Histopathological Tissue Based Cancer Tumor Detection

Chenyu Wang, Hector Rincon, Ifrah Javed, Justin To and Srila Maiti

AGENDA

01
PROJECT MOTIVATION
Research Question and Application

02
DATASET AND AUGMENTATION
Data Source and Feature Engineering

03
OUR APPROACH
Iterative Process, Metrics

04
FINAL MODEL
Architecture and Performance

05
FUTURE WORK
Further experimentation and questions

01

PROJECT MOTIVATION

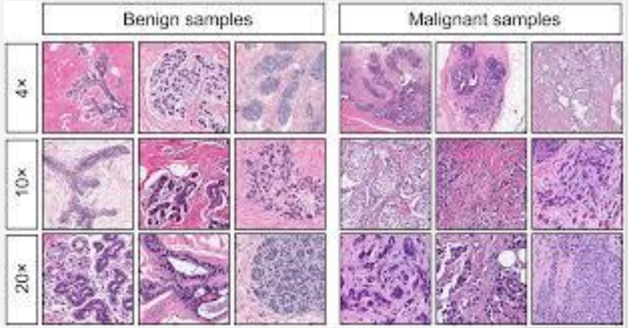
PROJECT BACKGROUND AND NEED

KEY CANCER TREATMENT FACTORS

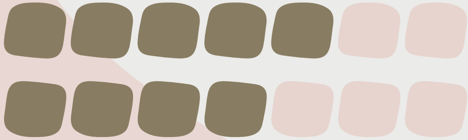
CORRECT DIAGNOSIS

EARLY INTERVENTION

CURRENT DIAGNOSTIC METHOD

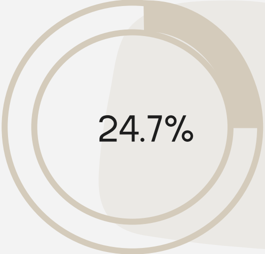


DEATH RATE



609,360
Deaths out of
1.9 million
cases in 2022

MISDIAGNOSIS RATE



RESEARCH QUESTION

"CAN WE PREDICT IF A TUMOR IS BENIGN OR MALIGNANT BASED ON A HISTOPATHOLOGICAL TISSUE IMAGE?"

PROJECT APPLICATION

PATIENT PRESENTS SYMPTOMS

Start of care.

BIOPSY

Surgical process to remove cells and create a histopathological image.

OUR ALGORITHM

A diagnosis is provided by running our algorithm.

TREATMENT

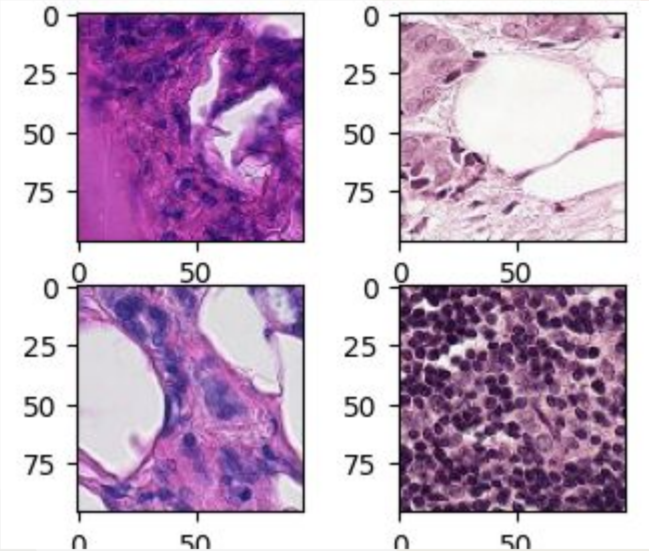
Treatment plan is decided based on diagnosis.

02

DATASET & AUGMENTATION

DATASET

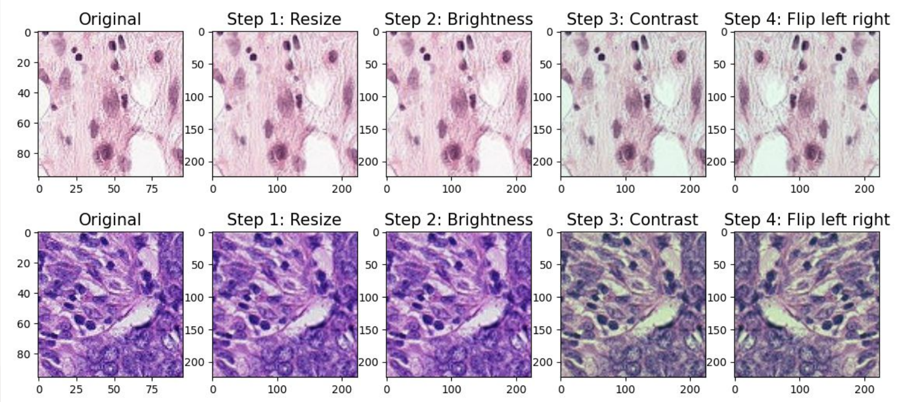
- Working with image data of pathological cells for cancer detection
- A version of the PCAM data set (https://github.com/basveeling/pcam)
- Cleaned from duplicates by Kaggle for the Histopathologic Cancer Detection competition
- Dataset contains 277,483 total images (220,025 train / 57,458 test)
- Test images are not labeled so we repurposed the 220K images and made the split on that subset
- Positive label: center 32x32px (out of 96x96 image) region of a patch contains at least one pixel of tumor tissue



[1] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, M. Welling. "Rotation Equivariant CNNs for Digital Pathology". arXiv:1806.03962

[2] Ehteshami Bejnordi et al. Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. JAMA: The Journal of the American Medical Association, 318(22), 2199–2210. doi:jama.2017.14585

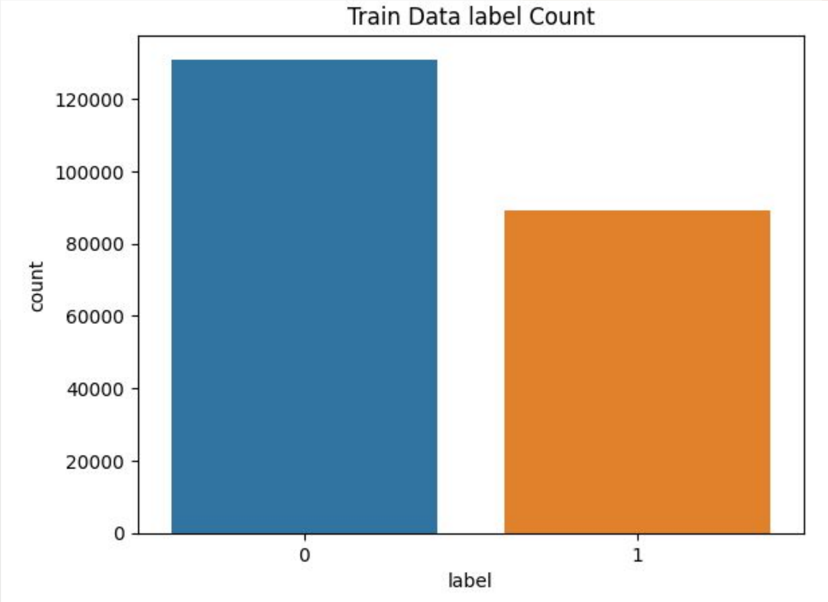
DATA AUGMENTATION



- Converted to grayscale (no info from color data)
- Attempted to use ImageDataGenerator (does not add images, but returns the new transformations)
- Aug. steps:
  - Contrast
  - Brightness
  - Random flips
  - Rotation (90, 180, 270) comparison
- Attempted to crop to 32x32/64x64 center region
- Limitations/calls of judgement

UNBALANCED DATA

- The training dataset had an imbalance of the positive vs negative examples (negative overrepresented).
- Solution: undersample the negative examples (took a random sample of # of positive examples available in the dataset)



03

OUR APPROACH

PAST RESEARCH

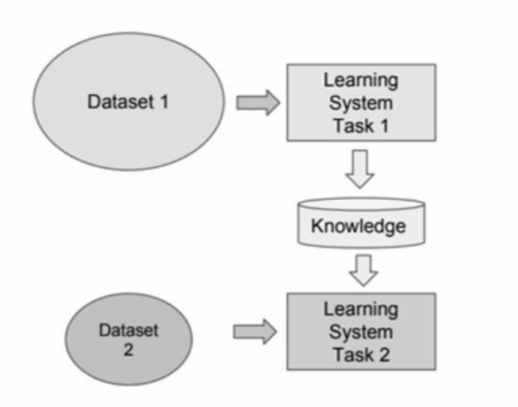
CNN → Transfer Learning → Ensemble

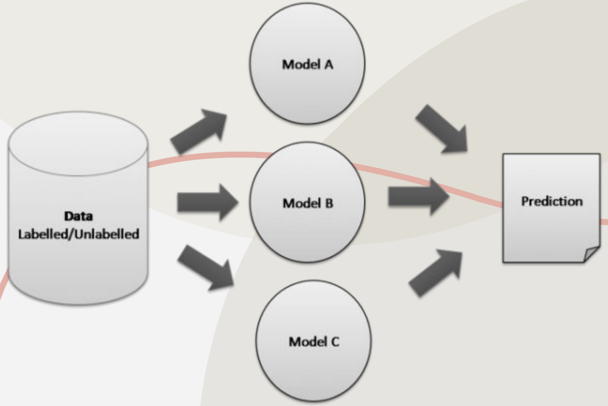
The easier and faster aspect with pre-trained CNN model

Reduce the generalization error of the prediction

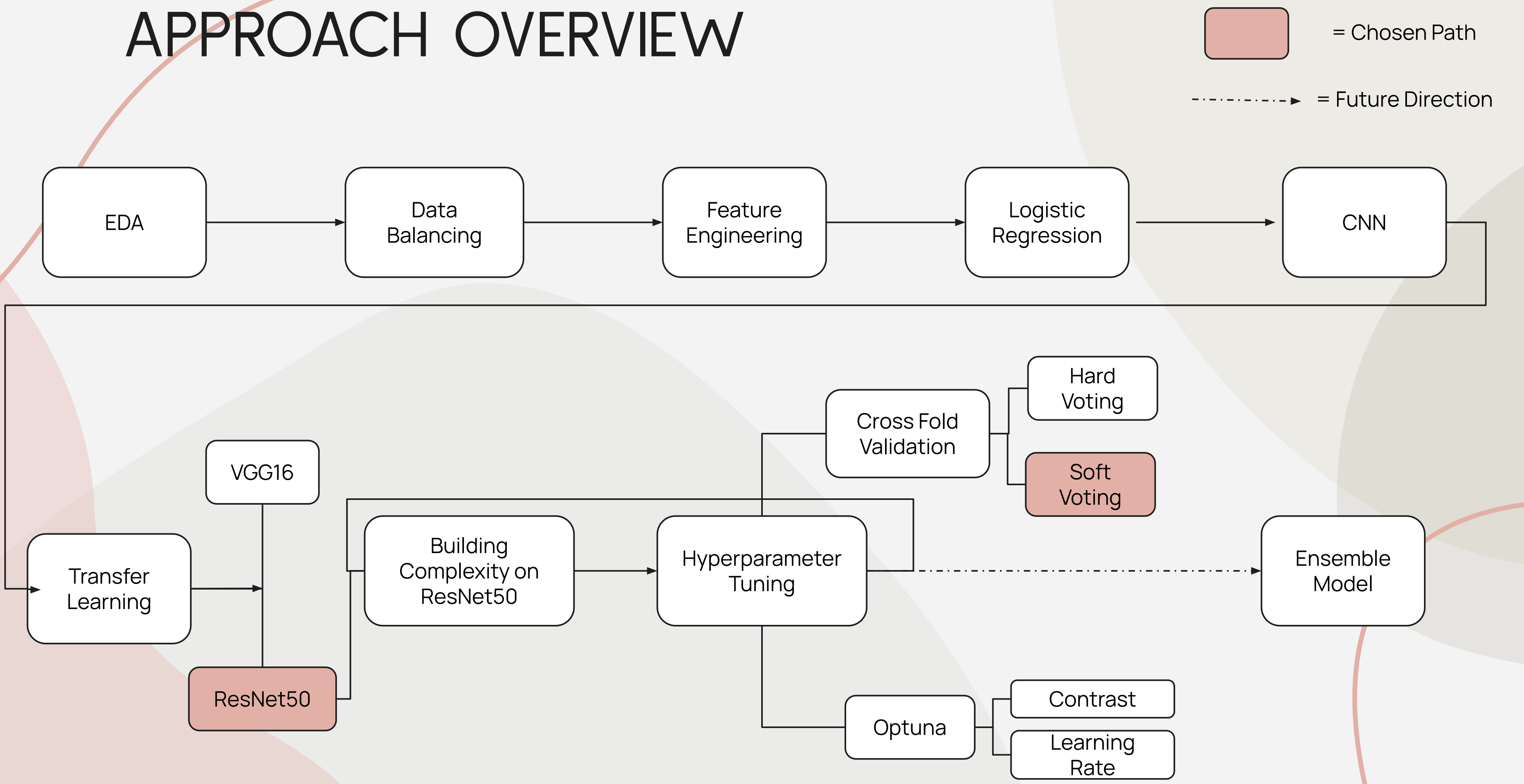
Transfer learning models:

AlexNet, VGG, ResNet, Inception, Xception, ...





APPROACH OVERVIEW



EVALUATION METRICS

METRIC	RATIONALE
Validation Accuracy	How well the model performs on validation data, validation accuracy is generally an indication if the model is overfitting or not.
Test Accuracy	How well the model performs on test data.
Validation False Negative Rate	How many true positives are wrongly misclassified as benign in the validation data
Test False Negative Rate	How many true positives are wrongly misclassified as benign in the test data
Validation AUC	FPR vs TPR rate, denotes how well the classifier can differentiate between benign and malignant class in the validation data, higher the number better it’s
Test AUC	FPR vs TPR rate, denotes how well the classifier can differentiate between benign and malignant class in the test data, higher the number better it’s

PERFORMANCE OF PREVIOUS MODEL EXPERIMENTS

Iterations	Model	Number of Parameters	Test Accuracy	Test AUC
1	Logistic Regression	22,564,225	63.74%	64.51%
2	CNN	4,906,753	80.55%	80.56%
3	CNN with dropout layer	4,906,753	79.77%	79.28%
4	CNN with BatchNormalization	4,773,738	82.73%	80.75%
5	CNN, multiple convolutional layer and BatchNormalization with asymmetric filtering	4, 779, 530	83.74%	91.60%
6	VGG16	14, 714, 688	84.11%	84.28%
7	Mobilenet	3, 229, 889	91.15%	97.29%
8	Xception + Mobilenet	23, 122, 793	90.75%	97.19%

04

FINAL MODEL

FINAL MODEL: RESNET

MODELS TESTED

VGG16

- **13 Conv2D** + **3 Dense**
- **Small kernels** (3x3 or 1x1):
  - multiple layers w/o pooling → mimics lager kernels
  - but **more non-linearity** & **less parameters**
- **Increasing channel size**
- Reduces effort on testing layer adding

ResNet50

- **Improved VGG**, with **48 Conv2D** + 2 Pooling
- Deeper networks:
  - VGG shows deeper **better**
  - **degradation** can happen, unrelated to overfitting
- Adds **shortcut connections**
- **Improved performance over VGG**

Xception & MobileNet

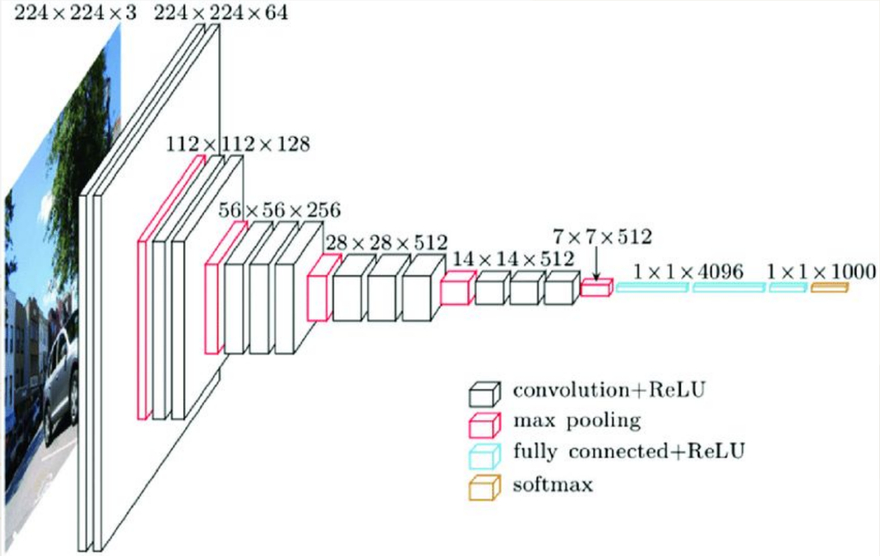
- **Improved Inception** model:
  - main idea: multiple convolutions, then concatenating them
- **Depthwise separable convolution layers** for faster computation

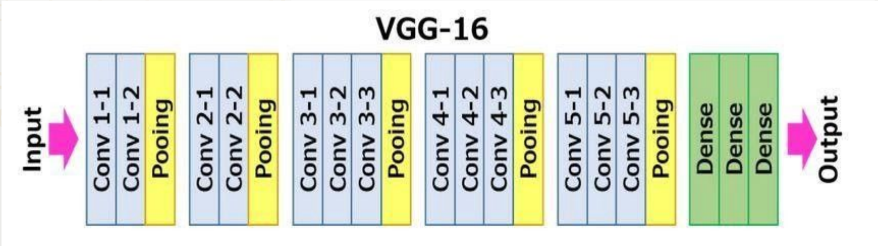
FINAL MODEL: RESNET

MODELS TESTED

VGG16

- **13 Conv2D** + **3 Dense**
- **Small kernels** (3x3 or 1x1):
  - multiple layers w/o pooling → mimics lager kernels
  - but **more non-linearity** & **less parameters**
- **Increasing channel size**
- Reduces effort on testing layer adding





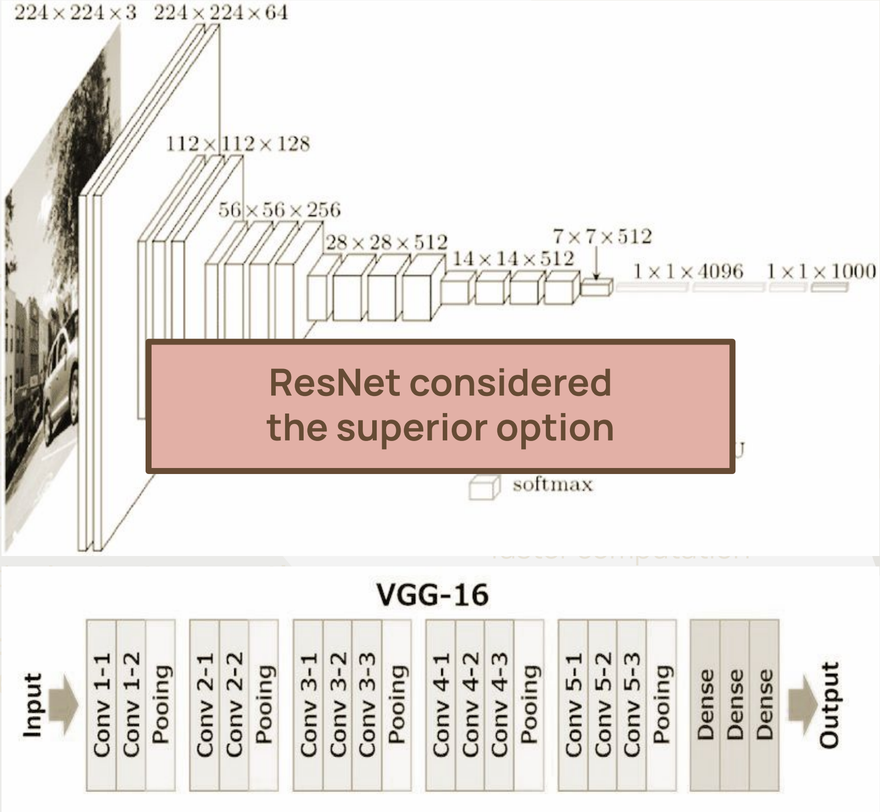
Reference: Very deep convolutional networks for large-scale image recognition; by Simonyan & Zisserman; 2014

FINAL MODEL: RESNET

MODELS TESTED

VGG16

- **13 Conv2D** + **3 Dense**
- **Small kernels** (3x3 or 1x1):
  - multiple layers w/o pooling → mimics lager kernels
  - but **more non-linearity** & **less parameters**
- **Increasing channel size**
- Reduces effort on testing layer adding



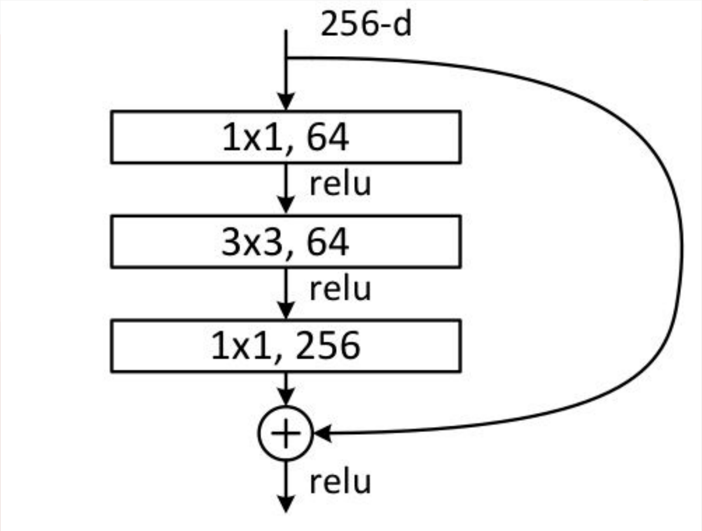
ResNet considered the superior option

Reference: Very deep convolutional networks for large-scale image recognition; by Simonyan & Zisserman; 2014

FINAL MODEL: RESNET

MODELS TESTED

VGG16 ⇨ **ResNet50** ⇨ Xception & MobileNet



- **Improved VGG**, with **48 Conv2D** + 2 Pooling
- Deeper networks:
  - VGG shows deeper **better**
  - **degradation** can happen, unrelated to overfitting
- Adds **shortcut connections**
- **Improved performance over VGG**

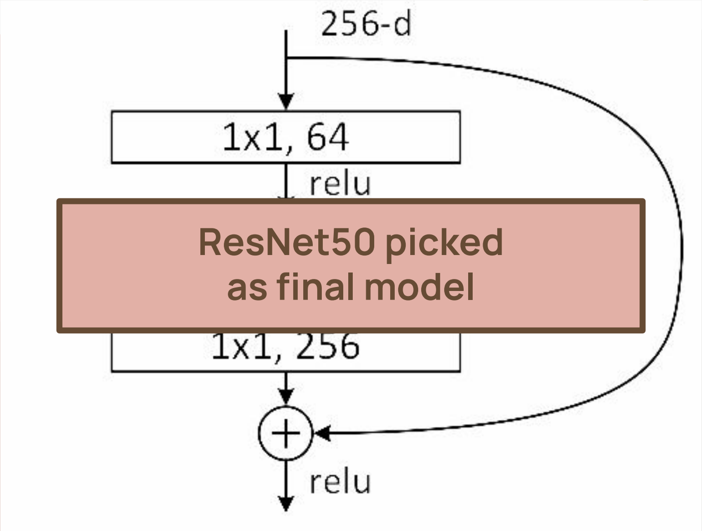
- **Improved Inception** model:
  - main idea: multiple convolutions, then concatenating them
- **Depthwise separable convolution layers** for faster computation

Reference: Deep Residual Learning for Image Recognition; by K He et al; 2016

FINAL MODEL: RESNET

MODELS TESTED

VGG16 ⇨ **ResNet50** ⇨ Xception & MobileNet



Reference: Deep Residual Learning for Image Recognition; by K He et al; 2016

ResNet50

- **Improved VGG**, with **48 Conv2D** + 2 Pooling
- Deeper networks:
  - VGG shows deeper **better**
  - **degradation** can happen, unrelated to overfitting
- Adds **shortcut connections**
- **Improved performance over VGG**

Xception & MobileNet

- **Improved Inception** model:
  - main idea: multiple convolutions, then concatenating them
- **Depthwise separable convolution layers** for faster computation

FINAL MODEL: ARCHITECTURE

BASE TRANSFER LEARNING MODEL

- **ResNet50**:
  - 50 layers total
  - 3 channels input
  - at least 32 x 32
  - trained on ImageNet

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		
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	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

Source: Deep Residual Learning for Image Recognition; by K He et al; 2016

FINAL MODEL: ARCHITECTURE

FOR EACH K-FOLD:

- **ResNet50**
- After the 50 layers:
  - Global **Average Pooling**
  - **Dropout** (rate at 0.1)
  - Flatten
  - Dense FC (2048 → 1; logits for soft voting)
- All layers **fully trainable**, given:
  - Large dataset
  - **Difference between medical & general images**
- Optimizer: **Adam with eta = 0.001**
- # of **epochs: 10**

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 96, 96, 3)]       0

 resnet50 (Functional)       (None, 3, 3, 2048)        23587712

 global_average_pooling2d (G  (None, 2048)             0
 lobalAveragePooling2D)

 dropout (Dropout)           (None, 2048)              0

 flatten (Flatten)           (None, 2048)              0

 dense (Dense)               (None, 1)                 2049

=================================================================
Total params: 23,589,761
Trainable params: 23,536,641
Non-trainable params: 53,120
_________________________________________________________________
```

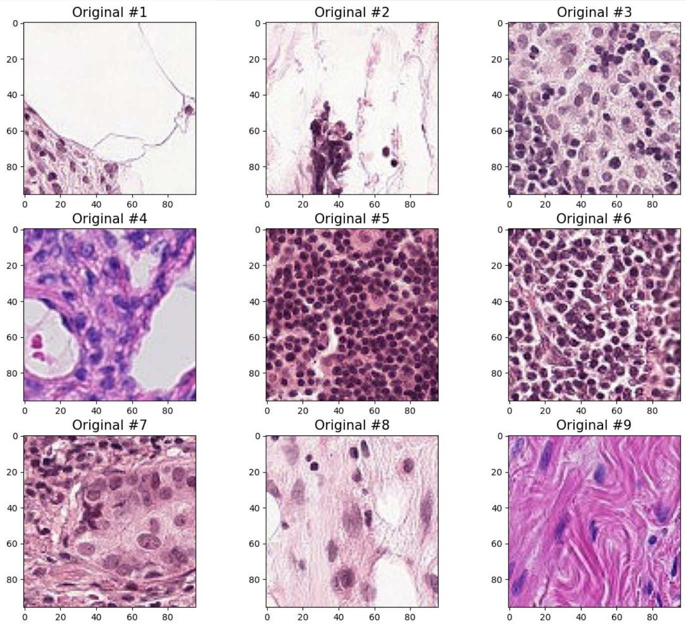
FINAL MODEL: IMAGE PREPROCESSING

Image Augmentation Steps

- Unlike general image processing tasks, **medical images** (specifically histopathological tissue films) are:
  - more **standardised** (color, contrast, etc)
  - **color carries no info** (reflects dye color)
  - other **random effects less likely** (e.g. size changes, non-right-angled shifts, random shear, etc.)

- Selected augmentations:
  - **random flipping** left-right
  - **right-angled rotations** (90°, 180° & 270°)
  - **rotated images added** to training set:
    - equally valid images
    - allows model to focus on patterns not depending on orientation

- Consistent with **experimentation**



First 9 Original Images

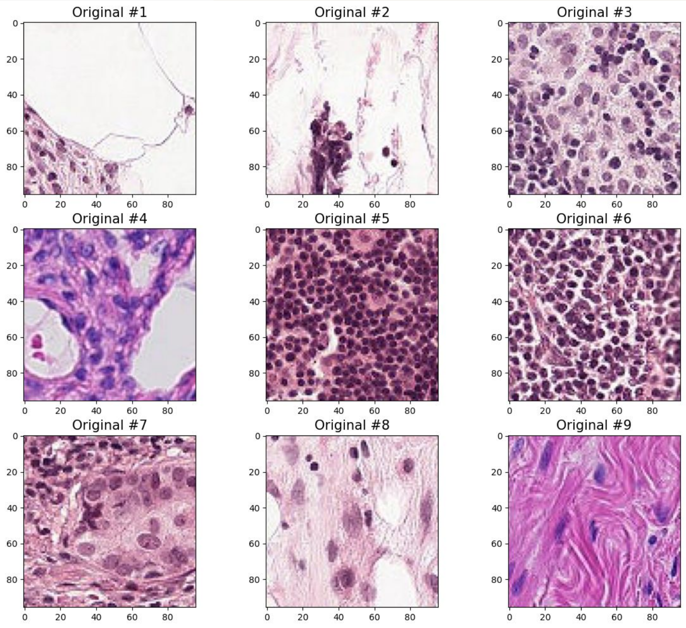
FINAL MODEL: IMAGE PREPROCESSING

Image Augmentation Steps

- Unlike general image processing tasks, **medical images** (specifically histopathological tissue films) are:
  - more **standardised** (color, contrast, etc)
  - **color carries no info** (reflects dye color)
  - other **random effects less likely** (e.g. size changes, non-right-angled shifts, random shear, etc.)

- Selected augmentations:
  - **grey-scaling** (then tripled it back to three channels)
  - **random flipping & right-angled rotations** (90°, 180° & 270°)
  - **rotated images added** to training set:
    - equally valid images
    - allows model to focus on patterns not depending on orientation

- Consistent with **experimentation**



First 9 Original Images

FINAL MODEL: IMAGE PREPROCESSING

Image Augmentation Steps

- Unlike general image processing tasks, **medical images** (specifically histopathological tissue films) are:
    - more **standardised** (color, contrast, etc)
    - **color carries no info** (reflects dye color)
    - other **random effects less likely** (e.g. size changes, non-right-angled shifts, random shear, etc.)
- Selected augmentations:
    - **random flipping** left-right
    - **right-angled rotations** (90°, 180° & 270°)
    - **rotated images added** to training set:
        - equally valid images
        - allows model to focus on patterns not depending on orientation
- Consistent with **experimentation**

	Contrast Factor Applied to Images		
	0	3	5
Avg Val Loss (Epoch 10)	**0.3010** (sd: 0.046)	0.3908 (sd: 0.096)	0.3890 (sd: 0.144)
Avg Val Acc (Epoch 10)	**90.76%** (sd: 1.43%)	89.28% (sd: 1.80%)	88.78% (sd: 3.63%)
Test Acc (Ensemble)	**94.48%**	93.42%	92.32%
AUC ROC (Ensemble)	**0.9854**	0.9805	0.9777

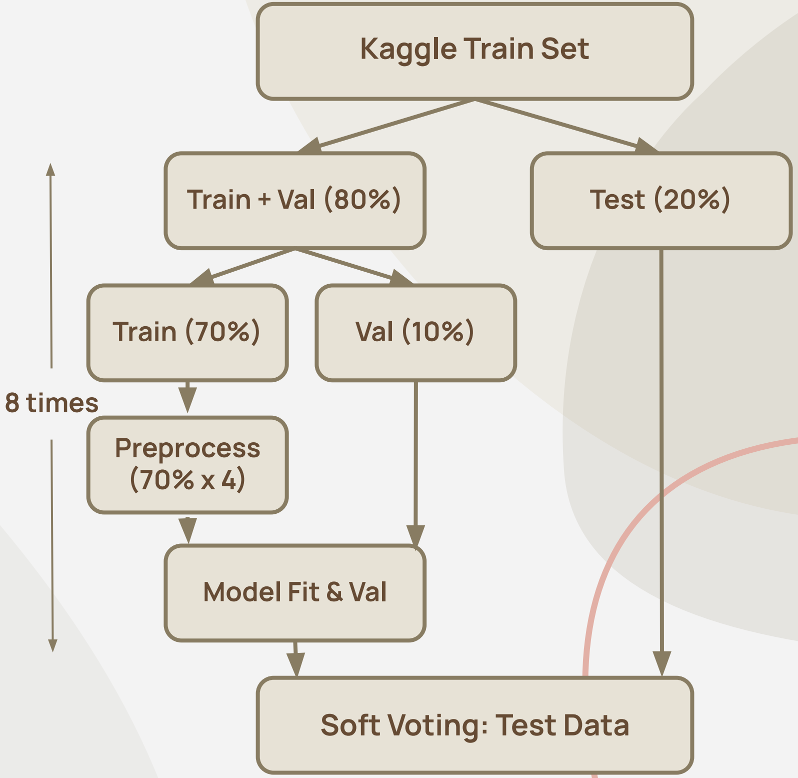
FINAL MODEL: K-FOLD & VOTING

K-FOLD PROCESS

- **Stratified** K-fold at **k = 8**:
  - K-value: **tidy train-val-test split (7:1:2)**
  - Stratified: **balanced classes** in train & test sets
  - Image **preprocessing**: on **training set only**

VOTING

- **Soft-voting:**
  - All 8-folds predict on test data
  - Final decision based on **average probabilities**
  - Consistently **better than hard voting:**
    - lack of extreme outliers (models all similar)
    - Improvement of ~0.495 in AUC ROC score
- Ensemble improves acc by ~3.69%, AUC by ~0.191



FINAL MODEL: K-FOLD & VOTING

K-FOLD PROCESS

- **Stratified** K-fold at **k = 8**:
  - K-value: **tidy train-val-test split (7:1:2)**
  - Stratified: **balanced classes** in train & test sets
  - Image **preprocessing**: on **training set only**

VOTING

- **Soft-voting:**
  - All 8-folds predict on test data
  - Final decision based on **average probabilities**
  - Consistently **better than hard voting:**
    - lack of extreme outliers (models all similar)
    - Improvement of ~0.495% in AUC ROC score
- Ensemble improves acc by ~3.69%, AUC by ~0.191 (with diminishing returns)

	Voting Mechanism		Improvement: Soft - Hard
	Hard	Soft	
Models with Contrast = 0			
- Test Acc	94.48%	94.52%	**+0.04%**
- AUC ROC	0.9805	0.9854	**+0.0049**
Models with Contrast = 3			
- Test Acc	93.42%	93.47%	**+0.05%**
- AUC ROC	0.9758	0.9805	**+0.0048**
Models with Contrast = 5			
- Test Acc	93.32%	92.60%	**+0.29%**
- AUC ROC	0.9699	0.9777	**+0.0079**

** True positive and true negatives show strict improvements as well*

FINAL MODEL: K-FOLD & VOTING

K-FOLD PROCESS

- **Stratified** K-fold at **k = 8**:
  - K-value: **tidy train-val-test split (7:1:2)**
  - Stratified: **balanced classes** in train & test sets
  - Image **preprocessing**: on **training set only**

VOTING

- **Soft-voting:**
  - All 8-folds predict on test data
  - Final decision based on **average probabilities**
  - Consistently **better than hard voting:**
    - lack of extreme outliers (models all similar)
    - Improvement of ~0.495 in AUC ROC score
- Ensemble **improves acc by ~3.69%, AUC by ~1.91%** (with diminishing returns)

	Single Model (Avg)	Ensemble Model	Improvement
Models with Contrast = 0 (8 folds)			
- Test Acc	90.83% (0.013%)	94.52%	**+3.69%**
- AUC ROC	0.9664 (0.009)	0.9854	**+0.019**
Models with Contrast = 3 (8 folds)			
- Test Acc	89.47% (0.013%)	93.47%	**+4.00%**
- AUC ROC	0.9565 (0.009)	0.9805	**+0.024**
Models with Contrast = 5 (8 folds)			
- Test Acc	89.02% (0.032%)	93.47%	**+4.45%**
- AUC ROC	0.9523 (0.021)	0.9777	**+0.025**

FINAL MODEL: K-FOLD & VOTING

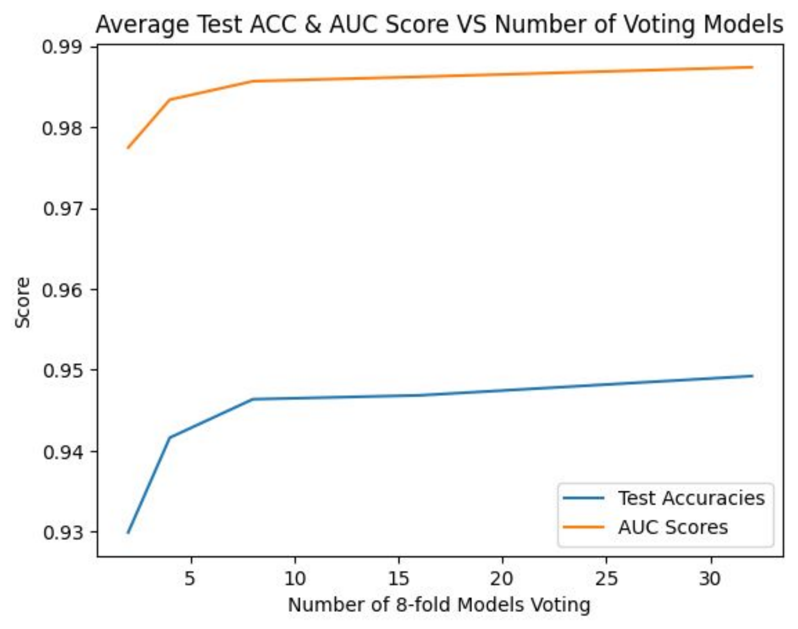
K-FOLD PROCESS

- **Stratified** K-fold at **k = 8**:
  - K-value: **tidy train-val-test split (7:1:2)**
  - Stratified: **balanced classes** in train & test sets
  - Image **preprocessing**: on **training set only**

VOTING

- **Soft-voting:**
  - All 8-folds predict on test data
  - Final decision based on **average probabilities**
  - Consistently **better than hard voting:**
    - lack of extreme outliers (models all similar)
    - Improvement of ~0.495 in AUC ROC score
- Ensemble **improves acc by ~3.69%, AUC by ~0.191** (with diminishing returns)

	Single Model (Avg)	Ensemble Model	Improvement
[illegible]	[illegible] (0.032%)	[illegible]	
- AUC ROC	0.9523 (0.021)	0.9777	



FINAL MODEL: OPTUNA

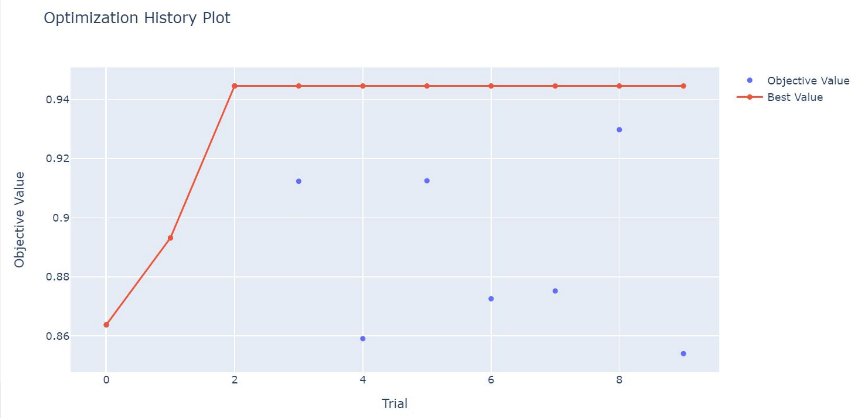
Model Tuning Using Optuna

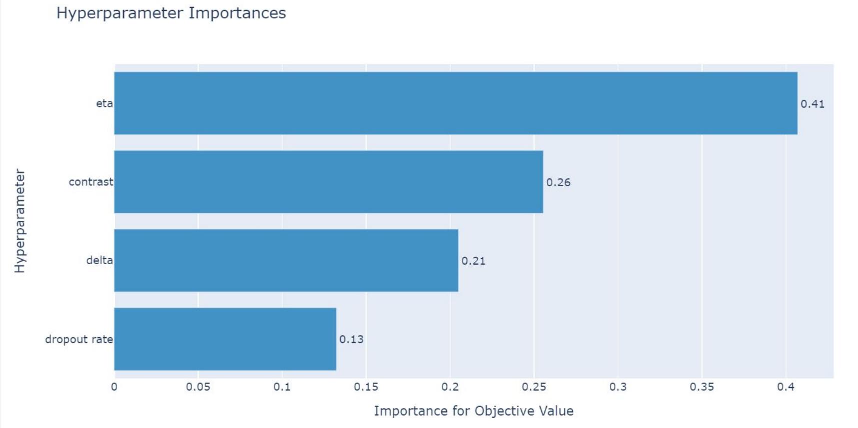
- **Optuna:** using Bayesian model for **hyperparameter tuning**

- Study run with **10 trials**, **optimize on AUC**:
    - **Learning rate**: 0.1 - 0.001, log-scale
    - **Dropout** rate: 0.1 - 0.5, float
    - **Contrast** factor: -1 - 5, integer
    - **Delta**: 0 - 0.5, float
    - Reduced image augmentation to 2x (from 4x) due to memory issues

- Best trial parameters:
    - **Learning rate**: 0.00121
    - **Dropout** rate: 0.2497
    - **Contrast** factor: -1
    - **Delta**: 0.2150

- More testing required for convergence





FINAL MODEL: LEARNING RATE

Learning Rates

- **4 sets** of numbers **tested**:
  - 0.001 (Adam default)
  - 0.00121 (Optuna rate)
  - 0.01
  - 0.1
- At **10 epochs:**
  - Best result: 0.001
  - Minor diff between 0.001 & 0.00121
  - More optuna trials may be needed
- Other issues:
  - Insufficient time to tune epoch num
  - Rate decay unexplored

	Learning Rate			
	0.001	**0.00121**	**0.01**	**0.1**
Avg Val Loss (Epoch 10)	0.3010 (sd: 0.046)	**0.2882 (sd: 0.063)**	0.3060 (sd: 0.042)	0.4134 (sd: 0.136)
Avg Val Acc (Epoch 10)	90.76% (sd: 1.43%)	**90.93% (sd: 1.31%)**	88.42% (sd: 2.03%)	83.03% (sd: 5.43%)
Test Acc (Ensemble)	**94.48%**	94.16%	93.17%	90.08%
AUC ROC (Ensemble)	**0.9854**	0.9830	0.9793	0.9777

FINAL MODEL: LEARNING RATE

Learning Rates

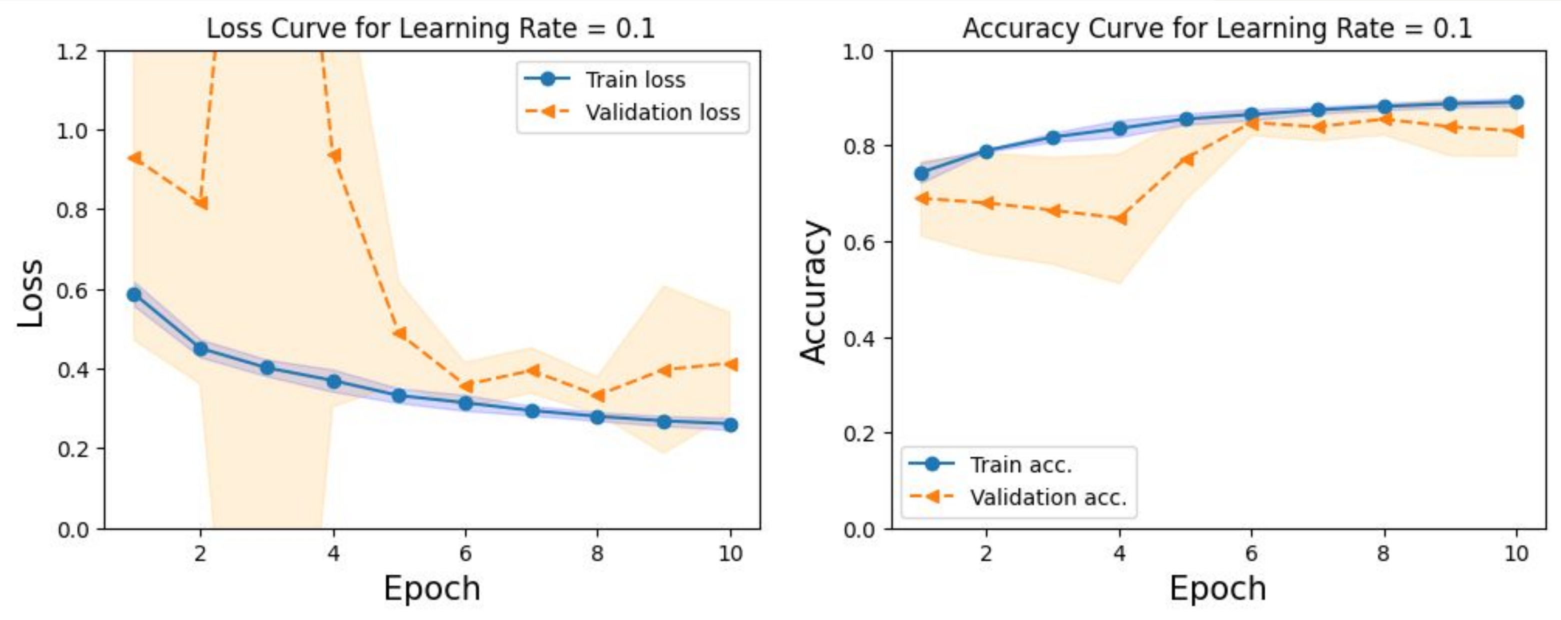
- **4 sets** of numbers **tested**:
  - 0.001 (Adam default)
  - 0.00121 (Optuna rate)
  - 0.01
  - 0.1
- At **10 epochs:**
  - Best result: 0.001
  - Minor diff between 0.001 & 0.00121
  - More optuna trials may be needed
- Other issues:
  - Insufficient time to tune epoch num
  - Rate decay unexplored

Learning rate of 0.001 picked

	Learning Rate			
	0.001	0.00121	0.01	0.1
Avg Val Loss (Epoch 10)	0.3010 (sd: 0.046)	0.2882 (sd: 0.063)	0.3060 (sd: 0.042)	0.4134 (sd: 0.136)
Avg Val Acc (Epoch 10)	90.76% (sd: 1.43%)	90.93% (sd: 1.31%)	88.42% (sd: 2.03%)	83.03% (sd: 5.43%)
Test Acc (Ensemble)	94.48%	94.16%	93.17%	90.08%
AUC ROC (Ensemble)	0.9854	0.9830	0.9793	0.9777

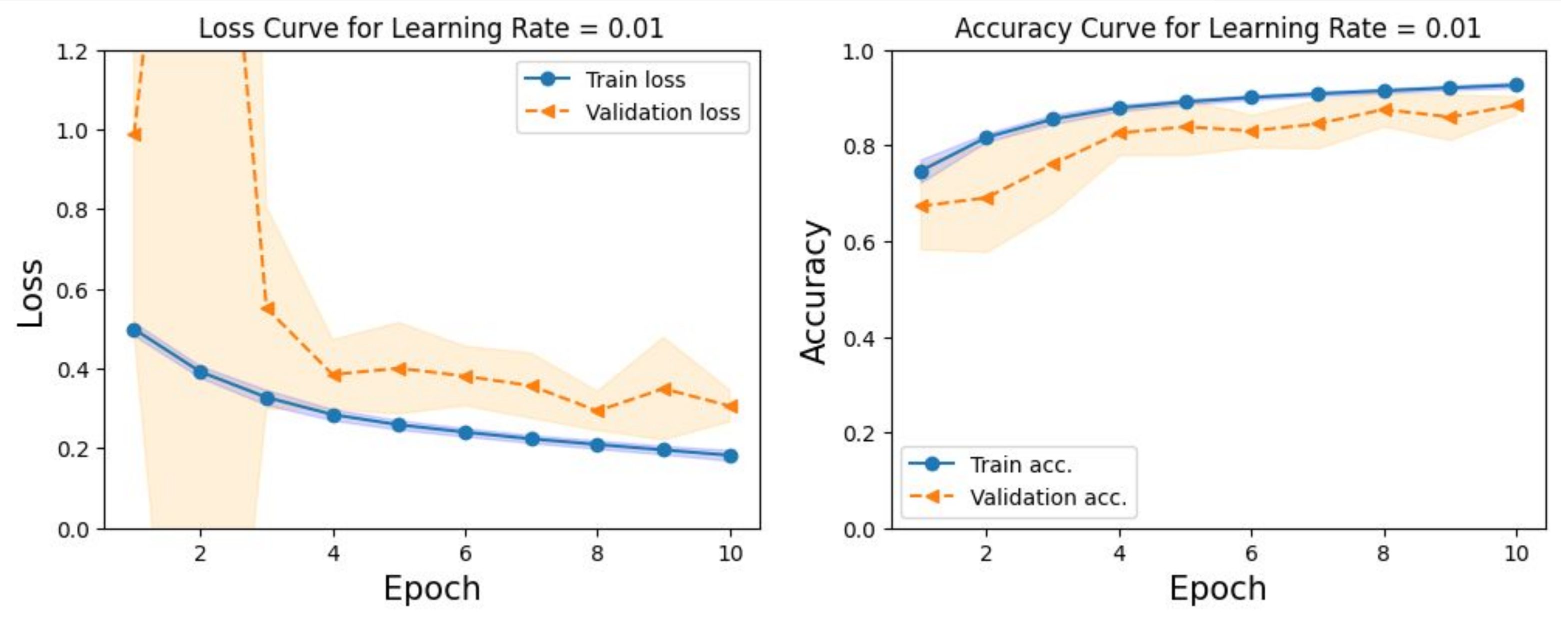
LEARNING RATE: 0.1

Loss and Accuracy Curves



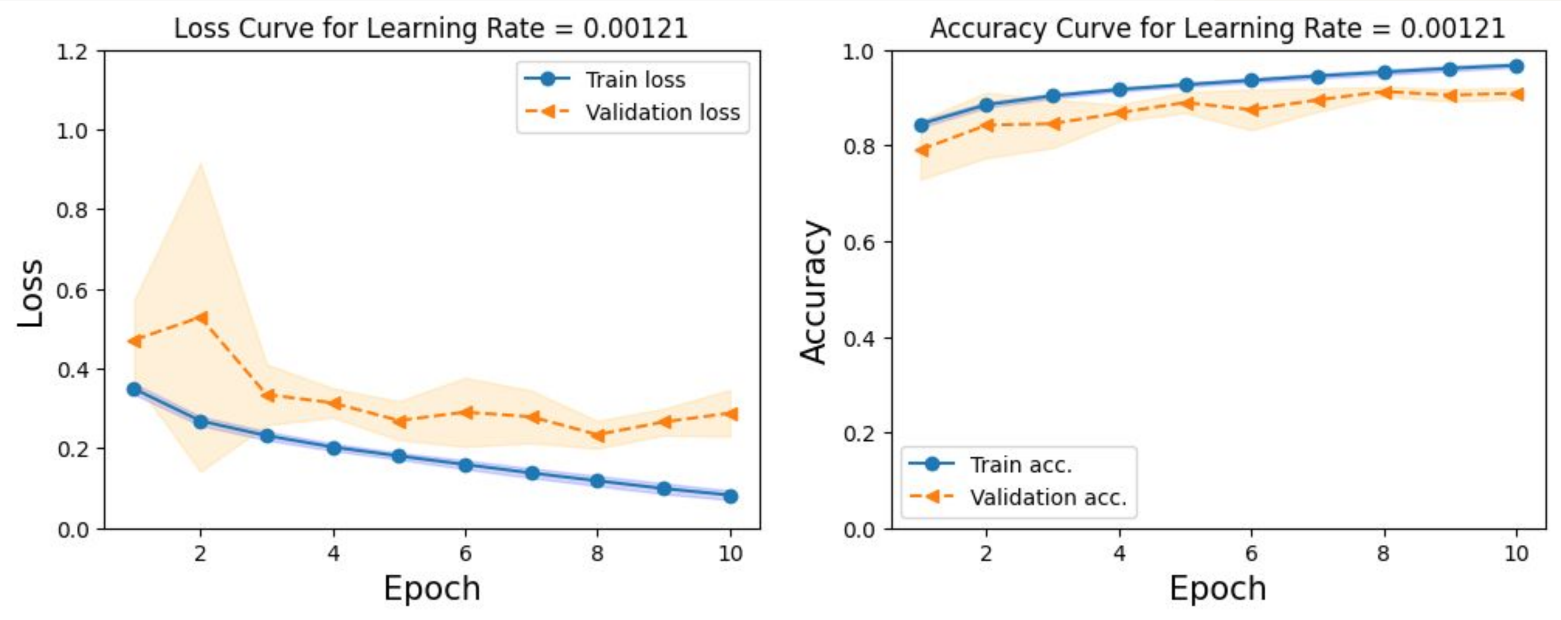
LEARNING RATE: 0.01

Loss and Accuracy Curves



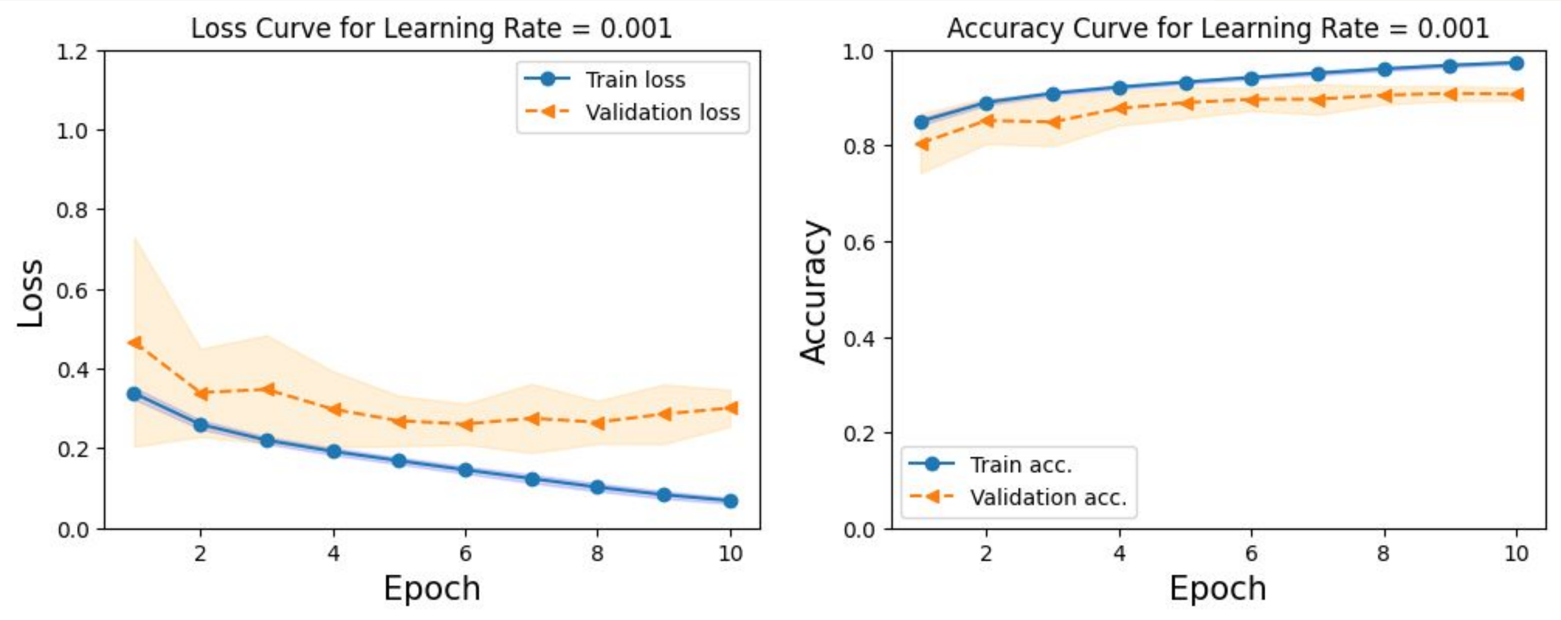
LEARNING RATE: 0.00121

Loss and Accuracy Curves



LEARNING RATE: 0.001

Loss and Accuracy Curves



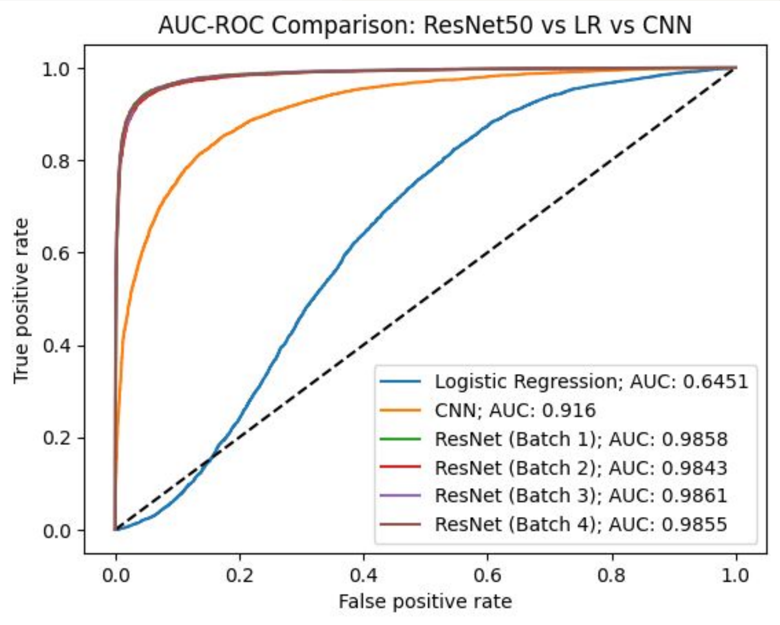
LEARNING RATES: CONFUSION MATRIX COMPARISON

Consistently Good

	Learning Rate			
	0.001	0.00121	0.01	0.1
True Positives	**8307 (93%)**	8273 (93%)	**8337 (94%)**	7452 (84%)
False Negatives	**604 (7%)**	638 (7%)	**574 (6%)**	1459 (16%)
True Negatives	**8538 (96%)**	8509 (95%)	8267 (93%)	**8602 (97%)**
False Positives	**373 (4%)**	402 (5%)	644 (7%)	**309 (3%)**

FINAL MODEL: PERFORMANCE SUMMARY

	Performance (over 4 batches)
Overall test accuracy	94.48% (sd: 0.21%)
- Precision	93.22% (sd: 0.47%)
- Recall	95.71% (sd: 0.87%)
F1 score	0.9445 (sd: 0.003)
AUC-ROC score	0.9854 (sd: 0.0007)

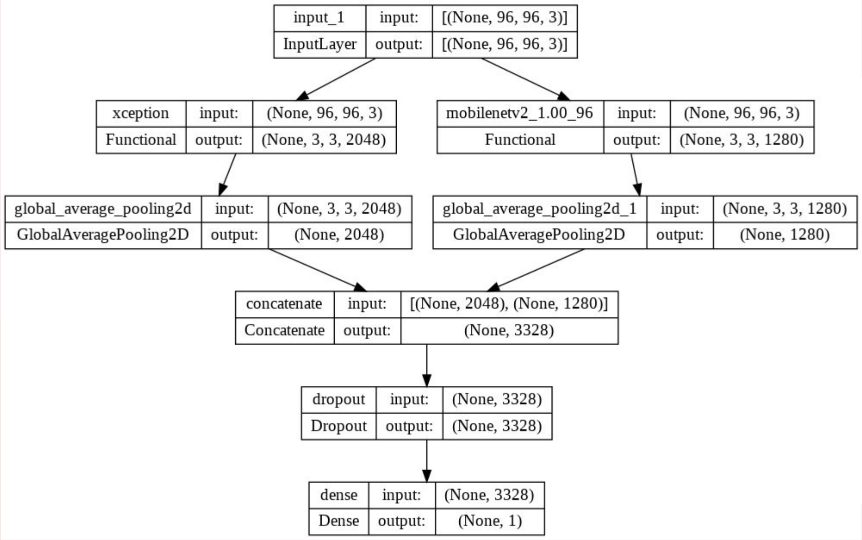


05

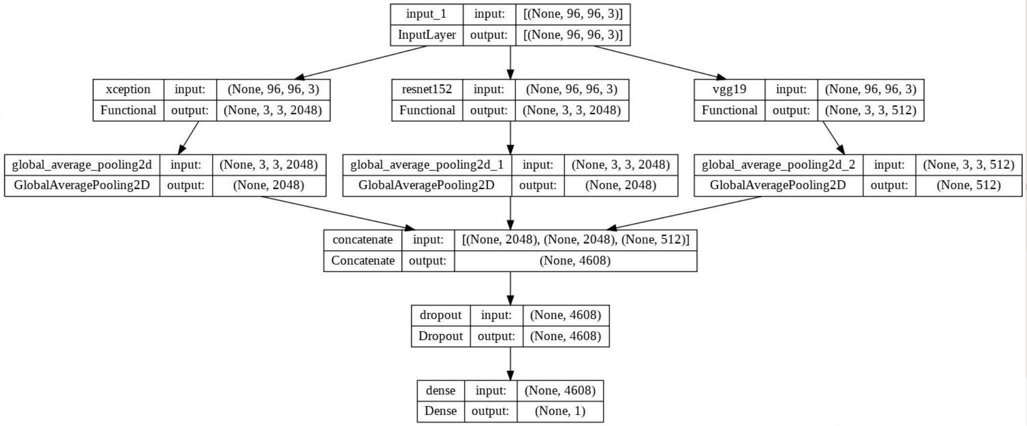
FUTURE WORK

ENSEMBLE MODELS

Xception + MobileNet



Xception+ResNet+VGG19



06

REFERENCES

REFERENCES

Gurcan MN, et al. Histopathological image analysis: a review. IEEE Rev Bimed Eng. 2009; 2:147-71, https://ieeexplore-ieee-org.libproxy.berkeley.edu/document/5299287

Xin, M., Wang, Y. Research on image classification model based on deep convolutional neural network. *J Image Video Proc.* 2019, 40 (2019). https://link.springer.com/article/10.1007/s13755-018-0057-x

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*. doi:10.48550/ARXIV.1512.03385 https://arxiv.org/abs/1512.03385

Chollet, F.. (2016). Xception: Deep Learning with Depthwise Separable Convolutions. https://arxiv.org/abs/1610.02357

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. https://arxiv.org/abs/1905.11946

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8172839/

https://www.cancer.org/research/cancer-facts-statistics/all-cancer-facts-figures/cancer-facts-figures-2022.html#:~:text=The%20Facts%20%26%20Figures%20annual%20report,deaths%20in%20the%20United%20States.)

CONTRIBUTIONS

Chenyu Wang

- Codes & slides
- Literature review:
  - model
  - architectures
- Data preprocessing
- Baseline model
- CNN model tuning
- PCA try and error
- Ensemble model:
  - Efficientnetb7 + inception+vgg19
- Brainstorming

Hector Rincon

Base Resnet model code
Densenet201 exploration experiments
Data preprocessing
Code deduplication/DRYing
Slides
Optuna experimentation

Ifrah Javed

-Slides
-Literature Review
 - Ensemble Models
 - Project Need
-Baseline Model
 - Optimization for LR Model
- GridSearch Optimization
- Learning Rate Optimization
- Brainstorming, general meeting planning, notes and setting up timeline and setting up team tools (Git, Google Folder, Presentation Slides)

Justin To

- RESNET model:
  - tuning
  - KPIs
- Experiments:
  - K-fold & voting
  - image preprocessing
  - learning rates
  - Optuna
- Preprocessing:
  - code review
- Literature review:
  - model architectures
- Brainstorming
- Codes & slides

Srila Maiti

Set up the initial structure of the code, reusable functions, classes, EDA, automating download of data from Kaggle

Baseline Logistic Regression Model, 5 CNN version experiments

Initial transfer learning model experiment with VGG16

Transfer Learning model experiments with Mobilnet, Restnet152, Xception and Mobilenet combination, InceptionRestnet, Densenet XGboost and image augmentation experiments and ensemble methods

Literature Review, coding, slides creation, brainstorming ideas

QUESTIONS?

THANK YOU!

07

BACKUP SLIDES

FINAL MODEL: OVERVIEW

Model Architecture

- ResNet50
- Trainable parameters: ~23.54 million

Image Preprocessing

- Greyscale, then duplicate by 3 (ResNet takes in 3 channels)
- Random flipping
- All images rotated and added to original (i.e. 4x training size)

K-fold & Voting

- 8-folds, i.e. train-validation-test split is 7-1-2
- Train data then multiplied by 4 (i.e. total = 62,380 x 4 = ~ 249, 520)
- Soft-voting (i.e. average probabilities, not average votes)

FINAL MODEL: OVERVIEW

Model Architecture

- ResNet50
- Trainable parameters: ~23.54 million

Image Preprocessing

- Greyscale, then duplicate by 3 (ResNet takes in 3 channels)
- Random flipping
- All images rotated and added to original (i.e. 4x training size)

K-fold & Voting

- 8-folds
- Train data then multiplied by 4 (i.e. total = 62,380 x 4 = ~ 249, 520)
- Soft-voting (i.e. average probabilities, not average votes)

FINAL MODEL: OVERVIEW

Model Architecture

- ResNet50
- Trainable parameters: ~23.54 million

Image Preprocessing

- Greyscale, then duplicate by 3 (ResNet takes in 3 channels)
- Random flipping
- All images rotated and added to original (i.e. 4x training size)

K-fold & Voting

- 8-folds
- Train data then multiplied by 4 (i.e. total = 62,380 x 4 = ~ 249, 520)
- Soft-voting (i.e. average probabilities, not average votes)

FINAL MODEL: RESNET

MODELS TESTED

VG

- 13 Conv2D +
- Small kernels
  - multiple la
    pooling →
    kernels
  - but more r
    less param
- Increasing c
- Reduces effort on testing layer adding

- Improved performance over VGG

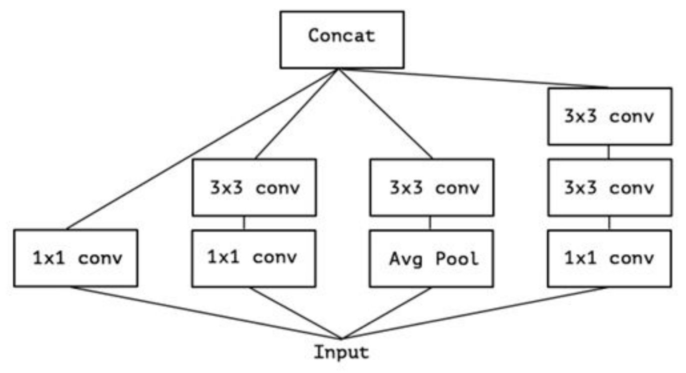


Figure 1. A canonical Inception module (Inception V3).

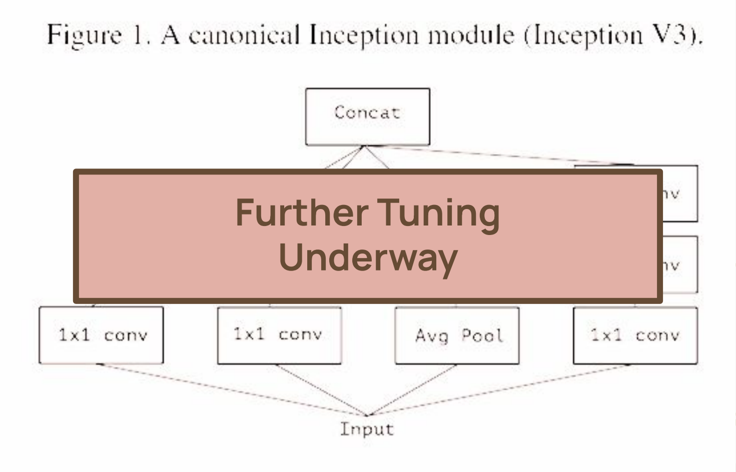
Reference: Xception: deep learning with depthwise separable convolutions; by F Chollet; 2016

Xception & MobileNet

- **Improved Inception** model:
  - main idea: multiple convolutions, then concatenating them
- **Depthwise separable convolution layers** for faster computation

FINAL MODEL: RESNET

MODELS TESTED



Further Tuning Underway

Xception

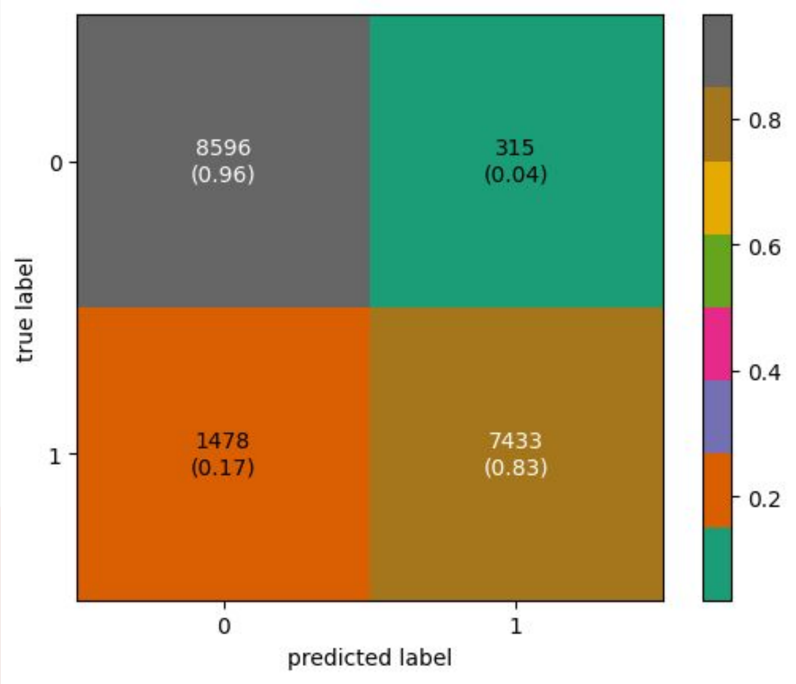
- **Improved Inception** model:
  - main idea: multiple convolutions, then concatenating them
- **Depthwise separable convolution layers** for faster computation

Reference: Xception: deep learning with depthwise separable convolutions; by F Chollet; 2016

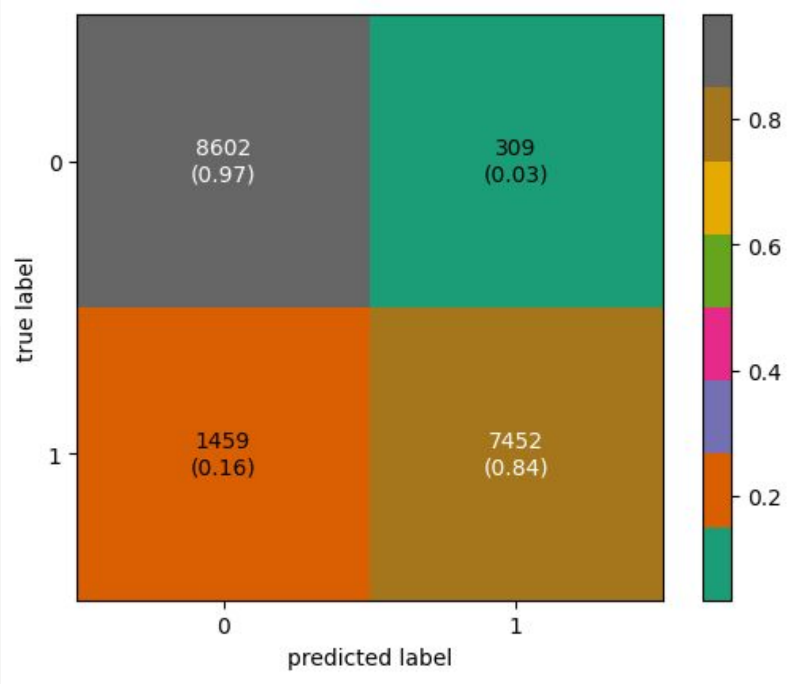
LEARNING RATE: 0.1

Confusion Matrix

(i) Hard Voting



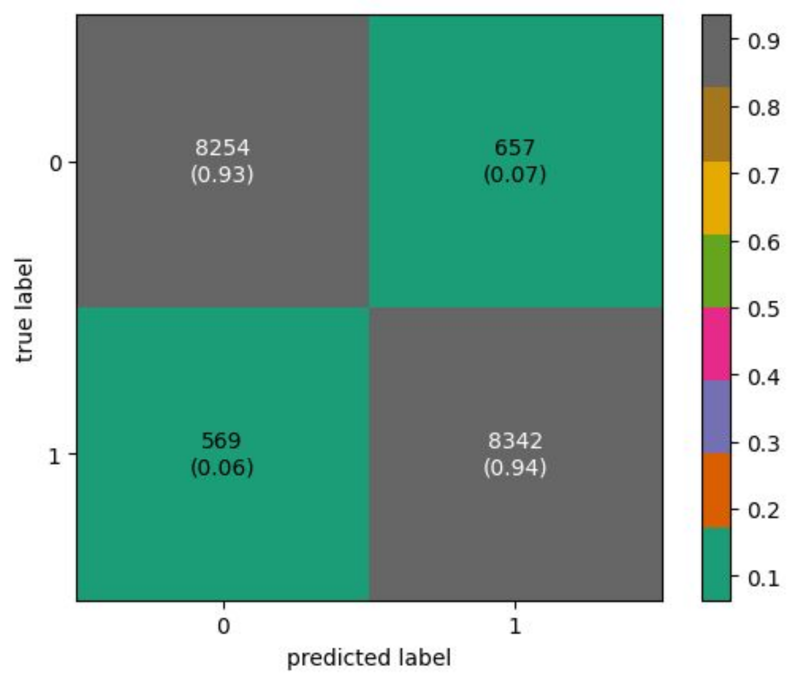
(ii) Soft Voting



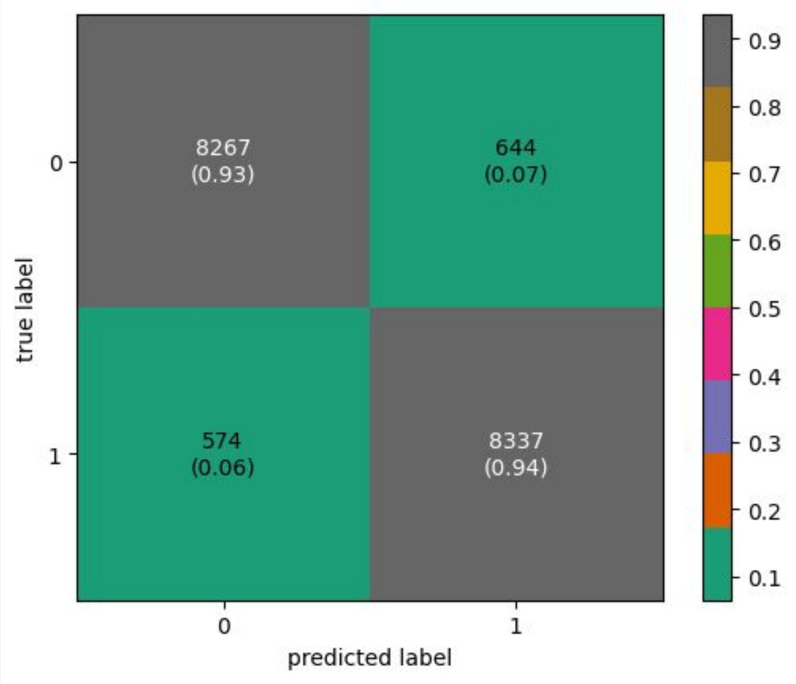
LEARNING RATE: 0.01

Confusion Matrix

(i) Hard Voting



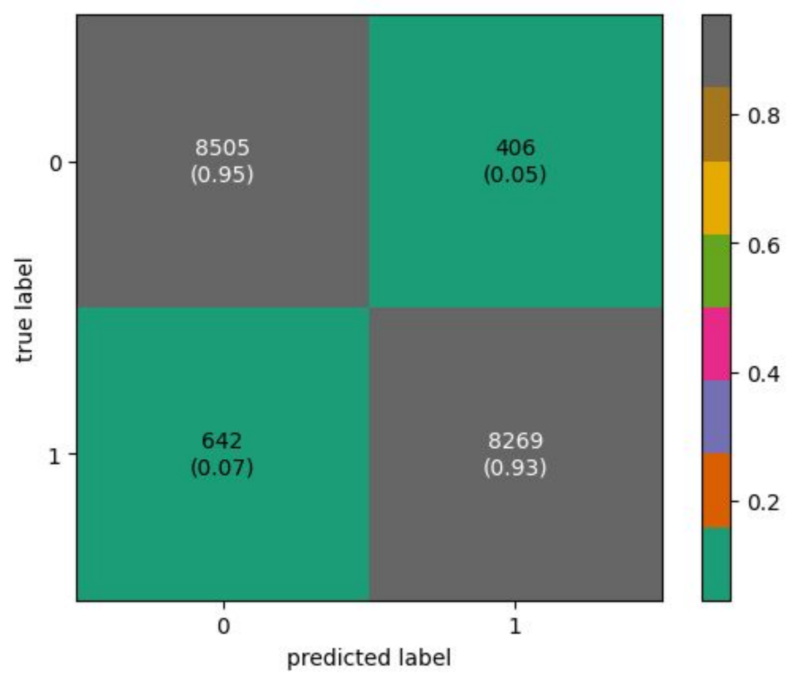
(ii) Soft Voting



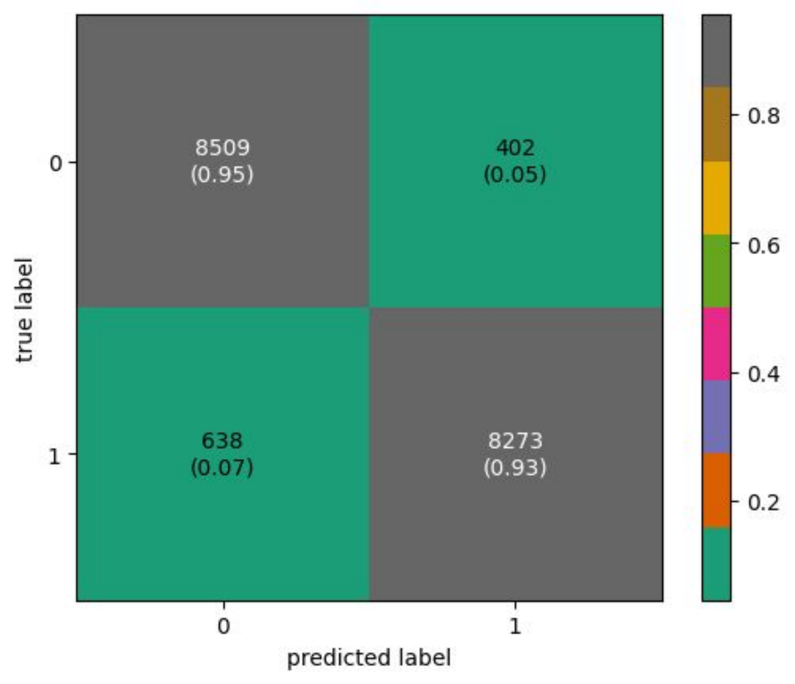
LEARNING RATE: 0.00121

Confusion Matrix

(i) Hard Voting



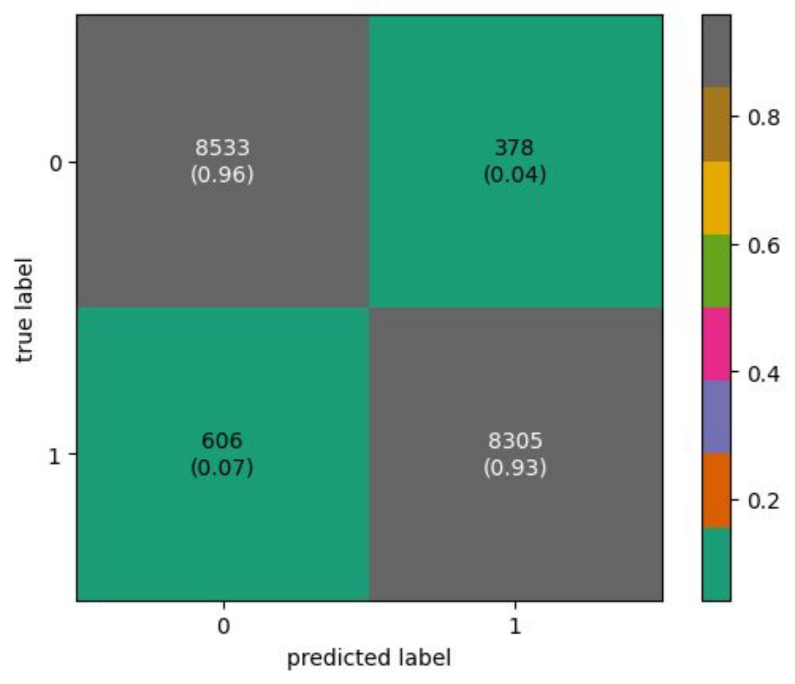
(ii) Soft Voting



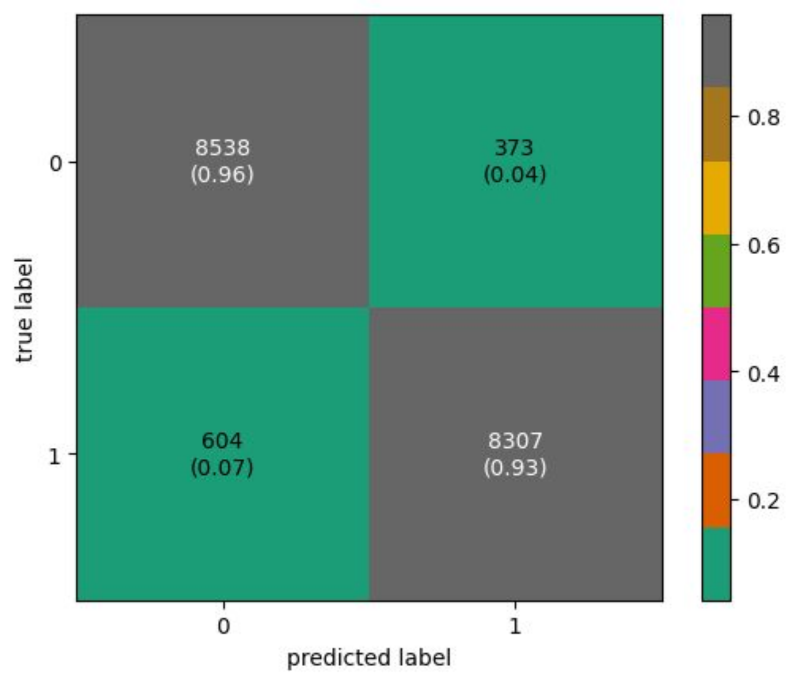
LEARNING RATE: 0.001

Confusion Matrix *(Average of 4 batches of 8 models)*

(i) Hard Voting



(ii) Soft Voting



DIFFERENT TRANSFER LEARNING MODELS EVALUATION

Available models							
Model	**Size (MB)**	**Top-1 Accura**	**Top-5 Accura**	**Parameters**	**Depth**	**Time (ms) pe**	**Time (ms) per inference step (GPU)**
Xception	88	79.00%	94.50%	22.9M	81	109.4	8.1
VGG16	528	71.30%	90.10%	138.4M	16	69.5	4.2
VGG19	549	71.30%	90.00%	143.7M	19	84.8	4.4
ResNet50	98	74.90%	92.10%	25.6M	107	58.2	4.6
ResNet50V2	98	76.00%	93.00%	25.6M	103	45.6	4.4
ResNet101	171	76.40%	92.80%	44.7M	209	89.6	5.2
ResNet101V2	171	77.20%	93.80%	44.7M	205	72.7	5.4
ResNet152	232	76.60%	93.10%	60.4M	311	127.4	6.5
ResNet152V2	232	78.00%	94.20%	60.4M	307	107.5	6.6
InceptionV3	92	77.90%	93.70%	23.9M	189	42.2	6.9
InceptionResNetV2	215	80.30%	95.30%	55.9M	449	130.2	10
MobileNet	16	70.40%	89.50%	4.3M	55	22.6	3.4
MobileNetV2	14	71.30%	90.10%	3.5M	105	25.9	3.8
DenseNet121	33	75.00%	92.30%	8.1M	242	77.1	5.4
DenseNet169	57	76.20%	93.20%	14.3M	338	96.4	6.3
DenseNet201	80	77.30%	93.60%	20.2M	402	127.2	6.7
NASNetMobile	23	74.40%	91.90%	5.3M	389	27	6.7
NASNetLarge	343	82.50%	96.00%	88.9M	533	344.5	20
EfficientNetB0	29	77.10%	93.30%	5.3M	132	46	4.9
EfficientNetB1	31	79.10%	94.40%	7.9M	186	60.2	5.6
EfficientNetB2	36	80.10%	94.90%	9.2M	186	80.8	6.5
EfficientNetB3	48	81.60%	95.70%	12.3M	210	140	8.8
EfficientNetB4	75	82.90%	96.40%	19.5M	258	308.3	15.1
EfficientNetB5	118	83.60%	96.70%	30.6M	312	579.2	25.3
EfficientNetB6	166	84.00%	96.80%	43.3M	360	958.1	40.4
EfficientNetB7	256	84.30%	97.00%	66.7M	438	1578.9	61.6