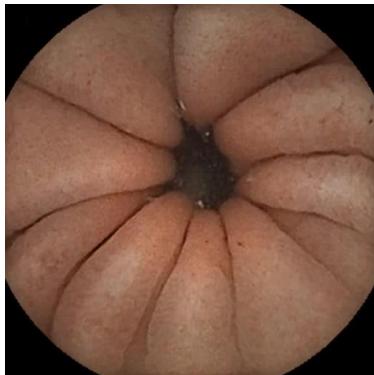


Andrew Forrester

INSIGHT  
AI Fellow

# AI Vision for Endoscopy

Automated Image Analysis

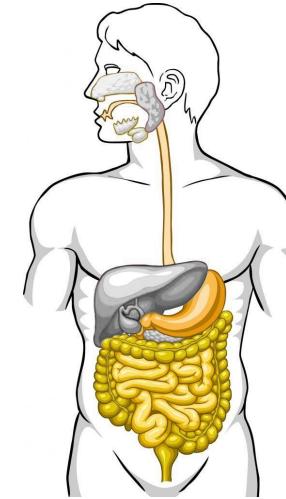


Andrew Forrester

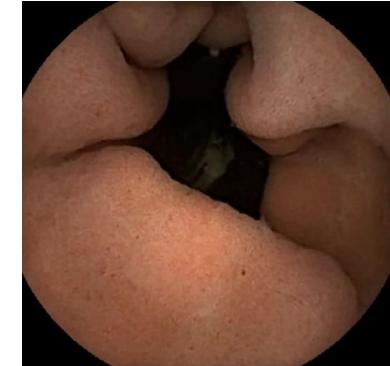
INSIGHT  
AI Fellow

# Context

## Capsule Endoscopy



$$(\sim 2 \text{ FPS}) \cdot (\sim 12 \text{ hours}) = (\sim 100k \text{ images})$$



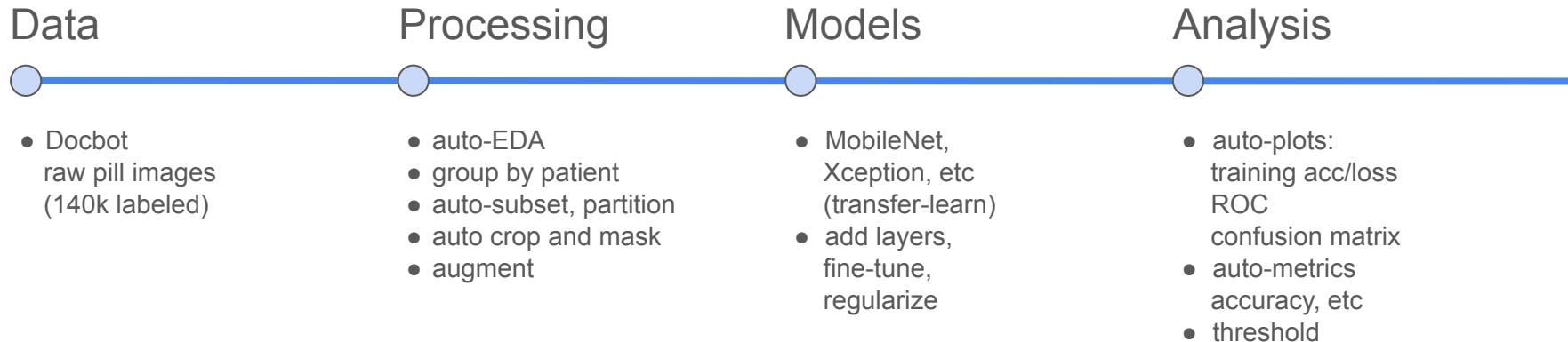
# Solution

Python package:

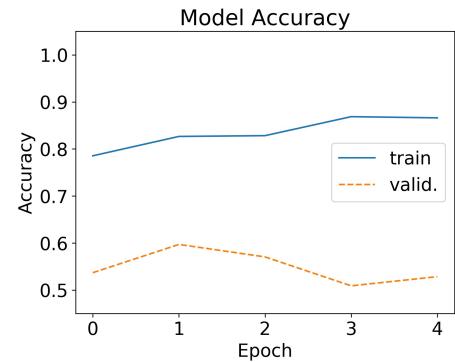
- functional classifiers (CNNs)
- framework for rapid iteration
- automated evaluation



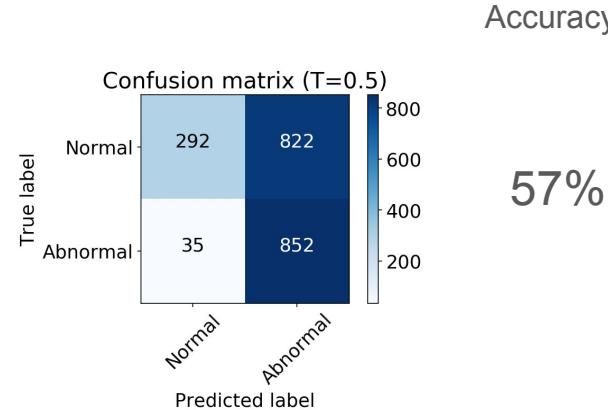
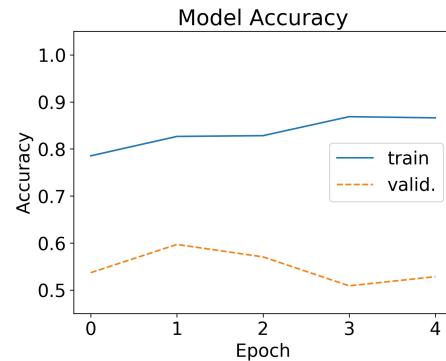
# Pipeline



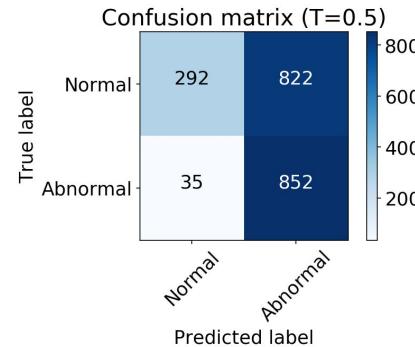
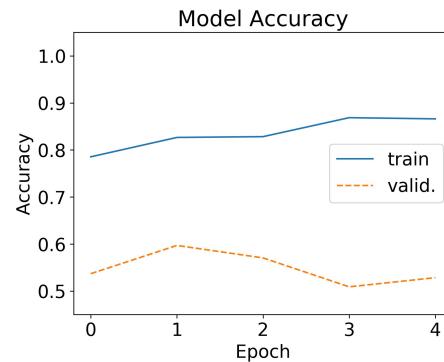
# Results



# Results

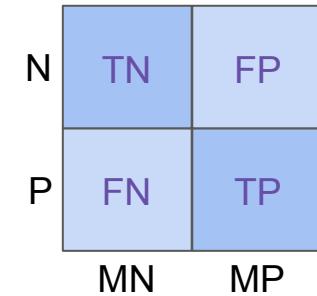


# Results

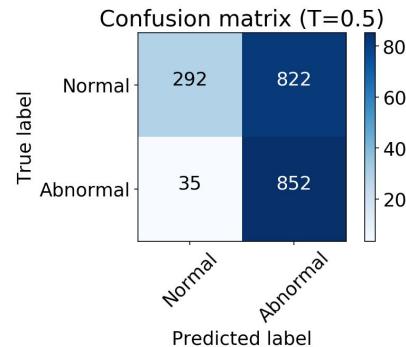
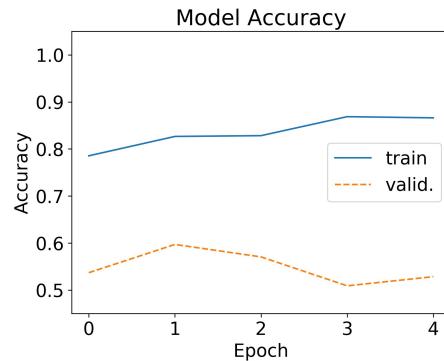


Accuracy

57%

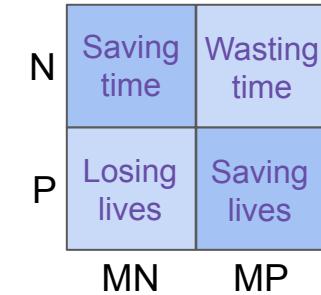


# Results

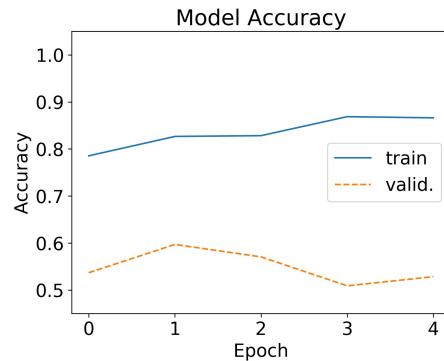


Accuracy

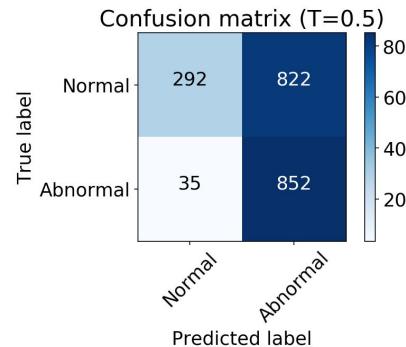
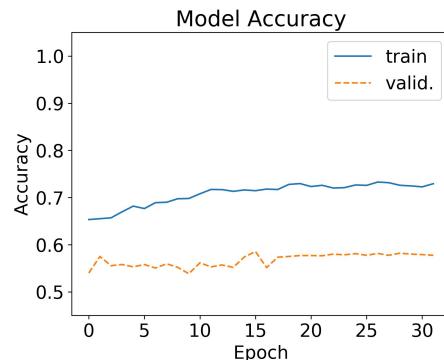
57%



# Results



More regularized

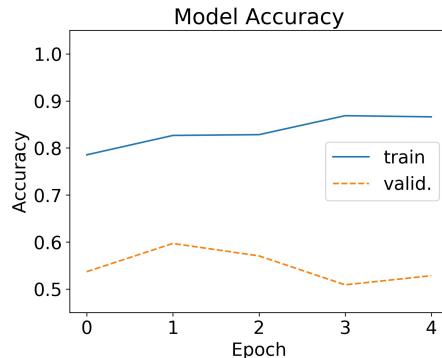


Accuracy

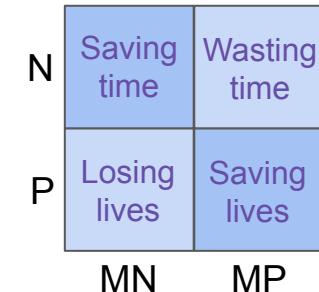
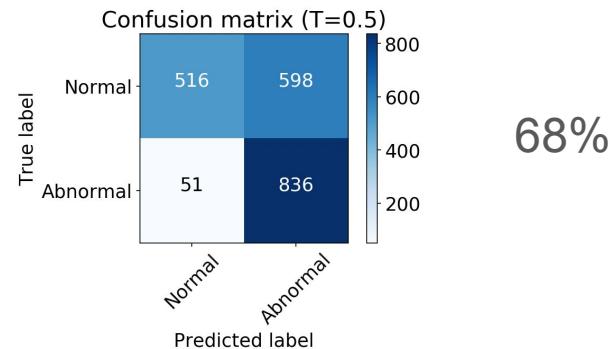
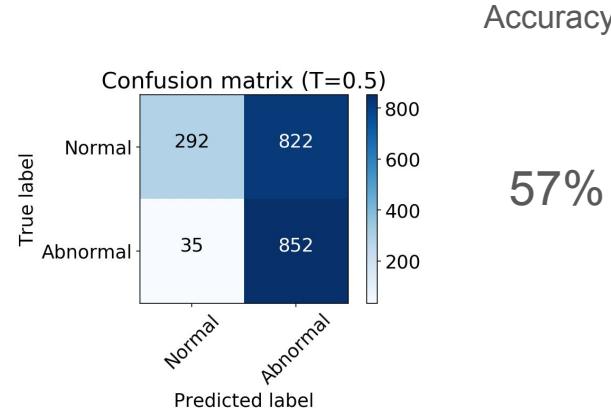
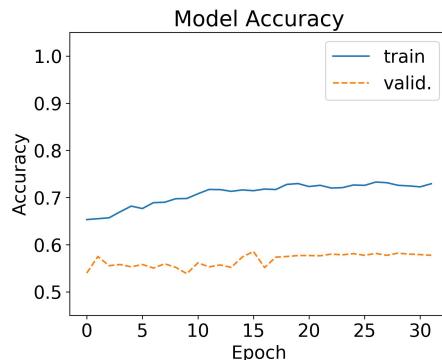
57%

N	Saving time	Wasting time
P	Losing lives	Saving lives
MN		
MP		

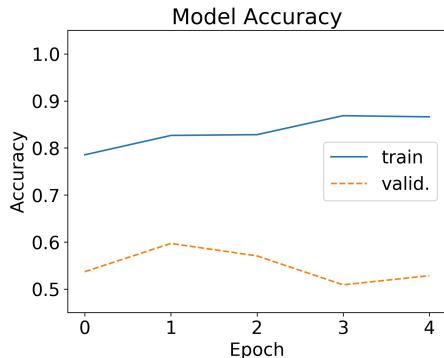
# Results



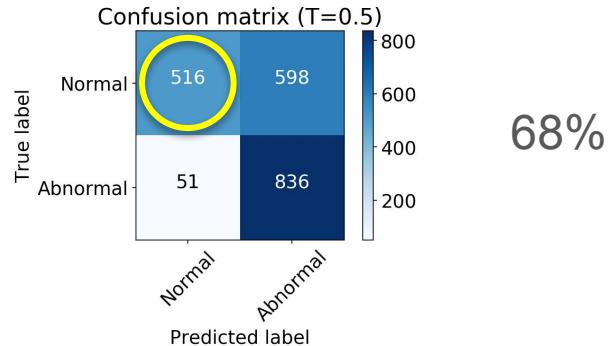
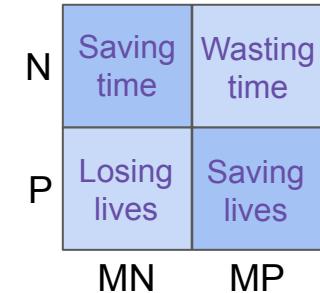
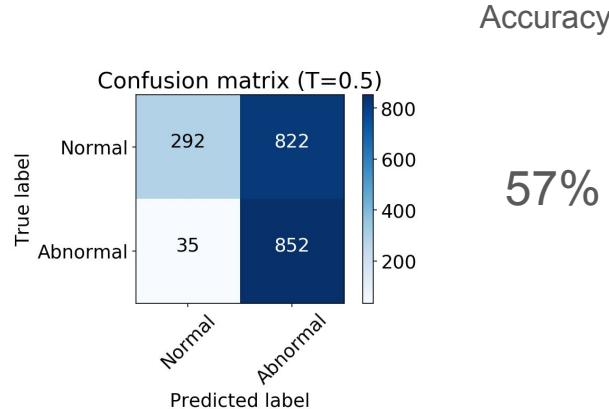
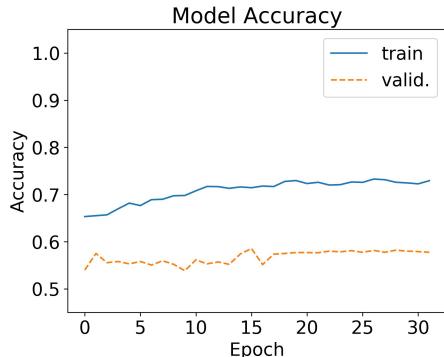
More regularized



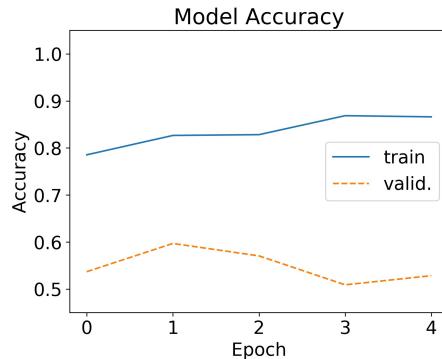
# Results



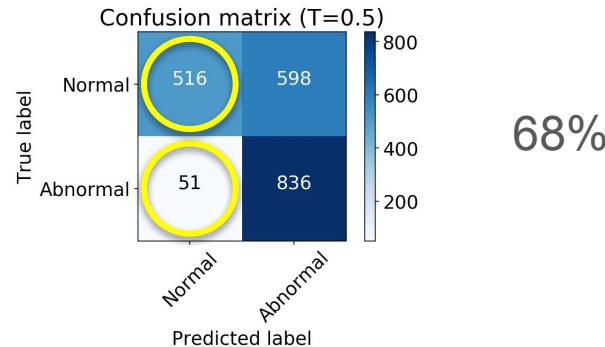
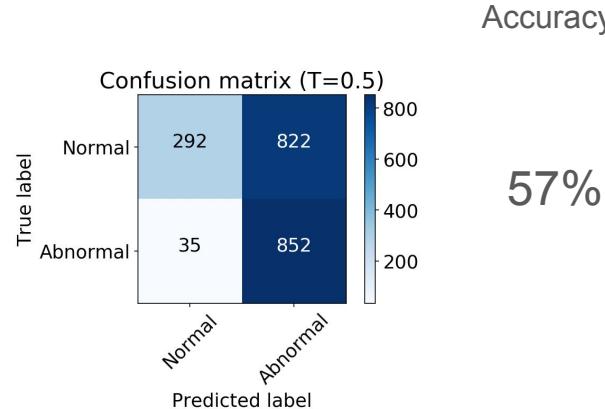
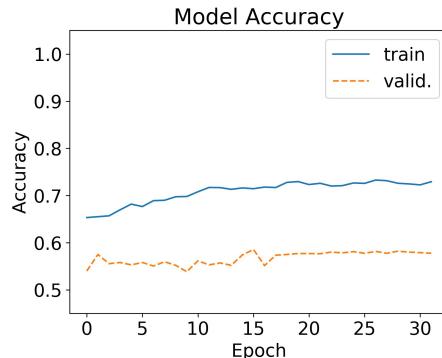
More regularized



# Results



More regularized



N	Saving time	Wasting time
P	Losing lives	Saving lives

MN      MP

Read time: -14%

Lives lost: +46%

# Thank you!

## Andrew Forrester



Physics PhD



NLP & AGI

Interests: Philosophy in action  
Society and its organization  
Alternative education



# Appendix

# Lessons Learned

- Data
  - Quality of data / mis-labeling
  - Reorganize pipeline for raw data, file/folder name/structure
- Analysis
  - Progress ← verification, evaluation
  - Many experiments ← organization, automation
- Development
  - Balance custom dev. with usage of existing tools
  - Beware of “code creep”, re-evaluate strategy
  - Continuous refactoring, retooling
- Tools
  - Efficiency/automation requires an investment
  - Learning curve  
AWS EC2, S3; git-lfs; tmux

# Next Steps

- Data
  - Classify by region, hierarchy of categories
  - Advise / quality control on labeling, raw pipeline
- Analysis
  - Further quantify training improvement
  - Systematize, automate hyperparameter tuning (AutoML, SigOpt?)
- Development
  - Explore existing tools
  - Refactor code, possibly re-tool

# Next Steps

- Data
  - Classify by region, hierarchy of categories
  - Advise / quality control on labeling, raw pipeline
- Analysis
  - Further quantify training improvement
  - Systematize, automate hyperparameter tuning (AutoML, SigOpt?)
- Development
  - Explore existing tools
  - Refactor code, possibly re-tool

Object Detection /  
Segmentation /  
Recognition

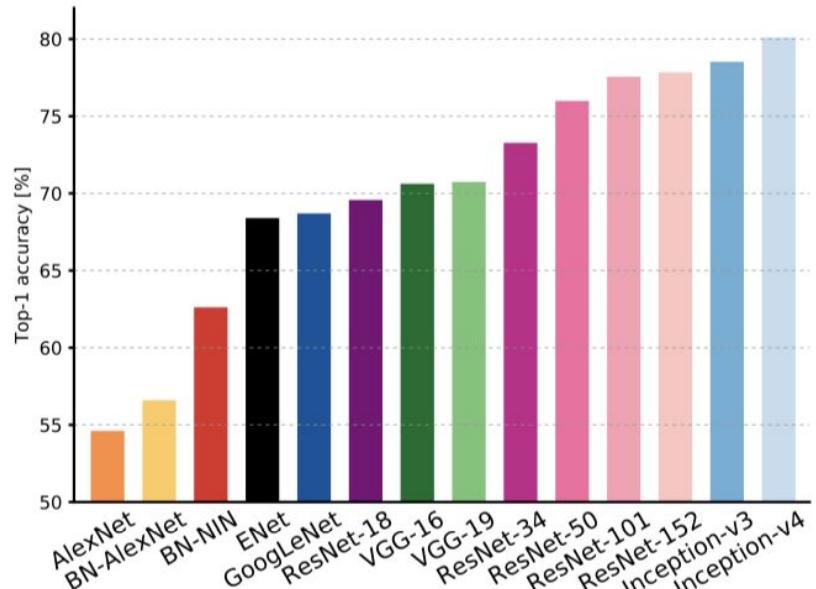
# Models

## Documentation for individual models

Model	Size	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth
Xception	88 MB	0.790	0.945	22,910,480	126
VGG16	528 MB	0.713	0.901	138,357,544	23
VGG19	549 MB	0.713	0.900	143,667,240	26
ResNet50	99 MB	0.749	0.921	25,636,712	168
InceptionV3	92 MB	0.779	0.937	23,851,784	159
InceptionResNetV2	215 MB	0.803	0.953	55,873,736	572
MobileNet	16 MB	0.704	0.895	4,253,864	88
MobileNetV2	14 MB	0.713	0.901	3,538,984	88
DenseNet121	33 MB	0.750	0.923	8,062,504	121
DenseNet169	57 MB	0.762	0.932	14,307,880	169
DenseNet201	80 MB	0.773	0.936	20,242,984	201
NASNetMobile	23 MB	0.744	0.919	5,326,716	-
NASNetLarge	343 MB	0.825	0.960	88,949,818	-

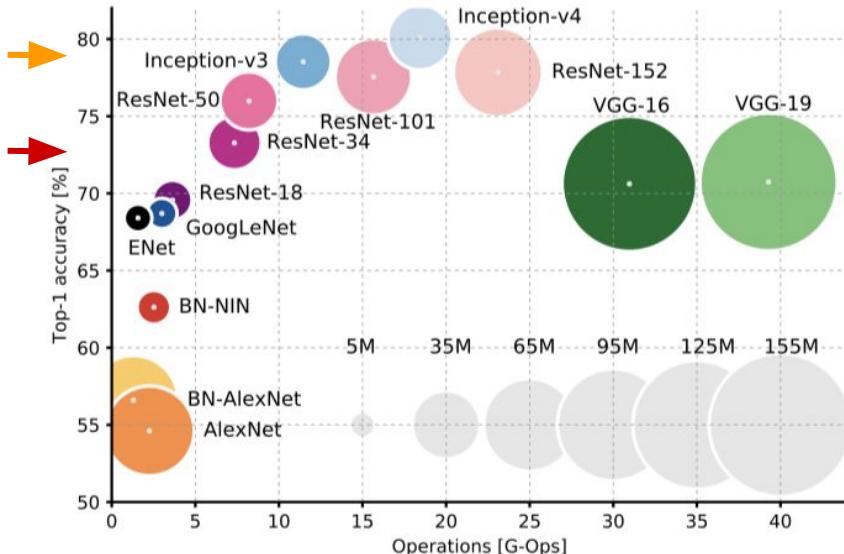


# Models



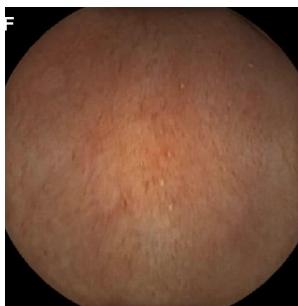
MobileNetV2

Xception



# Data Processing Example

- Anonymize
- Decrease size of inputs / network
- Erase irrelevant noise



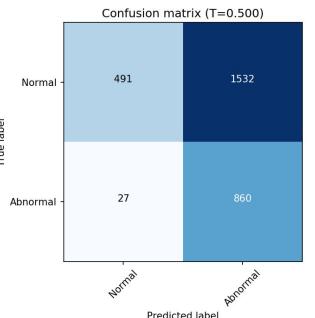
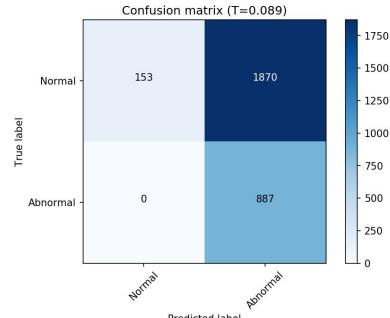
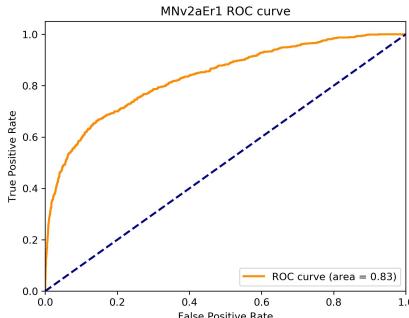
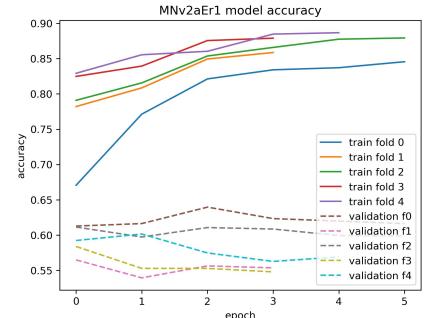
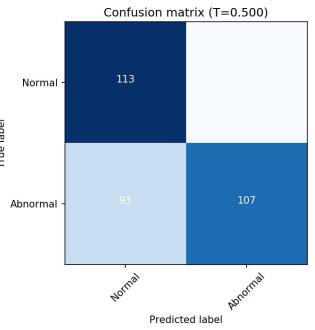
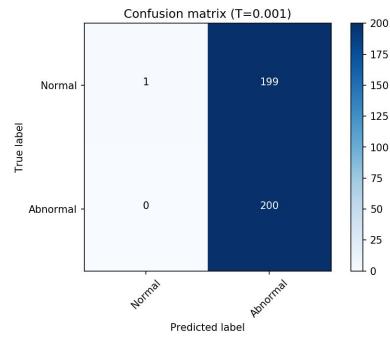
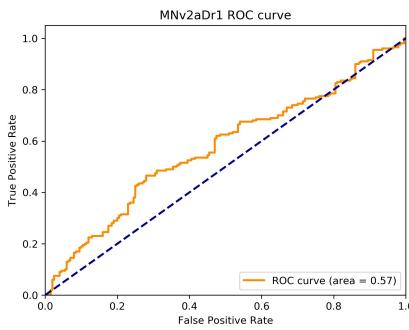
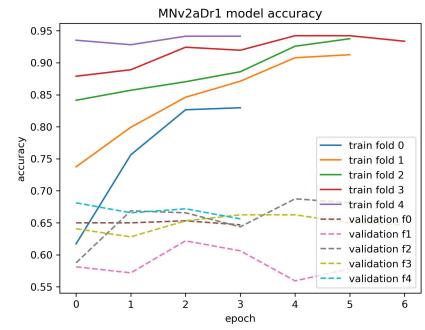
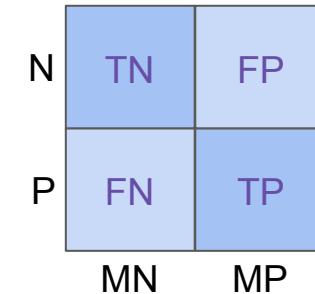
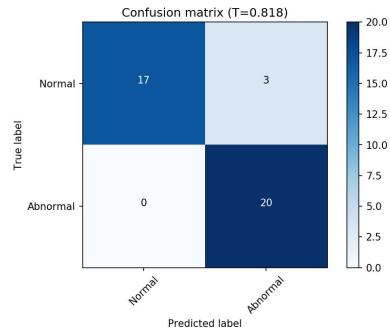
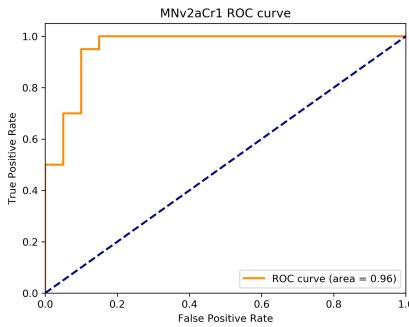
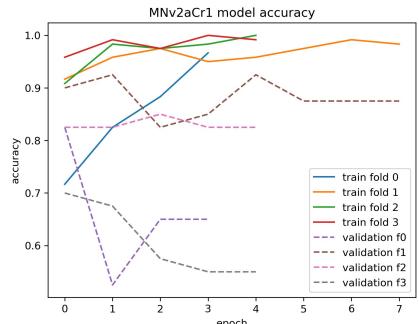
crop

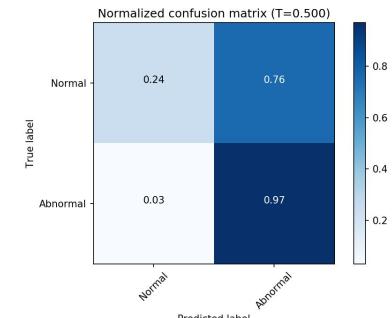
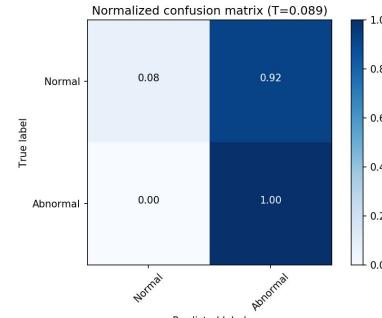
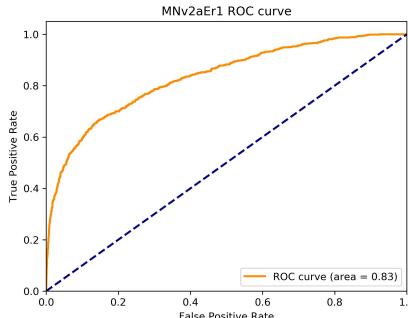
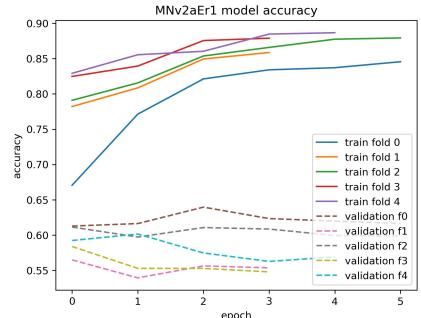
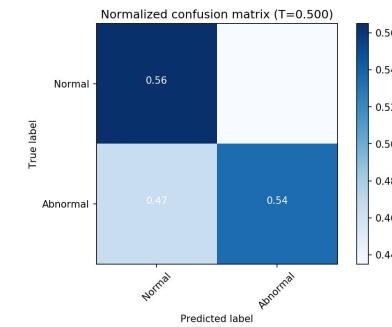
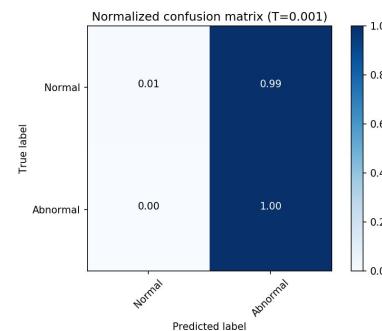
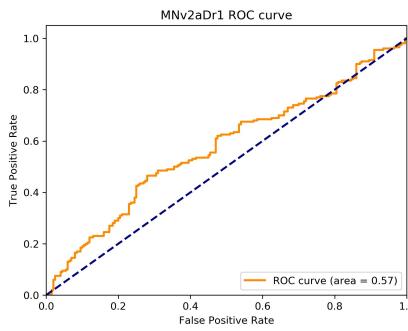
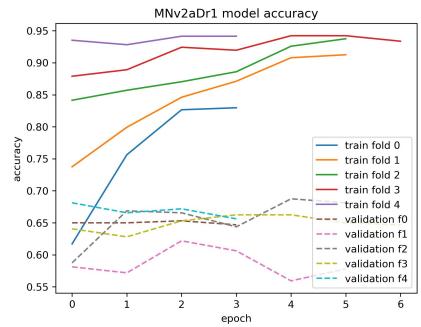
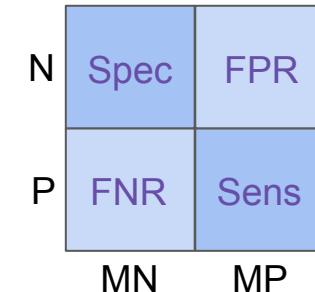
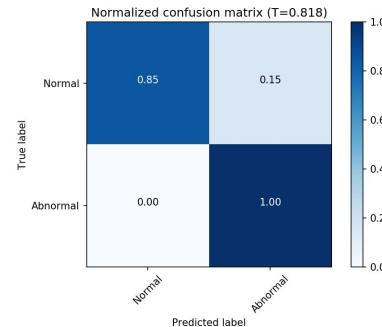
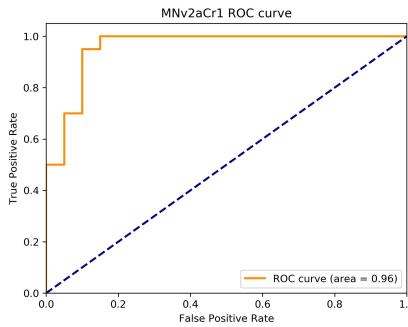
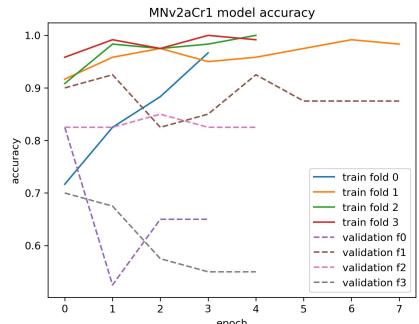


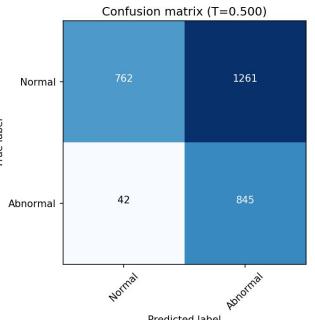
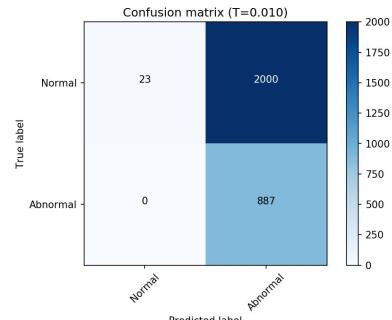
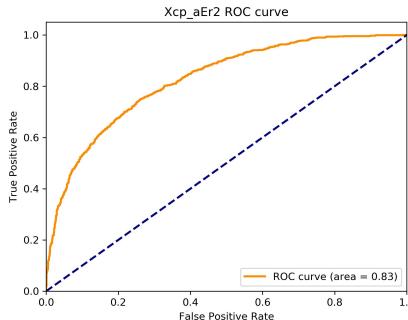
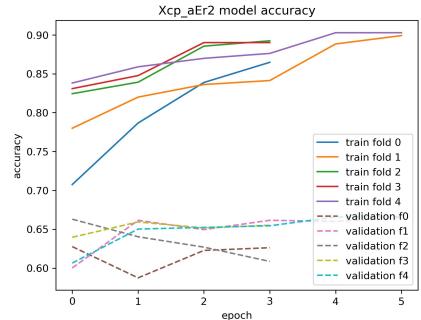
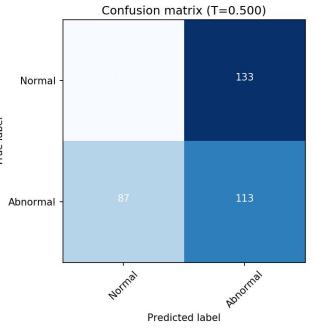
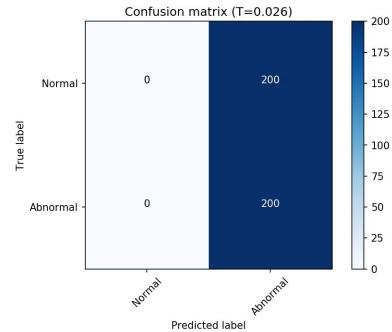
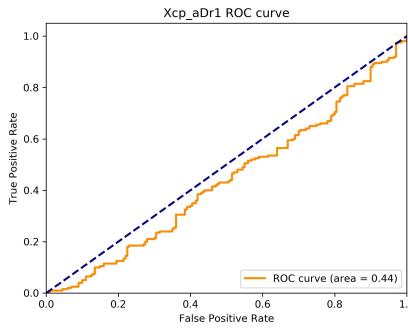
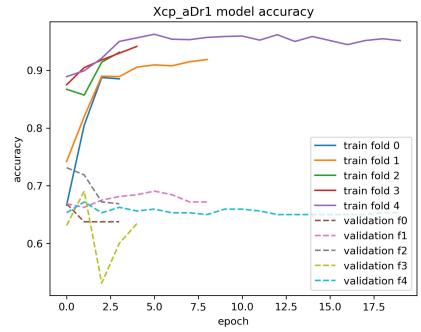
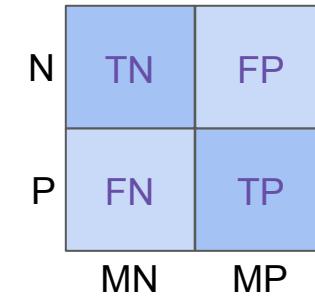
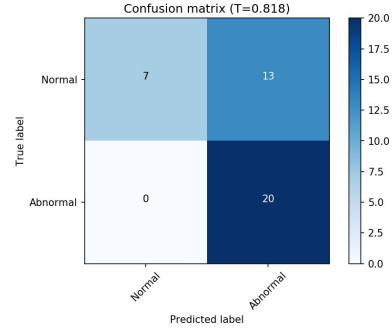
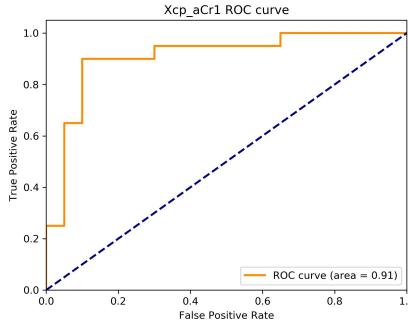
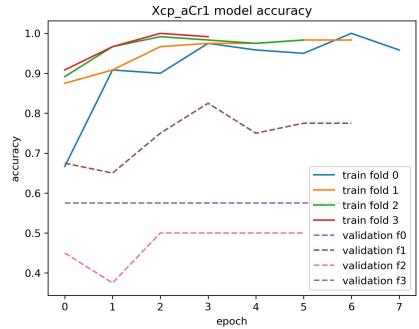
mask (green)

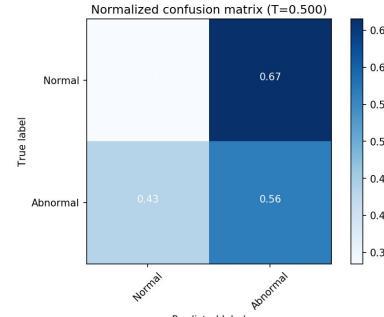
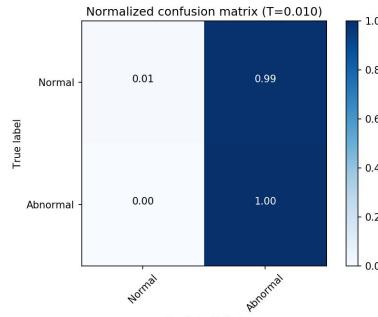
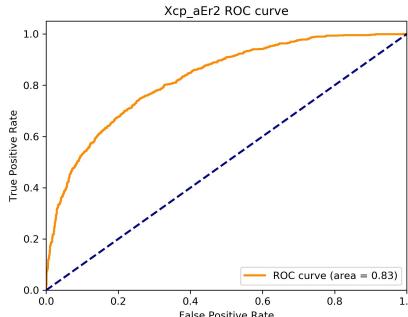
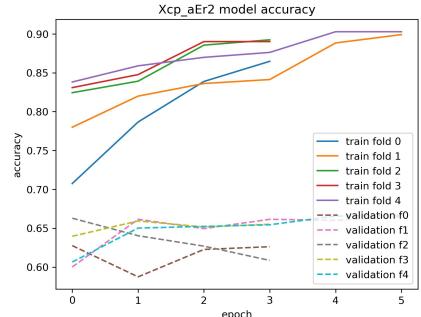
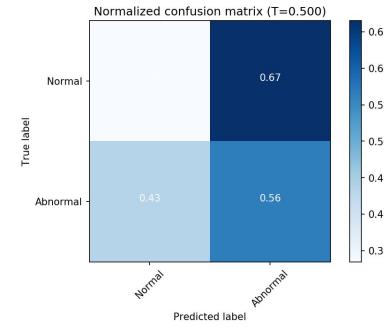
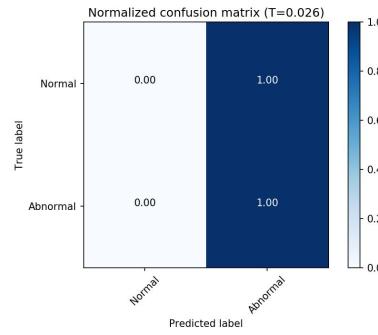
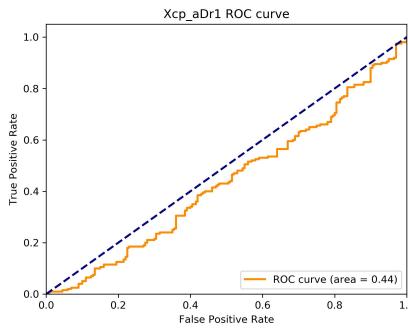
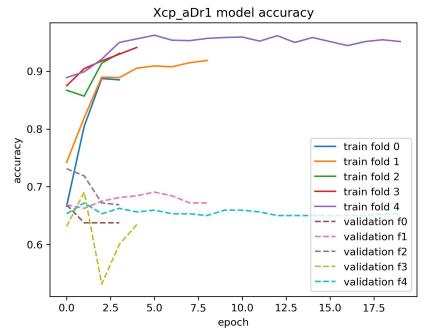
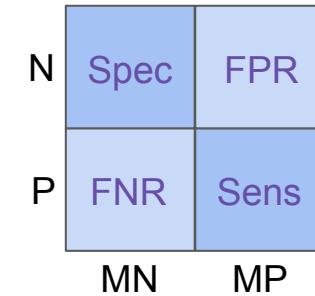
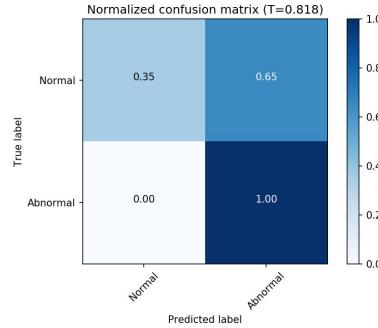
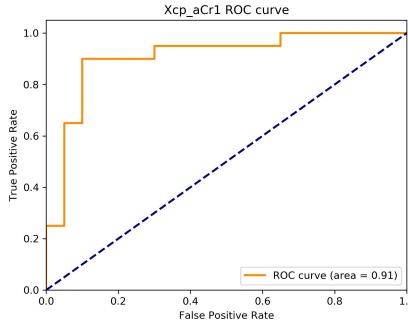
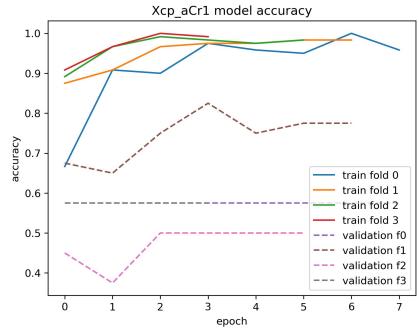


mask (black)



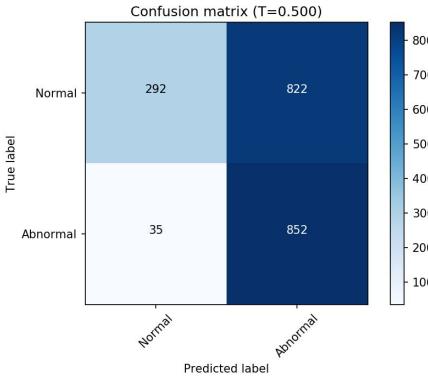
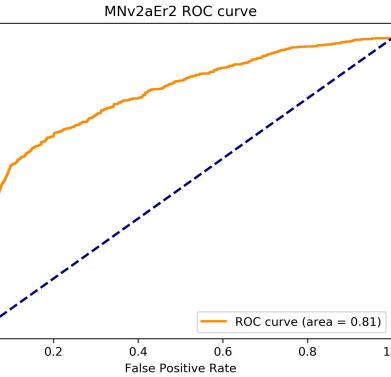
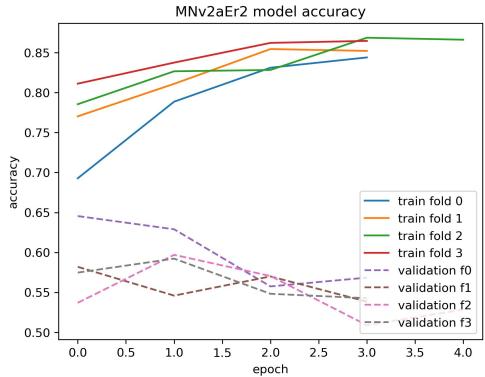






# MobileNetV2

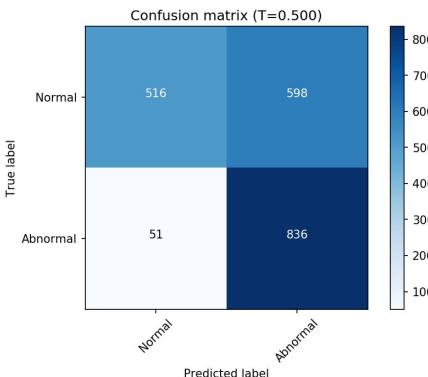
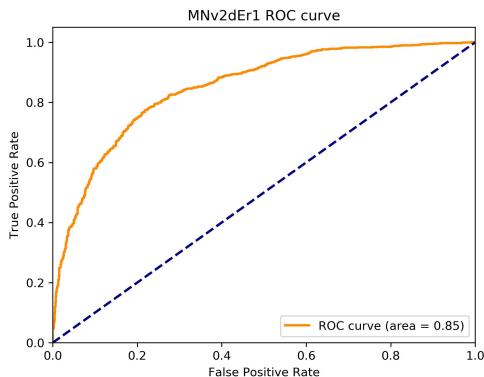
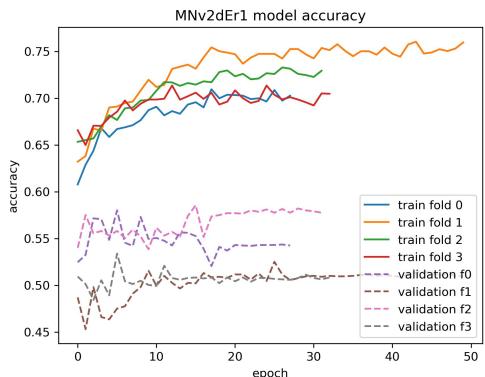
a)



Accuracy

57%

More regularization



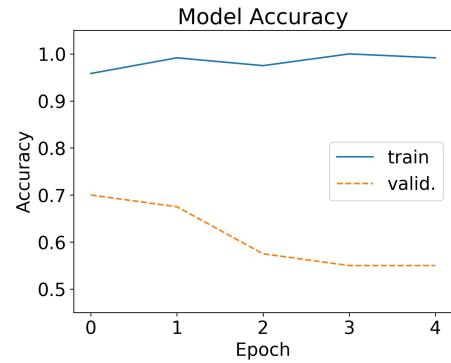
68%

TRAIN								
MobileNet	Speed (imgs/sec)	Acc		Sens		Spec		
Xception	tot time	thresh1	thresh2	thresh1	thresh2	thresh1	thresh2	
C	0.9	93	80	100	100	85	60	
cpu:	0.4	68	55	100	100	35	10	
45 min	3 min							
2 hr	10 min							
D	1.2	50	55	100	54	0.5	57	
	0.2	50	45	100	57	0	34	
	30 min							
	2 hr 15 min							
E	1.4	36	46	100	97	8	24	
	0.5	31	55	100	95	1	38	
	2 hr							
	6 hr							
INFER								
C	13	95	70	100	100	90	40	
	8	78	55	100	100	55	10	
15 s	3 s							
45 s	5 s							
D	40	50	50.5	100	52	0	49	
	15	50	49	100	78	0	20	
2 min 15 s	10 s							
7 min 30 s	30 s							
E	59	30	45	100	96	0	22	
	17	30	60	100	91	0	46	
	50 s							
1 hr 50 min	3 min							

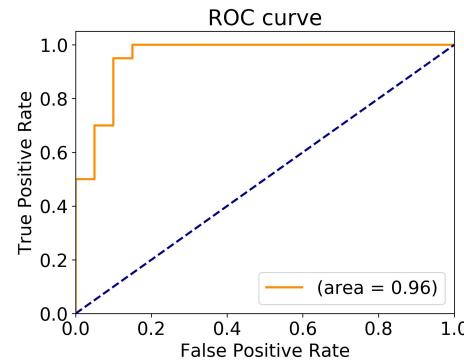
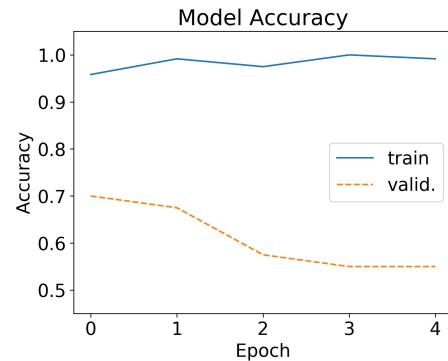
# Results

- MobileNetV2
- 200 images
- Colon: Normal vs Abnormal (blood)

# Results



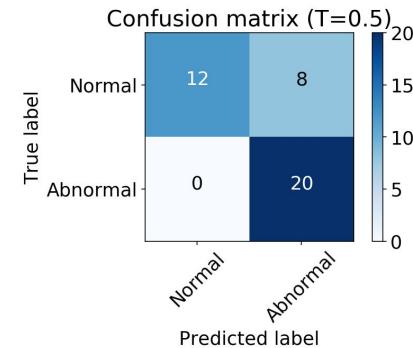
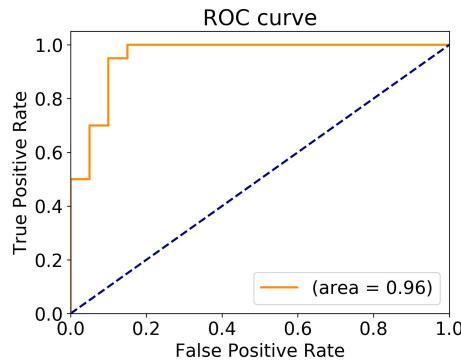
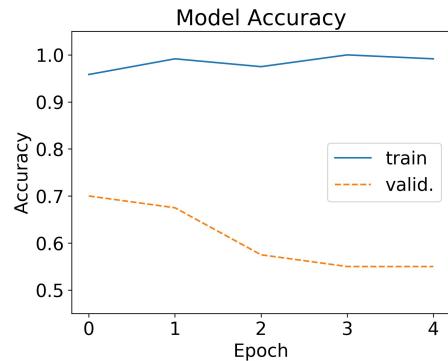
# Results



# Results

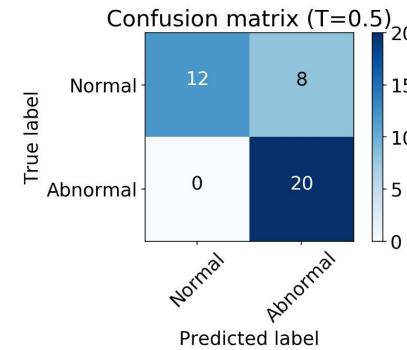
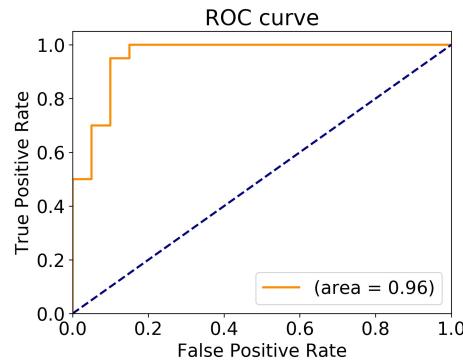
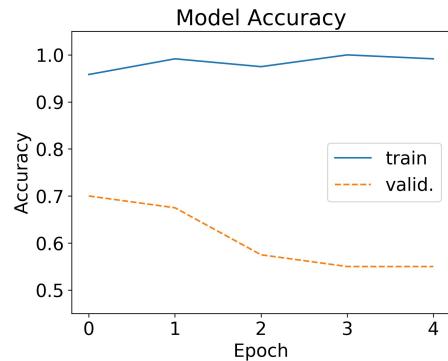
Accuracy

80%

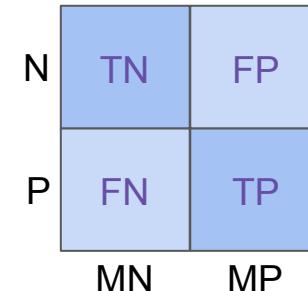


# Results

Accuracy



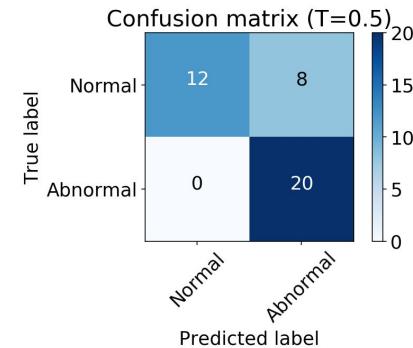
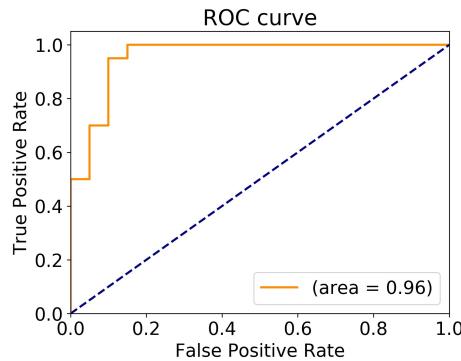
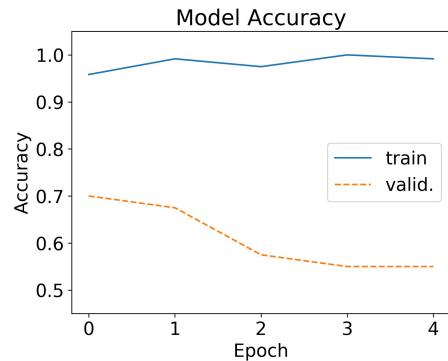
80%



# Results

Accuracy

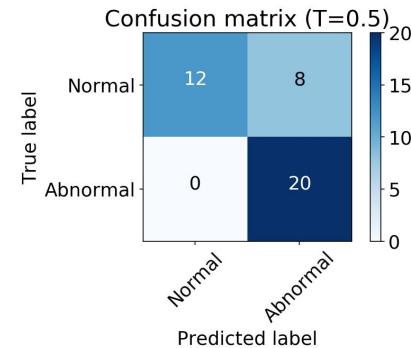
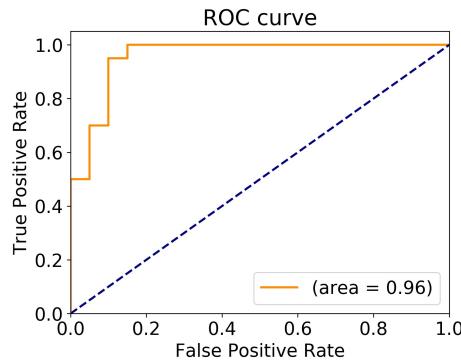
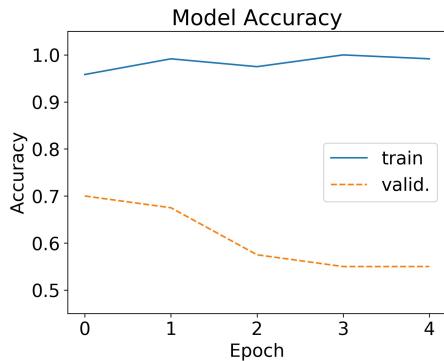
80%



# Results

Accuracy

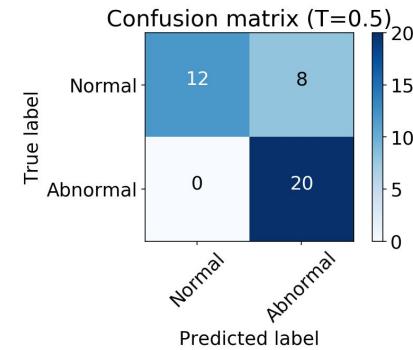
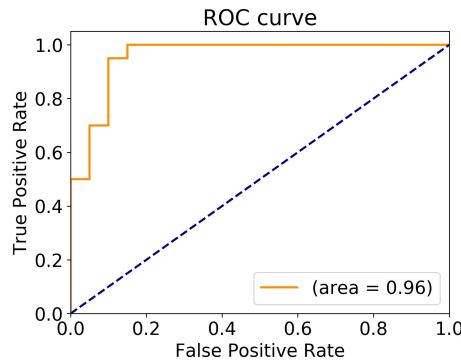
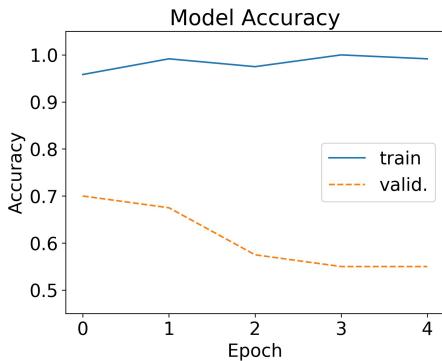
80%



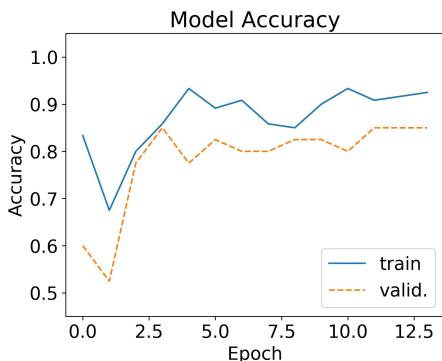
More regularized

# Results

Accuracy

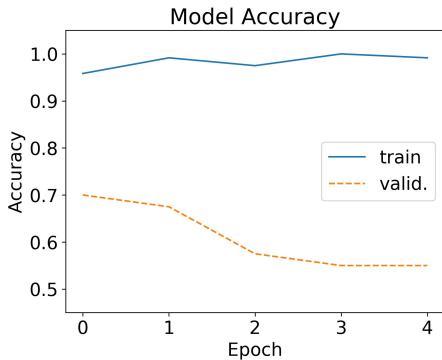


More regularized

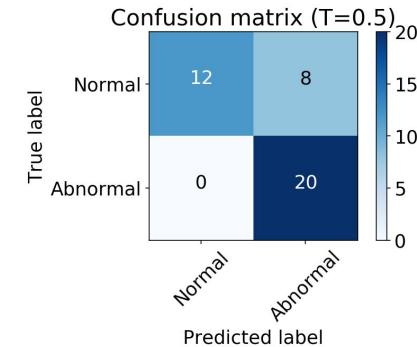
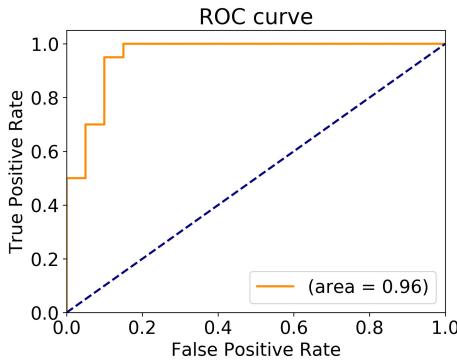


# Results

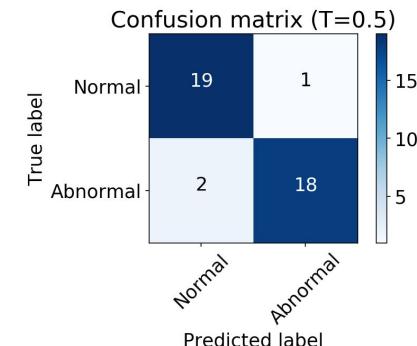
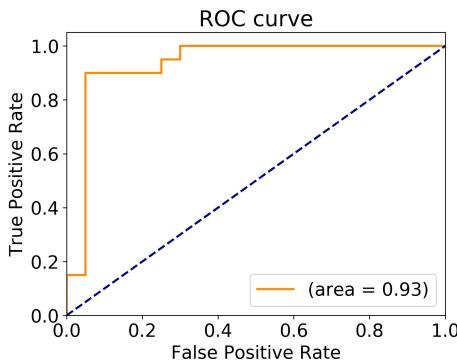
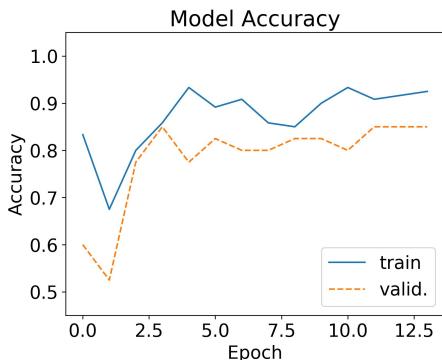
Accuracy



More  
regularized



80%



85%

# Goal



Python package:

- abnormality classifier
- region classifier



GitHub

Performance:

- higher accuracy ( $> 0.96$  AUC)
- need high **sensitivity** (few false negatives)
- want high **specificity** (few false positives)
- want high speed

